HAAGA-HELIA
University of Applied Sciences

SAP® ERP Upgrade and Testing

Case: Haaga-Helia UAS

Titta-Maria Silvennoinen

Bachelor's Thesis

DP in Business Information Technology

2011

HAAGA-HELIA
University of Applied Sciences

Abstract

Business Information Technology

Author or authors Titta-Maria Silvennoinen	**Group or year of entry** 2011
Title of report SAP® ERP Upgrade and Testing. Case: Haaga-Helia UAS.	**Number of pages and appendices** 126 + 27
Teacher/s or supervisor/s Jarmo Harmonen	

The necessity for testing of the new upgraded SAP® ERP 6.0 training environment at Haaga-Helia University of Applied Sciences led to the need for this thesis paper. The practical aim of the testing part was to ensure that the functionality and business processes required in the SAP ERP basics course of the Business Information Technology DP, having started in the autumn semester 2010, would not be affected by the upgrade and the course exercises of the previous implementation of the course, which functioned as test cases, would still be suitable.

The research method used for this thesis was a qualitative research with an empirical part. The empirical part consisted of a single-case study – i.e. testing the functionality and processes of the SAP ERP 6.0 application at Haaga-Helia UAS. Testing was conducted during the summer 2010. The test result tables for each test case can be found in the thesis appendices.

The theoretical part based on relevant literal and online sources introduced the field of study to the readers, so that they are better able to comprehend the significance of the empirical testing part and its results. The idea of this thesis was not to invent something new or improve something old - instead, the purpose was to provide an overview of the ERP upgrade as a post-implementation phase phenomenon and highlight relevant factors related to SAP ERP upgrade and, at the same, time provide the framework for the empirical part.

This thesis project indicated that upgrading an SAP ERP system can be a complex, business-changing project that deserves much of the same careful planning and execution as a new implementation. As a whole, this study provides a good information package on SAP ERP upgrade and testing as well as on other numerous factors that should be considered when planning, managing and executing an upgrade. In general, the duration, complexity and cost of an upgrade project depend on many factors such as the complexity of the SAP ERP landscape; the number of modules and the complexity of functionality in use; the extent of modifications; the number of interfaces to other applications; how far behind the client's installed version has fallen from the latest vendor release; technical requirements; and the skills of the upgrade team.

Most likely this thesis will be useful for those who are interested in ERP software and issues related to its post-implementation phase, and to those who are interested in upgrading their SAP ERP solution. The objectives of the empirical testing part were achieved and therefore the thesis is also useful for the sponsor.

Keywords
ERP upgrade, SAP ERP, testing, ERP maintenance, ERP life cycle

Table of contents

1 Introduction

All companies, irrespective of whether the company is a multi-national, multi-million euro enterprise or a small company with a six number turnover, need one or more information solutions to manage and optimize company's business operations and store important data. Nowadays, the most commonly used solution by companies is so called Enterprise resource planning software - ERP. It is an integrated commercial off-the-shelf (COTS) software package, typically comprised of several modules to support common business activities such as finance, sales, distribution, production or human resources. Each of these modules share information that is housed within the database structures across an entire enterprise to facilitate business planning and decision making. There are a large number of ERP vendors competing in the market, some of them offering more than just one solution. They can for instance provide completely different solutions for small to midsize organizations or large enterprises. Most solutions are designed to cover a range of industry segments, though some of these ERP solutions are tailor-made for certain industry segments only. The SAP® ERP packaged software that is subject of this study is SAP AG's enterprise resource planning solution for midsize and large organizations in all industries.

The world around is in constant change and as world changes also requirements for the ERP systems change. This means that once implemented system is also in constant change. Companies need to enhance or upgrade their ERP solutions in order to keep the system running smoothly in evolving technical environments and to improve their business processes to keep up with competition. In regards to ERP lifecycle concept all efforts and steps taken to upgrade and attain better benefits once the system is implemented take place during so called post implementation phase. Given the significant complexity of SAP ERP systems, upgrade projects are time demanding and complex. In order to avoid potential errors that may affect key application functionalities and disrupt critical business processes, SAP ERP users are required to perform comprehensive tests after each upgrade before going live.

This study discusses the ERP upgrade post-implementation phase phenomenon in general, its complexity and challenges of testing. Also a practical testing part is included – that is testing the functionality and certain processes of the SAP ERP 6.0 application at Haaga-Helia University of Applied Sciences.

1.1 Case study description and motivation for the research topic

A job market demand for trained enterprise resource planning professionals has motivated many universities to offer courses that combine business process knowledge with enterprise software expertise. The Finnish Haaga-Helia University of Applied Sciences (UAS) among number of other universities in Finland and globally have joined alliances with ERP software vendors such as SAP in order to be better able to provide tools and resources to teach their students as well as enable hands- on classroom and lab experiences with ERP software. Haaga-Helia UAS has already a ten year history of co-operating with SAP, ever since the first five Finnish Universities and Polytechnics joined the SAP University Alliance (SAP UA Finland 2010; SAP 2010a). Nowadays, SAP ERP is used in several courses of different degree programs (DP) at Haaga-Helia – in information technology DP as well as in business and management assistants DPs.

The SAP University Alliances program has established five University Competence Centers (UCCs) around the world that provide the member schools access the full suite of SAP software and maintenance, operations and technical support of the SAP software - thus eliminating the need for individual universities to make large investments in technical infrastructure (SAP SDN 2010; SAP UA EMEA 2010a).

The UCC Magdeburg in Germany acts as the SAP application service provider for the Haaga-Helia UAS (KKA 2010, 5). They are hosting an SAP ERP system, IDES version, for Haaga-Helia. IDES (Internet Demo and Evaluation System) is basically a fully functional SAP ERP system but it has been pre-configured and pre-populated with lots of master and transaction data, so that typical real-life business processes can be executed immediately in the class room (SAP HELP 2010).

During the summer of 2010, when no SAP-related courses are held, Haaga-Helia is upgrading the IDES from SAP ERP ECC 5.0 to the latest version 6.0. For Haaga-Helia upgrading the IDES environment provided by the UCC is fairly easy since the environment is a typical client server environment. The applications are operated and databases hosted at UCC servers and UCC actually takes care of upgrading SAP software when requested (SAP UA EMEA 2010b).

Due to the enormous complexity of a typical SAP system, every upgrade requires extensive testing. Even though, in this case we are not upgrading any critical SAP production system and no development and testing environments are needed as in general, sufficient testing is still required to make sure that the new IDES-environment works properly and all the functionality will be available for the SAP-courses starting in the autumn 2010 at Haaga-Helia.

The necessity for testing of the new upgraded SAP ERP 6.0 environment led to the need for this thesis work. The topic was proposed by instructor Jarmo Harmonen from Haaga-Helia. The researcher is very interested of learning more of ERP software in general and especially SAP ERP, so the proposed topic found its goal. Harmonen provided a set of test cases for the researcher and they form the empirical part of the thesis work. The theoretical part will provide necessary background information about ERP systems in general, ERP life-cycle and post-implementation phase. After that a deeper insight into SAP ERP, its maintenance activities and upgrade issues is provided. Also the testing activities related to upgrade are discussed thus providing a theoretical frame for the empirical part.

Other motivations and reasons why the thesis topic SAP ERP Upgrade and Testing is important:

- SAP is one of the largest ERP software providers.
- The topic is very current: ERP market today is considered to be a mature market and many companies have shifted /are shifting into post-implementation phase in their ERP life cycle (Ng & Gable 2010, 65; Worrell J.L. 2008, 2). It means that they need to enhance or upgrade their ERP solu-

tion in order to keep the system running smoothly in evolving technical environments and improve their business processes.

- Upgrade is one of the important activities in the ERP software lifecycle: while first time ERP implementation happens only once, ERP upgrade will occur many times after the first ERP implementation.
- Companies and organizations spend a significant amount of money on each ERP upgrade project (Zhao 2007, 4).
- Relatively little research attention has been given to ERP software upgrade (Zhao 2007, 4; Worrell 2008, 2; Otieno 2010, 12; Martinsen 2010, 6).
- According to feedback from SAP customers, testing is a major cost driver and challenge in upgrading (Riedel 2009, 122).

1.2 Objectives and research questions

The practical aim of the thesis is to test the upgraded SAP ERP 6.0 environment of Haaga-Helia UAS to ensure that functionality and business processes required in the SAP ERP basics course of the Business Information Technology DP starting in the autumn semester 2010 were not affected and the course exercises of the previous implementation of the course, that function as test cases, are still suitable.

Naturally, theoretical foundation for the practical part is studied and presented. The purpose of this part is to describe the field of study based on relevant research papers, books and other reliable sources and provide a theoretical framework for the practical part. The purpose is to introduce the study field to the reader so that she/he is better able to comprehend the significance of the practical part.

Main research problem areas and questions here are:

ERP lifecycle:

- What is the relationship of an ERP upgrade to ERP lifecycle?

Upgrading SAP ERP:

- What is an upgrade?
- What are the reasons for an upgrade?
- How does upgrading to a new version differ from implementation?
- What should be considered when upgrading the ERP system?
- Are there any risks related to an upgrade? How to avoid risks?
- Can any upgrade success factors be specified?
- Are there any other important factors related to SAP ERP upgrade?

Testing activities related to upgrading projects:

- Is testing necessary after upgrading? Why?
- What kind of testing activities are related to upgrade projects?

Who will benefit from this research? The researcher herself aims to get deeper understanding of the upgrade process of the SAP ERP application and improve the ERP knowledge in general during the thesis project. In addition, this paper is an attempt to provide information also to others who are interested in ERP software and issues related to its post-implementation phase, and to those who are interested in upgrading their ERP solution. The paper will not only benefit the sponsor but will give the insight about upgrading and testing SAP ERP to students, researchers, upgrading organizations and tester as well.

1.3 Scope

The theoretical part has a significant role in this thesis, because it helps the reader to understand the practical part and its results. The results are gathered during the testing of the new SAP ERP 6.0 upgrade. The idea of this thesis is not to invent something new or improve something old - instead, the purpose is to provide an overview of the ERP upgrade as a post-implementation phase phenomenon and highlight relevant factors related to it and at the same time provide the framework for the empirical part. This research does not cover upgrade tools or technologies related to version upgrade or give any detailed technical instructions for installing an upgrade.

Every upgrade requires testing. The practical aim of the empirical part is to test the upgraded SAP ERP 6.0 environment of Haaga-Helia UAS to ensure that functionality and business processes required in the SAP ERP basics course of the Business Information Technology DP starting in the autumn semester 2010 were not affected by the upgrade and the course exercises of the previous implementation of the course, that function as test cases, are still suitable. There were total of six test cases. The scope of the empirical part is explained more in detail in the test plan in Chapter 3.1.

1.4 Methodology

The research method used for this thesis is a qualitative research with an empirical part. The empirical part consists of a case study - that is testing the functionality and processes of the SAP ERP 6.0 application at Haaga-Helia UAS. A case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident (Yin 2002). Case studies can deal with either single or multiple cases. This study is a single-case study. Case study research is the most common qualitative method used in information systems (Orlikowski & Baroudi, 1991, in Mayers 2010).

The purpose of a case study can differ in different researches. Stake (1995) divides case studies into intristic or instrumental according to their aims. In an intristic case study the interest is only in understanding the particulars of the case, whereas in an instrumental case study the interest is in understanding something more general than the case, in other words to learn more about other cases as well (Stake 1995, 3). The case itself is of secondary interest and plays a supportive role in order to facilitate our understanding of something else in order to pursue the external interest (Stake 1995, 10).

In this study the empirical part of the research is more characterised by instrumental ambitions for the case study. That means that the case study is done to provide a general understanding of an SAP upgrade phenomenon using Haaga-Helia as a particular case. In this kind of situation, the analysed case is used as an example of a group of other cases that could also have been selected for analysis. The key purpose is to describe, conceptualise and learn from SAP ERP upgrade phenomenon.

Technical environment and tools for the case study (testing) are described in Chapter 3.1.2.

2 Theoretical foundation

In this section, the research theory is presented. First is given a short introduction to enterprise resource planning and ERP software. Next SAP ERP software and main concepts related to it are presented. Then the upgrading of ERP software in general is discussed and upgrade defined in the ERP life cycle context. Here the research questions addressed are: What is the relationship of an ERP upgrade to ERP lifecycle? What is an upgrade? How does upgrading to a new version differ from implementation? Finally, the research provides an overview how to plan, manage, execute and test SAP ERP upgrades. The latter part tries to answer the research questions: What are the reasons for an upgrade? What should be considered when upgrading the ERP system? Are there any risks related to an upgrade? How to avoid risks? Can any upgrade success factors be specified? Are there any other important factors related to SAP ERP upgrade?

2.1 Enterprise resource planning (ERP)

ERP stands for Enterprise Resource Planning. When people refer to ERP, they usually mean ERP software package. Even though ERP is most frequently used in the context of software, it in fact refers also to organizations business strategy of managing its business resources – e.g. how it integrates and optimizes production, distribution and finance (Salonen 2006, 13). Every business is obligated to have strategies for the use of business resources or it isn't a business for very long. Even the smallest businesses need to schedule staff, order materials, provide facilities, plan cash flow, pay bills and salaries. All that data could be managed by pencil and paper or even by electronic spreadsheets, but it would be a very ineffective ERP system requiring an army of employees to do it. (Getman 2008.) This is why large software applications have been developed to help companies implement ERP in their organization. Indeed, the fundamental aim of ERP software is to "pour" business processes into software, that is, to link technology with operational workflows within an enterprise (Muir & Kimbell 2010, 37).

2.1.1 ERP software – general

According to many authors (e.g. Monk 2009: 17-21; Olson 2009: 11-12, Sumner 2005: 2-3, Rashid & Hossain & Patrick 2002: 1-16) the roots of the modern ERP systems can be traced back to forty years in manufacturing industry where material requirements planning and shop-floor controlling were the main concern at the time. The MRP systems develop in 1970s were only focused on the production line, manufacturing and inventory control. Later on, in 1980's MRP systems were enhanced to so-called MRP II systems by adding tools for sales planning, customer order processing, capacity planning and production scheduling. In 1990's they functionality was further enhanced by other functionality such as finance.

At first ERP systems were so-called legacy systems that were developed in-house. They typically consisted of separate systems optimized for each department's tasks of the company. Also the data of each department was in separate data store so, for example, the finance department may have had a different set of customer data than the sales department. As a result, a sales person was unable to access the accounting database to view billing data or the shipping department database to check whether an order had already been processed. Not only does this duplicate data input take twice the time, but it also significantly increases the chance for data entry errors. Such isolated framework hampered the productivity, speed and performance of the overall organization and in competitive business environment it was essential to find a way to avoid this inefficient division of information. The answer was integration of different functions into one single ERP system with one unified database. SAP played a pioneering role in this development. (Muir & Kimbell 2010, 41.)

Today, ERP system is an enterprise-wide information system designed to coordinate all the resources, information, and activities needed to complete business processes. An ERP system is bought from a vendor and this differentiates it from in-house developed systems. Typically, ERP software is made up of many software modules. Each module mimics a major functional area (business function) such as production, sales, distribution, finance, accounting, human resources, of an organization. The data of various business functions is integrated in a single database. The common database can allow every department of an organization to store, process, monitor and report information in real-time. The information should be reliable, accessible, and easily shared. (Storani 2008) The real-time operational data that ERP systems provide enable managers to make better decisions and improve responsiveness to customer needs (Sumner 2005; 2).

In principal, the modular software design makes it possible for a company to select the modules they need. Most vendors' ERP software is flexible enough that you can install some modules without buying the whole package. Many companies, for example, may want to install a finance or HR module first and leave the rest of the system for later (Olson 2009, 5). Furthermore, the modularity enables to mix and match modules from different vendors, and even add new modules of their own to improve business performance, but according to Sumner (2005, 2) most companies implement a single ERP package rather than selecting different modules from different ERP vendors. Monk (2009, 29) also points out that if a company uses modules from different vendors, additional software programming must be done to get the modules to work together, for instance using batch data transfer processes that are performed periodically. In this case, the company no longer has accurate data available in real-time across the enterprise. Therefore, a company's level of data integration is highest when the company uses one vendor to supply all of its modules. On the other hand, larger enterprises may want to use modules from different vendors and possibly integrate other third-party software to their ERP system. The scale of requirements that must be fulfilled by these software solutions depends on the size of the enterprise and on the diversity of its business activities (Muir & Kimbell 2010, 37).

Formerly, ERP systems were used in larger and more industrial types of companies. But as ERP systems evolved their popularity soared in the nineties. This was partially due to the massification of the computer networks and the dissemination of the low cost client-server architectures but also because more companies saw benefits of an integrated enterprise-wide system and come to realize that having such system may help them sustain their competitive advantage and to be ahead of other players in the market . However, today ERP is used in wide variety of industries and organizations to support their operational processes. In fact, ERP systems are used in almost any type of organization – large or small.

Most ERP vendors offer so-called best practices, which are defined as "simply the best way to perform a process" (Sumner 2005, 2). These are generally evolved over the years into a set of generally accepted best practices specific to particular business, e.g. if your company is in the textile business, there would be a set of documented best practices for your kind of business. Best practises are not static but will keep evolving into the future in response to changes in the business environment. A comprehensive ERP solution, like SAP ERP, would be expected to enable your company to implement these best practices in its core business processes. (Ittoolbox 2011.) So when an organization acquires packaged software, it defines processes and practises for the system. This is fundamentally different from traditional in-house system developing, in which the systems designer defines new business requirements and implements software to conform to these requirements. Therefore, one of the fundamental decisions in implementing an ERP package is whether to change (re-engineer) the organization's business processes to fit the software or to customize the software to fit the organization's business practices. (Sumner 2005, 42.) The re-engineer business processes around the best practices of the ERP system happens basically by configuring many changeable parameters that are built in the system in a way that they would match the organizations requirements the best possible way. For example, an organization can select whether it wants to employ LIFO or FIFO inventory accounting method, to recognize revenue by geographical unit, product line, or distribution channel and whether to pay for shipping costs when a customer returns a purchase. Configuring an ERP system is largely a matter of balancing the way the customer wants the system to work with the way it was designed to work. (Wikipedia 2011a.) The configuration settings should be carefully planned prior to implementation, because once the system is in place, trying to configure while retaining data integrity is expensive and time consuming (Monk 2009, 35). In customizing, new features (e.g. fit to some local regulations) or interfaces are added to the standard ERP software, this means rewriting part of the code. ERP software should only require none or only little customization in order to fit to the requirements of an organization or the idea and benefits of fully functional off-the-shelf software package disappear. Even if customizing offers the potential to obtain competitive advantage compared to companies using only standard features, it can be complicated and expensive, and delay implementation. Nevertheless, customization offers the potential to obtain competitive advantage vis a vis companies using only standard features. (Wikipedia 2011a.)

Not too long ago, companies still had limited options when it came to ERP vendors and products. ERP or enterprise resource planning software options were only developed by less than a dozen of vendors and product offerings were slim. The dominant vendors back then were only three, namely SAP, Oracle and IBM targeting their products to large companies. (ERP 2011a.) All large vendors nowadays provide similar functionality, with some having strengths in certain areas (Monk 2009, 26). In the past 15 years, many new players have entered the market to target small and midsized companies and to fill in the industry specific demand. Also the biggest ERP vendors have increased their product offerings to capture this huge market opportunity. Today the ERP market is deeply fragmented. For small and medium sized enterprises the number of vendors is around 10.000 globally (3gerp 2011). There are even open source ERP solutions available. ERP software runs on various operating systems and hardware, from UNIX, Linux, and Windows to mainframe. Different ERP programs base on different programming languages depending on vendor. (Muir & Kimbell 2010, 34.)

Like all technology, ERP software is constantly changing. In today's global economy to stay competitive companies have to constantly analyze, redesign and automate business processes, and have information available as quickly as possible to make critical decisions in a short amount of time. ERP vendors have responded to this trend by extending ERP's capabilities into more areas like decision support, management reporting and data mining. Internet connectivity is another area in which ERP capabilities are expanding. ERP vendors continue to improve software and internet connections that integrate a business' internal operations, while also integrating the business with its dealers, vendors, and customers. (Monk 2009, 41-42.) In fact, several other systems have been pushed to the market, such as e-commerce, CRM (Customer Relationship Management), SCM (Supply Chain Management) and BI (Business Intelligence), which attempt to solve the information needs that appear at the boundaries of the organizations and also to provide information and strategic metrics for the upper management, in a friendly way. Web services, or as it is frequently called service oriented architecture (SOA) is also one area that is receiving a lot of attention. Web services enable different systems to exchange data without complicated interface links over the web. Another trend in ERP development and use involves vendors making the software available to companies over the internet, known as hosted ERP or Web-deployed ERP. In this kind of case, an ERP vendor acts as application service provider (ASP) for several client companies. When a company chooses to buy this kind of hosted service to use an ERP system, the software is not purchased by or installed on a server at the client company. Instead, it is maintained on the vendor's host computer, where clients access it through an internet connection. Thus, an ASP can provide ERP software with a much lower start-up cost, making it possible for smaller companies. (Encyclopedia of Small Business 2007.) Software offered using an ASP model is also sometimes called software as a service (SaaS).

To summarize, an ERP system is a large commercial business software package that enables an organization to automate and integrate its business processes to combine information from various sources into a single database and access that data in a real-time environment. The result of having a single integrated system is that it will increase the organizations efficiency by eliminating many redundant activities that would be required to keep different systems synchronised.

2.1.2 SAP ERP 6.0

The SAP® ERP application is one kind of an enterprise resource planning software package manufactured by SAP AG. Thus, all the functions and advantages of ERP discussed earlier are also applicable to it. It is targeted mainly to requirements of mid-size and large organizations in all industries and sectors. SAP is a global leader in ERP software, headquarters in Germany. SAP's applications are used by a significant number of companies and other organizations in more than 120 countries. (SAP 2010b.) But quite often, when people refer to SAP, they usually mean SAP ERP software.

SAP ERP is the core application in SAP's more comprehensive offering for large corporations - the SAP® Business suite. SAP ERP, as the entire Business suite, is modular, which means that individual applications can be purchased, installed, and run separately, but that all extract data from the common database (Monk 2009, 24). Other possible modules included in the Business suite are Customer relationship management (CRM), product lifecycle management (PLM), supply chain management (SCM), supplier relationship management (SRM). The latest version incorporates add-on software packages for 25 industries. (Muir & Kimbell 2010, 34.) Other software shipped by SAP includes SAP® BusinessOne for small-sized businesses and SAP® All-in-One for medium-sized businesses and SAP® Business ByDesign™ (SaaS). (Sens 2008, 4.)

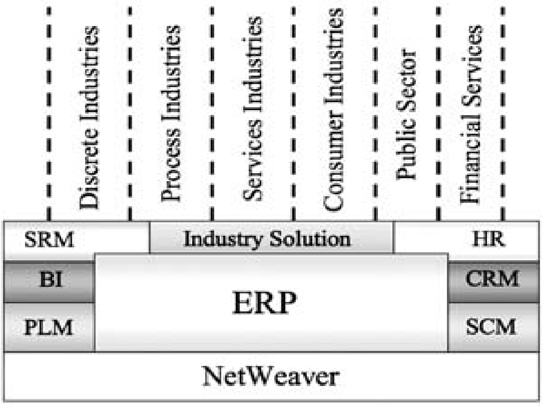
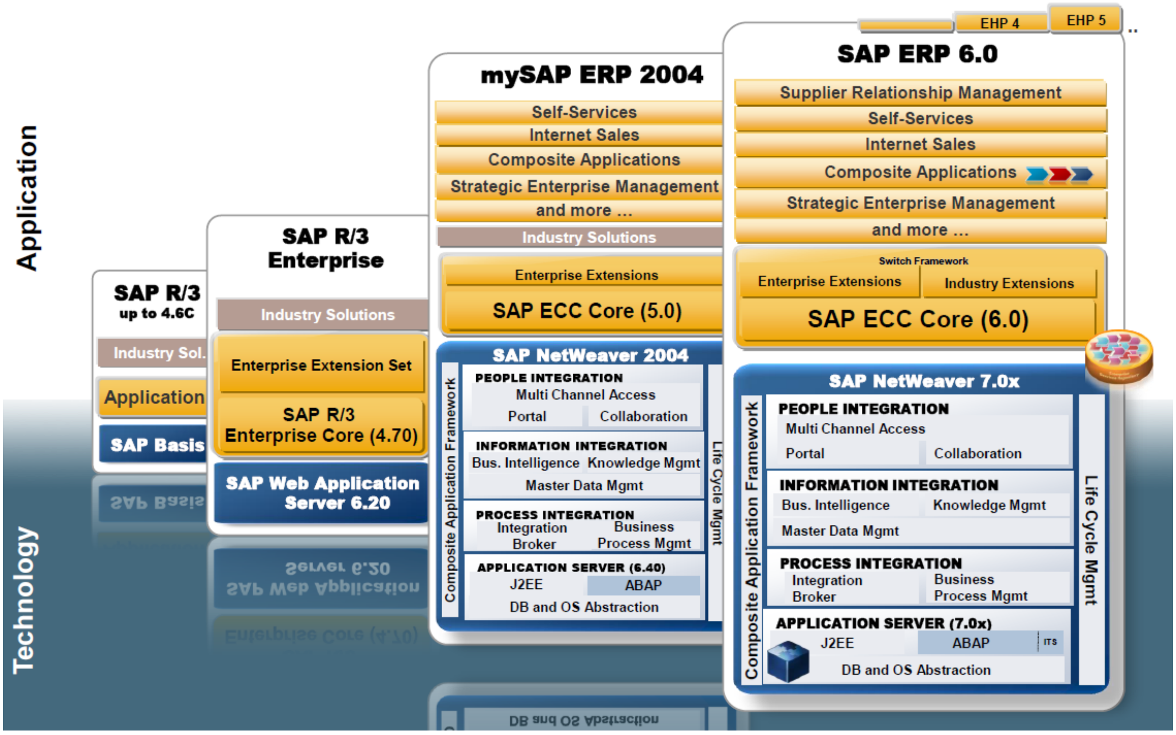


Figure 1: SAP business solutions and components. (Sens 2008, 3).

The applications run on the SAP NetWeaver® technology platform that supports both ABAP™ (SAP's internal programming language) and Java programming languages. However, the NetWeaver platform supports open architecture, which means that its products can be run with a mix of products from other vendors. (Muir & Kimbell 2010, 34.) In fact, the NetWeaver platform serves the same role as SOA (Monk 2009, 42). The open architecture also makes it easy for companies to integrate their hardware products, such as bar code scanners and mobile phones (Monk 2009, 25). At the heart of the open-standards approach to SOA promoted by SAP is the concept of enterprise services - the break down processes into smaller process steps for reusable enterprise services that can easily be adapted to changing business needs (SAP Support Portal 2011b, 14). With NetWeaver SAP offers integration tools like Process Integration PI and Business Intelligence BI (Sens 2008, 4).

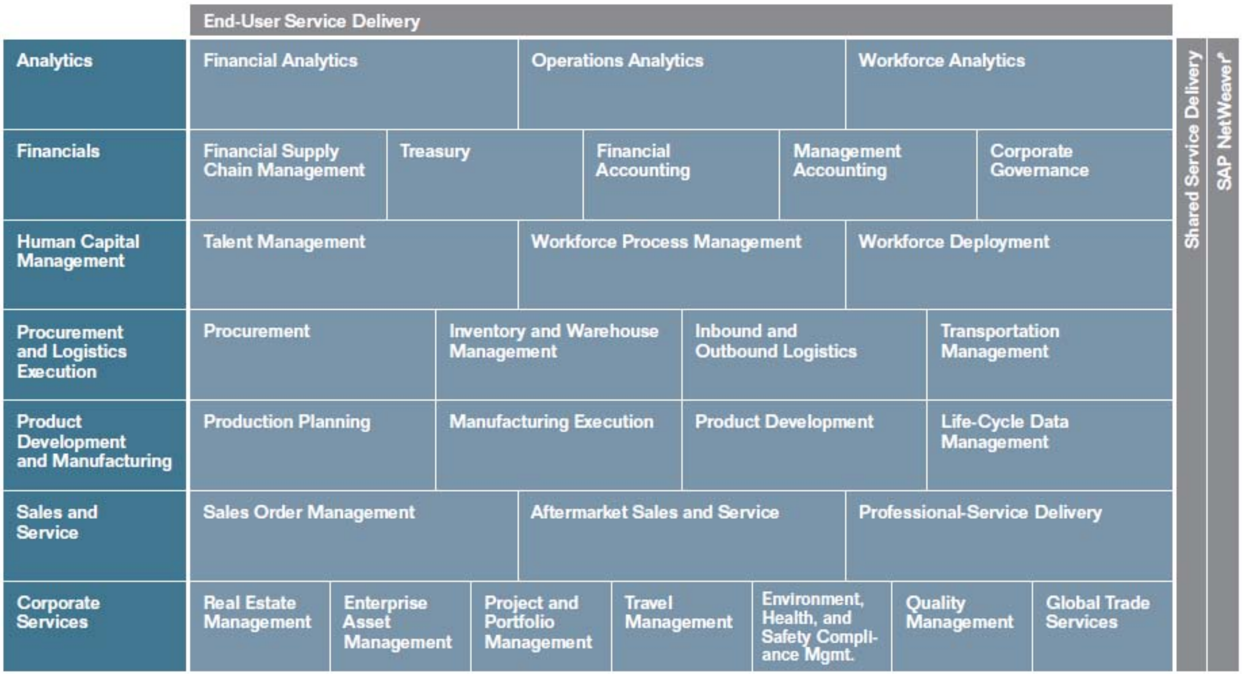
SAP ERP 6.0 is the latest technical upgrade to existing SAP ERP systems and it draws more than 35 years of experience with more than 43 000 SAP customers worldwide (Riedel 2010, 17). SAP released its first software R/2 for mainframes already in 1979, but the real success started with the second R/3 client-server software in 1992. Many companies are still currently running earlier versions SAP R/3 4.6c, mySAP and ECC5. Customers running SAP R/3 4.6c will either need to upgrade to ERP 6.0 before the end of 2013 or extend their agreements on more expensive customer-specific maintenance. SAP had planned to end the support already at the end of 2010, but as a result of pressure from customers support was extended until March 2013 (Computerworld UK 2010). The names of the individual SAP products are subject to changes quite often. It is for this reason that names like R/3, mySAP, ERP, and ECC are used for the same product. The same applies for XI and PI, for BW and BI and for SAP NetWeaver 2004S and NetWeaver 7.0. (Sens 2008, 2.) SAP ERP is differing from R/3 mainly because it is based on SAP NetWeaver that supports Java and SOA.



© Copyright 2010. SAP AG. All rights reserved.

Figure 2. Evolution of SAP R/3 to SAP ERP 6.0. (Source: SAP Support Portal 2010a.)

Recently, managers have begun to think in terms of business processes rather than business functions. Business processes can be defined as ”a collection of activities that create an output that is of value to a customer. Creating the output might involve activities from different functional areas" (Monk 2009, 236). Apparently, as response to this SAP have introduced with SAP ERP a new 'solutions' concept instead of 'modules' still used in earlier version R/3, that were more business function oriented. However, the solution concept is quite new so SAP modules are still described in many books and also the conversation turns into the traditional modules very quickly: Financial Accounting (FI), Controlling (CO), Materials Management (MM), Sales and Distribution (SD), Production Planning (PP), Plant Maintenance (PM), Project System (PS), Investment Management (IM) and Quality Management (QM). SAP ERP's new solutions map is illustrated below. Figure 3 provides an overview of the solution areas that SAP ERP comprise, each covering several business processes.



End-User Service Delivery

Analytics	Financial Analytics	Operations Analytics	Workforce Analytics				
Financials	Financial Supply Chain Management	Treasury	Financial Accounting	Management Accounting	Corporate Governance		
Human Capital Management	Talent Management	Workforce Process Management	Workforce Deployment				
Procurement and Logistics Execution	Procurement	Inventory and Warehouse Management	Inbound and Outbound Logistics	Transportation Management			
Product Development and Manufacturing	Production Planning	Manufacturing Execution	Product Development	Life-Cycle Data Management			
Sales and Service	Sales Order Management	Aftermarket Sales and Service	Professional-Service Delivery				
Corporate Services	Real Estate Management	Enterprise Asset Management	Project and Portfolio Management	Travel Management	Environment, Health, and Safety Compliance Mgmt.	Quality Management	Global Trade Services

Shared Service Delivery

SAP NetWeaver®

Figure 3: SAP ERP 6.0 solution map. (Source: SAP 2011)

Even though SAP ERP is packaged software and it covers nearly all possible business operation, it still needs to be configured after installation to meet the requirements of each customer. It has been said that SAP ERP is very flexible and offers numerous configuration options in implementation phase, but changing the settings later on is very troublesome. That is why the configuration settings should be carefully planned prior to implementation. SAP ERP's design incorporates best practises. Best practises comprise expert knowledge from various industries or relating to specific processes in an enterprise. SAP makes these best practices available in the form of technical and business-related documentation and pre-configured content in SAP systems. These can be regarded as "turnkey" resources that facilitate and accelerate the implementation process. (Muir & Kimbell 2010, 49.) Moreover, companies that implement best practices can reduce time needed for configuration, documentation, testing and training. In addition, best practices reduced risk by 71% when compared to other software implementations. (Selchert 2004, 8-10.)

In SAP's terminology the configurations are referred as customizing settings, e.g. standard predefined features like payment terms or material groups the client wants to use in their system or setting organizational structure. Typically settings are customized in SAP ERP's Implementation Guide (IMG) when an SAP ERP system is implemented. No standard code or software functionality is affected customizing these settings, therefore they don't affect the upgrade project scope. Only in case an organization wants to change current functionality or take new functionality in use offered by a new release, it may have to adjust the customizing settings during upgrades. (Riedel 2009, 81.) Customer-specific configurations are called as custom developments and they are prefixed with the letter Z or Y. This lets anyone working on the system know that this is a copy of standard configuration and has been changed for the customer's needs. An example of such a Z configuration is customization of the standard Sales Order caller OR. This OR has default settings that trigger certain functions within SAP. If it can be customized to meet the client's needs. This is done typically by copying the OR and renaming it to ZOR and then customizing it according the specific needs. SAP will not overwrite any of these document types during upgrades, but these would require extensive testing in the context of a new version upgrade (Riedel 2009, 82).

In top of the flexible configuration capabilities in SAP ERP, many customers still take the advantage of modification possibilities SAP ERP offers to meet their business requirements. Modifications are changes to SAP code which are initiated and developed by SAP user organizations. SAP discourages modifications and they are generally not supported under service agreements and SAP will not support any problems they cause with the system. Modifications are overwritten during upgrades and the ones that are still needed after the upgrade have to be adjusted (Riedel 2009, 83). This is why maintenance and upgrading of heavily modified SAP ERP system is expensive and difficult (SAP Support Portal 2011b, 9). This matter will be discussed later in this research.

An organization's SAP system landscape is part of a much larger IT landscape that might include all kinds of different third party software, such as office desktop software, emails, web applications and other interfaces. SAP system landscape in turn can consist of SAP ERP system, but it can also contain various other interconnected systems such as SCM and CRM. (Sens 2008, 231) SAP ERP, like all SAP systems running on SAP NetWeaver, is based on the three-tier architecture concept. This concept is specific to the architecture of one system in the landscape and should not be confused with the 'three system landscape' concept explained in the next paragraph. A three-tier architecture system consists of a database layer, an application layer and a presentation layer. The database layer serves to store and retrieve data for the application layer. The application layer provides business rules, functions, application logic and serves as a data receiver/deliverer from/to the database servers and operating system. The presentation layer usually is graphical user interface (GUI) or browser executed on the client (e.g. PC or mobile device) for data entry or accessing system functions. (SAP SDN 2008, 3-4.). All the common databases and operating system combinations are supported by SAP.

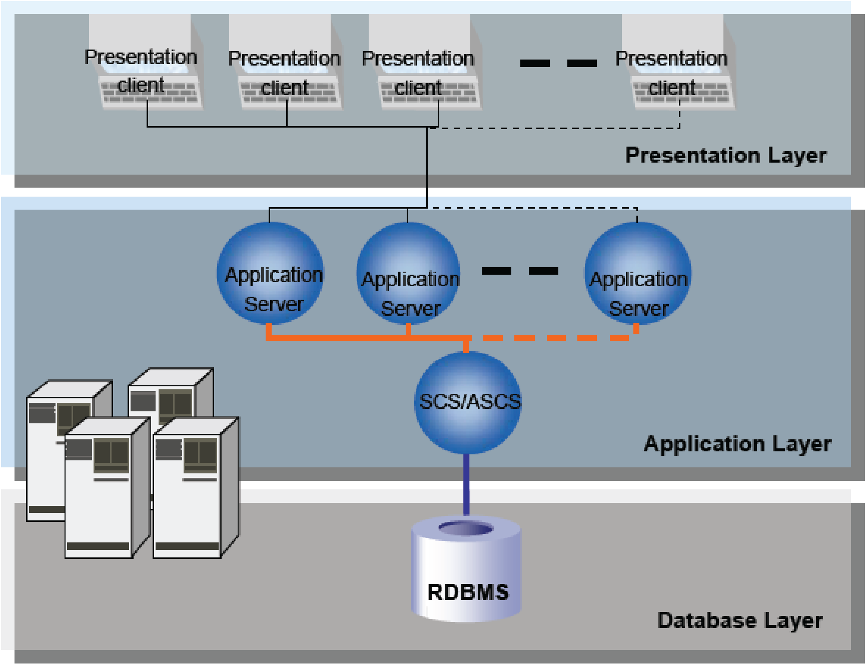
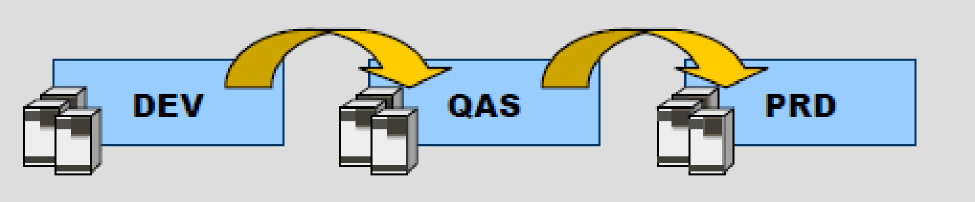


Figure 4. Three-layer architecture. (Source SAP SDN 2008, 4)

For the application layer SAP advises to use of three-system landscape; DEV (for development and customizing), QAS (for quality assurance and testing), and PRD (for productive “live” system). These are often referred also as instances. The system landscape is basically the set-up or arrangement of your SAP servers. This three-system landscape allows installation, customizing and testing of upgrades in a fashion that is isolated from the production system. The landscape may contain even more systems, e.g. for training. Thus, the training system would be a complete replica of the productions system, enabling the employees to experience the system and use it without the fear of pulling it down or making errors. Large organizations very often have also a so called sand box system that allows anyone to play around with the system without worrying too much about what will be the impacts. (Portougal & Sundaram 2006, 114-115.) Moreover, each of these instances may consist of more than one client. Clients are typically identified by a three-digit number, such as 100, 110, 200, 210, etc. Each client within an instance share the SAP ERP program, but will have separate master records and its own set of data tables (client dependent data). For example, if a sales order is created in client 100 and another sales order is created in client 110, they are totally independent. Similarly a customer master created in client 110 does not mean that the same customer master would exist in another client unless explicitly created. Typically production instance have only one client. Some customizing settings will affect all of the clients in an instance (client independent settings), but some settings will only affect the client in which it has been performed (client dependent settings). Customizations are typically transported to the production system in ”pipeline”. They are first performed in the DEV, and then they are transported to the QAS, and once tested in the test system, they are moved over to the production. In order to transport configuration changes to another SAP instance, the changes must be included in a “change request” that is then released to the receiving system. (ERPDB 14.5.2009.)



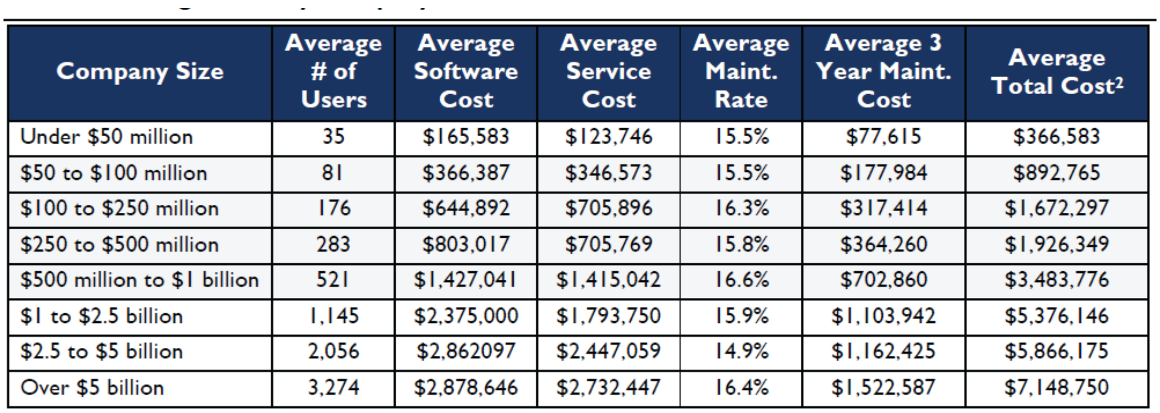
© Copyright 2008. SAP AG. All rights reserved.

Figure 5. A typical SAP ERP three-system landscape and order in which changes are transported. (Source SAP SDN 2008, 2)

SAP ERP system landscape design alters during the period of the upgrade (Sens 2008, 215). More complex the landscape, more time and effort is required to perform the upgrade. Each system adds an additional upgrade. Therefore, coordinating that all systems in the landscape are correctly and appropriately upgraded is very important. (Riedel 2009, 62-63.) More about this in Chapter 2.5.1.

The initial investment to acquire and implement an ERP system is substantial. In addition to the cost of the software, many companies discover that they need to buy new hardware to accommodate such powerful programs. According to Monk (2009, 31) for a fortune 500 company, software, hardware, and consulting costs can easily exceed $100 million and full implementations of all modules can take years. But even after the system is up and running, the costs continue to mount as the ERP system needs to be maintained and upgraded. Large companies can spend $50 million to $100 million on upgrades (Monk 2009, 31). As the company grows, the number of users goes up, along with the total cost of software and services; one can naturally expect a correlation between size of the ERP deployment and costs. Aberdeen's (2008, 3) research illustrates this correlation among midsize companies (see **Error! Reference source not found.**)

Table 1. Average cost of ERP by company size.



Company Size	Average # of Users	Average Software Cost	Average Service Cost	Average Maint. Rate	Average 3 Year Maint. Cost	Average Total Cost²
Under $50 million	35	$165,583	$123,746	15.5%	$77,615	$366,583
$50 to $100 million	81	$366,387	$346,573	15.5%	$177,984	$892,765
$100 to $250 million	176	$644,892	$705,896	16.3%	$317,414	$1,672,297
$250 to $500 million	283	$803,017	$705,769	15.8%	$364,260	$1,926,349
$500 million to $1 billion	521	$1,427,041	$1,415,042	16.6%	$702,860	$3,483,776
$1 to $2.5 billion	1,145	$2,375,000	$1,793,750	15.9%	$1,103,942	$5,376,146
$2.5 to $5 billion	2,056	$2,862097	$2,447,059	14.9%	$1,162,425	$5,866,175
Over $5 billion	3,274	$2,878,646	$2,732,447	16.4%	$1,522,587	$7,148,750

Another research on midsize companies conducted by CFO and Agresso (2009, 3) estimates similarly that a typical midsize company may spend an average of more than $1.2 million each year to maintain, modify, and update its ERP system. However as the total price of go up as companies grow, cost per user generally should scale down due to bargain power. According the Aberdeen's table above, small organizations pay for 3 year maintenance nearly five times more per user ($2218 vs. $ 465) than larger organizations (Aberdeen includes upgrades in the 3 year maintenance costs).

2.2 Upgrading ERP software - general

2.2.1 Definition of upgrade in the ERP life cycle context

Like any information system, the life cycle of ERP represents various stages which the software passes through. In information systems literature, ERP life cycle is often referred to the systems development life cycle (SDLC) (e.g. by Sumner 2005, 41-42), that nearly all information technology professionals are familiar with. Stages in it vary based on methodology, but mostly include planning and requirements definition, analysis, design of the new system, implementation, and post-implementation support such as maintenance and security. (Peslak & Subramanian & Clayton 2008, 25). The SDLC model is mainly based on software development and thus covers the stages from the view point of the software vendor that is responsible of developing and maintaining its standard ERP software for a large customer base. However, the commercial off the shelf software ERP is suggested to have a life cycle of its own, because the development (adaption: configuration and customization) and maintenance activities apply to the client's installed version only (Peslak & Subramanian & Clayton 2008, 25). Therefore, a number of researchers present models specifically for the life cycle of ERP systems (Sullivan L. 2009, 33).

The Hedman's (2003, 9) model is presented here next, because it illustrates well this double perspective. His model is based on various life cycles developed by other researchers. The figure below has been modified by adding the activities related to each phase that Hedman (2003, 62) names later in his dissertation. The life cycle on the developer level includes the stages when the standard version of the ERP system is produced. On the client level the stages of adopting the system into the client's organisation are included.

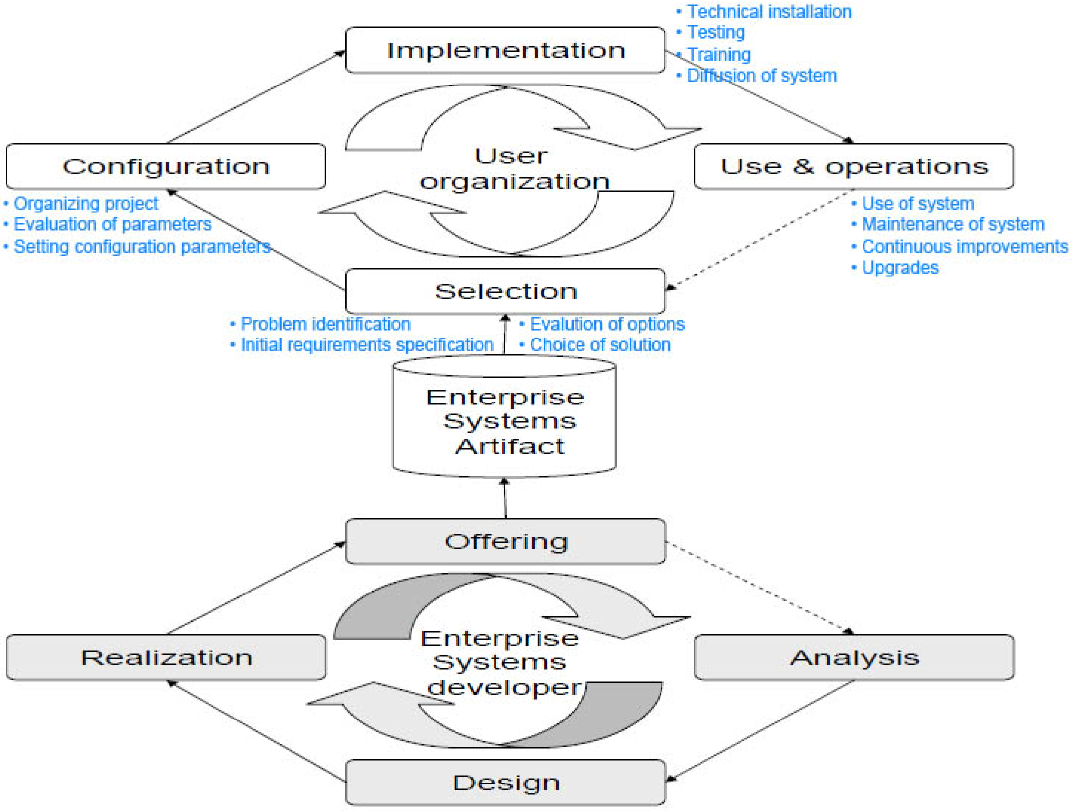
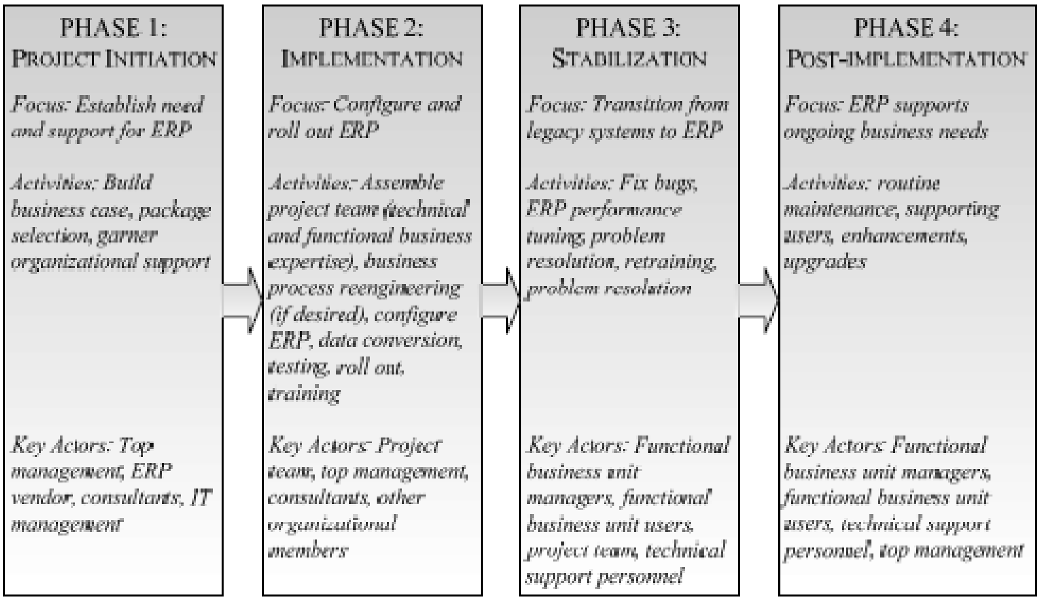


Figure 6. Hedman's (2003, 9 & 62) vendor-side and client-side life cycles of enterprise system.

This thesis paper focuses on the upper client-side ERP life cycle and there on the last post-implementation phase that is referred as 'Use & Operation' by Hedman. Within this paper the term post-implementation will be used in order not to confuse the reader. According to Hedman (2003, 63) this last phase "involves the use and administration of the system until it is terminated and replaced by another solution". However, Hedman (2003, 8) also confirms what was said earlier about ERP being in an ongoing state of change. He says that ERP should be continuously evaluated and improved in the post-implementation phase and therefore different phases in the life cycle should iterate refining the initial solution. The post-implementation phase includes the implementation of additional functionalities including integration with other information systems, e.g. data warehouse, customer relationship management, supply-chain management, and e-commerce (Hedman 2003, 63). In his model upgrade activities happen during the post-implementation phase.

Some early ERP life cycle structures presented ended with the implementation phase and therefore presented a limited view of the ERP life cycle without considering the on-going maintenance, support, and continuing organizational change that are characteristic of ERP projects. One of the first models that emphasized the importance of the post-implementation phase in the ERP life cycle was a four-phase Enterprise System Experience Cycle by Markus and Tannis (2000). It consisted of four distinct phases: chartering phase, project phase, shakedown phase, and onward and upward phase. In this model the post-implementation phase is divided into two separate phases (a) the shakedown phase represents the period between go-live and when normal operations become stable and routine on the new ERP system and (b) the onward and upward phase covers the remaining operational life of the ERP system until it is upgraded or replaced. Achieving expected results, improving business processes, and routine maintenance and upgrades of the ERP system are success indicators associated by Markus & Tannis with the onward and upward phase. (Sullivan 2009, 31-37.) Following the conceptual framework of the Markus and Tanis model, the focus of this thesis is on upgrading activities within the onward and upward phase.

In one later study, Worrel (2008, 6-7) has adapted Markus and Tannis model by naming the phases differently (Figure 7), but the main concept of the separate phases is still the same. According to him, the final stage of the ERP lifecycle, post-implementation, concentrates on issues that occur as a result on ongoing operations and use of the ERP to support business transactions and decisions.



PHASE 1:
PROJECT INITIATION

Focus: Establish need and support for ERP

Activities: Build business case, package selection, garner organizational support

Key Actors: Top management, ERP vendor, consultants, IT management

PHASE 2:
IMPLEMENTATION

Focus: Configure and roll out ERP

Activities: Assemble project team (technical and functional business expertise), business process reengineering (if desired), configure ERP, data conversion, testing, roll out, training

Key Actors: Project team, top management, consultants, other organizational members

PHASE 3:
STABILIZATION

Focus: Transition from legacy systems to ERP

Activities: Fix bugs, ERP performance tuning, problem resolution, retraining, problem resolution

Key Actors: Functional business unit managers, functional business unit users, project team, technical support personnel

PHASE 4:
POST-IMPLEMENTATION

Focus: ERP supports ongoing business needs

Activities: routine maintenance, supporting users, enhancements, upgrades

Key Actors: Functional business unit managers, functional business unit users, technical support personnel, top management

Figure 7. ERP Lifecycle by Worrel (2008, 7).

According to Otieno (2010, 26) the research that is most closely related to ERP software upgrade is software maintenance. Kidd (2001, in Otieno 2010, 26) for instance defines software maintenance as "the activities performed on software after the program has been installed". Otieno says that both ERP upgrade and maintenance serve a similar functional purpose and main commonality is the importance of responding to users' business needs. This is surely why some researchers, like Peslak & Subramanian & Clayton (2008, 27) and Wenrich & Admand (2009, 61), call the final phase as maintenance phase. There exists also maintenance models, such as standards published by the IEEE (Institute of Electrical and Electronics Engineers) and SEI (Software Engineering Institute), but they are designed for internally maintained software and seem not to be fully applicable to large packaged off-the shelf ERP software (Ng & Gable 2010, 74; Ng & Gable & Chan 2003b., 234; Wenrich & Ahmad 2009, 60). According to Ng and Gable (2010, 76) maintenance models e.g. do not take into consideration the vendor role in maintaining the system. In addition, when making ERP upgrade decisions, an ERP client-organization must consider not only its internal organizational needs, but also future vendor maintenance support, upgrade compatibility, and business vision. Weinrich and Ahmad (2009, 61) say that they lack the interplay of vendor-initiated change and user organization change requests. Moreover, according to them maintenance models do not include steps for weighing immediate implementation through customization versus waiting for the next up-grade, or whether to re-apply previous client specific customizations after the installation of new upgrade that overlay customizations or replace them with the delivered corresponding functionality.

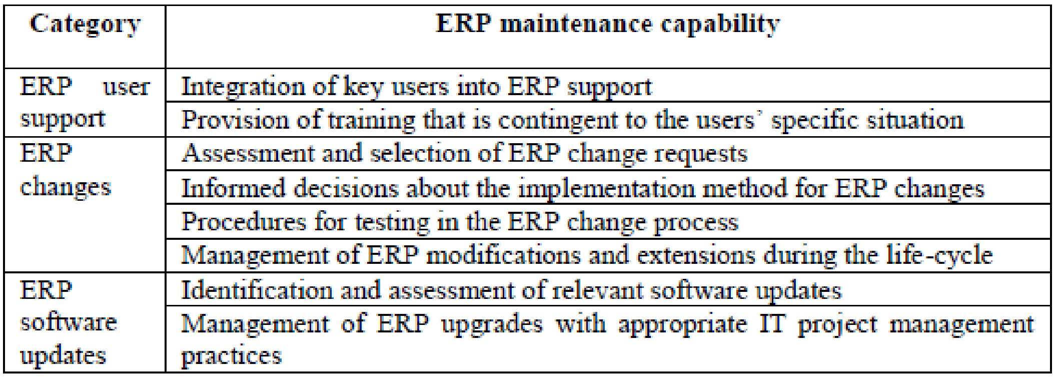
In this research we consider upgrades as part of the maintenance activities, but adhere to using the term post-implementation about the last phase in the ERP life cycle. The reason is that the phase after the initial ERP implementation is generally referred as a post-implementation phase in the literature, even if in various models named differently. And with reference to Ng & Gable & Taizan (2002, 100) ERP maintenance is defined as "post-implementation activities related to the packaged application software undertaken by the client-organization from the time the system goes live (i.e. successfully implemented and transported to the production system) until it is retired from an organization's production system".

According to Ng et al. (2002, 100), ERP maintenance activities include:

- Responding or handling user-support requests (initiated by an ERP-using organization's system users),
- Implementing internal change-requests (initiated by an ERP-using organization's system users and IT staff),
- Upgrading to new versions/releases (introduced by the vendor), and
- Performing patches (support provided by the vendor).

Hecht (2011, 188) has conducted recently a comprehensive review of the ERP post-implementation literature. Hecht states that one of the key IT services provided by today's IT organizations in the ERP post-implementation phase is the maintenance of existing ERP systems. According to her ERP maintenance covers various activities to support and extend and existing system and those include the handling of user requests and user training, the implementation of changes to the existing functionality of the system, or the implementation of patches and software updates provided by the ERP vendor. So her interpretation of maintenance activities and upgrade as one of the key activities in the post-implementation phase is very similar to Ng et al. above. The table below present Hecht's preliminary findings of three ERP maintenance categories, in which upgrade is included as a type of ERP software update.

Table 2. ERP maintenance capabilities derived from the ERP post-implementation literature (Hecht 2011, 189).



Category	ERP maintenance capability
ERP user support	Integration of key users into ERP support
	Provision of training that is contingent to the users' specific situation
ERP changes	Assessment and selection of ERP change requests
	Informed decisions about the implementation method for ERP changes
	Procedures for testing in the ERP change process
	Management of ERP modifications and extensions during the life-cycle
ERP software updates	Identification and assessment of relevant software updates
	Management of ERP upgrades with appropriate IT project management practices

Next clarification to terminology, because as seen above and often in the literature the terms referring to upgrade are used interchangeably, without defining them more precisely.

In standard language, ‘upgrade’ and ‘update’ mean broadly the same “to raise to a higher grade or standard”, “improve the quality” or “bring up to date”. In that sense, the last three maintenance activities specified by Ng et al. or last two maintenance categories by Hecht above could be considered as upgrading activities, because they aim to improve and enhance the ERP software characteristics. But in relation to software the ‘upgrade’ specifically refers to a ‘version upgrade’. Also referred as release, if discussed from the vendor-perspective. It means implementing of a new standard version provided by the vendor that should provide added enhancements over an earlier version, to the client’s installed version. Further, upgrade is a new version from the same vendor (as the installed version) (Ng & Gable & Chan 2003a, 1053), otherwise it would be a new implementation.

If an organization decides to improve its ERP system by integrating other modules or information systems such as e-commerce to it, it can be considered as a maintenance activity. However, it is not an upgrade but an extension that falls under the ERP changes in Hecht’s category above. Also Zhao (2007, 10) points out that extension refers to changes made by “add-ons,” third party vendor “bolt-ons,” and extensions to current systems. In addition, vendors provide also smaller system upgrades that are generally referred as patches (as above by Ng et al.), updates, support packages, minor releases, enhancement packages, bug fixes or hot fixes, but they are much simpler to implement than a version upgrade. This kind of maintenance is often referred as ‘routine’ or ‘traditional’ maintenance in the literature. Otieno (2010, 27) says that the major difference between version upgrade and routine upgrade is that “ERP system upgrades usually replace the previous version with a new version, whereas maintenance usually targets specific functions that need to be corrected or perfected”. According to Ng & Gable & Chan (2003, 1046-1047), ”the upgrade process is very similar to the patch maintenance procedure, the main differences being that upgrade requires more thorough planning and business justification, more effort for impact analysis and reapplication of previous modifications or user-enhancements (if the new version has not incorporated the required functionality), longer time to complete, more money and resources to implement, and serious consideration of potential system downtime".

One important point is that since the upgrades are provided by the vendor, they are vendor-introduced and at the same time applicable to a large customer base. This fact differentiates them from the internally initiated change requests (cf. Ng et al. 2002 internal change requests and Hecht's ERP changes) that aim to fit ERP in organization's own business environment and that can result in client-specific code customizations in client's installed version. Modification in turn can complicate the implementation of vendor's standard version upgrade. This topic will be discussed later in this research. This research focuses specifically on SAP ERP version upgrade.

To summarize the above said, there are a number of ERP lifecycle models and they differ slightly from model to model. However, there is always one stage in the recent models that include upgrading of the software. In this research we preferred to call it the post-implementation phase, because the phase immediately after the initial ERP implementation is generally referred as a post-implementation phase in the literature. It begins immediately after go-alive (when the system is taken into use) and continues until the organization removes the ERP software from use (Escobar & Toma 2007, 9; Sullivan 2009, 8). Upgrade is one of the important maintenance activities in the post-implementation phase of the ERP software lifecycle. Upgrade is a new standard version of the software packaged provided by the ERP vendor issued to a large customer base that should provide added enhancements over an earlier version. The ERP upgrade activities discussed in this paper is from the ERP-using organization's (client) perspective, where the activities pertain to the client's installed version. Based on the above said, the compact definition of ERP upgrade is formed as follows:

> An ERP upgrade is a maintenance activity in the post-implementation phase of the ERP life cycle in that a software package provided by the vendor replaces the client's installed version with a newer improved version from the same vendor.

In recent decades, companies across the world have implemented ERP software (Salmeron & Lopez 2010, 1952) and ERP market today is considered to be mature (Ng &

Gable 2010, 65; Worrell 2008, 2). With the maturing use of ERP systems most companies are now in post-implementation phase in their ERP life cycle. It means that organizations are forced to continuously assess and improve their maintenance capabilities in order to operate ERP systems efficiently, to ensure a high quality and an effective usage of the ERP system (Hecht 2011, 188).). It means that they need to enhance or upgrade their ERP solution in order to keep the system running smoothly in evolving technical environments and improve their business processes. Therefore, the topic is very current and the post-implementation phase is gaining more importance (Esteves & Bohórquez 2007, 420 in Hecht 2011, 187), but still the volume of ERP studies concentrates mainly on the pre-implementation and implementation phases, with little discussion of post-implementation (Zhao 2007, 3; Law & Chen & Wu 2010, 297), even though the high failure rates of ERP projects, even after a successful system installation, indicate that the post-implementation stage is very critical for the success of ERP projects (Kouki 2009, vii). It's after go-live that determines if the ERP system "makes or breaks" (Martinsen 2010, 5). Therefore, understanding post-implementation of ERP will help organizations extend their software's life (Law &, Chen & Wu 2010, 297) and succeed long after the ERP implementation (Zhao 2007, 3). Even less research attention has been given to ERP software upgrade (Ng & Gable & Chan 2003b., 234–235; Zhao 2007, 4, 9; Worrell 2008, 2; Otieno 2010, 12; Martinsen 2010, 6), despite the fact that it is one of the key activities in the post-implementation phase and organizations spend a significant amount of money on each ERP upgrade project (Zhao 2007, 3-4). Otieno (2010, 5) says that the absence of information about ERP upgrades is somewhat surprising given the general consent that upgrading is essential for companies to survive and compete. Zhao (2007, 4) believes that one possible reason could be that upgrade is perceived to be a smaller project (compared to the first time ERP implementation), and another reason could be that little theory has been developed regarding the topic of ERP upgrade. This research hopes to narrow this gap.

2.2.2 Initial ERP implementation vs. ERP upgrade projects

In the previous chapter the upgrade was defined as one of the important activities in the ERP software lifecycle, but how does it differ from the initial ERP implementation[1]? Main difference of course is that while first time ERP implementation happens only once, ERP upgrade can occur many times after the first ERP implementation. According to Betty and Williams (2006, in Zhao 2007, 78) the most common thought about ERP upgrade is that it is similar to initial ERP implementation but with smaller scope. Therefore, many organizations have underestimated the difficulty of upgrade projects. This caused project delays or even unsuccessful upgrade projects. The following differences between initial ERP implementation and upgrade projects consist mainly on Zhao's (2007, 78-81) observations he bases on his own research and literature review.

1. **Project objectives.** The main objective for initial ERP implementation is to seek optimal business solutions for the organization, therefore, controlling project time might not be considered as important as introducing best practices, novel functionalities to the organizations. But in case of an ERP upgrade, top management may try to complete the project as soon as possible and rush to use the new version.

2. **Scope and complexity**. As a rule, initial ERP implementation is larger and more complex than upgrade projects. Implementations encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, running, testing, and making necessary changes (SearchCRM 2011). Also typically, in ERP implementation projects, configurations options have to be set before the system can be taken into use. Many companies also choose to customize the vendor ERP system to satisfy their business needs. This in turn requires more resources and includes more tasks for both IT and managerial issues, whereas upgrade projects are simpler. However, especially functional upgrades, which allow companies to use additional features provided in the new version, usually require some configuration.

3. **Project duration.** Initial ERP implementation requires longer time. This is obvious because of its bigger scope and higher complexity. Initial implementations last generally 15 to 32 months, upgrade projects 5 to 8 months.

4. **Budget.** Initial ERP implementation is costly. On average, ERP upgrade projects only cost 18% of the initial ERP project cost.

5. **Top management support.** Top management is more involved in initial ERP implementation than in upgrade projects because they consider upgrade projects less risky. During initial ERP implementation they try to control resources, project technology, project scope and structure, and IT infrastructure very carefully to minimize risks and ensure the successful implementation. There are several committees formed

[1] The word deployment is sometimes used to mean the implementation (SearchCRM 2011).

to support the initial ERP implementation, such as top management committee, steering committee whereas, most of the time individuals, such as CIOs or IS managers, are the only support from top management in the organization for upgrade projects.

6. **Project communication and employees.** There is less communication and less full time project employees in ERP upgrade projects.

7. **Implementation strategy.** Most companies seem to choose a big-bang strategy for their ERP upgrade projects. For initial implementation only ca. 60 % choose a big-bang strategy and the rest a phased implementation.

8. **Training.** After many lessons learned from ERP implementation failures and numerous ERP studies conducted by researchers and ERP vendors, organizations have recognized the importance of training in initial ERP implementation and schedule training programs and reserve money for that, ca. 30 % of total costs, in their budgets. In ERP upgrade projects, training programs are typically compresses covering only 5% of total costs in the budget.

9. **Business process re-engineering.** In ERP upgrade projects, time seems to be critical and since additional BPR delay the project, very often BPR is not considered as a critical success factor in ERP upgrade projects. Therefore, it is not surprising that according to Panaya (2010a, 3) the most organizations are starting out with a technical upgrade. Consequently, the new functionality offered by new version is not analyzed at all, even though it might offer an opportunity to eliminate customizations from client's installed version where such capabilities are available in the newer version.

The point above illustrates the real life situation but it is not always the ideal case how upgrade projects should be handled. These issues will come up later again when upgrade critical success factor are discussed.

2.3 Planning SAP ERP upgrade

Upgrading applications is an important issue for SAP ERP using organizations. Basically, they have three questions to answer – why, when and how? The organization must define why they want to upgrade their system, does the organization have new demands or is the vendor affecting the upgrade initiative. The organization must also decide when they want to upgrade, is it right after the new version is released or should they wait for a more mature and tested version. The organization also has to consider, if they want to implement the standard upgrade and get rid of the customer-specific modifications or do they still have to continue to adjusting them in the new version. This chapter outlines challenges and other important considerations related to upgrade planning.

2.3.1 Why upgrade ERP software?

The initial investment to acquire and implement an ERP system is substantial. So why attempt to upgrade once implemented system that already works? Even the conventional wisdom says "if it ain't broke, don't fix it". Especially because ERP upgrades are often seen as costly and complex projects that require intensive resources (Otieno 2010, 115; Sens 2008, 87) and they are also a source of risk (Panaya 2010a, 3). Sens says that the process is sometimes even feared. The fact however is that ERP system cannot remain static after their initial implementation. Business processes for which a company adopted its ERP system at implementation are not necessarily the same processes it needs to track today. Companies grow and change, open and close new business lines and facilities or consolidate operations, obtain new partners or outsource functions. Reporting requirements increase as companies expand across borders. As has been described by Monk (2009, 31), ERP system never seems to be fully implemented but it is in an ongoing state of change. Rapidly changing business environment and evolving technology require continuous checking and upgrading of the ERP software to new version (Zhao 2007, 3). There might be many reasons why an organization might decide to upgrade to the latest version. Some of the most common reasons are introduced next.

Business needs. Organizations demand for new functionality to improve operational excellence, enable innovations, and support new business models (Riedel 2009, 35) that are necessary to keep the competitive position. Upgrading can support also a company's business strategy by enabling flexibility and efficiency to innovate and adapt to this ever-changing environment. New technological developments such as SOA may help organizations to streamline business processes across SAP and third-party applications, as well as across business units, suppliers, and customers. (Riedel 2009, 40.) ERP upgrade decisions may also be influenced by an organisation's strategic orientation toward new technology. An organization that prides itself as an exploiter or innovator of technology will probably adopt the latest upgrade as soon as the new software is released in the market. Some assume that quickly migrating to new versions is key to competitive advantage (Worrell 2008, 12). Vendors, who stand to profit by selling or licensing ERP software, also actively market new versions well before their ability, promising new features and better performance that have not yet been built or properly tested. Such pre-announcements can affect the timing of customer's decision to upgrade. On the other hand, an organization that considers itself as a follower of technology will probably wait to upgrade until migration is absolutely necessary and the new version has been thoroughly tested by earlier adopters. (Otieno 2010, 28.)

Maintaining a stable IT environment

Any disturbance or instability in the IT landscape can affect the performance of the entire company. Therefore, the ERP software must remain compliant with your company's underlying technology, including your operating system, database version, and hardware. Operating systems and database versions usually have shorter maintenance periods than SAP software releases and require more regular upgrades than SAP applications. Thus, older versions such as SAP R/3 may not be compatible with newer operating systems and database versions or may not be able to utilize the enhanced functionality and performance of newer version. A technical upgrade to a latest version helps to stay up to date with the underlying technology. Therefore, many organizations see upgrading as a normal activity within the on-going functioning of an IT department in a business and a necessary investment for maintaining a stable SAP software landscape (Riedel 2009, 34, 42).

Reducing the level of customization to lower the cost and effort of business setup and testing. Many clients have modified the ERP code heavily to meet their own business requirements, but the heavily modified system becomes increasingly expensive and difficult to upgrade. Besides SAP internal estimates indicate that up to 60% of custom developments are not even used by customers. An upgrade can serve as an opportunity to clean up the software system by eliminating unused customizations and help to reduce costs of future upgrades. (Riedel 2009, 47.)

Consolidating the software landscape to reduce technical infrastructure and system software costs. In case SAP software landscape has grown unnecessarily complex, an upgrade can serve as an opportunity to consolidate it and at the same time streamline hardware and server infrastructure and get rid of possible legacy systems. This, in return, reduces the effort needed for ongoing system administration tasks (such as providing backups or applying new support packages). Furthermore, a more consolidated SAP software landscape increases business efficiency and reduces expense and complexity of future upgrades. (Riedel 2009, 47.)

Legal compliance and regulatory requirements. ERP software such as SAP ERP integrate operations with financial and HR processes. Upgrades cover regularly occurring changes in legal requirements (such as tax adjustments) as well as newer compliance standards developed to meet the needs of a global economy. (Riedel 2009, 41.) For example, the Single Euro Payments Area (SEPA) affected many European companies when it introduced new payment standards and rules for euro payments. SAP supported this new functionality from the version R/3 4.6C upwards (SAP Support Portal 2011a), so if a company had an older version an upgrade to a newer version was required.

Vendor's release schedule or support policies. Sometimes upgrade of an ERP software appears to be inevitable. Vendors typically establish 'sunset dates', after which vendors discontinue to support a particular version. Thus, organisations that require vendor support are pressured to upgrade before the vendor's sunset date. Although an organisation may choose to operate unsupported software, ERP products eventually become incompatible with other software or require new functionality. Unless organisations develop their own software or buy new software, their only option becomes upgrading to a new, vendor-supported version. However, organisations do not need to upgrade to every new version of software because vendors typically support multiple versions at the same time. (Otieno 2010, 6-7.)

According to the survey conducted by Panaya Inc. (2010a, 11-12), based on 145 responses from SAP customers and system integrators worldwide, the end of maintenance seems to be the primary reason (54%) for an upgrade. Other reasons cited in their survey are seen in the figure below. End of maintenance is a primary reason to upgrade for 74% of version 4.6 users, versus 63% for version 4.7 and 43% for 5.0.

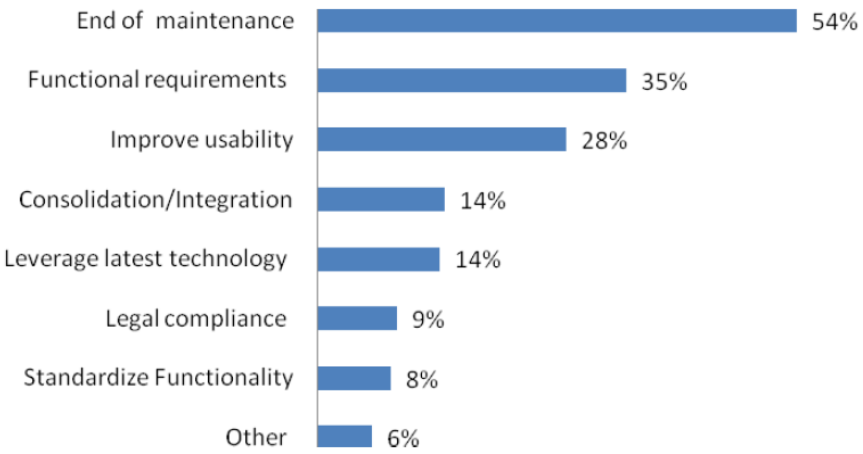


Figure 8. Reasons for SAP upgrade according to Panaya's survey (2010a, 11).

Also Yachin (2009,4) and Sens (2008, 87) confirm that the end of current maintenance licensing is the most common reason for upgrades nowadays, whereas benefits from new features available in the new version upgrade (functional requirements) used to be still few years ago (Martinsen 2010, 19). In this light, it is not surprising that according to Panaya (2010a, 11) the most organizations (68%) are starting out with a technical upgrade. This type of upgrade does not include any functional changes or the addition of new functionalities (which are enabled in functional upgrades). Instead, it offers a rapid, low-cost, and low-risk alternative that is aimed at retaining existing functionalities.

2.3.2 The downside of not upgrading

When justifying an upgrade, an organization should also consider the question, what are the risks of not upgrading? According to Otieno (2010, 113), long upgrade cycles (length of period between ERP system upgrades in an organization more than 3 years) are likely to add additional risks. Upgrades tend to be more difficult, more complex, and have a greater impact on the user environment when long period of time has elapsed. Also Vaucouleur (2008, 7) states that if upgrades are skipped, upgrading will be harder later on. Martinsen (2010, 19) says that postponing new version upgrades will restrain benefit-realization from the ERP system and this will result in some opportunity cost. Also Riedel (2009, 48) thinks that the competitive position and differentiation of the business may be threatened if the company is unable to adapt its business processes as required by own organization or by outside factors beyond own control. He adds also risks such as technological obsolescence and the incompatibility of the current hardware, operating systems and database versions (Riedel 2009, 48).

Furthermore, if an organization falls further and further behind the latest version, it risks also the loss of standard vendor support. With an integrated ERP environment, the entire business is at risk if the ERP system becomes de-supported. Or if the vendor continues to provide technical support after the sunset date, it might have to be purchased at an additional charge. In addition, post-sunset customers might be given lower priority than customers with more recent versions. Moreover, when an organization finally decides to upgrade, it might be impossible to upgrade straight to the latest version, like in the case of KPLC. When KPLC decided to upgrade from version 2.11H to the latest version SAP R/3 version 4.7, the organisation was informed that it was impossible to upgrade directly to version 4.7. The only way out was to first upgrade to version 4.6 and then to version 4.7. When KPLC enquired if they could upgrade to mySAP, it was informed that mySAP was a new product and hence regarded as a new installation. (Otieno 2010, 113-15.)

Despite the above factors, surprisingly large number of organizations have decided not to upgrade and stick with the software version they are currently running. According to the Forrester Research survey of 900 ERP users, 72% of ERP customers are in a holding pattern, with no specific plans to invest in their ERP systems in 2011. While only 19% plan to upgrade or expand existing ERP implementations, that figure is down from 22% in 2010 and from 29% in 2009. Approximately half of ERP customers are currently on releases that are two versions behind the current release; these may be four years old or more. It is surprising firstly because many organizations invested millions in buying and implementing ERP, and now depend on it as the core system. And secondly, ERP customers who are on expensive vendor maintenance contract are already paying for the latest release of the application. Reason for not upgrading might be that after spending a couple of decades implementing ERP, the most businesses believe that that job is now finished, and businesses want to spent the money now on technologies that are seen as more innovative and value-added, such as mobility, consumer-oriented technologies and social networking. Future will show if the ERP upgrades can keep up with the latest development. (Moad 25.1.2011.)

2.3.3 When to upgrade?

As described in the previous section, there can be different factors that drive ERP upgrade decision. For example, the primary driving force behind an upgrade decision might be lack of vendor support for one organization but acquiring the latest functionality might be the critical factor for another. Organizations today rely heavily on their ERP software to run critical business activities and ERP is typically considered as a long time investment. Therefore, as new versions are released regularly, the organization's question is not whether to upgrade, but when to do so (Mukherji & Rajagopalan and Tanniru 2006, 1685). Although upgrades are an integral part of ERP maintenance, the timing of a version change seem to be defined rather arbitrarily in the user organizations (Kankaanpää & Pekkola 2010, 2; Mukherji et al. 2006, 1685). Unlike upgrading of a small stand-alone package, such as a word-processing software, purchasing and implementing ERP upgrade is a costly and time demanding process. It is therefore not optimal for an organization to upgrade its ERP each time a new version is introduced. (Ng & Chan 1999, 102.) A few researchers have developed models to determine the optimal timing for upgrade, but according to Kankaanpää and Pekkola (2010, 9), there is no indication that organizations are using such models for determining the suitable upgrade timing.

One example is the analytical ERP upgrade model of ERP life-cycle developed by Ng and Chan (1999) for determining the optimal timing for upgrade and to complete the upgrade implementation. Their model considers initial investment costs of the ERP and investment cost of the new version (purchasing, implementation and training), total maintenance cost of the existing ERP system from time 0 until the new upgrade version is fully implemented and then the model sums all to the total cost of the entire ERP life cycle. Furthermore, the model also considers user dissatisfaction cost. The mathematical formulas seem a bit complicated at short glance, but the implications Ng and Chan (1999, 109-110) draw from the model are introduced next shortly. According to them, the decision to upgrade should not be dependent upon the initial cost already invested in the ERP. Even if an organization feels that the expected return on the huge sums invested initially has not yet been realized and therefore is reluctant to upgrade, the upgrade should not be prolonged, if the new version could realize the expected benefits better than the existing version. Indeed, the model suggests that the upgrade should be completed even earlier if user dissatisfaction costs are high. Also if new customizations can be avoided by upgrade, the new version should be adopted earlier. Furthermore, their model suggests that an organization oprating in a volatile business environment, should wait for later versions to be available before it decides to upgrade and when it does it should complete the upgrade in a fast schedule by allocating more resources. The model also indicates that in an organization where user dissatisfaction is high, the main issue is not which version should be adopted but to speed up with upgrade implementation so that users could make use of the new version earlier. Results also suggest that if an existing version costs more to maintain than a new version, upgrade should be performed. Respectively, if the newer version to be upgraded costs more to maintain, an organization should delay its decision to upgrade. Finally, their model suggests that with a more efficient upgrade team, an organization cannot only wait to adopt later version of ERP that include more functionality but also complete the upgrade within a shorter schedule. This result highlights the importance of having experienced people involved in the implementation of the upgrade.

Another example is the economics-based technology upgrade model by Mukherji et al. (2006). Their model states that the optimal time for an upgrade is when the gap between the version in use and the new available version reaches a critical threshold. This means that the organization has to compare technology and change management costs to opportunity costs. Technology cost means the costs of adopting a new version. Change management cost refers to the cost of upgrade deployment activities, e.g. the time used for learning new routines and training the users. Opportunity cost is the cost of lost opportunity. It can manifest itself through the decreased productivity or the loss of revenue due to the decision not to adopt new technology or to adopt an ill-suited version. (Kankaanpää & Pekkola 2010, 5.) The model suggests that the organization should wait until the technology cost of the new version decreases since the latest version is the most expensive immediately after its release. However, this may be the case for most software in the market, but in the context of ERP software this might not be applicable. Furthermore, if the estimated change management costs are high, upgrade should be postponed. However, if a company does not invest in new technology, it may lose the opportunity for higher productivity. In addition, it is stated that if the loss of vendor support involves a very high cost for the organization and outweighs upgrade adoption costs, the optimal strategy is to upgrade. One of the important implications of their study is that ”leap frogging” is the most efficient upgrade strategy in most cases. This means that organization should not invest in every new version but instead wait until there is a need for change, skip unnecessary versions, and upgrade when a clearly beneficial version is available (Mukherji et al. 2006, 1684).

The findings of Kankaanpää and Pekkola (2010, 9) study on 12 mid to large size companies and public organizations in Finland also support leapfrogging as the most efficient and used technique for defining ERP upgrade timing. They however point out, that successful leapfrogging requires in-depth knowledge about the functionalities of the current version and organization’s own needs. At the same time, this requires that the organization has to actively follow the development of the software in order to be able to compare the functionalities with the functionalities of their current version. The literature analysis of the study suggests that the timing of the upgrade is dependent on the availability of a suitable version, the customer’s need for upgrade and economics. Additionally, their empirical findings suggest that, from the customer’s point of view, the timing of an IS upgrade is guided by four determinants, namely business interests, business calendar, on-going and planned development projects, and the vendor (see Figure 9. Determinants of IS upgrade timing by Kankaanpää and Pekkola (2010, 9).).

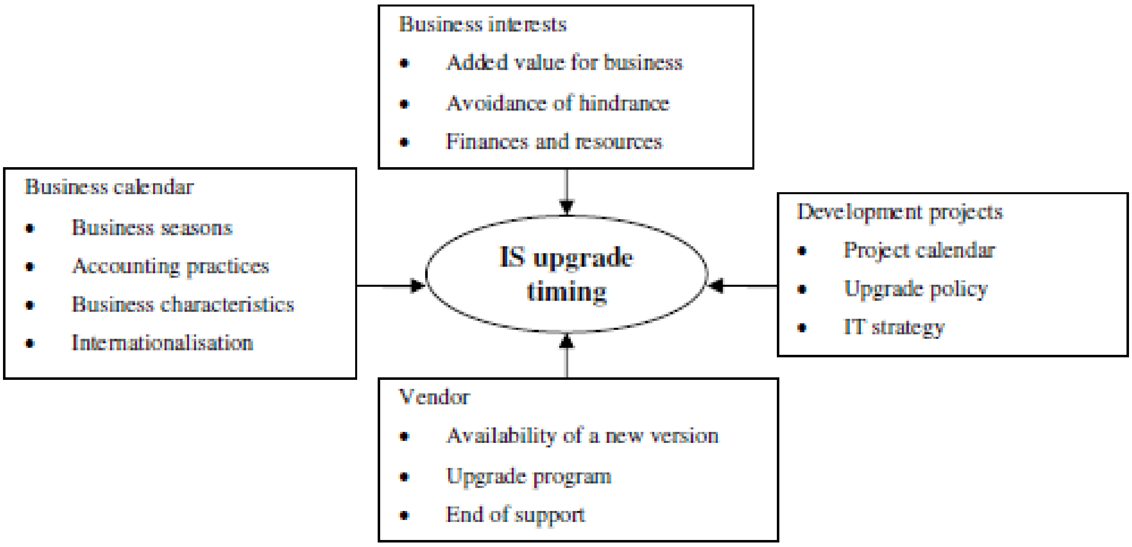


Figure 9. Determinants of IS upgrade timing by Kankaanpää and Pekkola (2010, 9).

According to Kankaanpää and Pekkola (2010, 6-10) , the most significant determinant for upgrade timing is the business value of the upgrade. This is strongly related to risk management, expected business benefits, and avoiding hindrance to business. Naturally, organization's financial situation and availability of resources has a word whether to upgrade. Hence the timing of the version change, is commonly decided in conjunction with organization's overall IT strategy and budgeting. Technological advancement and wear have also their impacts on upgrade timing. Business calendar, business issues and development projects on their half define what are the possible times for upgrade in the user organization are. The business calendar, that is the annual business activities of a company, limits time-slots for upgrade. As a rule of thumb, upgrades are avoided at the end of the fiscal year and the beginning of the reporting period, high-business season and during holidays. Also the level of internationalization and industry specific business characteristics define each organization's own business calendar. For example, the time frames for system upgrades are very small for a global retail company that is open 24/7. Upgrades are preferably made when the potential hindrance for business is at lowest. Hence, different kinds of systems have different windows of opportunity for their upgrades. Accounting systems, for instance, have strict rules when any change can be made. This means that financial administration systems are frozen during the closing of the books. It is also advisable to plan and schedule ERP upgrades in conjunction with other development activities in order to avoid overlapping projects or to combine projects when joint benefits are anticipated. ERP vendors propose version changes and upgrades from their own business perspectives and schedules which do not necessarily match with the customers' interest, needs or business calendar. But despite of the dependency on the vendor, the availability of a suitable version and resources should be the dominating preconditions for an upgrade. According to Kankaanpää and Pekkola, the organization may postpone the upgrade up to the point when it has to be made as e.g. vendor's support services are stopped.

But for how long can you postpone the upgrade and what it the ultimate point? The discussion about the upgrade timing is concluded with the list of five clear signs provided by Stackpole (17.12. 2008), an ERP manufacturing professional:

1. Upgrade an ERP system that is more than five years old.
2. Upgrade when ERP system integration is difficult.
3. Upgrade when an ERP system is missing the "modern" features and functions required to efficiently run the business.
4. Upgrade when employees, partners and consultants are not using the system anymore or are not available to fix it.
5. Upgrade when it is obviously time, whether the hard upgrade ROI (return on investment) is clear or not.

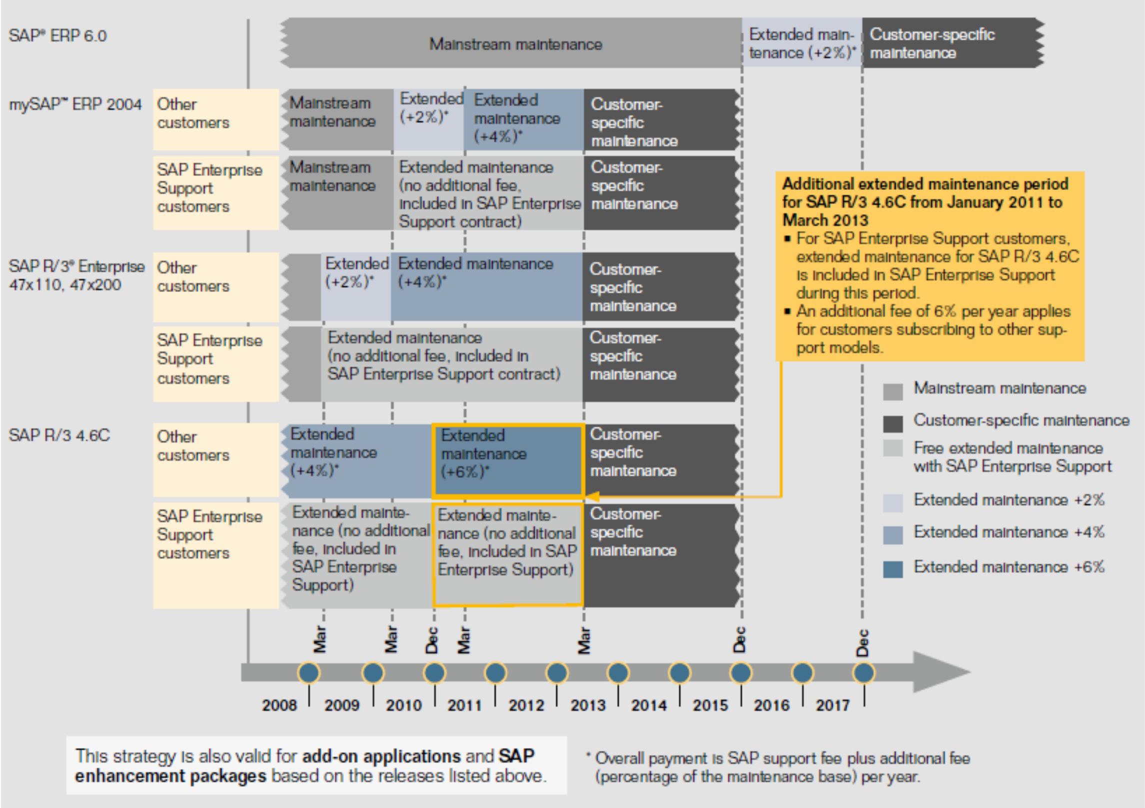
2.3.4 Release and maintenance strategy and upgrading paths

The ongoing upgrade, maintenance, and support of ERP systems differ significantly from a traditional in-house software system. Unlike the implementation of in-house developed application systems, ERP package adoption and maintenance is not a task manageable by any client organization on its own. Some improvement activities, like internal change requests and modifications, are purely in-house issues. But more often ERP activities are affected by the vendor's product plans and support policies. Vendor driven M & S include e.g. technical support services, distribution of software patches, and also minor and major software releases. ”In general, the most ERP user organizations subscribe to the vendor's M&S service. The rationale behind this is twofold. Firstly, ERP expertise is precious to many client organizations, and a subscription to the vendor's M&S program is an important means for client organizations to secure such expertise when needed. Secondly, it entitles a company to support services, software patches, and new releases in the future.” Obviously, these M & S services provided by vendors do not come cheap. But despite the high cost, the most client organizations continue to subscribe since they see that the saving obtained do not justify the risk. (Law &, Chen & Wu 2010, 299.) Therefore, as also described earlier in the chapter 2.3.1, the main reason for upgrading ERP software is maintaining the system at current vendor-supported levels. Depending on the vendor, upgrades can occur multiple times in a year or once every several years. (Bortoulus 3.9.2010.) In recent years, researchers have discovered a trend for ERP vendors to launch new releases more frequently. In the 1990s, the interval between ERP releases was ca. 3 years, but this has recently declined to 1.5–2 years. That means a release will be removed from the support list sooner than before. (Law &, Chen & Wu 2010, 299.)

SAP’s Release and Maintenance Strategy for SAP ERP determines the availability of new releases (including enhancement packages), the length and conditions of their maintenance, and the dependencies between individual releases. The SAP release and maintenance strategy is usually issued to clients up to two years in advance and the most current strategy can be obtained from http://service.sap.com/releasestrategy (logon requires a user account). The SAP ERP application is the follow-up product to SAP R/3 software. SAP ERP 6.0 is the latest and current target release (also often referred as go-to release) for customers considering upgrades of their current SAP ERP software (SAP Support Portal 2010b, 7).

There are three distinct maintenance phases for SAP ERP defined under the SAP Maintenance Strategy: mainstream maintenance, extended maintenance and customer-specific maintenance. Mainstream maintenance is the original support period for a given release. Typically SAP recommends upgrading before you reach the end of the mainstream maintenance phase. After the mainstream maintenance period has concluded and the client chooses not to opt an upgrade, the client has the options of choosing either extended maintenance (at an additional fee, but with all benefits of the mainstream maintenance phase) or entering the customer-specific maintenance period automatically. During customer-specific maintenance, some restrictions apply to the scope of support. In addition, during different maintenance phases clients receive support according to their support agreement. Agreement offerings for SAP ERP include the basic SAP Standard Support and more comprehensive SAP Enterprise Support that provides proactive support (e.g. planning, blueprinting, and realization) in addition to all the features of the SAP standard support option. Clients can select their support level based on their individual business needs and desired depth of SAP support. (SAP Support Portal 2011c.)

The following illustration will tell which SAP ERP versions SAP supports, when certain ERP versions will be withdrawn from support (sunset dates) and against what additional charges the support can be extended.


SAP® ERP 6.0
Mainstream maintenance
Extended main-tenance (+2%)*
Customer-specific maintenance
mySAP™ ERP 2004
Other customers
Mainstream maintenance
Extended (+2%)*
Extended maintenance (+4%)*
Customer-specific maintenance
SAP Enterprise Support customers
Mainstream maintenance
Extended maintenance (no additional fee, included in SAP Enterprise Support contract)
Customer-specific maintenance
SAP R/3® Enterprise 47x110, 47x200
Other customers
Extended (+2%)*
Extended maintenance (+4%)*
Customer-specific maintenance
SAP Enterprise Support customers
Extended maintenance (no additional fee, included in SAP Enterprise Support contract)
Customer-specific maintenance
SAP R/3 4.6C
Other customers
Extended maintenance (+4%)*
Extended maintenance (+6%)*
Customer-specific maintenance
SAP Enterprise Support customers
Extended maintenance (no additional fee, included in SAP Enterprise Support)
Extended maintenance (no additional fee, included in SAP Enterprise Support)
Customer-specific maintenance
Additional extended maintenance period for SAP R/3 4.6C from January 2011 to March 2013
▪ For SAP Enterprise Support customers, extended maintenance for SAP R/3 4.6C is included in SAP Enterprise Support during this period.
▪ An additional fee of 6% per year applies for customers subscribing to other support models.
Mainstream maintenance
Customer-specific maintenance
Free extended maintenance with SAP Enterprise Support
Extended maintenance +2%
Extended maintenance +4%
Extended maintenance +6%
Mar
Mar
Dec
Mar
Mar
Dec
Dec
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
This strategy is also valid for add-on applications and SAP enhancement packages based on the releases listed above.
* Overall payment is SAP support fee plus additional fee (percentage of the maintenance base) per year.

© Copyright 2010. SAP AG. All rights reserved.

Figure 10. Release and Maintenance Strategy for SAP ERP 6.0 and its predecessors. (Source: SAP Support Portal 2010b, 16).

Maintenance strategy for SAP R/3 4.6C Customers

• Mainstream maintenance for SAP R/3 4.6C ended in December 2006, with extended maintenance offered until the end of March 2013. (SAP had planned to end the support already at the end of 2010, as a result of pressure from customers support was extended without until March 2013 (Computerworld UK 2010)).

- For customers on standard support models, an additional fee applies (an additional 4% in 2010, and an additional 6% per year from January 2011 to March 2013).
- Beginning in April 2013, SAP will provide customer-specific maintenance for SAP R/3 4.6C.

Maintenance strategy for SAP R/3 Enterprise Customers (47x110, 47x200)

- Mainstream maintenance for SAP R/3 Enterprise software ended in March 2009,with extended maintenance offered until the end of March 2013.
- For customers on standard support models, an additional fee applies (an additional 2% from April 2009 to March 2010, and an additional 4% per year from April 2010 to March 2013).
- Beginning in April 2013, SAP will provide customer-specific maintenance for SAP R/3 Enterprise.

Maintenance strategy for mySAP ERP 2004 Customers

- Mainstream maintenance for mySAP™ ERP 2004 ended March 2010, with extended maintenance offered until the end of March 2013.
- For customers on standard support models, an additional fee applies (an additional 2% per year from April 2010 to March 2011, and an additional 4% per year from April 2011 to March 2013).

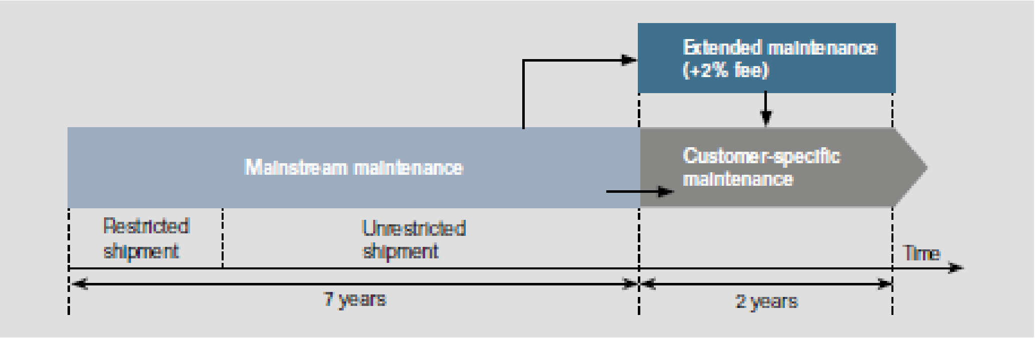
Figure 1 shows the details specific to each release. For clients with older releases than SAP R/3 4.6C (SAP R/3 3.1I–4.6B) SAP offered extended maintenance for these releases through the end of 2006. Since then, these releases have been in customer-specific maintenance. (SAP Support Portal 2010b, 15-17.)

Maintenance strategy duration

Maintenance strategy duration rules for SAP ERP 6.0 have the following durations:

- Seven (7) years of mainstream maintenance
- Two (2) years of extended maintenance at an additional 2 % fee
- Thereafter, customer-specific maintenance. (SAP Support Portal 2010b.)

This 7-2 maintenance applies to SAP ERP releases as of November 2008 (6.0) and mainstream maintenance window under this strategy continues until December 2015 and after that extended maintenance window to December 2017. The older releases released 2004-2007 fall under the 5-1-2 strategy, where five years of mainstream maintenance and 1 year of extended maintenance at an additional 2% fee and next 2 years of extended maintenance at an additional 4 % fee. Thereafter, the release enters customer-specific maintenance. Longest mainstream maintenance window under this strategy until March 2013 (2007 releases). (SAP Support Portal 2011d.)



© Copyright 2011. SAP AG. All rights reserved.

Figure 11. 7-2 maintenance strategy duration for SAP ERP 6.0 release. (Source: SAP Support Portal 2011f, 46)

Today, many organizations are in different stages of upgrading to SAP ERP 6.0. Approximately half of ERP clients are currently on releases that are two versions behind the current release; these may be four years old or more (Moad 25.1.2011).

SAP provides upgrade paths for SAP releases. Usually clients can upgrade directly (in one step) from one release to any other subsequent SAP release as long as the releases are in the mainstream maintenance phase or extended maintenance phase. Depending on technological constraints, an upgrade to a release that is several releases removed from a client's current release may have to be performed in more than one step.(SAP Support Portal 2011f, 47.) Currently the earliest SAP R/3 release that will support a direct upgrade to SAP ERP 6.0 and include SAP enhancement package 5 in the upgrade process is SAP R/3 4.6C. (SAP Support Portal 2010b, 18)

Mainstream maintenance and Scope of Standard Support

The deliveries of new releases of the licensed software are covered by the SAP support contract during the mainstream maintenance phase. According to general information provided by SAP, Standard support generally includes the following upgrades (among other things):

1. **Notes.** The development and use of software means the existence of bugs, and SAP ERP is no different from any other software. SAP's corrective bug fixes are called OSS notes (or simply 'notes'). Each note is aimed at solving a specific problem across all releases in which this problem exists. SAP Notes also document related issues, customer questions, and recommended solutions (e.g. customizing settings). Notes are released regularly and frequently – between 2,000 and 3,000 notes are released every month just for SAP ERP. Notes are grouped into support packages and support package stacks. (Panaya 2010b, 1, 8.) It is relatively easy to analyze and assess its impact when implementing individual notes. Therefore, implementations using this policy can be carried out in an ongoing manner without demanding extensive testing effort and with minimal risk of unexpected impacts. (Panaya 2010b, 3.)

2. **Support packages.** Since individual implementation of notes is labour-intensive, and an individual note often requires previous notes to be implemented first, SAP groups notes into support packages, which customers can implement in a single run (Panaya 2010b, 8, 3). Support packages target a specific application component and are released sequentially. For example, the EA-HR ERP component provides HR-related functionality. Every month a new support pack is released for this component. (Panaya 2010b, 8) A single support package contains, on average, about 400 notes (SAP Support Portal 2011c.) Implementing support packages requires extensive testing compared to single notes and also some level of code freeze throughout the implementation project (Panaya 2010b, 5).

3. **Support package stacks.** There are dependencies between ERP components that create subsequent dependencies between support packages. To tackle this reality, SAP introduced the concept of support package stack that aligns support packages across all ERP components. Support package stacks are released (roughly) on a quarterly basis. (Panaya 2010b, 8.) A single support package stack contains around 8,000 notes (SAP Support Portal 2011c). For example, Support Package Stack 15 contained 8368 notes in 36 support packages (Panaya 2010b, 8). SAP strongly recommends regular application of support package stacks at least once a year (SAP Support Portal 2011f, 44).

4. **Technology updates.** Technology updates to enable compatibility with changed technologies of third-party operating systems, databases and browsers. (SAP Support Portal 2011f, 89)

5. **Enhancement packages (EHP).** SAP introduced this new upgrade deployment concept for SAP ERP 6.0. Enhancement packages are a new SAP technique to deliver new functionality; corrections and legal changes will be deliv-

ered still in support packages (Riedel 2009, 20). An organization must upgrade to SAP ERP 6.0 before it can use enhancement packages (SAP Support Portal 2011b, 34). SAP ERP 6.0 is now considered as the base release that is planned to remain stable over the coming years. So enhancement packages will deliver new functions on top of SAP ERP 6.0. New functionality can be implemented by customer selectively. (Riedel 2009, 25.) With regard to licensing, enhancement packages are treated the same as release upgrades; the delivered functionality is covered under the terms of client's maintenance and licence agreement (SAP Support Portal 2010b, 29). Since this enhancement package concept is a very remarkable change in the way SAP delivers upgrades, it is devoted an own chapter 2.3.5 in this research.

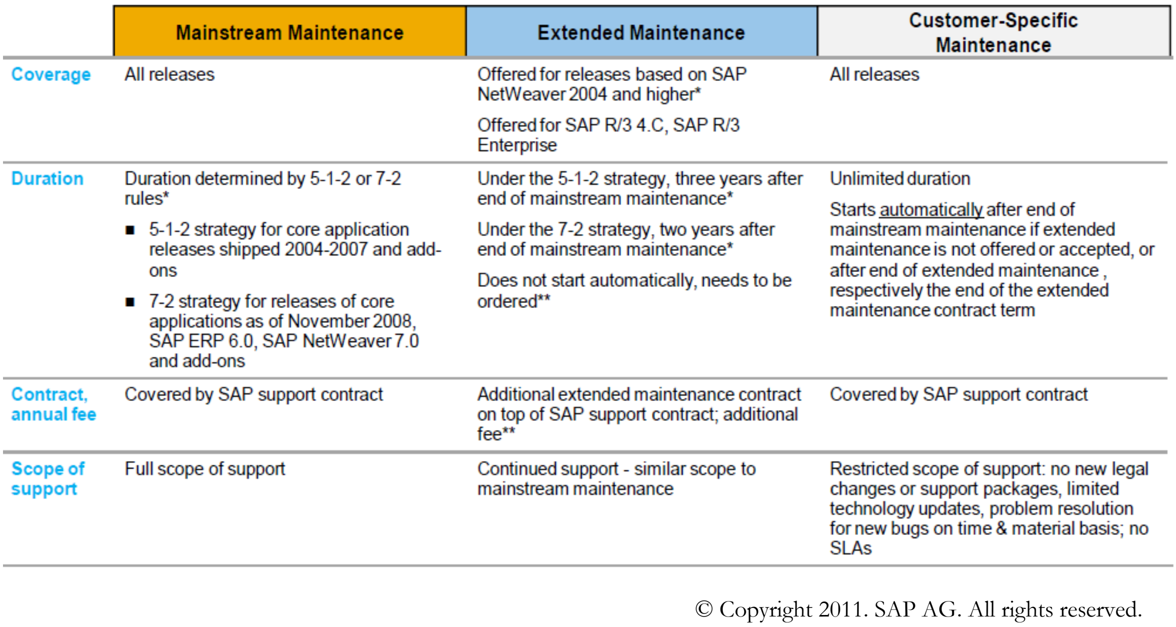
6. **New software releases.** Ultimately, all the above mentioned minor upgrades and functional enhancements are incorporated into a new software release. When upgrading from SAP R/3 to SAP ERP 6.0, client's existing SAP R/3 contract will be converted in their entirety to a new contract, this means that the client needs to pay an additional charge. However, a percentage of client's SAP R/3 licenses may be credited to the new contract. (SAP Support Portal 2010b, 29.) New software releases, require considerably more effort by clients than the other type of upgrades mentioned above.

Organization's own upgrade policy defines at what level and when (single note, support package, or support package stack, enhancement packages, releases) corrections and improvements are implemented.

After the mainstream maintenance period has concluded, you can either plan to upgrade, or have the options of choosing either extended maintenance or entering the customer-specific maintenance period.

Extended maintenance. After the end of mainstream maintenance, SAP offers extended maintenance for SAP ERP releases. During extended maintenance, the scope of support is similar to the scope during mainstream maintenance. Extended maintenance is an optional offering and requires a separate, additional contract on top of your support agreement because it is available against an additional fee (SAP Support Portal 2011c.) However, please note that SAP has decided to waive this additional fee for SAP Enterprise Support customers for several releases. In addition, extended maintenance does not have to be ordered for the entire extended maintenance period. Extended maintenance can be ordered for one or multiple successive quarters, for example, when the client plans to upgrade in the course of that extended maintenance period. Thus, extended maintenance provides more flexibility in deciding on the right time for an upgrade. Extended maintenance is recommended, if client installed version requires legal changes, such as typically provided for HR or FI/LO or if the technology stack will be changing or fixes provided by support packages are required. (SAP Support Portal 2011e.)

Customer-specific maintenance. If you do not order extended maintenance or when your extended maintenance contract expires, client's release will automatically enter into the customer-specific maintenance phase (SAP Support Portal 2011e). There is no need to apply for an additional contract, but the customer continues to pay the annual support fee (no extra charge) for the support option he has (for example SAP Enterprise Support). Customer-specific maintenance does not have an expiry date. (SAP Support Portal 2011c.) During this phase, customers receive support services similar to those offered in the mainstream maintenance phase, with some restrictions. For example, SAP does not deliver new support packages or updates to cover legal changes, and technology updates are limited. In addition, problem resolution is customer specific, which means customers are charged for solving problems not yet known to SAP. (SAP Support Portal 2011f, 46.) This means in other words, that if the client encounters a problem during customer-specific maintenance and cannot find a solution by itself, it can create a message, and SAP's support specialists will investigate it. If the problem is already known to SAP, the client will for example be advised to implement a note in which the problem is solved. If the problem is "new", that is, not yet known to SAP, problem resolution is customer-specific and may be liable to a fee based on time & material. (SAP Support Portal 2011e.) Moreover, one-step upgrade to latest release may not be available. As a rule, SAP does not provide new or additional upgrade paths for a start release in customer-specific maintenance. Upgrade paths that were available during mainstream maintenance or during the period when extended maintenance was offered are still available during customer-specific maintenance. (SAP Support Portal 2011c.)



	Mainstream Maintenance	Extended Maintenance	Customer-Specific Maintenance
Coverage	All releases	Offered for releases based on SAP NetWeaver 2004 and higher* Offered for SAP R/3 4.C, SAP R/3 Enterprise	All releases
Duration	Duration determined by 5-1-2 or 7-2 rules* ■ 5-1-2 strategy for core application releases shipped 2004-2007 and add-ons ■ 7-2 strategy for releases of core applications as of November 2008, SAP ERP 6.0, SAP NetWeaver 7.0 and add-ons	Under the 5-1-2 strategy, three years after end of mainstream maintenance* Under the 7-2 strategy, two years after end of mainstream maintenance* Does not start automatically, needs to be ordered**	Unlimited duration Starts automatically after end of mainstream maintenance if extended maintenance is not offered or accepted, or after end of extended maintenance , respectively the end of the extended maintenance contract term
Contract, annual fee	Covered by SAP support contract	Additional extended maintenance contract on top of SAP support contract; additional fee**	Covered by SAP support contract
Scope of support	Full scope of support	Continued support - similar scope to mainstream maintenance	Restricted scope of support: no new legal changes or support packages, limited technology updates, problem resolution for new bugs on time & material basis; no SLAs

© Copyright 2011. SAP AG. All rights reserved.

Figure 12. SAP Maintenance Strategy – general rules. (Source: SAP Support Portal 2011d, 4)

2.3.5 SAP Enhancement package concept

SAP's traditional practice of shipping a full version upgrade every other year has been superseded by a new method of delivering enhancement packages one or two times a year. In other words, this new practise provides new business functionality that was traditionally released through SAP version upgrades in smaller doses. SAP enhancement packages deliver only optional new or improved business functionality that are either relevant for all industries or targeting particular industries. Thus, delivered functionality can be deployed selectively depending on client's business needs. Enhancement packages do not replace support packages that continue to be mandatory containing bug fixes, and legal or tax changes. (SAP Support Portal 2010b, 12; Riedel 2009, 25 ; Sens 2008, 81-82.) The enhancement packages aim to simplify the upgrade process and minimize disruption to production system as well as reduce risk and testing efforts. SAP provides testing templates for every business function to simplify the testing of client organizations (Riedel 2009, 136; SAP Support Portal 2010c).

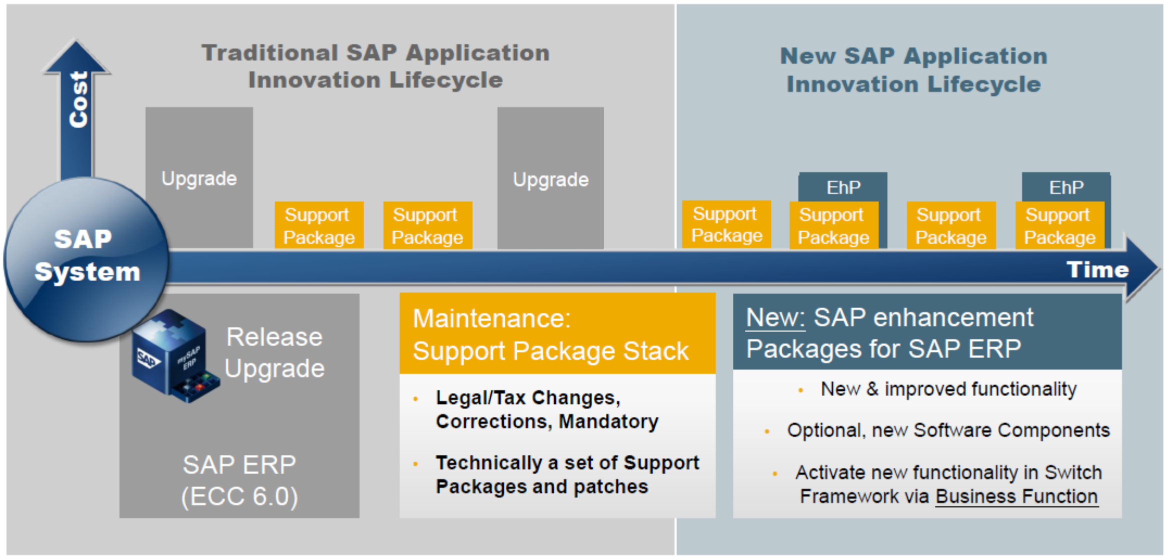
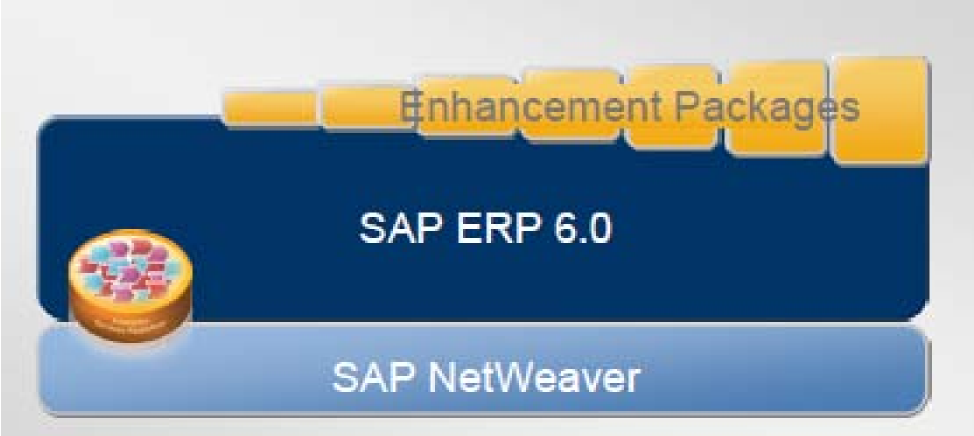

Traditional SAP Application Innovation Lifecycle
New SAP Application Innovation Lifecycle
Cost
SAP System
Time
Upgrade
Support Package
Support Package
Upgrade
Support Package
EhP
Support Package
Support Package
EhP
Support Package
Release Upgrade
SAP ERP (ECC 6.0)
Maintenance: Support Package Stack
Legal/Tax Changes, Corrections, Mandatory
Technically a set of Support Packages and patches
New: SAP enhancement Packages for SAP ERP
New & improved functionality
Optional, new Software Components
Activate new functionality in Switch Framework via Business Function

Figure 13. SAP's traditional practice of delivering upgrades vs. the new enhancement package concept. (Source: SAP Support Portal 2010c, 51).

Enhancement packages represent a major development undertaking – they are not minor in term of what they can deliver. For example, enhancement package 3 involved a similar amount of development effort as SAP ERP 6.0 itself. The amount of new functionality delivered by each enhancement package is therefore significant and can be compared to a full release. However, the differentiating factor between an enhancement package and previous full releases is the installation and activation functionality/technology that lies behind enhancement packages. (Riedel 2009, 27.) Therefore, with regard to licensing, enhancement packages are treated the same as release upgrades - enhancement packages can be licensed separately, if not covered under the terms of client's maintenance and licence agreement (SAP Support Portal 2010b, 29). The functionality that has been added to the system through enhancement packages will become part of the next major SAP release.

An organization must upgrade to SAP ERP 6.0 before it can use enhancement packages (SAP Support Portal 2011b, 34). SAP ERP 6.0 is now considered as the base release that is planned to remain stable over the coming years. So enhancement packages will deliver new functions on top of SAP ERP 6.0. (Riedel 2009, 25.)



© Copyright 2010. SAP AG. All rights reserved.

Figure 14. SAP ERP enhancement packages are cumulative. (Source: SAP Support Portal 2010c, 15)

Currently five enhancement packages have been delivered for SAP ERP 6.0. At the moment, estimated release date for EHP 6 is Q4 2011. SAP enhancement packages are cumulative. This means when implementing an enhancement package you get all the innovation delivered with the previous enhancement packages. For example, if you implement enhancement package 5 for SAP ERP 6.0, you get, and can activate, all innovation that was delivered with the first four enhancement packages for SAP ERP 6.0 as well. (SAP Support Portal 2011b, 13 ; SAP Support Portal 2011f, 50 .) However, if you install enhancement package 5 after having previously installed enhancement package 4, you need only install the delta functionality. (Riedel 2009, 27-28.)

Full system upgrades and installations can be extremely costly in terms of time, money, and resources, not to mention the disruption to your day-to-day business activities. Enhancement packages are aimed at bringing the flexibility to install only those portions of software that directly impact the process or functions client wants to improve. After these specific components are in place you can re-activate the impacted processes, and your system will react only to those changes." (Muir & Kimbell 2010, 350.)

Installing SAP Enhancement Packages

When installing an SAP enhancement package or including it in the upgrade, SAP recommends the following approaches:

- **Selective installation of software components.** Install only selected parts of the enhancement package. Select only those software components that are needed from a business point of view. As a result, customers can isolate the impact of software updates and bring new functionality online faster through shortened testing cycles.

  After client has installed functionality from an enhancement package, it cannot reverse the installation. If client is not sure if it wants to leverage dedicated functionality, use a sandbox system to explore the new functionality.

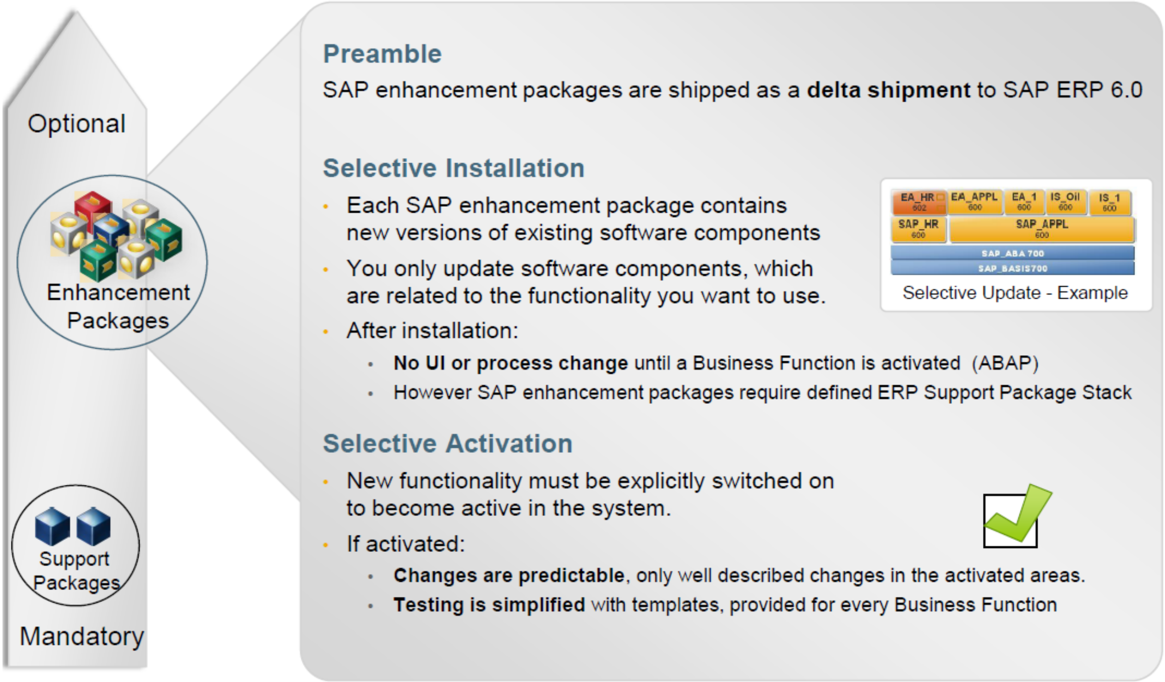
- **Selective activation of business functions.** New functionality must be explicitly switched on to become active in the software. Activate only the business functions that you require. As a result, changes are predictable affecting only the activated areas. As long as you do not activate a business function, the installation of an enhancement package has no impact on existing business processes or user interfaces.

  After you have activated a business function using the switch framework, you cannot reverse the activation.

- **SAP Enhancement Packages and Support Packages.** Always install SAP enhancement packages together with a support package (in one queue), because support packages deliver corrections not only for base release but also for the functionality (e.g. programs, screens, and DDIC objects) included in the enhancement packages. That is also why, the delivery of SAP enhancement packages is synchronized with the delivery of support packages. Concurrent installation also helps to minimize the effort of installation and testing as well as reduce the downtime.

  However, it should be noted that support packages can be installed independently of enhancement packages. This means that regular maintenance is done by applying support packages and only optionally new functionality is added through enhancement packages.

  (Riedel 2009, 25, 28; SAP Support Portal 2011b, 45; SAP Support Portal 2010b, 14.)



Preamble

SAP enhancement packages are shipped as a **delta shipment** to SAP ERP 6.0

Selective Installation

- Each SAP enhancement package contains new versions of existing software components
- You only update software components, which are related to the functionality you want to use.
- After installation:
  - **No UI or process change** until a Business Function is activated (ABAP)
  - However SAP enhancement packages require defined ERP Support Package Stack

Selective Activation

- New functionality must be explicitly switched on to become active in the system.
- If activated:
  - **Changes are predictable**, only well described changes in the activated areas.
  - **Testing is simplified** with templates, provided for every Business Function

© Copyright 2010. SAP AG. All rights reserved.

Figure 15. Enhancement packages can be installed selectively. (Source: SAP Support Portal 2010c, 52)

2.3.6 Determining an upgrade approach

The upgrade justification process will ultimately determine the upgrade approach organization takes (Riedel 2009, 35). The decision for an SAP software upgrade is usually determined by a mix of business, technology, and strategic criteria as discussed previously in the chapter 2.3.1 . SAP supports the following three types of upgrade approach:

Technical upgrade

A technical upgrade focuses on a purely technological upgrade, without implementing new functionality that would change user behaviour or business processes (Martinsen 2010, 44). An example of this is the upgrade from SAP R/3 4.6C to SAP ERP 6.0 without installing and activating enhancement package functionality. Therefore, the impact of technical upgrade on organization's business and business processes is very limited (SAP Support Portal 2010b, 19).

The intention of a technical upgrade is to replace software components with newer versions. Two techniques can be distinguished: partial and complete replacement. During a partial replacement, only certain components are replaced with new executables and objects as in case of support packages. When a full upgrade version is installed, all objects and software components are replaced with newer ones; that is complete replacement. (Sens 2008, 117) A technical upgrade offers also an opportunity reduce landscape complexity through consolidation of instances and servers, as well as Unicode conversion and database upgrade. (Riedel 2009, 37.)

A technical upgrade brings the SAP ERP system into the maintenance mainstream of the higher version (Riedel 2009, 37). It thus ensures the SAP support and maintenance. Given that the majority of organizations are upgrading due to end of maintenance, it is not surprising that most organizations (68%) are starting out with a technical upgrade according to Panaya (2010a, 11)

Performing a technical upgrade is a relatively fast and low cost effort with manageable impact and minimized disruption. It protects IT system stability and provides the foundation for the technological, business and strategic improvements in SAP ERP software later on. (Riedel 2009, 36; SAP 2007, 7; SAP Support Portal 2010b, 19)

Functional upgrade

An important consideration for version upgrades is the new functionality offered by the target release. The primary goal of this approach is to achieve business benefits and improve operational excellence (SAP Support Portal 2010b, 19). The functional upgrades will become more common now that SAP has disconnected the technology layer from the functional layer (Sens 2008, 91). Instead of implementing a full new upgrade version, functional upgrade of SAP ERP can be implemented nowadays by activating new business functions provided by SAP enhancement packages (see 2.3.5).

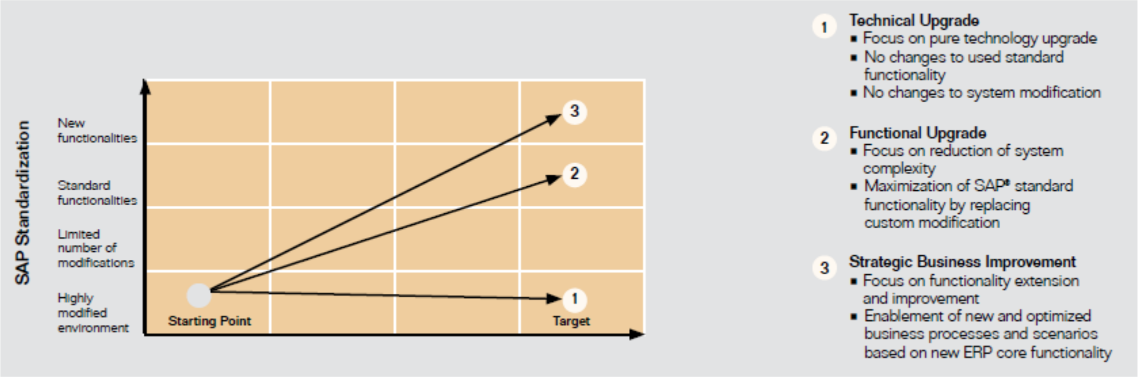
This upgrade starts with a technical upgrade but on top incorporates additional business functionality. Even though it is possible to combine technical and functional upgrade in a single project, SAP recommends that the implementation of the new business functionality is performed after a technical upgrade. This way the overall stability of the system can be first ensured. (Riedel 2009, 37.) Most organizations seem to follow this advice, because according to SAP's own statistics more than 80% of clients approach functional upgrade as a separate project and according to Panaya (2010a, 11) 74 % of organizations plan doing functional or strategic upgrade, but as a second phase following the technical upgrade.

A functional upgrade provides also the opportunity to reduce the system complexity thus simplifying upgrades and reducing costs in the future. This is carried out by clearing the system of unused client-specific modifications and custom developments or replacing them where possible with standard SAP ERP functionality (SAP 2007, 7; SAP Support Portal 2011b, 38; SAP Support Portal 2010b, 19).

Strategic upgrade

Strategic upgrade (or Strategic Business Improvement Upgrade) approach involves major reengineering of business processes (SAP 2007, 7) and implementation of new strategic business software or components on the basis of SAP ERP (Riedel 2009, 38). Scope is very much dependent on the business needs of the organizations. This approach is driven by need to exploit SOA and to implement new enterprise services that should offer greater flexibility to optimize business processes e.g. by simplifying daily business tasks by creating role-based user interfaces (SAP Support Portal 2010b, 19). The aim is to build a business process platform that provides more flexibility to business process innovation and enable latest technologies. For example, the organization can integrate heterogeneous environments and legacy systems with SAP ERP. (Riedel 2009, 38.) SOA also enables organizations to shorten application innovation lifecycles and implement strategic business enhancements at their own speed (SAP Support Portal 2011b, 38).

As explained above, most of the new opportunities provided with SAP ERP are usually implemented in subsequent projects after the completion of the technical upgrade. Hence, the technical upgrade is also required before the strategic upgrade.


SAP Standardization
New functionalities
Standard functionalities
Limited number of modifications
Highly modified environment
Starting Point
Target
1
2
3
1 Technical Upgrade
▪ Focus on pure technology upgrade
▪ No changes to used standard functionality
▪ No changes to system modification
2 Functional Upgrade
▪ Focus on reduction of system complexity
▪ Maximization of SAP® standard functionality by replacing custom modification
3 Strategic Business Improvement
▪ Focus on functionality extension and improvement
▪ Enablement of new and optimized business processes and scenarios based on new ERP core functionality

© Copyright 2007. SAP AG. All rights reserved.

Figure 16. Possible upgrade approaches. (Source: SAP 2007, 7)

2.3.7 Cost and effort estimate

Cost

Organizations go through great efforts and costs each time a new version upgrade is needed. As mentioned earlier in the chapter 2.1.2, ERP maintenance costs a significant amount of money. The general rule is that as the organization grows, the number of users goes up along with the total cost of upgrade. In addition, ERP solution needs to be more comprehensive when companies get larger. Exact numbers of ERP version upgrade costs are hard to come by because the total effort to fully complete an ERP upgrade project varies based on technical, organisational, and operational factors and what type of upgrade approach (technical, functional, strategic) the organizations decides to adopt. On average however, (based on KPLC, KENGEN and Bamburi estimation) an ERP upgrade costs 30% of the initial ERP implementation project cost and can take more than a year to complete as organizations restructure their business processes and update their technology infrastructures (Otieno 2010, 157). Ng, Gable and Chan (2003b, 234) point to similar estimates. Martinsen (2010, 17) refers to a AMR research from 2004 which states that the upgrade cost include near 50% of the software license fee plus 20% of the original implementation cost. Not surprisingly, Panaya (2010a, 17) reports that the median projected budget increases with the number of the production users, from $175,000 for organizations with less than 100 users to $2M for organizations with over 10,000 users. In any case, the cost of ERP upgrade is continuous as upgrades typically happen many times during the ERP lifecycle (Zhao 2007, 4).

The cost of upgrading to a new release depends on a number of factors, including the extent of modifications, complexity of ERP system landscape, complexity of functionality, interface, skills and technical requirements, therefore exact calculation are hard. In addition, the chosen upgrade approach affects the total costs, e.g. functional upgrade may require longer implementation time as well as more user training and assimilation of new business processes. Hardware and technical environment changes may also be required. Holm and Mattson (2008, 10) say that organizations find it easiest to measure direct costs such as capital and labour costs but have difficulty measuring indirect costs such as system downtime. Therefore it is not surprising that according to recent survey by SAP 69% of organizations said that cost and effort estimates are the major challenges they face during the upgrade planning phase (SAP Support Portal 2011b, 32)

Some of the indicators and examples used in a cost estimate may be:

- Costs of internal IT and business staff
- General project and change management aspects (involves about 10% of the total project effort)
- Time for general project application adjustments (for example, one hour for each simple adjustment of SAP software modifications or custom developments)
- Testing effort (involves up to 40% of total project effort, depending on the degree of application adjustments and the maturity of existing test procedures)
- Training effort, if applicable – depends on the number of users affected, the training concept used (classroom versus e-learning), and the extent of functional enhancements
- Costs for external assistance (e.g. IT consultants)
- Additional hardware costs, if applicable (for example, server adjustments, hard disk, or memory)

- Software costs - additional license costs for SAP ERP or third-party software, if applicable and if the costs were not already covered by maintenance and service .
- Down time costs – costs and losses caused by non- availability of the production system, for example, in the case of 24/7 production.

(Riedel 2009, 99; SAP Support Portal 2011b, 38-39.)

Duration

Even if ERP might eliminate the need to upgrade many separate enterprise applications an upgrade is still complicated and time demanding. According to Vaman (2007, 313) major application-release upgrades require on an average three-to-six months to execute and two-to-four person-years of effort. Martinsen (2010, 17) and Zhao (2007, 12-13) cite sources that estimate the ERP upgrade duration between eight and nine months with a team equivalent of one full-time employee per 35 business users. Zhao emphasizes that lack of experience may cause the upgrade project to prolong. According to SAP upgrade experience database, the average duration of a technical upgrade to SAP ERP 6.0 is about three to five months (SAP Support Portal 2011b, 37).

As expected, the project time effort also correlates to the number of users. According to Panaya[2] (2010a, 3) upgrade effort is highly correlated to the number of production users. The median effort in person days for organizations with less than 100 users is 147 person days, while the median for organizations with over 10,000 users is 2,500 days. The upgrade effort is also highly impacted by the same factors than cost.
In fact, according to Panaya the use of industry solutions almost doubles project effort. Therefore, the Panaya survey respondents reported overall project duration ranging from 5 weeks to 120 weeks.

[2] Panaya is a consulting company and not a neutral research organization, although it is not affiliated with software vendors. This does not invalidate the results, but as a policy matter we should retain some skepticism toward the findings.

The more changes have been built into the SAP ERP system (e.g. through modifications, custom developments and customizing), the higher the cost and effort required to upgrade (Riedel 2009, 59). How much more cost these alterations add to the total cost of upgrade is difficult to precisely measure. Vaucouleur's (2008, 7) empirical survey points to a range between 10 and 15 per cent of the original alteration cost for a single upgrade, but these numbers are derived from informal discussions with ERP practitioners.

When estimating cost and effort customers may draw on experience from previous upgrade projects or on benchmark data provided by SAP (such as Upgrade experience database) or its partners (SAP Support Portal 2011b, 37-38). SAP as well as a number of other SAP partners (like Panaya) offer also upgrade tools and services that can help to estimate the effort and cost required for an upgrade. These services often include a system analysis using an automated tool that can extract data from the source system and perform an analysis of the upgrade effort. (Riedel 2009, 101.)

2.3.8 Technical considerations

In most cases, it is necessary for organizations to make adjustments to their IT infrastructure for an upgrade. In general, the impact on existing IT infrastructure depends on the following criteria:

- Upgrade path (the difference between the current and target release)
- Scope of functional enhancements in the target release
- Extent of utilization of existing IT infrastructure (e.g. number of users, data load)

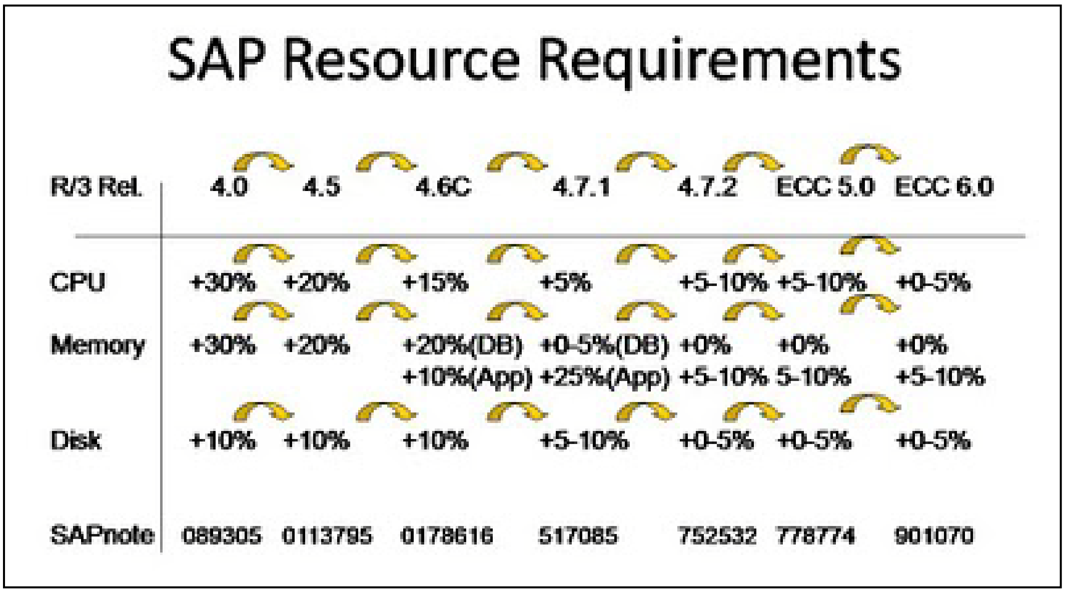
(SAP Support Portal 2010b, 25.)

These adjustments may include resizing the application and database servers, deploying new front-end components, making network adjustments to maintain system performance, upgrading or migrating the operating system and database platform, and converting to Unicode (SAP Support Portal 2011b, 44).

Hardware and sizing

For most customers, an upgrade to SAP ERP 6.0 will require an increase in CPU power and memory (Riedel 2009, 67), if the current hardware does not happen to be currently oversized and able to handle the workload of the new version (Sens 2008, 105). In fact, the general assumption that is also expressed by SAP is that every new SAP ERP version takes more computer resources, such as CPU, main memory, and disk space. When upgrade consists of a significant release jump, the new resource consumption is the sum of all steps. The table below shows the increase of resource requirement per upgrade version. For example, in case an upgrade is performed from source release 4.0 to target release 6.0, the new resource consumption of CPU may be as high as +65%.

Table 3. Additional hardware requirements per SAP ERP upgrade version. (Source: Boler Consulting 23.3.2008)



SAP Resource Requirements

R/3 Rel.	4.0	4.5	4.6C	4.7.1	4.7.2	ECC 5.0	ECC 6.0
CPU	+30%	+20%	+15%	+5%	+5-10%	+5-10%	+0-5%
Memory	+30%	+20%	+20%(DB) +10%(App)	+0-5%(DB) +25%(App)	+0% +5-10%	+0% 5-10%	+0% +5-10%
Disk	+10%	+10%	+10%	+5-10%	+0-5%	+0-5%	+0-5%
SAPnote	089305	0113795	0178616	517085	752532	778774	901070

SAP recommends that the organization should involve its hardware partner early in the upgrade project to assess and adjust the hardware requirements. It should also be noted, that additional hardware requirements exist for migration a non-Unicode system to Unicode. (Riedel 2009, 68.)

Unicode

Unicode is a character-encoding schema containing nearly all characters used worldwide. An application that uses the Unicode character schema is able to represent all languages and its characters (Sens 2008, 187-188). Therefore, SAP encoded the Unicode standard in the SAP ERP 6.0 release and different code pages and multidisplay, multiprocessing (MDMP) are no longer required like in the past. In order to make use of Unicode, the database needs to be converted to Unicode first (Sens 2008, 188). So if an organization employs global business processes or manages global master data, or if it plans to open the system to the Internet by allowing customers to enter contact data directly, it most likely should consider transition to Unicode (SAP Support Portal 2010b, 23-24). Unicode conversions should not be incredibly difficult, but they are time and resource intensive (Reed 2010, 2). Therefore an organization converting to Unicode needs to consider additional sizing requirements. Especially particular requirements exist for database hardware. Additional storage requirements depend on factors such as database Unicode encoding scheme, settings and compression, but for example for DB2 UTF-8 requires ca. 10% and SQL Server UCS-2 40-60% more space. Additional CPU and memory requirements also exist for Unicode. An average increase of 10-30% for CPU and 40-50% for main memory. (Riedel 2009, 69.)

Interfaces to SAP applications and third-party products

Typically, a number of other SAP solutions are also part of an SAP software landscape, for example SAP CRM, and SAP SCM. These applications are usually interconnected, with business processes running across and through them, therefore when planning an upgrade, an organization must know whether the upgrade have impact on other software in the landscape. (SAP Support Portal 2010b, 28.) In general, there are no dependencies (SAP Support Portal 2011b, 32) – but if there are, SAP provides tools such as upgrade dependency analyzer that informs of any known upgrade dependencies between the SAP solutions (SAP Support Portal 2010b, 28).

Third-party products are also subject to the SAP version upgrade, such as toolkits for archiving, auditing, security, forms printing, and monitoring. For each product, a release support certificate needs to be obtained from the vendor. If an upgrade of third-party product is required, it is important to determine where during the upgrade process this software upgrade can take place. (Sens 2008, 115) It is also important to document all interfaces always so that they can be checked during the upgrade project (SAP Support Portal 2011b, 47).

Database or operating system upgrade

In most cases, the SAP target release is also bound to a higher release of the RDBMS database version or operating system (Sens 2008, 111; Riedel 2009, 69).

Country-specific functionality

Not all country-specific functionalities are part of the SAP ERP standard product, but are instead provided as separate add-ons either by an SAP country subsidiary or by an SAP partner. In case such an add-on is used in the current system, it should be checked if the functionality will be available for the new version and when. (Riedel 2009, 69-70.)

Languages

The organization needs to pay special attention also if, besides English and German, additional languages are installed. In case, there are language functions that are not part of the SAP repository but are SAP name range language additions or other customer modifications, they are affected by the upgrade. If, during the upgrade process, these languages are not included into the upgrade, these languages are removed from the SAP ERP system. (Sens 2008, 136-137.)

Deploying New Front-End Components

SAP recommends that organizations use and deploy the latest release of its GUI. This might require an upgrade of hardware or the operating system of user PCs. (SAP Support Portal 2011b, 44.)

Adjusting the Network to Maintain System Performance

To avoid performance bottlenecks, an organization should consider if any network configurations are required. SAP publishes Front-End Network Requirements for all SAP Business Solutions. This white paper summarizes key recommendations for an optimized network configuration. However, it is strongly recommended that the organization conducts measurements on the system to perform adequate network sizing. (SAP Support Portal 2011b, 45.)

2.3.9 Other important considerations related to SAP ERP upgrade

Modifications

A key consideration when upgrading is how much the organization has modified the SAP ERP software. Customer-created modifications - that is, changes made to SAP repository objects – directly affect the complexity of an upgrade, because modifications are overwritten during upgrades and the ones that are still needed after the upgrade have to be adjusted. (Riedel 2009, 82-83) Therefore the following must be considered:

- The upgrade process for a modified SAP ERP implementation is more time consuming than an upgrade of an ERP implementation with no modifications;
- The timeframe and cost of upgrades increases exponentially with the number of modifications due to validation and testing;
- Modifications are not supported by the vendor and must be maintained and updated by the organisation.

(Otieno 2010, 154-155.)

Each new release requires the organization to evaluate the effects of modification and do to adjustments repeatedly. Therefore in the first place, organizations should avoid over-modifying their SAP ERP software. In fact, most academics and practitioners agree that the system should be implemented as vanilla as possible, i.e. as close to "out of the box" and with as few modifications as possible. Others however argue that vanilla implementation, in which an organization adopts vendor's industry 'best practices' embedded within the ERP, homogenizes business processes too much and eliminates competitive advantage. (Worrell 2008, 3.) Secondly, since the modifications to be carried over from one version to the next are the biggest technology headache that organizations face in upgrades, organizations should carefully review the new functionality offered in the new target release to determine if some of the modifications could be eliminated (Otieno 2010, 159). Furthermore, they should abstain from adding any new modifications (post-implementation modifications) to the target release.

An organization should always document any done modifications. These previous documentations serve as foundation for the new upgrade. The reason for the modification should always be specified, otherwise it is hard to determine if the modification is still needed in the new version. In the worst case, the upgrade will include modifications that are not needed and hence affect the project in a negative way. (Martins 2010, 22.) SAP's own estimates indicate that up to 60% of modifications are not even being used (SAP Support Portal 2011b, 47).

Modifications that are still needed after the upgrade must be adjusted. This basically means, that all the customer-initiated code previously written has to be re-modified and tested to ensure it works in the new version. And that takes time. SAP estimates that 30 % of total upgrade time will be spent updating modified code and approximately as much more time is required for testing (SAP 2007, 9). In Otieno's (2010, 159) research the company KPLC had very heavily-modified ERP software, and its upgrade project required approximately 80% of software developers' and 66% of a business analysts' time and effort. SAP discourages modifications and they are generally not supported under service agreements and SAP will not support any problems they cause with the system. Therefore, organizations should also make sure, that there are skilful coders available to handle the re-modifications. If the organisation's modification is a kind used by multiple SAP ERP clients, there is the possibility that the SAP could be persuaded to include this in their next product release but this could require significant negotiating leverage and is uncertain (Otieno 2010, 155).

In order to avoid potential errors that may affect key application functionalities and disrupt critical business processes, SAP ERP users are required to perform comprehensive tests before going live. Traditionally, organizations have been dealing with this challenge by using a manual trial-and-error approach to test their modifications. This means that all change-prone objects have been identified in the development environment or sandbox, then re-modified to comply with SAP ERP 6.0, and then all adjustments tested before proceeding to QA and production. With this traditional method a vast amount of time is required, as well as development and testing resources. (Panaya 2011a) There are also automated tools provided by SAP and SAP's partners that are supposed to identify and also handle the code that needs re-modification before the upgrade can be smoothly performed (Riedel 2009, 83).

Authorizations

It is important to consider that an upgrade may have impact also on authorizations and other security issues within the upgrade system environment and therefore require both careful planning and testing. For example, new functions and protocols (such as access via HTTP), or a new operating system or database software might cause this. (Riedel 2009, 84-85.) Authorization roles are user permissions to perform certain activities in an SAP ERP system. For example, new version may include new authorization requirements and old authorization roles will require adjustments simply to be able to provide the same accessibility as in the source version. Typically, adjustments are required in the areas such as financial, vendor/ customer management, and DMS (document management system). (Panaya 2011a.)

Training

Depending on the scope and approach of the upgrade project, the organization might have to provide training for users. In case of functional upgrade, power users and end users have to learn the new business functionality in order effectively assimilate the new release. But the organization needs to ensure also that the project team (project manager, administrators, business process experts, developers and database experts) is adequately trained before they can participate in the upgrade project. Training is important to ensure the success of the entire upgrade project. For this reason, it is important to develop an appropriate training plan to ensure that employees understand the changes to their business processes and job responsibilities. (Riedel 2009, 187-189.)

External assistance

If organization's internal resources are limited or the staff lacks the skills required for an upgrade, the organization should consider hiring external assistance to keep the project on time and minimize the risk. Often external partners have experience and special skills in upgrade projects, e.g. upgrade project management, testing and quality assurance, verifying modifications. The downside is increased costs when compared to the internal resources. (Riedel 2009, 86-89.)

2.4 Managing SAP ERP upgrade project

SAP ERP upgrades can be complex and demanding, but according to Sens (2008, 87) an upgrade project is like all other projects—it's all about skills, commitment from the business, and good project planning. In the scope of this research, the project management cannot be discussed in detail, but this chapter highlights some essential considerations.

2.4.1 Building a project team

The success of the upgrade project depends heavily on the project people and the way the project is organized. Project team should include members with several skill sets and different backgrounds (Sens 2008, 197). The number of people in the project team depends on a number of factors: the usage of the SAP ERP application in the company, system landscape complexity and, of course, the number of locations and countries involved.

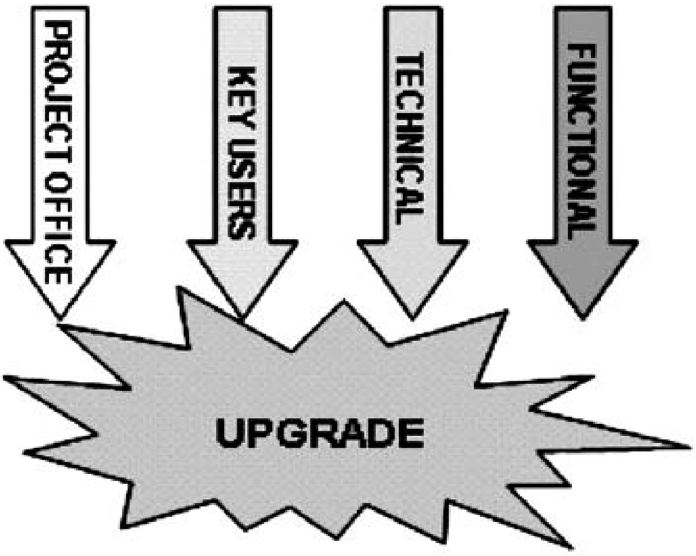


Figure 17. Different skills are required during an SAP upgrade project. (Sens 2008, 197)

In most organizations, projects are sponsored by a steering group and lead by one or more project managers (Sens 2008, 208). ERP upgrade project should always have the support from the top management (Wenrich & Ahmad 2009, 58). The steering group monitors the project, provides support and helps to ensure that the project stays within the organization's strategic goals.

The project manager is the head of the project, who manages and coordinates all team members, is responsible that project runs smoothly and reports to senior management (Riedel 2009, 119). Planning the project and its budget is also responsibilities for the project manager (Sens 2008, 206). The project manager should be experienced in managing complex ICT or business affecting projects (Sens 2008, 197).

Especially in case of a functional upgrade, when upgrade has major impact on the business processes, strong involvement of the business is required. Key users and business process experts have broad business process knowledge; therefore they are the ones who should design business processes. These business process experts should also be responsible for integration testing and coordination, as well as the performance of user acceptance tests in their functional area. In addition, they should take lead in defining functional training requirements for the end users. (Riedel 2009, 120.) Each business process that is implemented in SAP should have an owner. This "Business Process Owner" (BPO) is directly responsible for the way the business process has been implemented in SAP ERP. If customizing and objects that represent business processes need to be changed and tested, it is the process owner who should initiate this. (Sens 2008, 221.)

Technical people manage the IT aspects of the upgrade across the system landscape. They are responsible for installing upgrades and managing SAP ERP system landscape during the entire upgrade project, including hardware, operating systems, other SAP software, third-party software and interfaces, and database software. They also develop and maintain custom developments and modifications. (Riedel 2009, 120).

Depending of the upgrade approach and scope, an upgrade project can last for several months. Therefore it should be ensured that team members, internal and external, are available when they are required during the upgrade project.

2.4.2 Standards and procedures

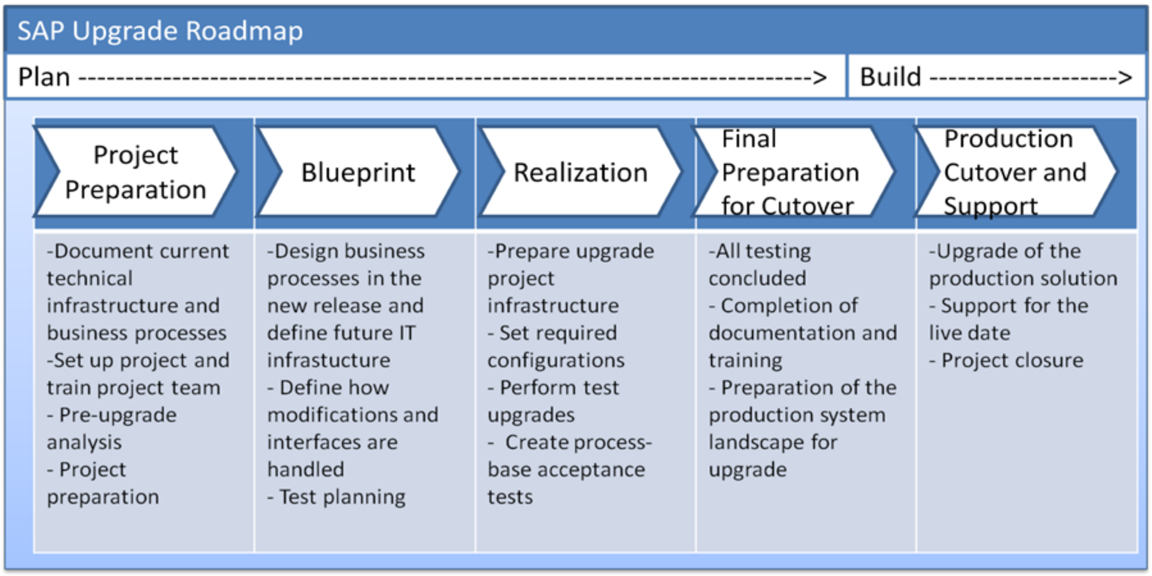
To facilitate the efficient and transparent execution of critical project tasks and overall progress of the project, the following comprehensive standards, procedures, and project elements are necessary:

- Standards for project documentation, problem solving, progress tracking, and procedures for escalation
- Project approach and scope (for example, technical upgrade only)
- Roles and responsibilities of the internal and external resources involved
- Setup of project landscape and code freeze procedures
- Guidelines and procedures for testing and training strategy.

(SAP Support Portal 2011b, 37.)

SAP recommends its own SAP Upgrade Road Map methodology to successfully complete an upgrade project. The SAP Upgrade Road Map is a detailed guideline for project managers to plan and execute all relevant activities of a typical SAP ERP upgrade. It provides best practises for basic project management and functional and technical aspects to upgrade an entire SAP ERP system landscape (Riedel 2009, 109-110). Its checklists and templates can be used to accelerate, optimize and tracks the primary project activities. It also defines activities that are directly or indirectly related to testing (Riedel 2009, 133). The upgrade roadmap basically follows the same sequence as the SAP ASAP methodology, which is used for SAP implementations (Sens 2008, 283). The SAP Upgrade Road Map content is delivered with SAP Solution Manager application, but there is also offline version available that does not allow access to any application-specific upgrade toolbox (Riedel 2009, 206). Offline version is available at http://service.sap.com/upgraderoadmap.

The road map defines five phases the upgrade project follows (see figure below).



SAP Upgrade Roadmap

Plan --> Build ------------------->

Project Preparation	Blueprint	Realization	Final Preparation for Cutover	Production Cutover and Support
-Document current technical infrastructure and business processes -Set up project and train project team - Pre-upgrade analysis - Project preparation	-Design business processes in the new release and define future IT infrastucture - Define how modifications and interfaces are handled - Test planning	-Prepare upgrade project infrastructure - Set required configurations - Perform test upgrades - Create process-base acceptance tests	-All testing concluded - Completion of documentation and training - Preparation of the production system landscape for upgrade	-Upgrade of the production solution - Support for the live date - Project closure

Figure 18. SAP Upgrade Road Map. (Adapted from SAP Support Portal 2011b, 42)

In the first Project preparation phase, the present state of the system landscape is analyzed, including the systems that should not be upgraded. All the documentation from the initial ERP implementation has to be collected and updated if necessary, including business processes affected by the upgrade. Also consistent picture of the custom developments and modifications is created. During the first phase project plan is prepared that includes activities, resources and budget. (Riedel 2009, 212.) The second Blueprint phase is used to perform a gap analysis between the current and target SAP release functionality. In this phase is also defined how modifications and interfaces are handled and if existing test cases can be used or are new test cases required (Riedel 2009, 212-213). During the third phase, a couple of test upgrades are performed and all additional configurations for the new release are done. Process-based test plans are created. In the final preparation and cutover phase, system and integration testing is concluded to ensure that business processes run smoothly after the upgrade. Training materials are finalized. The fifth phase includes the actual cutover of the productive environment to the target release.

2.4.3 Upgrade success factors

Identifying risks related to an upgrade project can be a challenge for managers, especially because there are different ways in which they can be described and categorized. Often terms as ”risk factors”, “critical success factors”, ”best practices”, ”upgrade project errors”, and ”pitfalls” are used to convey the same concept. So the following list of upgrade success factors homogenizises that kind of factors found in the literature influencing SAP ERP upgrade. The list is not an all-inclusive list, but is based on ERP upgrade literature review and it identifies many of the factors that could threaten the successful outcome of an upgrade project. Some of the factors in the list have been already discussed in earlier chapters of this research and factors regarding the testing will be discussed in the next chapter, therefore it should be considered in conjunction with what has been said in other parts of this research.

Release maturity. Organization should not adopt the new release until it is mature enough. Wait at least until the first major software support package of cumulative fixes is released, which usually takes six-to-five months following initial release availability. Otherwise, you risk wasting time on endless cycles of bug fixes and rework involving the vendor, project team and consultants. (Search Manufacturing ERP 3.8.2010; Vaman 2007, 314)

Upgrade timing. Arrange the timing of the upgrade so that the disruption to business is minimal (for example, avoid disruption of the year-end accounting cycle). Plan upgrades to avoid loss of vendor-version support, based on vendor-support guidelines. (Vaman 2007, 314.)

Top management support. Approval and support from top management and steering group is very important for the success of the project, because they provide leadership and necessary resources (Martinsen 2010, 32; Zhao 2007, 40).

Business driver. Align the upgrade strategy to business objectives in order to highlight the business value that can be realized . The organization should look for new functionality that the vendor has developed since the last upgrade and leverage full potential of these capabilities to improve inefficient business processes. (Kimberling 26.7.2010; Vaman 2007, 315.) Even a technical upgrade should have business drivers to justify it, such as instance and database consolidation (goal: cost savings), the organization should refrain upgrading just to avoid extended maintenance fees from SAP (Reed J. 2010, 1). It may be difficult to obtain top management support for an upgrade with no business case, because they usually want to see return on the upgrade investment (Wenrich & Ahmad 2009, 62).

Project scope. Project scope should not be too extensive. The scope of the project and the number of changes introduced should be limited and after the project has started it is important to resist adding elements to the upgrade that fundamentally change the project scope. (Riedel 2009, 150, 158). The organization should decide in advance whether their upgrade will be just a technical upgrade or one that includes major new functionality as well. If it is just a technical upgrade, the project team will be mostly people from IT, with some user involvement for testing. But if the organization is changing or adding new business processes, the scope of the project will be much greater, and the business process experts must be included from the start. (Paul 19.1. 2009.)

Change management. During the upgrade project it is important to define and adhere to reliable change management procedures for custom developments and modifications if they have to be incorporated to the system during the upgrade project. Namely, larger companies cannot afford to have too many weeks of code freeze due to requirements from the business side. Therefore, it is vital to define when and how these customer-specific developments are implemented and when there is an explicit code freeze (no changes are allowed until the productive instance goes live). (Riedel 2009, 155-156; SAP Support Portal 2011b, 37.)

Adequate planning. Upgrade project should be planned as carefully as the initial implementation (Riedel 2009, 157). Even organizations whose ERP implementations were triumphant often find themselves facing problems during upgrade because they did not follow the same careful process (Wenrich & Ahmad 2009, 61). A thorough plan concentrates to develop realistic costs and timeframes and to minimize system downtime. A complete inventory and analysis of application modifications, interfaces, output and data conversion requirements should be performed as a basis for developing a detailed upgrade plan, and determining effort levels and resource requirements. (Vaman 2007, 314.) The upgrade plan should also include the resources required to accomplish each milestone. All team members should be listed, including those whose involvement is limited. (Paul 19.1. 2009.) Project team should be able to provide sufficient capacity and a suitable skill set (SAP Support Portal 2011b, 37). Upgrades can be never over planned. Very often it is small things that can cause problems, such as running out of disk space when making copies of the instances e.g. for testing (Paul 19.1. 2009).

Project management. Powerful and excellent project management regarding people, equipment and materials and control over technical, cost and time constraints is a key to the success of any ERP upgrade project. Project management activities span from the beginning of the project to the closing of it. Part of the powerful management is that the upgrade project is well documented and appropriate project management methodologies and tools are used. (Zhao 2007, 34, 40.) It is important to agree on project standards, procedures, milestones and guidelines and adhere to these (SAP Support Portal 2011b, 37). Furthermore, project management should try to manage resistance towards change in the current system (Martinsen 2010, 32).

Communication. Communication is one of the most exacting and difficult tasks in any ERP project (Martinsen 2010, 32). A successful ERP upgrade needs to actively engage IT staff, vendors, users, and executives to work together and share information and knowledge (Law & Chen & Wu 2010, 300). One major purpose of communication is to inform the whole organization about the expectations or goals of the change - why change is needed, what the new ERP system will look like, and what will happen if change does not occur (Zhao 2007, 24). Furthermore, it is essential to share information about results and progress in each project phase between project members and the organization (Martinsen 2010, 32). It is particularly important to ensure that everyone involved is well informed about key dates and the duration of the upgrade project (Riedel 2009, 151).

Modification reversal. During an upgrade, it is important to evaluate any modifications made to ERP source code. Analyze the upgrade version's improved functionality. Determine if new, delivered capabilities can replace all or some custom ERP source code modifications. Subsequent upgrades will be less error prone, easier to manage and the SAP ERP will be less costly to support when modifications are minimized. (Vaman 2007, 315; Wenrich & Ahmad 2009, 62.) If the organization decides to keep a lot of modifications in the existing system, it should not assume that they will work in new version without problems (Reed 2010, 2). Kept modifications will require a lot of recoding and testing, therefore the time and effort required should be reviewed in the project plan accordingly (Paul 19.1. 2009).

External support. Experienced upgrade consultants can be used to accelerate the upgrade process, supplementing internal resources. In addition, upgrade consultants can reduce the risks of cost overruns, business disruption and performance problems. (Vaman 2007, 315.) Consultants can have good experience in a certain field of industry, detailed knowledge about the SAP ERP itself or for instance about the upgrade project management. Consultants can be hired to accompany a certain time period or help with different stages of the upgrade project. (Zhao 2007, 24) The organization should conduct thorough evaluations of possible partners and choose those that have experience and customer references in that particular industry and version of SAP ERP. (Reed 2010, 1.)

User involvement. It's so evident that it almost a cliché, but any ERP project will be doomed to failure without solid user buy-in (Paul 19.1. 2009). Inviting end users of the upgraded system to participate in the project activities will help users to accept the new version smoothly (Zhao 2007, 40; Martinsen 2010, 32). User input is also very important in assessing the functional scope of the project (Riedel 2009, 152).

Training. The quality and availability of training is another critical factor to the success of upgrade activities. (Law & Chen & Wu 2010, 300) Providing enough training to employees will help them to adapt the new system environment after change (Zhao 2007, 40). Strategic goals for upgrade will go unrealized if the users are not able to use the system correctly (Wenrich & Ahmad 2009, 62). However, it is not only end users who need training, but also those in lead roles needed for the upgrade. It should not be assumed that employees that are experts in older SAP ERP versions are able to lend a useful hand on an ERP 6.0 without prior training on new tools like Solution Manager. (Reed J. 2010, 3.) Functional upgrades can cause changes in addition to business processes also to organization's job descriptions. Such change is even more pronounced if the organization has not upgraded in several years. For this reason, it is important to develop a comprehensive organizational change management and training plan to ensure that employees understand the changes to their business processes and job responsibilities. (Kimberling E. 26.7.2010.)

Adequate testing. When facing a tight budget, organizations attempt to eliminate testing. Organizations should expect to spend 25% of the project schedule on testing. (Wenrich & Ahmad 2009, 62.) The organization should plan to have a similar test range like in a new install, including unit test, user acceptance tests, usability, performance and integrations tests (Search Manufacturing ERP 3.8.2010; Vaman 2007, 315). All changes to the system should be tested as thoroughly as possible so that their impact can be assessed before they are transported to the production instance. Therefore, it is important to invest in additional hardware and software to support proper testing environment . (Law &, Chen & Wu 2010, 300; Reed 2010, 3.) Also the testing the technical upgrade process repeatedly is important. Repeated upgrades help to prepare an upgrade ”script” (also known as “runbook”), which can be used perform the upgrade smoothly when the production cutover is reached (Riedel 2009, 158).

Upgrade tools. Use upgrade support tools provided by software vendor to plan, control and accelerate the process. Such tools are useful e.g. for analyzing custom developments and modifications, required system sizing and for migrating data from one version to the next. (Riedel 2009, 220-221; Vaman 2007, 315.) For SAP ERP invaluable tool throughout the upgrade project is e.g. SAP Solution Manager and the integrated SAP Upgrade Road Map.

2.5 Executing SAP ERP upgrade

This chapter looks at the execution phase of an upgrade. Based on two key upgrade challenges, its focus is on system landscape management and downtime minimization during an upgrade.

2.5.1 Managing the system landscape during an upgrade project

The SAP ERP system landscape design alters during the upgrade process and therefore good landscape management is required. Landscape management consists of a clear plan that tells exactly in what order the systems are upgraded and what the status is of a certain system during the upgrade period. (Sens 2008, 215.) As described in the chapter 2.1.2 a typical SAP ERP landscape consists of three instances: development, quality assurance and production. If an organization has other instances, it is recommended that you upgrade the system landscape in the same sequence in which the organization transports the changes and modifications to the production system (SAP Support Portal 2011b, 37).

Below the outlines are provided for how to set up and manage this kind of standard three-system landscape in an upgrade project in order to minimize upgrade risk and minimize the duration of the code freeze period. The code freeze period is the time during the upgrade when no changes (configuration changes, SAP notes, modifications or enhancements) are allowed to basic system until the productive system goes live (Riedel 2009, 155-156). The code freeze period will usually start after the development system has been established (Riedel 2009, 161). There are other ways to build the upgrade system landscape, but the illustrations below should help to understand the basic setup and how the landscape evolves throughout the upgrade process – how each instance is upgraded to the new version and the transport routes. Typically additional copies of systems are used to perform required activities, such as

- Provide support to the production system during the entire upgrade project.
- Possibility of a rehearsal of the upgrade before the actual production instance is upgraded.
- Ability to test all implemented business processes and interfaces in the new target release.
- Ability to test and manage all customer-specific modification and custom developments. (Sens 2008, 216.)

Phase breakdown below follows the phases of the SAP upgrade road map.

I Project Preparation Phase

The main purpose of an upgrade project is to upgrade the production instance. Therefore, it is recommended to prepare a sandbox system (UPS) as a copy of the productions system (PRD) first. Project activities during this phase are:

- ➢ Project management creates a detailed project plan, names the project team members and orders temporary hardware that is needed for upgrade project.
- ➢ Technical team prepares the sandbox system.
- ➢ Developers identify custom developments and modifications.
- ➢ Business people study new business functionality and start preparing test scenarios and planning test execution. (Riedel 2009, 162-163.)

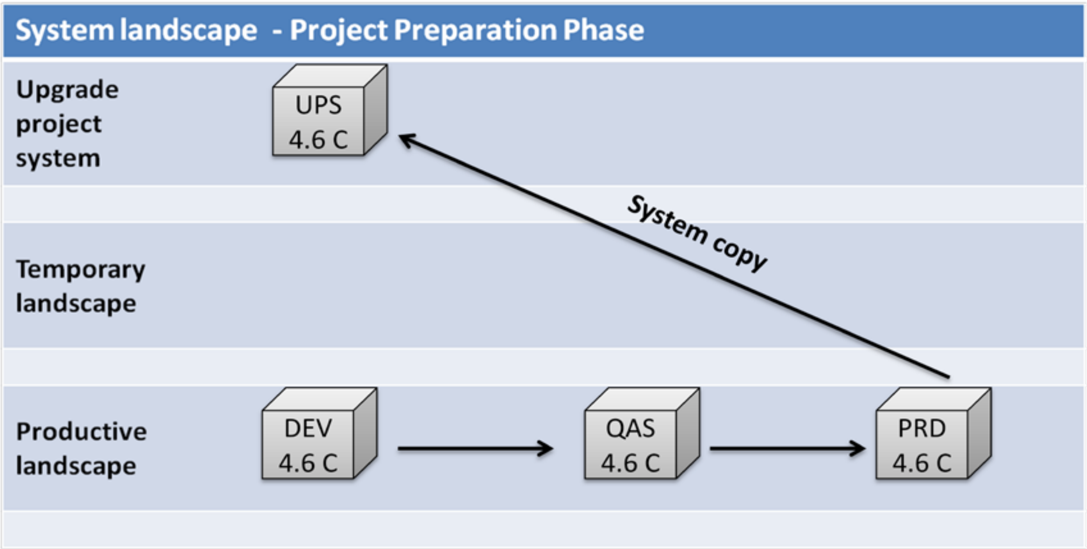


Figure 19. I Project preparation – duplication of the PRD instance. (Adapted from SAP Support Portal 2010c, 33)

II Blueprint Phase

In the upgrade blueprint phase the focus is on familiarization and testing of the new version in the sandbox. Project activities during this phase are:

- Technical team performs a technical upgrade to the UPS, but it should be noted that the UPS will not become part of the base SAP system landscape.
- Developers adjust custom developments and modifications.
- Business people carry out upgrade customizing and testing of business processes.

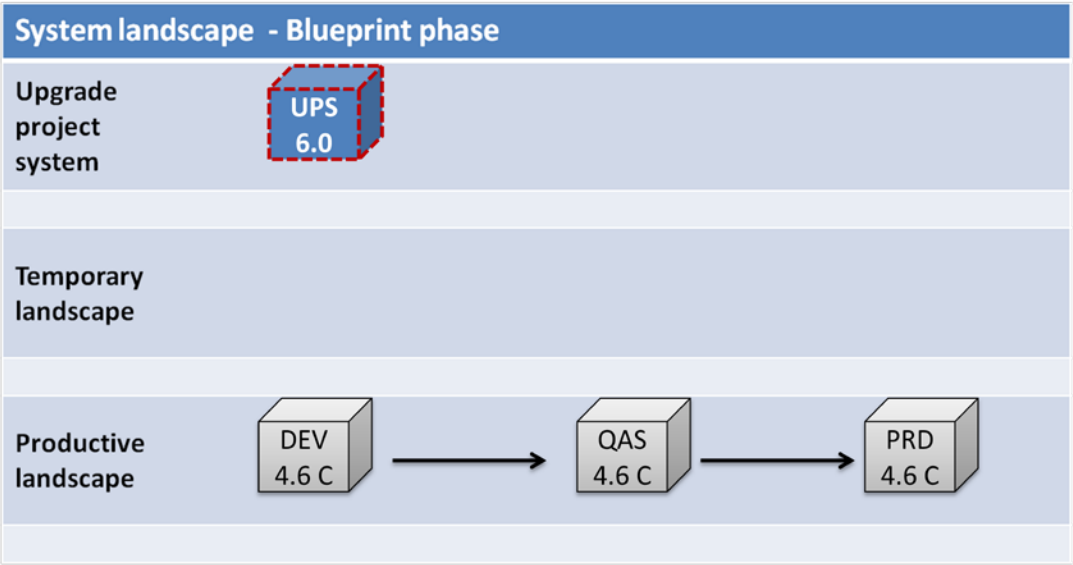


Figure 20. II Blueprint phase – upgrade of the sandbox instance. (Adapted from SAP Support Portal 2010c, 34)

At the end of this phase, business processes should be running properly in the sandbox system. There should also be detailed documentation of all actions performed and development team activities. Project management should refine project plan accordingly. (Riedel 164-165.)

III Realization phase

In the realization phase the development instance DEV is duplicated to DEV'. The base DEV is upgraded to the new version. From now on the base DEV will be the instance where the actual project work takes place (not sand box anymore), but the copy DEV' is required in order to support the production system in case of any problem. Imagine a situation where DEV was already upgraded to a new release and at the same time a serious error would be discovered in the PRD, which is still in the source release. As a rule all fixes are done first in DEV, then transported to QAS and from there to the PRD, but if the source release DEV' would not exist, it would not be possible. This is why a copy of source release DEV is required. It is important to consider a code freeze and abstain from any other changes to the system than unavoidable corrections. Otherwise, dual maintenance is required - all changes that are made to the DEV' must also be made in the upgraded DEV. Project activities during this phase are:

- Technical team first duplicates DEV to DEV' for maintenance and after that upgrades DEV to new version.
- Developers re-adjust custom developments and modifications and perform short unit testing.
- Business people carry out upgrade customizing and perform unit testing on business processes. (Riedel 2009, 165-167.)

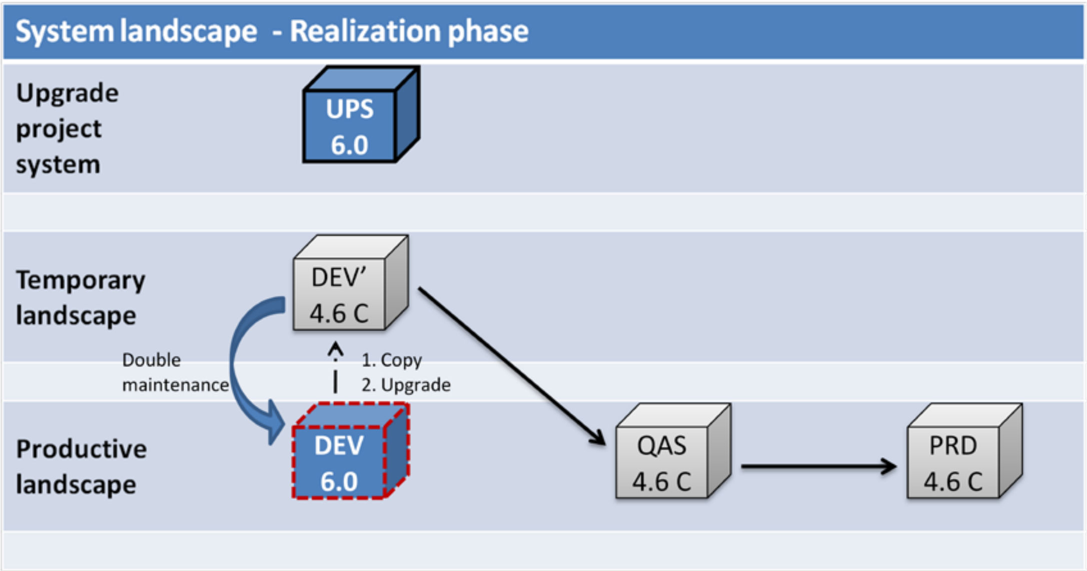


Figure 21. III Realization – duplication of the DEV and upgrade to new version. (Adapted from SAP Support Portal 2010c, 37)

At the end of this phase, the unit testing of custom developments and modifications should be completed.

IV Final preparation for cutover

In the final preparation for cutover phase, the quality assurance and testing instance QAS is duplicated to QAS' in order to support the production system and retaining the change transport flow from DEV' to PRD in case of any errors. Project activities during this phase are:

- Technical team first duplicates QAS to QAS' for maintenance and after that upgrades QAS to new version and transports project work to the QAS from DEV.
- Developers correct errors in custom developments.
- Business people carry out final integrations tests in QAS' and regression testing in the upgraded QAS. (Riedel 2009, 167-168.)

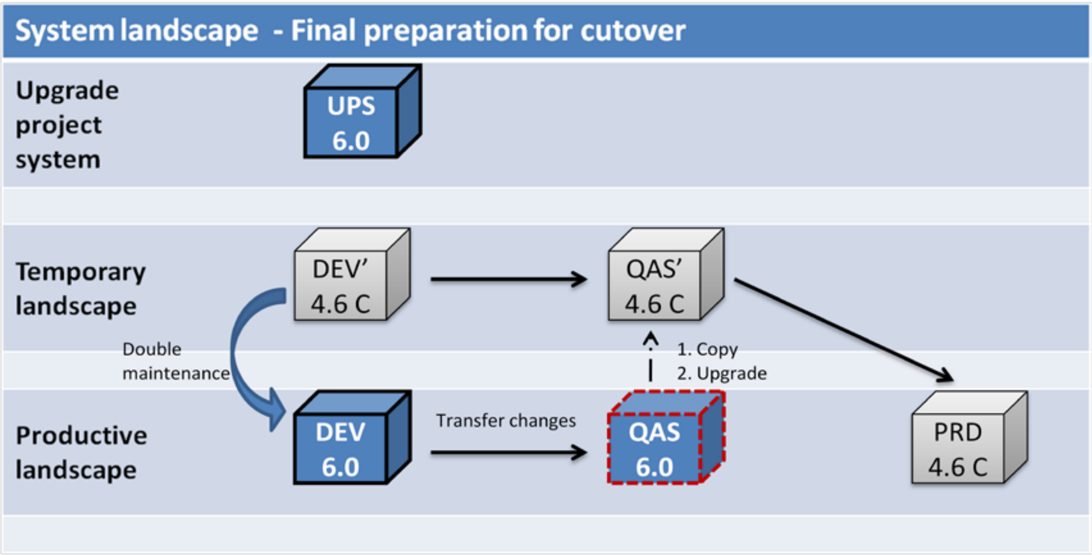


Figure 22. IV Final preparation for cutover – duplication of the QAS and upgrade to new version. (Adapted from SAP Support Portal 2010c, 38)

At the end of this phase, the testing of business processes should be completed.

V Production cutover & support

The production cutover and support is the final phase of the upgrade process and culminates with the go-live of the production system. Project activities during this phase are:

- Technical team upgrades the production system and restore the original change transport flow in the SAP system landscape.
  Business people sign off on the upgraded production system. (Riedel 2009, 168-169.)

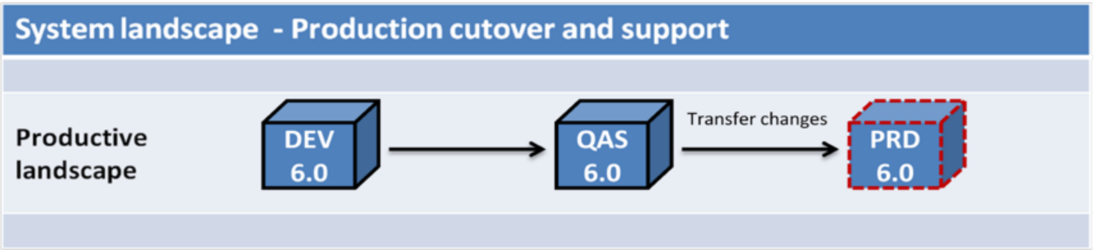


Figure 23. V Production cutover and support – upgrade of the production system. (Adapted from SAP Support Portal 2010c, 41)

At the end of this phase, the new SAP ERP version is in use in production system and temporary system landscape is removed. It might be however advisable to keep the DEV' available for some time after the upgrade so that in case of unexpected errors former functionality can be checked. Formally project is closed now, but there is still need for ongoing support of the upgraded system. (Riedel 2009, 169.)

2.5.2 Downtime

Downtime is the period during the upgrade when the production system is not available for end users. During this time technical upgrade with upgrade tools is performed, data is backed up and final tests are carried out. Downtime is a big challenge especially for the businesses whose system requires 24/7 availability (e.g. airlines, big production lines). Downtime should be consideration already in both planning and executing phase of an upgrade project. During the planning phase the maximum downtime available should be determined and precise upgrade tasks to be performed during the cutover should be specified. During execution phase when technical team performs technical upgrades to the sandbox, development and quality assurance instances, they should measure the time required to upgrade activities and test downtime minimization approaches. Even if an organization plans when the downtime takes place, what steps are required and estimates the time required, it should also be prepared for unexpected problems such as operating system failures or human errors. The first is called planned downtime and the latter unplanned downtime. (Riedel 2009, 164, 169-171)

An organization has little direct control over unplanned downtime, but planned downtime can be minimized through following considerations:

- Using an online backup instead of an offline backup strategy can further reduce total downtime, while technologies such as splitmirror backup can reduce backup downtime to zero.

- The time required to prepare the software for release to users can be increased depending on the languages installed, the other transports you want included in the system (e.g. number of clients), and the duration of user acceptance testing.

- Upgrading after hardware infrastructure has been reconfigured, enhanced, or resized (more hardware power), which increases the speed of database-related activities in particular (SAP Support Portal 2011b, 46).

- Doing a preupgrade cleanup of database tables affected by data conversion activities during the downtime (the size of the database has no direct impact on the duration of technical downtime) (SAP Support Portal 2011b,46).

- Having the latest version of the upgrade software tools.

- Choosing a "downtime-minimized" strategy instead of a "resource minimized" strategy (determined by the preconfiguration mode during the technical upgrade)

- Usage of SAP upgrade processes are supposed to reduce downtime: Incremental Conversion (ICNV), Customer-Based Upgrade (CBU), or Incremental Upgrade & Unicode Conversion (IUUC).

- SAP also offers services to help you reduce the production downtime as much as possible such as Near-Zero-Downtime approach.

  (Riedel 2009, 176-177; SAP Support Portal 2010b, 26; SAP Support Portal 2011b, 46)

In planning, you can consider the above things to reduce the downtime, but there is a trade-off between cost and time reduction. Investments in faster CPUs, better storage, new backup tools, and automated testing tools affect the overall cost of the upgrade.

2.6 Testing SAP ERP upgrade

The successful execution of an upgrade project depends on many factors. This chapter will give insight into one of the most important factors testing and answer the following research questions: Is testing necessary after upgrading? Why? What kind of testing activities are related to upgrade projects?

Commercial of the shelf package software testing differs from in-house software testing. The emphasis is to ensure that the product meets the user organization needs and that it is compatible with the environment in which it is used. (Snyder & Parth 2007, 207) But why ERP software should require testing at all, since vendors test all new versions before releasing then to the market? SAP ERP should contain very little errors, given the fact that it is used by thousands of companies all over the world. But when it is initially implemented, SAP ERP standard software is adapted to customer-specific business processes and IT environment, possibly encompassing complex interfaces to other SAP applications or third-party software, as well as custom developments and modification. Therefore, the overall SAP ERP system at each organization is unique. This is why, the standard version upgrade might affect customer specific SAP ERP systems differently and therefore upgrade must be tested thoroughly before go live. (SAP Support Portal 2008, 8.) Based on results from SAP's upgrade experience database, 74% of respondents say that testing is in fact a major challenge in SAP ERP upgrade projects. Although the actual amount of testing required during upgrade depends on complexity of the individual ERP landscape, the level of modifications and the functional scope of the project, there is no getting around the fact that a big part of the total upgrade effort will involve testing. Even in a purely technical upgrade thorough testing is essential. (SAP Support Portal 2011b, 47; SAP 2007, 9.) Adequate time for testing should already be included in the upgrade project plan schedule. According to Wenrich & Ahmad (2009, 62) organizations should expect to spend 25% of the project schedule on testing and according to Sens (2008, 95) testing can absorb 30% of the total project costs.

2.6.1 Purpose and focus of testing

The underlying concept to testing is quality assurance, which is delivered through testing (trying something out in conditions that represent the final live situation – a test environment). The purpose of testing is to provide confidence that the new or changed operations the upgraded ERP delivers are fit for purpose, function as supposed to with desired performance as well as remedy any problems and errors as early as possible since they are harder to diagnose and more expensive to fix in live production environment than if found in testing. (Taylor & Lacy & MacFarlan 2007, 115-116, 118.) If the upgrade is not well-tested it might jeopardize the continuity of operation of the SAP ERP software and therefore have a negative impact on the whole business. Also the acceptance of the new SAP release in the organization might be affected. (Sens 2008, 222.)

Although the goal is to test every possible aspect that could be affected by the upgrade, testing is always a spot check. It is impossible to run a 100% check due to limited amount of time and money. Therefore, the organization should carry out the change impact analysis to determine what has changed in the upgraded version and what should be tested. The focus of testing effort is very much dependent on the type of upgrade - purely technical or incorporating functional enhancements - and the scope of the project (Riedel 2009, 123-122). And as a rule, custom developments and modifications require more extensive testing (Riedel 2009, 82). But in general, five things should be tested, configurations, custom developments and modifications, data, user security profiles, and integrated business processes. While it is important to confirm that the new upgrade works as configured and modified, it is equally important to test it with the real-life data and transaction volumes. Also, security profiles should be tested to ensure that employees have the appropriate access and security to execute required business processes when the upgrade goes live. Last but not least, end-to-end business processes should be tested to ensure that all ERP and integrated non-ERP processes are operational. (Search Manufacturing ERP 3.8.2010.) In fact, according to Riedel (2009, 123) if the organization has limited resources, it should concentrate on the most used and most critical business processes. Although custom developments and modifications require also unit testing. The organization should also remember to test peripherals used for output such as printers and fax machines.

In summary, SAP ERP users are required to perform adequate testing before production instance is upgraded and upgrade adjustments transported there in order to minimize the risk of potential errors that may affect key application functionalities and disrupt critical business processes. Testing includes the testing of new or changed business processes and components and examines the behaviour of these in the target release and technological environment.

2.6.2 Test types

Testing can be broadly divided in two categories, functional and technical (SAP Support Portal 2008, 16).

Functional testing. This type of testing ensures proper functionality of the software, and therefore verifies correctness (Ray 2011, 174; Riedel 2009, 14). The focus here is exclusively on functional business processes and the purpose is to ensure that these work correctly and according to user expectations also after the SAP ERP upgrade (Anderson et al. 2009, 577). In functional testing, business processes are typically scripted into repeatable business cases (Anderson et al. 2009, 581). These should have been documented to some extent already during the initial implementation, but they should be reviewed and revised if required. In functional testing test cases consist of performing day-to-day tasks by an end user, such as Order-to-Cash business process that is applicable to most companies (Sens 2008, 209).

Technical testing. This type of testing tries to identify bottlenecks that slow down the system (Ray 2011, 174). Whereas functional testing ensures that a business process works correctly, technical testing ensures that a business process works quickly. The focus here is on how the system behaves during daily and high-load periods. (Anderson et al. 2009, 585)

Functional and technical testing can be further subdivided into the following:

FUNCTIONAL TESTING

- **Developer tests** Tests carried out by software developers at the lowest technical level. Besides functional aspects, these tests also include technical aspects (see below). (SAP Support Portal 2008, 16) Applicable to upgrade if customer-specific modifications need to be adjusted or new modifications adopted during the upgrade project.
- **Unit tests** (also called component tests)**.** Lowest level of functional tests which encompass individual transactions or a particular module from the SAP ERP (SAP Support Portal 2008, 16). Such a test only makes sense, if just one single unit (e.g. warehouse management) is affected by the upgrade (Sens 2008, 209). Or in case some configurations have been changed or customer-specific development done and possible errors in individual functionality has to be evaluated immediately. Indeed, unit testing is normally the first test that is completed during the configuration, and is focused towards the program's inner functions, rather than the integration. (Ray 2011, 174.)
- **Scenario tests.** Assessment of multiple related transactions within a module (solution area) or business process that spans multiple areas within SAP ERP, such as Order to Cash or Procure to Pay. (Riedel 2009, 125-126; SAP Support Portal 2008, 16). This testing is usually done in the development environment to prove out a requirement.
- **Integration tests.** This testing is similar to scenario testing except it is typically done in the QA environment and uses real data from production environment. An integration test examines correct execution of business processes spanning across various SAP ERP business areas or other SAP and third party applications (Riedel 2009, 126; SAP Support Portal 2008, 16). Testing is no longer function by function, but is now cross-functional. Therefore, the test teams for each business process must include members from each department the business process is relevant to. (Ray 2011, 179.) Because integration tests run over several different applications, interfaces and departments, they are rather complex to perform. Hence, they are necessary in order to guarantee the consistency of your IT landscape. (Sens 2008, 210)

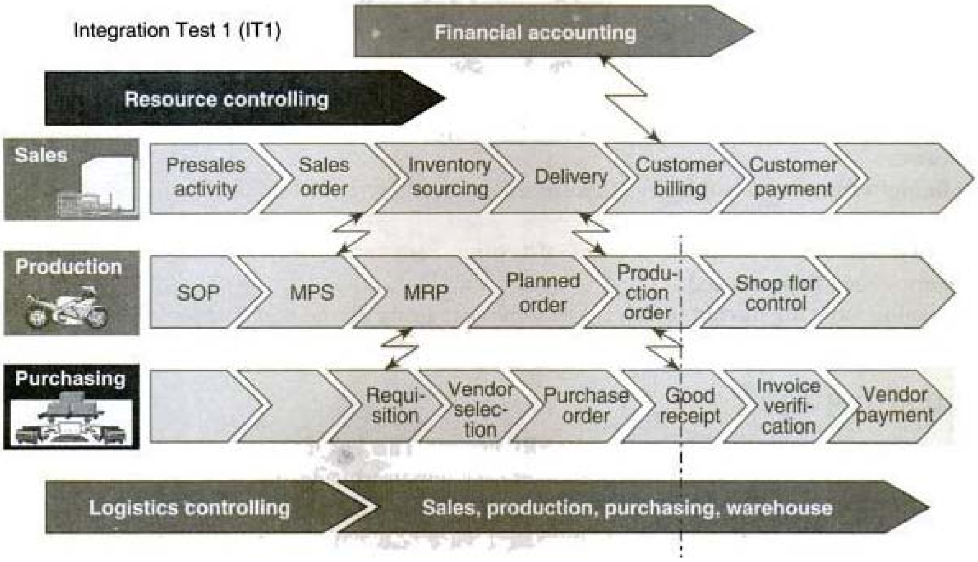


Figure 24. Integration tests run over several different applications and departments (Ray 2011, 180)

• **User acceptance tests**. Functional tests typically performed during the later part of the upgrade project where end-users test the scenarios/processes/transactions they are going to use in their daily work and ensure that these are working as per user satisfaction. (Ray 2011, 174.) Primary goals of user acceptance tests are to receive the formal acceptance of the business units (SAP Support Portal 2008, 16) and to detect usability problems, such as dialog design and system response time from an end user perspective (Riedel 2009, 126). User acceptance testing cannot be automated because it is highly dependent on the involvement of end users (Riedel 2009, 126).

• **Regression tests.** In the upgrade process the last test to ensure that previously tested upgrade and functionality still works as expected and no defects were introduced in production instance. Regression tests validate that the mapped functionality and core business processes were not affected (Riedel 2009, 126) and can be executed with correct system behaviour and results after the production system has been upgraded (SAP Support Portal 2008, 14) Basically in regression testing selected tests that were already run successfully in the testing instance are repeated to make sure that no new errors were introduced to the upgraded production instance.

TECHNICAL TESTING

• **Developer tests.** Tests carried out by software developers at the lowest technical level. The focus is on technical aspects such as correct handling of interfaces. (SAP Support Portal 2008, 14.)

• **Technical system tests.** System tests are known in the SAP environment as technical system tests. Instead of looking at the functionality of the software in isolation, technical system tests check the entire system, i.e. data consistency, databases, application servers, interfaces, network, printers etc. (Riedel 2009, 126; SAP Support Portal 2008, 14.)

• **Performance tests.** Performance testing is a technical system test that measures the throughput and response times of the system under test (SAP Support Portal 2008, 15). The purpose of performance tests - also called load, volume and stress tests- is to see whether SAP ERP infrastructure is still after upgrade capable of handling the expected workload. During such tests, a large number of transactions are processed against the SAP ERP application in order to see what the result will be. In most cases, these types of tests are performed through dedicated tools that are able to simulate a large number of client systems. Each client sends a certain amount of workload to the application. (Sens 2008, 211.)

• **Security tests.** Tests to check user access and authorizations, data security and other security related aspects (SAP Support Portal 2008, 15). For example, new functions and protocols (such as access via HTTP), or a new operating system or database software might cause this. (Ridel 2009, 84-85.) Authorizations and authorization roles are user permissions to perform certain activities in an SAP ERP system. For example, new version may include new authorization requirements and old authorization roles will require adjustments simply to be able to provide the same accessibility as in the source version. Typically, adjustments are required in the areas such as financial, vendor/ customer management, and DMS (document management system). (Panaya 2011a.)

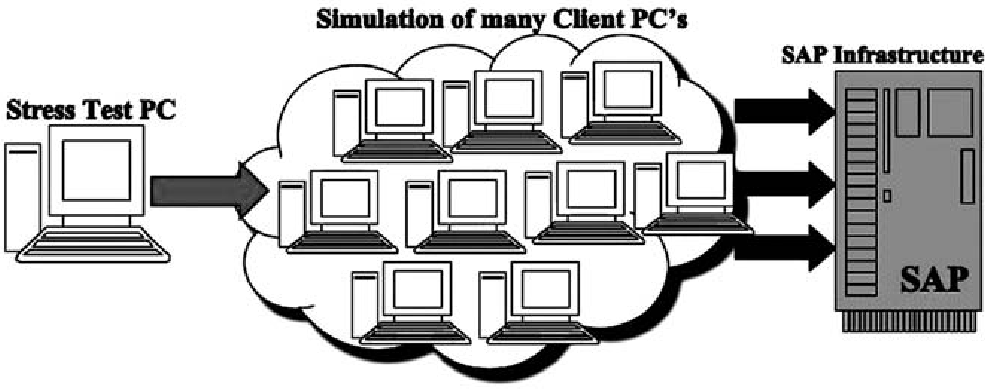
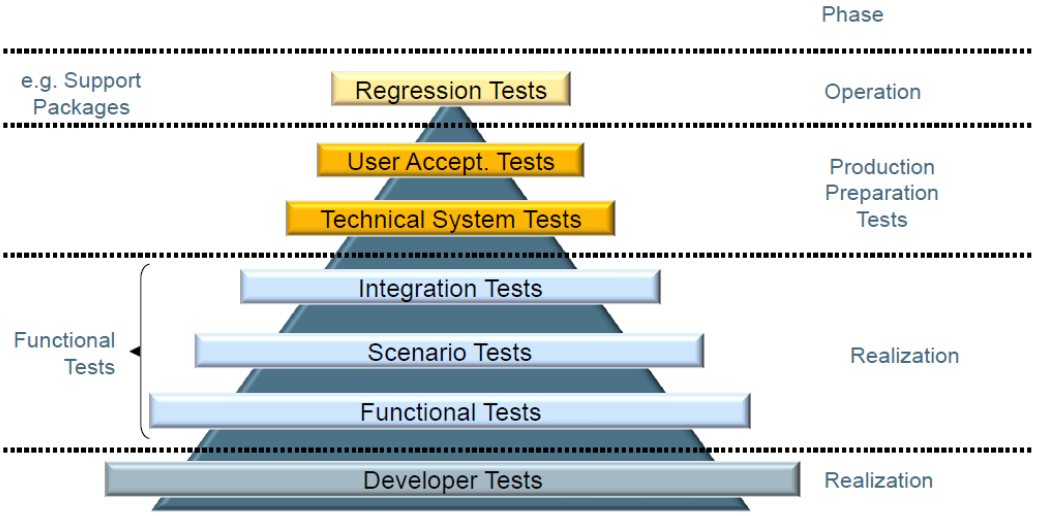


Figure 25. Performance testing using simulation tool. (Source: Sens 2008, 211)

Testing is hierarchical. It begins with each developer testing his or her pieces of software, and continues by testing increasingly large and integrated pieces of the product (Snyder & Parth 2007, 218). The test "pyramid" below shows how the test integration level increases from bottom to top. The integration level increases from unit testing (referred as functional tests in the figure) level of single objects to end-to-end integration test level where all SAP ERP business areas as well as other integrated SAP and non-SAP systems in the system landscape are tested. (Riedel 2009, 125.) The final tests before go alive are user acceptance tests.



© Copyright 2010. SAP AG. All rights reserved.

Figure 26. Increasing level of test increases towards the top of the test pyramid. (Source: SAP Support Portal 2010c, 40)

Testing is never done in the production instance. Testing takes always place on the SAP ERP instance implemented for testing, QAS / Test, or whatever the organization chooses to call it. Like explained in the chapter 2.5.1, many organizations first duplicate the current production instance and make it a sandbox where preliminary analysis of new functionality and needed adjustments regarding customizations, custom developments and modifications is carried out (change impact analysis). This requires some preliminary testing also. Sandbox is an isolated system, this means that changes made in the sandbox are never transported to other instances. Therefore, it is very important to record all the details of any change being made. The actual changes are made in the development instance (DEV). You can test the impact of upgrade change locally and then move it to testing instance (QAS) and test again. QAS is usually a recent copy of the production instance that has been upgraded to new release so that changes can be tested with realistic data and real-life environment. Once the testing is completed, the production instance can be upgraded and changes moved there. Once this happens, the new version goes live. All the changes done in one SAP instance are recorded into so called transport requests which can be then migrated to other instances without need to redo them.

TEST UPGRADES

Also the testing the technical upgrade process repeatedly is important. Repeated upgrades help to prepare an upgrade ”script” (also known as “runbook”), which can be used perform the upgrade smoothly when the ”real upgrade”, production cutover, is reached (Riedel 2009, 158; Sens 2008, 211). Ideally the organization has a sandbox system set up that replicates the production system as closely as possible. The rehearsal upgrades can be run on this system several times. For a small upgrade the organization should go through at least two upgrades (development and QAS instances). For more information see the chapter 2.5.1. As test upgrades are carried out, every single step should be documented in the runbook. The runbook can be used and refined during each upgrade. It should be tested also. The best way to do this is to execute the runbook against an SAP ERP test instance that is a copy of the production instance. Especially if production instance is very critical to the business in terms of downtime, the runbook should be very mature and well-tested. Also it is very important to record the exact elapse of the upgrade in the runbook. (Sens 2008, 212.)

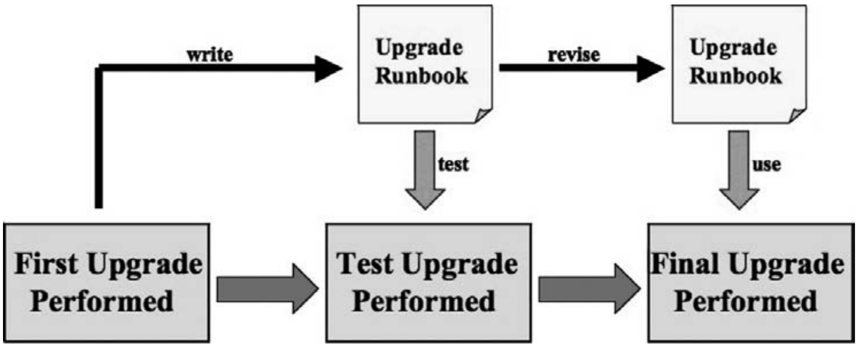


Figure 27. Upgrade runbook life cycle. (Source: Sens 2008, 212)

2.6.3 Manual and automated test approach

Adequate testing requires a lot of effort and staff. Traditionally testing has been carried out manually. In manual testing, after the test cases have been defined, all test activities are performed manually by testers, and the rest results are recorded manually. No test automation tools are used. (Ray 2011, 176.) Now, as ERP software landscapes are coming more complex and business processes more spread-out, even outside the company, automated testing tools have been developed to make testing faster and more effective. In automated testing approach, help of automated testing software is taken. These tools are generally called as CATT (Computer Aided Test Tool), recently eCATT (Extended Computer Aided Test Tools) and automated testing as CAST (Computer Aided Software Testing). (Sens 2008, 212; Taylor et al. 2007, 137.) After the creation of test cases, all test activities are carried out automatically by automated test software. After the testing, the results are recorded automatically by the testing tool. (Ray 2011, 176.)

Not everything can be automated and testing requires always lots of manual effort. According to Ray (2011, 176) automatic testing is especially useful for regression testing, in which test cases are run repeatedly after every version upgrade and support package upgrade to check that functionalities run as before. Manual testing is always used during user acceptance testing, and mainly used also during unit and integration testing. However, for any test, at least manual test cases have to be created. These test cases can be then re-used for test automation in future upgrades, although the effort to create an automated script is higher than to write a manual test description. If the organization invests in automated testing, it should also consider the licence costs and required know-how to use the tool. The return on investment comes with the reuse of automated test scripts because the cost of execution of an automated test script is less than the cost of manual execution. (Riedel 2009, 129-130.) It is recommended to have a team of experts handle the test automation, but user departments should provide or accept the test scripts and carry out error analysis (Riedel 2009, 127). A test script is a programming code created by automated test tool to carry out a specific test case (Wikipedia 2011b). Regardless of the approach, manual or automated, for both it is very essential to create accurate test cases. Without meaningful set of good test cases, that provide adequate coverage of the areas to be tested, automated testing will be no more effective than manual testing, even if it is more efficient (Riedel 2009, 130).

In general automated test makes sense if:

- "A test case is executed frequently.
- A test case consists of many test variants (same execution with different data).
- Substantial effort is required to prepare the test (e.g., entering many business transactions to obtain predefined, expected results)." (Riedel 2009, 130.)

Even if, automated testing can minimize cost of testing, there are also benefits to manual testing:

- "Greater flexibility in the design, creation and execution of test cases. Manual testers can use their expert knowledge to fill in gaps and make assumptions about test requirements.
- Identification of "real" issues. Tester can accurately identify whether functionality is not working due to a defect or whether there is a temporary issue such as a network connectivity problem.
- Testers can perform random tests that meaningfully extend the scope of the given test plan." (Riedel 2009, 132.)

SAP, as well as some third party vendors, offer testing tools for SAP ERP that e.g. help to define the test focus, manage relevant test cases, integrate test management and set up comprehensive reporting. SAP recommends that the organization uses SAP Solution Manager for the entire upgrade project and for testing in particular it offers:

- Documentation of business processes and assignment to SAP systems.
- Dynamic analysis of the impact of software changes on SAP business processes.
- Planning of manual and automated tests.
- Test execution and documentation.

- Incident handling.
- Monitoring of test execution.
- Test status reporting. (SAP Support Portal 2008, 22)

As mentioned earlier, the SAP Upgrade Road Map content is delivered with SAP Solution Manager application (also offline version available) (Riedel 2009, 206). The SAP Upgrade Road Map is a detailed guideline to plan and execute all relevant activities of a typical SAP ERP upgrade. These guidelines also include test case templates delivered as part of the enhancement packages for SAP ERP. Templates help to simplify the testing of new functions. There is a test case catalog for each business function that is included in the enhancement package. (Riedel 2009, 136.) Especially the Test Workbench and eCATT within SAP Solution Manager can be used for automated testing. (SAP Support Portal 2008, 31). These can be supplemented by a range of additional SAP products that are integrated with SAP Solution manager , such as SAP Test Data Migration Server (SAP TDMS), SAP Quality Center by HP and SAP Test Acceleration and Optimization (SAP TAO). Also many third-party test applications can be integrated with SAP Solution manager using existing interfaces. SAP offers also testing services such as testing project management and training. There are also services available from third-party consulting companies such as cloud-based services that help to identify potential errors caused by modified code and to prioritize what has to be tested and what not (Yachin 2009, 1-9). Despite the fact that test management tools have been available for years, according to Panaya[3] (2011b, 3) only half of the organizations seem to use these and other half still uses Microsoft Word / Excel or HP Quality Center. According the same survey, only 25% use automated testing.

[3] Panaya is a consulting company and not a neutral research organization, although it is not affiliated with software vendors. This does not invalidate the results, but as a policy matter we should retain some skepticism toward the findings.

2.6.4 Testing process

The testing process is shown schematically in Figure 28. The test activities are not carried out in a sequence. Several activities may be done in parallel, e.g. test execution begins before all the test design is complete.

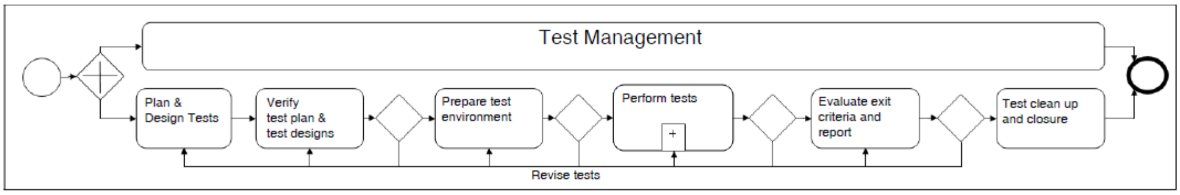

Test Management
Plan & Design Tests
Verify test plan & test designs
Prepare test environment
Perform tests
+
Evaluate exit criteria and report
Test clean up and closure
Revise tests

Figure 28. Testing process. (Taylor et al. 2007, 133, in Nurmi 2009, 59)

Test management. Effective management of testing activities is an important for ERP upgrade. First of all, for the upgrade project manager is it vital to ensure that:

- Appropriate test activities and resources are included in project plans.
- Testing resources (people, tools, licences) are allocated if required.
- The project understands the mandatory and optional testing deliverables.
- The testing activities are managed, monitored and controlled. (Taylor et al. 2007, 122.)

Secondly, the test mangers', working under project manager, are responsible for managing the entire testing process. They job is to plan, control and report all test activities. These activities include e.g.:

- Planning the test resources
- Prioritizing and scheduling what is to be tested and when (milestones, delivery dates)
- Monitoring progress of test activities
- Management of problems and errors
- Checking that incoming errors and their documentation are processed
- Test metrics collection, analysis, reporting and management. (Taylor et al. 2007, 133.)

Plan and design tests. The SAP ERP upgrade testing should starts with planning and design well in advance, although testing activities themselves may not occur until the end of the project. Proper planning requires a significant amount of time. The test plan is a written document that details major testing tasks, estimation of the time required for each task and their schedule during the upgrade. Also the types of testing to be performed should be identified (e.g. unit, integration, performance) along with approach (manual or automated) and testing results evaluation criteria for each type of testing (e.g. pass/fail). (Snyder & Parth 2007, 208-210.) In which environment testing will be done, who will create test data, who will do the testing, what authorizations are required for testers, what are the deliverables and how testing results are reported etc. are also part of it (Ray 2011, 108, 177). In addition, some bigger organizations may have a higher-level document called a test strategy.

After the comprehensive test plan is written, it is translated into more detailed test cases. A test case normally consists of a unique identifier, requirement references from a design specification, preconditions, events, a series of steps to follow, input, output, expected result, and actual result. Steps can be stored in a word processor document, spreadsheet or some other repository. Test cases can be brief or very detailed. Larger test cases may also contain prerequisite states or steps, and descriptions. (Wikipedia 2011b.) It is impossible to test everything. Otherwise there would be thousands of test cases, therefore test design aims develop test cases that measure the correct things. It is important to avoid focusing too much on the lower level testing, even though it is often easier. (Taylor et al. 2007, 126.) Test cases should cover at least all the most critical business processes. Because the requirements of understanding the individual business processes of the organization, the involvement of business experts will be necessary for test case creation, as for much of the entire test panning (Riedel 2009, 126-127). However, capturing business process knowledge from key users is not always easy, and according to Panaya (2011b, 3) survey 52% of organizations regard this as the top testing related challenge. Without meaningful set of good test cases, that provide adequate coverage of the areas to be tested, it doesn't matter how much time is used for testing. In other words, what is tested is more important than how much is tested. Test cases should also be designed so that they are easy to audit, easy to run repeatedly ideally even by less-experiences users, not just by business experts. This frees up more experienced staff to concentrate on other priorities. (Riedel 2009, 85.)

The collection of test cases is called a test catalog. For example, sales and distribution test catalog could contain a set of test cases such as a test case for sales order creation, a test case for availability check and a test case for credit check. All test cases relevant for the upgrade, manual or automated, are collected into the test plan to be performed during the upgrade. A test package in turn, is a person and period-oriented view of a test plan. It contains all tests which a tester should perform in a specified period. The same test case can be assigned to several test packages. (Ray 2011, 176-177.) If relevant test case is not available in the test catalog, new test case must be created. Often, insufficient and incomplete test catalog prevents SAP customers from performing accurate testing (SAP Support Portal 2011b, 47).

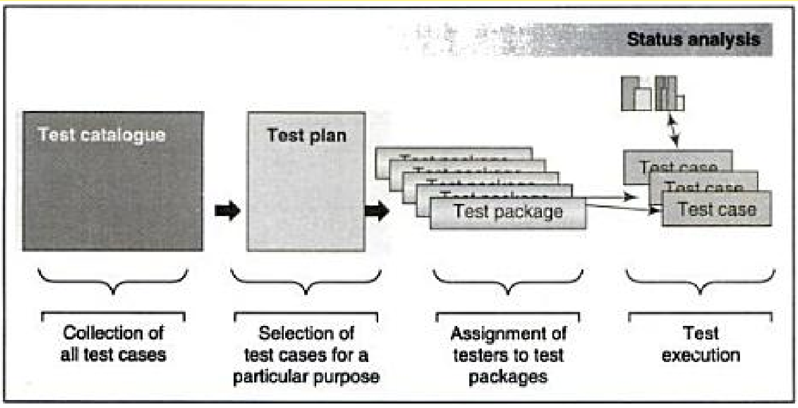

Status analysis
Test catalogue
Test plan
Test package
Test case
Test case
Test case
Collection of all test cases
Selection of test cases for a particular purpose
Assignment of testers to test packages
Test execution

Figure 29. Testing terminology. (Source: Ray 2011, 177)

Verify test plan and test designs. Test plan and test cases needs to be approved by business process owners and members of the test team (Ray 2011, 179). This is to ensure that the test cases are complete and deliver adequate test coverage over the business processes and interfaces. (Taylor et al. 2007, 141)

Prepare test environment. The organization needs to organize the system landscape to provide one or more SAP ERP instances on which to carry out relevant tests. For each testing instance has to be prepared with sufficient test data (master data and transaction data) and of course all the relevant users have user account and necessary security authorizations. (Riedel 2009, 128.) Chapter 2.5.1, Managing the system landscape during an upgrade project, describes a typical system landscape for an upgrade project. If more hardware is required such as servers, this should be considered in advance so that infrastructure is in place by the time testing starts (Ray 2011, 179).

Perform tests. The test cases agreed in test plan (test model) are carried out either manually or using automated tools and testing procedures. The tester is supposed to do each of the steps specified in test case and document the output such as whether the order number was created and, if so, the actual order number in case it has to be reviewed in detail at a later time. Also warnings or errors displayed by the SAP ERP should be noted. Especially, in case of performance testing or performance issues, basic performance metrics (such as "wall clock" execution time), should be noted. (Anderson et al. 2009, 592.) Tester also must record that whether each of the steps is successful or if it failed. If a test fails, the reason must be carefully documented. However, if possible the tester should continue with other test cases according the test plan. When incident or issue is resolved, the same tester should retest the test. In the exit criteria evaluation stage the actual result is compared to the expected results. The result may be interpreted in terms of pass/fail. (Taylor et al. 2007, 133-134). The typical test flow is illustrated in the chart below.

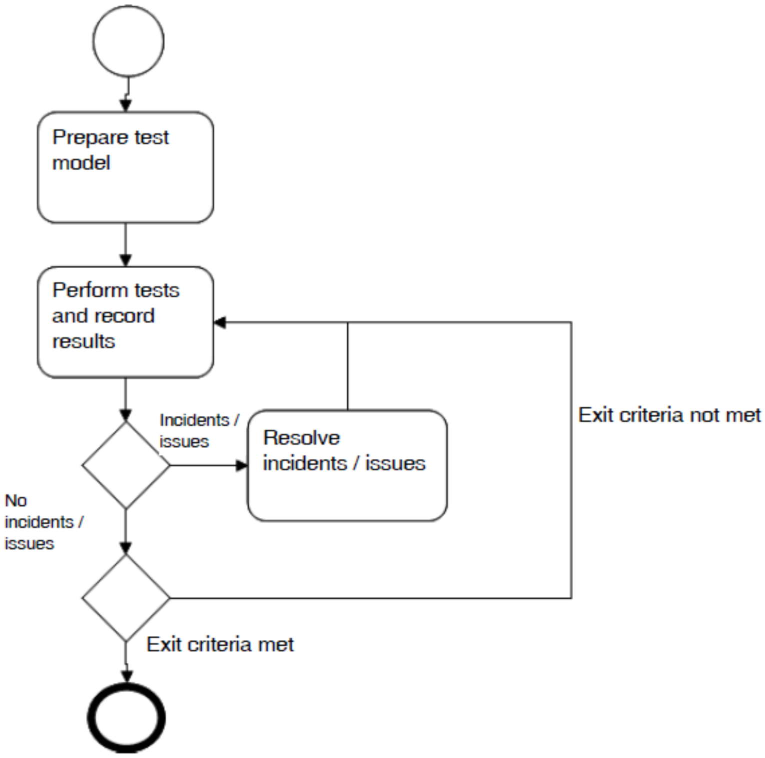


Figure 30. Performing test. (Taylor et al. 2007, 135, in Nurmi 2009, 60)

Evaluate exit criteria and report. After the tests have been executed, test results are compared to the expected results, the test report is produced, test metrics gathered and the results of the test summarized. Test completion is decided based on meeting the exit criteria defined during the planning phase. (Taylor et al. 2007, 133.)

Test clean up and closure. After all testing has been completed, the test environment is cleaned up or initialized for future use and all test documents and test data archived. Also at the same time the testing success and procedures should be analyzed and improvements for the future testing should be noted. (Taylor et al. 2007, 133-134.)

2.6.5 Test activities in the SAP Upgrade Road Map

Organizations using SAP ERP can use the upgrade project phases defined in the SAP Upgrade Road Map to assist through testing activities. However, this can be different depending on organization's individual requirements and therefore the following table is not definite but helps to understand how the testing effort and resources can be organized during an upgrade project.

Table 4. Different testing activities at different phases of the upgrade project. (Data gathered from Riedel 2009, 134-136; 165-168.)

Project Phase	Type of testing	Activities	Objective
Project preparation		• Define test strategy and concept • Select the test tools • Define testing roles and responsibilities	• Test focus defined • Framework defined • Testing team set up
Upgrade blueprint	• Developer tests when adjusting custom developments and modifications • For business process analysis business processes can be tested (Sandbox)	• Perform a business process analysis to identify most used/critical processes • Review new and changed business processes in the new release to determine the need for new or changes test cases • Identify business functions to be installed and activated from the enhancement package • Assign test templates provided by SAP for the enhancement package • Create test plan and select existing test cases from the test catalog or create new ones if required • Define test standards for reporting, defect management, and test exit criteria.	• Relevant test case have been assigned to the project • Initial knowledge transfer to the project team completed.
Upgrade realization	• Unit testing	• Upgrade test instance to the target release	• Unit testing of custom development and modi-

	• Acceptance (DEV’)	• Create test cases • Provide test data • Create test plans and packages. • Build sequence for the execution order of test cases • Assign test packages to testers • Carry out knowledge transfer to tester • Test and report	fications finished • Acceptance tests finished
Final preparation for cutover	• Integration • Performance (QAS’)	• Prepare regression tests	• Final integration tests finished • Performance tests finished • Regression tests defined
Production cutover and support	• Regression		• Regression tests finished • Production instance ready for use

3 Empirical part

The practical aim of the empirical part is to test the upgraded SAP ERP 6.0 environment of Haaga-Helia UAS to ensure that functionality and business processes required in the SAP ERP basics course of the Business Information Technology DP starting in the autumn semester 2010 were not affected by the upgrade and the course exercises of the previous implementation of the course, that function as test cases, are still suitable. The course instructor Jarmo Harmonen provided a set of test cases for the researcher and they form the basis for the testing. The theoretical framework and testing activities related to SAP ERP upgrade were discussed in Chapter 2.6.

The SAP ERP environment of Haaga-Helia does not exactly correspond with typical real-life SAP ERP environments in organizations, because Haaga-Helia uses SAP ERP IDES system (Internet Demo and Evaluation System) hosted by University Competence Center Magdeburg in Germany. IDES system is basically a fully functional SAP ERP system but it has been pre-configured and pre-populated with lots of master and transaction data. It is a standard implementation, i.e. it does not include any custom developments or modifications that would affect the upgrade testing process. Also the entire SAP ERP system landscape is operated by UCC Magdeburg – SAP ERP application, servers, databases, instances and clients. Therefore, UCC Magdeburg actually carried out the upgrade from SAP ERP 5.0 to the latest version 6.0. Haaga-Helia was provided with a new client connection to the upgraded SAP ERP IDES 6.0 instance.

So from the Haaga-Helia's view point there were no development and testing instances provided first for pretesting (technical /unit / acceptance /integration /performance) as in general, before the productive instance was upgraded. Therefore, since the testing is performed after production cutover, it can be regarded as regression testing. However, in Haaga-Helia's case the downtime is no issue, because testing is done during the summer semester, when there are no SAP courses running. And on the other hand, the client connection to the old SAP ERP version 5.0 is still available over the summer.

Every upgrade requires testing. And even though, we are not upgrading any critical SAP production system, adequate testing is still required to make sure that the upgraded IDES-environment works properly and all the functionality required in the future SAP-courses is working properly. In case of Haaga-Helia's IDES environment the testing does not need to be so extensive. In the following chapter first the test plan is presented. The actual test cases are not included in the research, but a description of each test case along with the result summary is provided after the test plan. The detailed test results of six test case exercises are presented in the form of a table and they can be found in the appendices.

3.1 Test plan

3.1.1 General description

The primary objective of the regression tests is to ensure that the required functionality in the SAP ERP system supports the business processes as defined in the test cases. Regression testing is not restricted to only the business processes functionality but involves also testing of master data, student authorization rights, performance, printing as well as usability and consistency of the course exercise instructions that are used as test cases. If defects or problems are found in the system during the testing, all possible corrections have to be retested. Usability and consistency remarks on test cases are noted in the test report. Secondary objective of testing is to become acquainted with the new version and learn new things about SAP ERP.

3.1.2 Environment

Tests are conducted at Haaga-Helia's premises in Pasila, so the technical environment corresponds the real-life technical course environment. Testing time is June-August 2010. The researcher works as the sole tester. The tester's user rights are the same than regular course students have. Haaga-Helia's SAP ERP instance consists only of one client for productive system, so the tests are performed in that and no special testing environment is available.

SAP version: SAP ERP 6.0
Interface: GUI , language English
Client (1): IDES Client 932

Tests are conducted manually so no special testing tools are required. Test results are documented using the desktop application MS Word.

The researcher worked as a tester. The tester is not a proficient but a student SAP user. She has taken similar SAP course in autumn 2009 in the Finnish speaking Information Technology Degree Program at Haaga-Helia UAS and completed during the course similar exercises.

3.1.3 Entry criteria

Entry criteria for regression tests are:

- Upgraded SAP ERP IDES environment is ready for testing
- User account and authorizations defined for the tester
- Test cases are ready, approved and handed over to the tester.

3.1.4 Scope

The pactical testing part consists total of six test cases – five course exercises and an assignment to perform and automate period closing.

1. *Course exercises.* The test cases consist of five exercises that were used in teaching the Business Information Technology Degree Program course *SAP R/3 Business Application and Business Development SYS8TF139-6*in Haaga-Helia University of Applied Sciences during the spring 2010. These exercises were originally designed for SAP R/3 version but later used with some minor modifications also in SAP ERP ECC 5.0 environment. The main purpose of the testing is to find out if the exercises in question are still suitable for the future SAP-courses using the new upgraded SAP ERP 6.0 IDES environment. All the five test cases were provided by the instructor Jarmo Harmonen.

Test case 1:	SAP R/3 User interface exercise
Test case 2:	SAP R/3 order-delivery process, Exercise RT 3a K2010 ver. 2.3.2010
Test case 3:	Exercise RT 3b part1: Master Data & part2: Production
Test case 4:	SAP Order-delivery process exercise
Test case 5:	SAP IMG assignment

Goals of the testing

The main goal of the testing is to ensure the following:

- Master data. Can all the in test cases required master data be found?
- Can the test cases and business processes in them be executed without problems?
- Printouts. Is it possible to print the business documents defined in the test cases on paper (order confirmation, purchase order, delivery note, invoice) ?
- Can all test case exercises be performed with student user authorization?
- Testing and evaluating usability and consistency of test case instructions, including instructions /screen shots referring to user interface.

2. *Assignment (Test case 6).* Performing and automating period closing according the instructions given in the 'UCC-tutorial on Period Closing in SAP ERP 6.00 Systems in UCC Environment' and verifying that it works. This is not a course exercise for students like five test cases above. Period closing is a revision security mechanism incorporated in SAP ERP and it is meant for productive use. The reason for it is to impede illicit postings to prior periods. Since there is no need for such a constraint in teaching environment, this security mechanism incorporated in SAP ERP can be automated. Otherwise it should be performed manually.

The tester will also point out if there are any other remarks on exercise steps that needs updating or clarification in tester's opinion. These observations are recorded in the Other remarks column of the result table. This way instructor will get valuable feedback information and can modify the exercises to be more consistent, understandable and up-to-date for the future students.

Out of scope

- Student is not responsible of solving the errors and problems related to the test cases that occur during the testing, but solutions can be suggested and recorded in the test result table.
- Tester will not report of any cosmetic issues in test case exercise instructions – such as spelling mistakes, illogicality or language used - if it doesn't significantly affect understanding of the exercise.

3.1.5 Acceptance criteria and methods

Tests are executed in two iterations. In the first testing round the tester follows carefully the step by step instructions in the exercises when executing tests and records if the test case step can be successfully executed or not (passed / failed) in the Results column of the test result table. If known, the reason for failing is indicated in the Notes column. Also warning or errors displayed by the SAP ERP software should be noted. Also if there are any performance or authorizations issues they are to be documented as well. Naturally, the tester is supposed to document the output of the test step such as an order number. However, in case one test case step fails, if possible the tester should continue with other steps. Only the steps that did not pass the first testing round, the ones that are marked to have failed, are being re-tested in the second testing round. Now the tester will try to solve how to proceed so that the test step can be successfully passed. The possible solution is marked in the Notes column. If the solution seems to work, the test step is marked as passed. If the tester is unable to find any solution, step will fail again.

All the steps that are handled in the second testing round (ones that failed in the first round) needs to be revised for the future courses. It is up to the instructor if he decides to revise the exercises according the suggested solutions in the second testing round or find another solution. For all the failed steps in the second testing round, the instructor will need to solve the problem.

In 'Other remarks' section of the result table tester will point out if there are any usability or consistency remarks on exercise instructions that needs updating or clarification in tester's opinion. This way instructor will get valuable feed-back information and can modify the exercises to be more consistent, understandable and up-to-date for the future students. Tester will not report any spelling mistakes or give other remarks on language used if it doesn't significantly affect understanding of the exercise.

Student is not responsible of solving the problems or defects related to the test cases that occur during the testing, but solution can be suggested and recorded in the test result table. Testing can be completed either after the first testing round if the entire test case could be executed successfully (all steps passed) and if results are documented or after the second testing round even if some steps of the test case failed as long as results are documented.

3.1.6 Deliverables

The test results presented in the form of a table and they are delivered to the instructor Jarmo Harmonen by e-mail latest on the 25th of August 2010. The tester should use the following template for reporting the test results:

[TESTING ROUND] **[Test case : name)**				**Date:**
Steps	**Result (passed/ FAILED)**	**Notes**	**Output /outcome**	**Other remarks**
1				
1.1				

3.2 Test case descriptions and result summaries

3.2.1 Test case 1: User interface exercise

Description

The user interface exercise is the first hands-on SAP exercise and it consists of 8 short sub-exercises. Its goal is to get the student acquainted with the SAP ERP user interface structure and learn how to find master data of customers, products, accounts and how to display sales orders and document flow.

Results summary

First seven sub exercises could be executed without any problems or incidents. The last sub exercise 8 steps failed, in the first testing round. A sales order could not be retrieved from the system and displayed, because the master data for that particular sales order was not transported to the data tables of upgraded SAP ERP. When Haaga-Helia's SAP ERP IDES system was upgraded the new system client number was provided by UCC Magdeburg. And as mentioned in Chapter 2.1.2, each client has separate master records and its own set of data tables (client dependent data). The sales order was created in the source version client, and since it was not transported to the target version client data tables, the data for it was missing. In the second testing round the problem was solved by displaying another sales order that existed in the client data tables as pre-populated IDES master data.

3.2.2 Test case 2: SAP ERP order-delivery process I

Description

SAP R/3 order-delivery process, Exercise RT 3a K2010 ver. 2.3.2010

In this exercise student is not only supposed to display already existing information but create new data into the system. The student will get acquainted with a typical logistic process flow from customer order to billing in the SAP ERP system. The process starts with an order, proceeds with stock check and purchase of necessary materials and finally ends with delivery to a customer as well as the billing. Before it is possible to start rhe order-delivery process all necessary background information has to be created.

During different phases the impact of different transactions will be examined. It will be also checked that all the related documents can be printed out from the system.

The procedure is following:

Necessary background information (= master data)

- Organizational units (already existing)
- Customer and vendor information (will be copied from existing master data)
- Material information (will be copied from existing material)
- Price lists, discounts, pricing procedures (will be defined)

Processing a new sales order

- Create a sales order
- Check of availability, delivery time
- Check status of stock
- Material requirement planning (MRP) for the ordered material
- Generate purchase order form MRP Purchase request
- Receive the ordered material to stock
- Check status of stock
- Generate delivery from sales order (material will be picked from stock and delivered to the customer)
- Check status of stock
- Examining of the sales order and its document flow
- Billing the customer
- Releasing the bill into financial accounting
- Examining of the sales order and its document flow

Results summary

In the first testing round all the test case exercise steps failed, because in the exercise required master data was missing. In SAP ERP system, before business processes can be carried out, necessary background information (=master data) such as material, customer, and vendor has to be entered into the system. Likewise in the previous test case 1, when it was asked to create a new material, a new customer and a new vendor by copying it from an existing reference master data, it could not be done, because the master data for these particular data items was not transported to the data tables of upgraded SAP ERP. When Haaga-Helia's SAP ERP IDES system was upgraded the new system client number was provided by UCC Magdeburg. And as mentioned in Chapter 2.1.2, each client has separate master records and its own set of data tables (client dependent data). The reference data was created in the source version client, and since it was not transported to the target version client data tables, the master data for it was missing. Furthermore, because of the missing master data the tester was unable to proceed with subsequent steps and therefore, they failed as well.

In the second testing round the missing master data problem was solved either by creating corresponding reference material or by using master data that existed in the client data tables as pre-populated IDES master data. After that the tester was able to proceed with the subsequent steps of the test case. In the steps where the tester was supposed to post the material number and goods issue documents, the system error message "Posting only possible in periods 2010/05 and 2010/04 in company code 1000" appeared. The error message is displayed, because the date of the testing is in 6/2010, and the system's current posting period is still 2010/5 and the period 4/2010 is not closed either. After the Test case 6. - Automate period closing - was executed in which first the posting period 5/2010 was closed and then the current posting period was changed to the actual calendar period, along with automating the period closing. This solved the problematic steps and the error message would not appear again in the future. The only thing that failed in the second testing round was printing of invoice on paper. The tester was unable to solve the problem within a reasonable time and therefore this step failed.

3.2.3 Test case 3: SAP ERP production process and BOM

Description

Exercise RT 3b part1: Master Data& part2: Production (SYS8TF139-06_RT3_b K2010 ver. 29.3.2010)

This test case exercise focuses mainly on two SAP ERP modules - Sales and Distribution (SD) and Materials Management (MM). The exercise is very similar to the previous test case exercise, but having production part added. The student will get acquainted with a typical logistic process flow from customer order to billing in the SAP ERP system when ordered material is a BOM (Bill of Material), which is a finished product (parent) made up of different inventory components (children). First all necessary master data has to be entered in the system. The actual business process starts with customer order, followed by stock status check. In exercise there is not sufficient stock of child components so they need to be ordered first in stock and then transfer to production, before the finished product can be manufactured and delivered to the customer.

The procedure is following:

Necessary background information (= master data)

- Organizational units (already existing)
- Customer and vendor information (will be created)
- Material information - BOM + main material + child components (will be copied from existing material)
- Price lists, discounts, pricing procedures (will be defined)

Processing a new sales order

The process flow of the test case exercise is illustrated in the following chart.

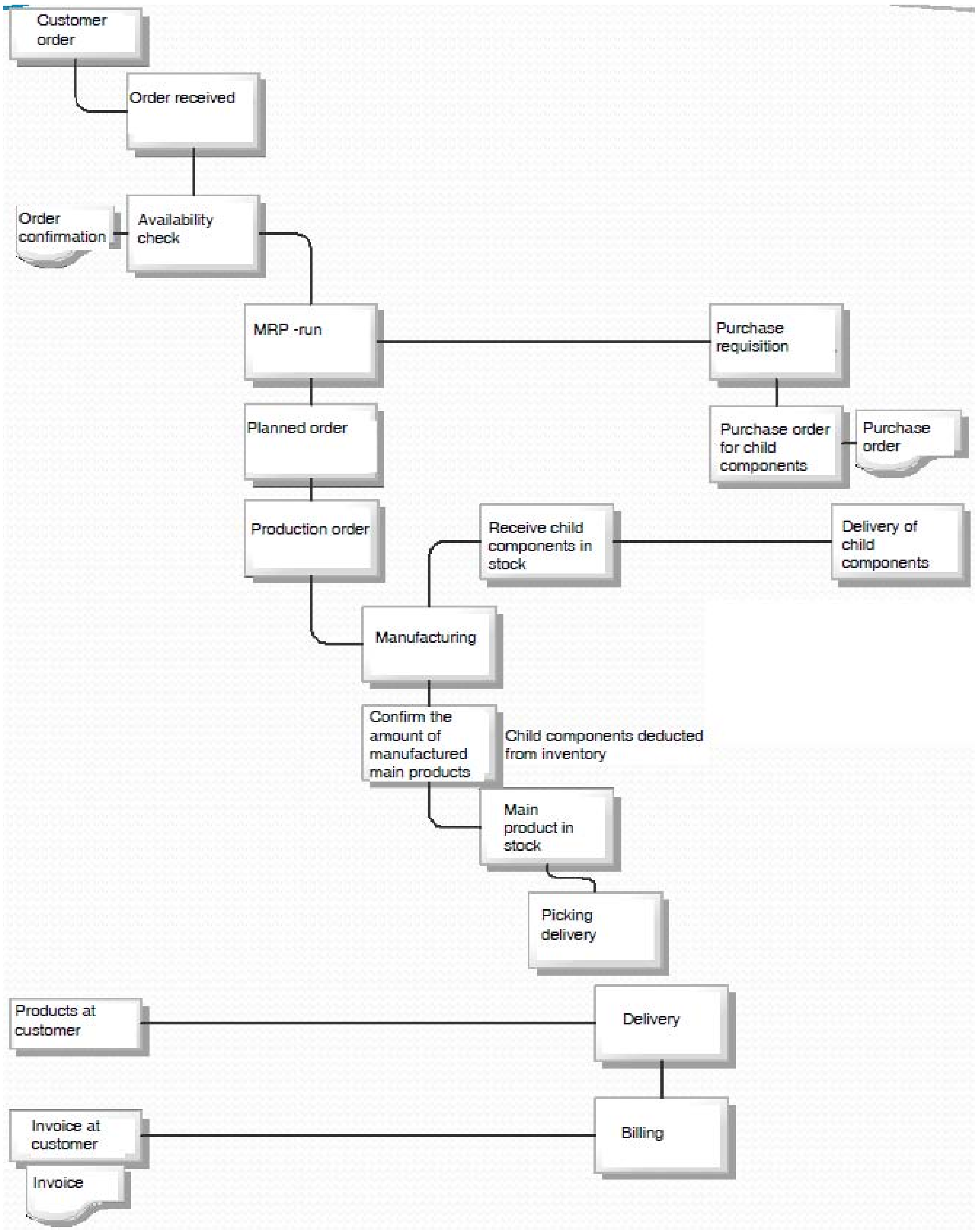


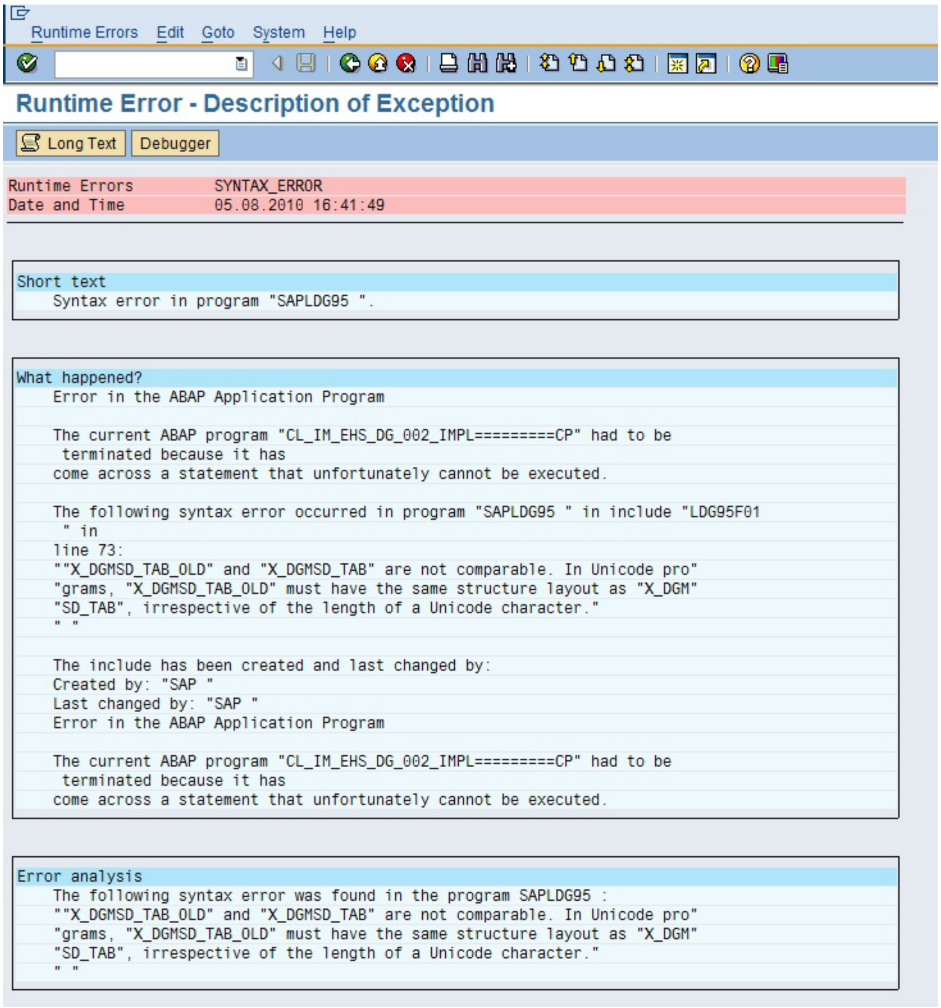
Figure 31. The process flow of the 3. test case exercise.

Results summary

In the first testing round all the test case exercise steps failed, because in the exercise required master data was missing like in the previous test cases 1 and 2. In SAP ERP system, before business processes can be carried out, necessary background information (=master data) such as material, customer, and vendor has to be entered into the system. When it was asked to create new material master records for main product and child components, by copying it from an existing reference master data, it could not be done, because the master data for these particular data items was not transported to the data tables of upgraded SAP ERP. When Haaga-Helia's SAP ERP IDES system was upgraded the new system client number was provided by UCC Magdeburg. And as mentioned in Chapter 2.1.2, each client has separate master records and its own set of data tables (client dependent data). The reference data was created in the source version client, and since it was not transported to the target version client data tables, the master data for it was missing. Furthermore, because of the missing master data the tester was unable to proceed with subsequent steps and therefore, they failed as well.

In the second testing round the tester tried to solve the missing master data problem first by using master data that was created in the test case 2 or that existed in the client data tables as pre-populated IDES master data. It functioned all right for child components. For BOM it seemed to work first, and the tester was able to proceed with the subsequent steps of the test case. But in the step where the tester executed the MRP-run and the system was to create purchase request for the child components, dependent request was created instead. The tester used a lot of time trying to solve the problem, which most likely was related to some production routing or MRP settings of the BOM. However, the tester was unable to solve the problem within a reasonable time and therefore all the other steps but the first one failed.

In addition, at some point during testing the Unicode the system error message related to some Unicode problems was displayed (see Figure 32). At later point when executing the test case 5 a message appeared informing that Unicode conversion has been performed. Supposingly, they had performed Unicode conversion or some updated regarding to it at UCC Magdeburg in between. In any case, the Unicode error message did not appear again during testing.



```
Runtime Errors         SYNTAX_ERROR
Date and Time          05.08.2010 16:41:49

Short text
    Syntax error in program "SAPLDG95 ".

What happened?
    Error in the ABAP Application Program

    The current ABAP program "CL_IM_EHS_DG_002_IMPL=========CP" had to be
     terminated because it has
    come across a statement that unfortunately cannot be executed.

    The following syntax error occurred in program "SAPLDG95 " in include "LDG95F01
     " in
    line 73:
    ""X_DGMSD_TAB_OLD" and "X_DGMSD_TAB" are not comparable. In Unicode pro"
    "grams, "X_DGMSD_TAB_OLD" must have the same structure layout as "X_DGM"
    "SD_TAB", irrespective of the length of a Unicode character."
    " "

    The include has been created and last changed by:
    Created by: "SAP "
    Last changed by: "SAP "
    Error in the ABAP Application Program

    The current ABAP program "CL_IM_EHS_DG_002_IMPL=========CP" had to be
     terminated because it has
    come across a statement that unfortunately cannot be executed.

Error analysis
    The following syntax error was found in the program SAPLDG95 :
    ""X_DGMSD_TAB_OLD" and "X_DGMSD_TAB" are not comparable. In Unicode pro"
    "grams, "X_DGMSD_TAB_OLD" must have the same structure layout as "X_DGM"
    "SD_TAB", irrespective of the length of a Unicode character."
    " "
```

Figure 32. System error message related to Unicode.

3.2.4 Test case 4: SAP ERP Order-delivery process exercise II

Description

SAP Order-delivery process exercise, Tuesday 13.4. & 20.4.2010

This test case exercise was almost identical with the test case 2, in which student carries out a typical logistic process flow from customer order to billing in the SAP ERP system. Only difference is that instructions are not so detailed, so the student will need to try to remember how to perform certain functions or check out instructions in the test case exercise 2. The process starts with the order, proceeds with stock check and purchase of necessary materials and finally ends with delivery to the customer as well as the billing. Before it is possible to start order-delivery process all necessary background information has to be created. During different phases the impact of different transactions will be examined. It will be also checked that all the related documents can be printed out from the system.

The procedure is following:

Necessary background information (= master data)

- Organizational units (already existing)
- Customer and vendor information (will be copied from existing data)
- Material information (will be copied from existing material)
- Price lists, discounts, pricing procedures (will be defined)

Processing a new sales order

- ➢ Create a sales order
- ➢ Material requirement planning (MRP) for the ordered material
- ➢ Generate purchase order form MRP Purchase request
- ➢ Receive the ordered material to stock
- ➢ Generate delivery from sales order (material will be picked from stock and delivered to the customer)
- ➢ Billing the customer
- ➢ Releasing the bill into financial accounting

Results summary

In the first testing round all the test case exercise steps failed, because in the exercise required master data was missing like in the previous test cases 1, 2 and 3. In SAP ERP system, before business processes can be carried out, necessary background information (=master data) such as material, customer, and vendor has to be entered into the system. When it was asked to create a new material, a new customer and a new vendor by copying it from an existing reference master data, it could not be done, because the master data for these particular data items was not transported to the data tables of upgraded SAP ERP. When Haaga-Helia's SAP ERP IDES system was upgraded the new system client number was provided by UCC Magdeburg. And as mentioned in Chapter 2.1.2, each client has separate master records and its own set of data tables (client dependent data). The reference data was created in the source version client, and since it was not transported to the target version client data tables, the master data for it was missing. Furthermore, because of the missing master data the tester was unable to proceed with subsequent steps and therefore, they failed as well.

In the second testing round the missing master data problem was solved by using the same master data created in test case 2 as reference material. After that the tester was able to proceed with the subsequent steps of the test case. The only thing that failed in the second testing round was printing of invoice on paper. The tester was unable to solve the problem within a reasonable time and therefore this step failed.

3.2.5 Test case 5: SAP ERP IMG assignment

Description

SAP IMG assignment Spring 2010

This test case exercises were divided into two parts:

1) First the IMG-tool (Implementation Management Guide) was used to create a new sales area. IMG-tool is used for performing the customizing settings in SAP ERP.
2) After that, a typical logistic process flow from customer order to billing is carried out, like in test cases 2 and 4, but now using the new sales organization.

In the first part, a new sales organization is created for a company code IDES AG and it consists of the following subelements: distribution channel, sales office and sales group. Sales and distribution organization hierarchy (SD module) is one of the most central configuration structures in SAP ERP system. This needs to be set during the initial implementation. However, it can be recustomized if changes occur in the organization structure. After the organization structure has been defined, every transaction and master data, such as customer, material, prices, are linked to the organization structure for which they were originally created in the system.

In the second part, the new sales area is used with order-delivery process already familiar from test cases 2 and 4. The process starts with the order, proceeds with stock check and purchase of necessary materials and finally ends with delivery to the customer as well as the billing. No manufacturing is required, because the ordered material is a trading good. Before it is possible to start order-delivery process all necessary background information has to be created. During different phases the impact of different transactions will be examined. It will be also checked that all the related documents can be printed out from the system.

The procedure is following:

Necessary background information (= master data)

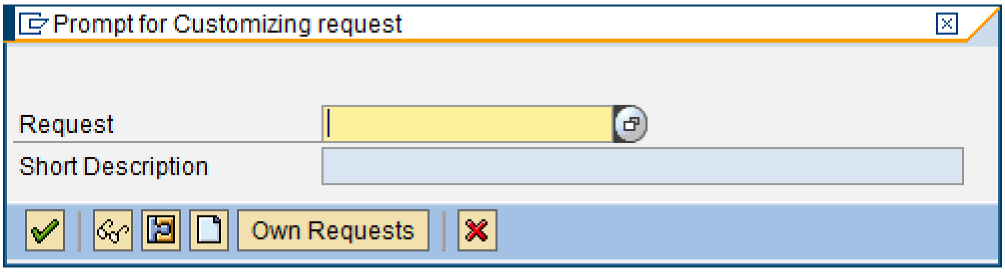
- Organizational units (others already existing, but sales area structure created)
- Customer and vendor information (will be copied from existing master data to new sales area)
- Material information (will be copied from existing material to new sales area)
- Price lists, discounts, pricing procedures (will be defined)

Processing a new sales order

- Create a sales order
- Material requirement planning (MRP) for the ordered material
- Generate purchase order form MRP Purchase request
- Receive the ordered material to stock
- Generate delivery from sales order (material will be picked from stock and delivered to the customer)
- Billing the customer
- Releasing the bill into financial accounting
- Examining of the sales order and its document flow

Results summary

In the first testing round all the test case exercise steps failed, because when trying to save the new sales organization a popup window "Prompt for Customizing request" appeared (see Figure 33.) There were no instructions in the exercise how to handle this situation. And since the tester was unable to create the new sales organization structure, also the subsequent steps failed.


Prompt for Customizing request

Request

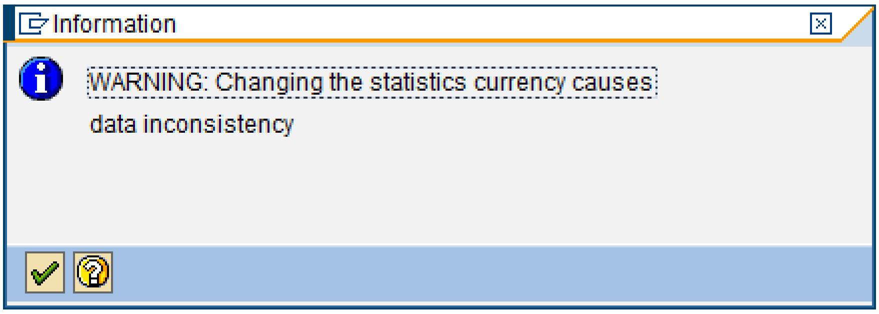
Short Description

Figure 33. Prompt for Customizing request.

In the second testing round, the tester solved the problem by creating an own customizing request, that was also used later on when building the sales organization structure further in the subsequent steps. Customizing requests are used in the SAP ERP environment to transport configuration changes to another SAP instances (refer to Chapter 2.1.2.). Typically customizations are first performed in the development instance, then they are transported to the quality assurance instance for testing, and only after that release to production instance. However, in this test case no transportation was required.

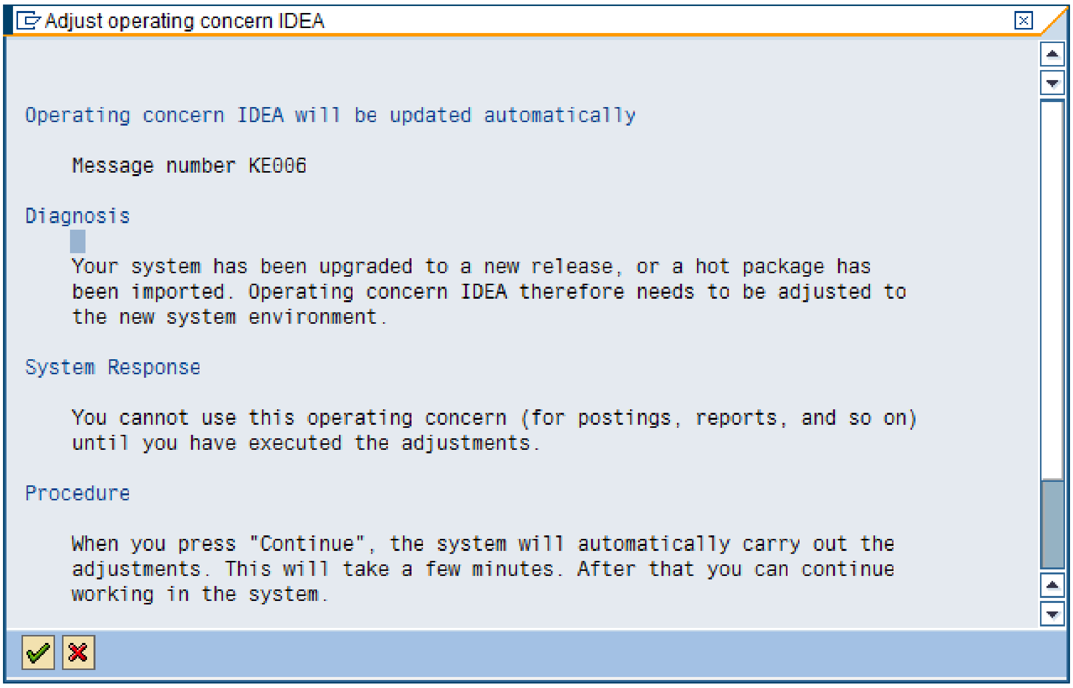
In the steps ”Assignment of enterprise structure”, when the new sales organization units should have been assigned correctly according the structure, the tester noticed that this activity case exercise instructions and screenshot did not match with the new upgraded ERP 6.0 version. The assignment activities of Sales and Distribution organization units in IMG had been replace by new transactions in SAP 6.0. The tester followed the new guidelines and new sales organization structure could be created as specified in the test case exercise.

At a later step, when a sales order was created, the following system message appeared.



WARNING: Changing the statistics currency causes data inconsistency

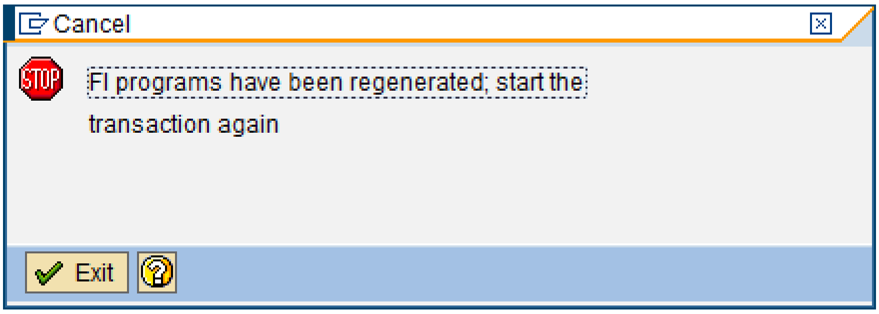
The tester clicked enter. The system displayed another message.



```
Operating concern IDEA will be updated automatically

    Message number KE006

Diagnosis

    Your system has been upgraded to a new release, or a hot package has
    been imported. Operating concern IDEA therefore needs to be adjusted to
    the new system environment.

System Response

    You cannot use this operating concern (for postings, reports, and so on)
    until you have executed the adjustments.

Procedure

    When you press "Continue", the system will automatically carry out the
    adjustments. This will take a few minutes. After that you can continue
    working in the system.
```

Figure 34. System message that appeared during testing.

The tester clicked enter once more. After that system carried out the updating automatically. Operation lasted a few minutes. After that the following message was displayed.



FI programs have been regenerated; start the transaction again

Figure 35. System messaged displayed after automatic system update.

After that the sales could be created without problems. The tester did not do anything that should have caused this automatic update. Supposingly, UCC Magdeburg had performed some updating activities in IDES environment in between and this is why the system messages appeared. Possibly updates were also related to Unicode conversion, because after the system performed automatic updates also the Unicode problem that occurred in the test case 3 disappeared.

The next steps in the test case could be executed without problems. However, the system performance was very slow (“wall clock” appeared for some time in every step). The only thing that failed in the second testing round was printing of invoice on paper. The tester was unable to solve the problem within a reasonable time and therefore this step failed.

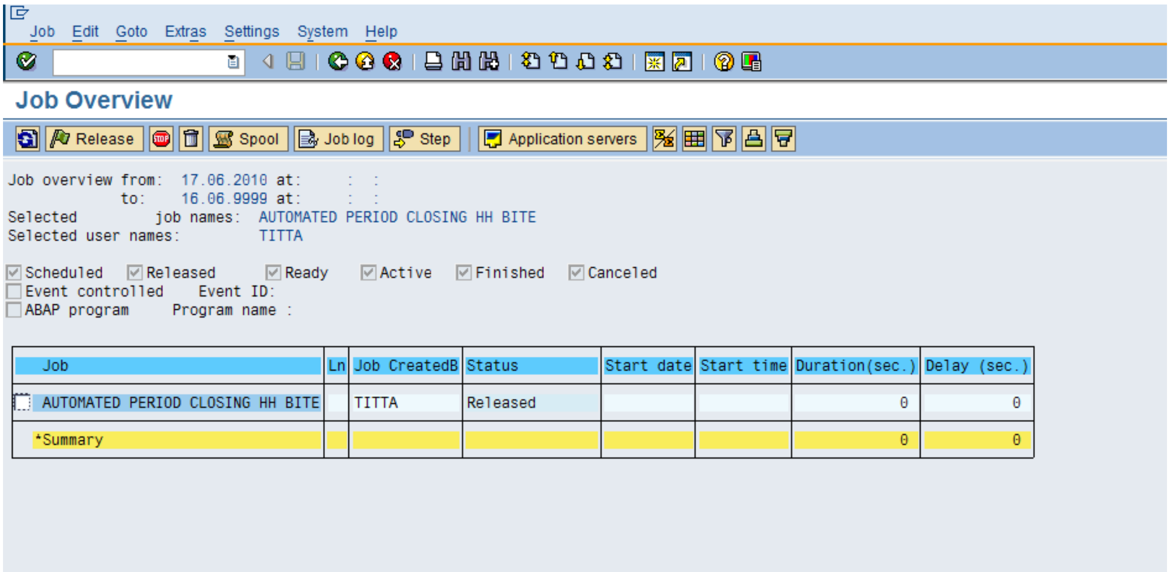
3.2.6 Test case 6: Automating period closing

Description

Period closing is a revision security mechanism incorporated in SAP ERP and it is meant for productive use. The reason for it is to impede illicit postings to prior periods. Automating period closing in SAP ERP required the following activities:

1. **Opening current posting period for the company code (FI module).** The booking periods for the company IDES AG were already opened until period 12/2015, so there was no need to create new periods.
2. **Close past period (MM module).** The current booking period was checked, and it was 5/2010, one period in the past of the actual company code period. Therefore the current posting period needed to be changed. This was done by closing the period 5/2010 and as result the current calendar month 6/2010 was set correctly as the current booking period.
3. **Automate scheduling (ABAP Workbench).**

After that current booking period was set to correspond the actual company code period, the automatic scheduling for period closing could be set. First step was to create the program that takes care of the scheduling, therefore the UCCPERI program was created as variant of pre-existing RMMMPERI program. The UCCPERI program was scheduled to perform period closing automatically on the “First day of current month”. Program was set to run as background job starting from 17.6.2010 (the testing date) till 16.6.9999.



```
Job overview from:  17.06.2010 at:     :  :
            to:     16.06.9999 at:     :  :
Selected          job names:  AUTOMATED PERIOD CLOSING HH BITE
Selected user names:          TITTA

☑ Scheduled  ☑ Released    ☑ Ready   ☑ Active  ☑ Finished  ☑ Canceled
□ Event controlled    Event ID:
□ ABAP program      Program name :
```

Job	Ln	Job CreatedB	Status	Start date	Start time	Duration(sec.)	Delay (sec.)
AUTOMATED PERIOD CLOSING HH BITE		TITTA	Released			0	0
*Summary						0	0

Figure 36. Screenshot of the job schedule that automates the period closing.

Results summary

All the steps could be tested without problems or errors. Functioning of the automatic period closing was verified on 1.7.2010, which was also the date the UCCPERI program was scheduled to perform period closing for the first time. The system had closed the period 6/2010 automatically and set the period 7/2010 as the current posting period for the company IDES AG, exactly as it was supposed to. Thus, the entire test case passed testing already in the first testing round.

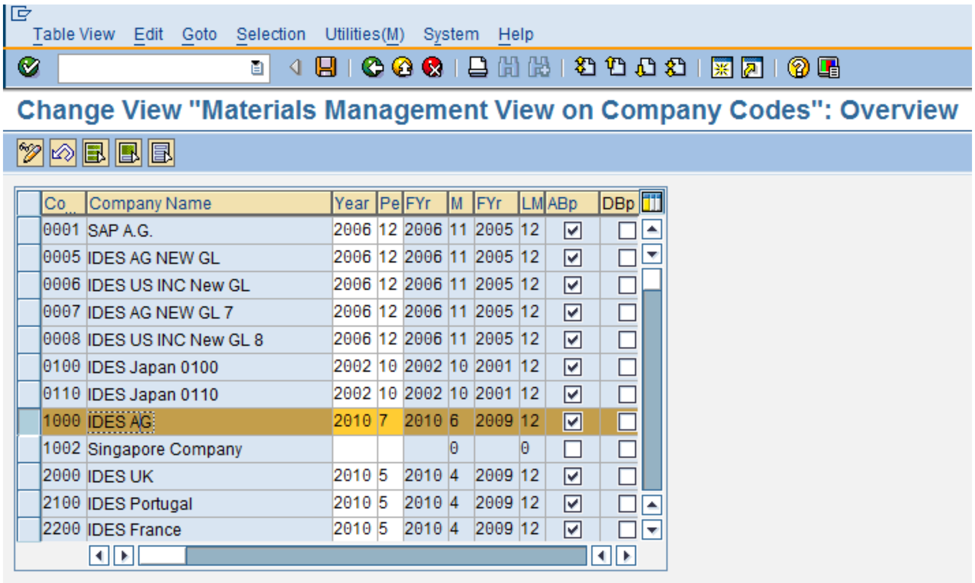


Table View Edit Goto Selection Utilities(M) System Help
Change View "Materials Management View on Company Codes": Overview

Co	Company Name	Year	Pe	FYr	M	FYr	LM	ABp	DBp
0001	SAP A.G.	2006	12	2006	11	2005	12	☑	□
0005	IDES AG NEW GL	2006	12	2006	11	2005	12	☑	□
0006	IDES US INC New GL	2006	12	2006	11	2005	12	☑	□
0007	IDES AG NEW GL 7	2006	12	2006	11	2005	12	☑	□
0008	IDES US INC New GL 8	2006	12	2006	11	2005	12	☑	□
0100	IDES Japan 0100	2002	10	2002	10	2001	12	☑	□
0110	IDES Japan 0110	2002	10	2002	10	2001	12	☑	□
1000	IDES AG	2010	7	2010	6	2009	12	☑	□
1002	Singapore Company				0		0	□	□
2000	IDES UK	2010	5	2010	4	2009	12	☑	□
2100	IDES Portugal	2010	5	2010	4	2009	12	☑	□
2200	IDES France	2010	5	2010	4	2009	12	☑	□

Figure 37. System performed the period closing successfully and set correctly 7/2010 as the current posting period.

4 Conclusion

A summary of the results of this research study are provided in this chapter. This research has tried to provide a deeper understanding of ERP upgrade and testing. The focus was on SAP ERP upgrade from the perspective of the user organization. The research was divided into two parts, theoretical and empirical. In the theory part, every effort was made to answer the research questions defined at the beginning of the research and to provide framework for the empirical testing part. The objective of the empirical part was to test the upgraded SAP ERP 6.0 environment of Haaga-Helia UAS to ensure that functionality and business processes required in the SAP ERP basics course of the Business Information Technology DP starting in the autumn semester 2010 were not affected and the course exercises of the previous implementation of the course, that functioned as test cases, were still suitable. To present the results first each research question will be answered, followed by the summary of the empirical testing part. The chapter is concluded with a research assessment, limitations and a recommendation for future research.

4.1.1 Answers to the research questions

The first research question was aimed at identifying the relationship of an ERP upgrade to ERP lifecycle. It was discovered that the commercial off the shelf ERP software has a life cycle of its own, since the development and maintenance activities apply to the client's installed version only. The ERP life cycle consist of different phases and the upgrade activities happen in the post-implementation phase of the life cycle. The post-implementation phase starts after the initial ERP implementation. Next was covered the second research question 'What is an upgrade' and as result the upgrade was defined as "a maintenance activity in the post-implementation phase of the ERP life cycle in that a software package provided by the vendor replaces client's installed version with a newer improved version from the same vendor". Then, it was discussed briefly, how upgrading to a new version differ from the initial implementation. The main difference of course is that while first time ERP implementation happens only once, ERP upgrade can occur many times after the first ERP implementation. Upgrade projects have a smaller scope and usually they are less complex. Therefore, they require also less time and money. On average, ERP upgrade projects only cost 18% of the initial ERP project cost. Often top management is less involved, there is less communication and resources and training in upgrade projects. However, this can cause project delays and even unsuccessful upgrade projects.

The reasons for an upgrade were discussed next in the theory part. The fact is that ERP systems cannot remain static after their initial implementation. The business processes for which a company adopted its ERP system at implementation are not necessarily the same processes it needs to track today. Rapidly changing business environment and evolving technology require continuous checking and upgrading of the ERP software to new version. But when making ERP upgrade decisions, an ERP client-organization must consider not only its internal organizational needs, but also future vendor maintenance support and upgrade compatibility. When justifying an upgrade, an organization should also consider the question, what are the risks of not upgrading? If an organization falls further and further behind the latest version, it risks the loss of standard vendor support. With an integrated ERP environment, the entire business is at risk if the ERP system becomes de-supported. Upgrades also tend to be more difficult, more complex, and have a greater impact on the user environment when upgrades are skipped. In addition, postponing new version upgrades may cause technological incompatibility problems, restrain benefit-realization from the ERP system and in the worst case threaten the competitive position. Organizations today rely heavily on their ERP software to run critical business activities and ERP is typically considered as a long time investment, therefore upgrade appears to be inevitable at some point of the ERP software life cycle. However, it is not optimal for an organization to upgrade its ERP each time a new version is introduced, because implementing ERP upgrade is a costly and time demanding process. Depending on the vendor, upgrades can occur multiple times in a year or once every several years. SAP's Release and Maintenance Strategy for SAP ERP determines the availability of new releases (including enhancement packages), the length and conditions of their maintenance, and the dependencies between individual releases. The mainstream maintenance is the original support period for a given release. For SAP ERP 6.0 it is 7 years. The 6.0 version was released in 2008 and for the moment the mainstream maintenance period continues till December 2015. Typically SAP recommends upgrading before you reach the end of the mainstream maintenance phase. After the mainstream maintenance period has concluded and the client chooses not to opt an upgrade, the client has the options of choosing either extended maintenance (at an additional fee, but with all benefits of the mainstream maintenance phase) or entering the customer-specific maintenance period automatically. The earlier versions SAP R/3 4.6C, SAP R/3 Enterprise software, mySAP ERP 2004 are still supported by SAP, but the mainstream maintenance for them has already ended. Extended maintenance for these versions is offered until the end of March 2013. Today, many organizations are in different stages of upgrading to SAP ERP 6.0. Approximately half of ERP clients are currently on releases that are two versions behind the current release; these may be four years old or more. SAP provides upgrade paths for SAP releases. Usually clients can upgrade directly (in one step) from one release to any other subsequent SAP release as long as the releases are in the mainstream maintenance phase or extended maintenance phase.

The research question, what should be considered when upgrading the ERP system, was rather vast. There are many factors to consider, whether related to IT, resources, project organization or technical considerations. Also the research questions about other important factors related to SAP ERP upgrade as well as risk and success factors are closely interconnected. Therefore in this research, the questions were discussed under the three main chapters – planning, managing, executing and testing an SAP ERP upgrade.

In the planning part, the upgrade justification process, why and when to upgrade, is important. This was just discusses above in connection with the previous research questions. Another important decision is to determine the upgrade approach organization takes. A technical upgrade approach focuses on a purely technological upgrade, without implementing new functionality that would change user behaviour or business processes. Performing a technical upgrade is a relatively fast and low cost effort with manageable impact and minimized disruption. A functional upgrade introduces new functionality. As of SAP ERP 6.0 instead of implementing a full new upgrade version, functional upgrade of SAP ERP can be implemented by activating just those functions provided by SAP enhancement packages that are needed from a business point of view. In generally, this approach requires more business process analyzing, testing and training, but the complexity can vary tremendously depending on the functional scope to be implemented. An organization should also try to estimate the cost and duration of upgrading to new release. Cost depends on a number of factors, including the extent of modifications, complexity of ERP system landscape, complexity of functionality, interfaces, skills and technical requirements, therefore exact calculations are hard. In addition, the chosen upgrade approach affects the total costs, e.g. functional upgrade may require longer implementation time as well as more user training and assimilation of new business processes. According to SAP upgrade experience database, the average duration of a technical upgrade to SAP ERP 6.0 is about three to five months. In most cases, it is also necessary for an organization to consider adjustments to their IT infrastructure for an upgrade. These adjustments may include resizing the application and database servers, deploying new front-end components, making network adjustments to maintain system performance, upgrading or migrating the operating system and database platform, and converting to Unicode. Other key considerations when upgrading are how much the organization has modified the SAP ERP software, how much is training required and is external assistance required. Customer-created modifications to the standard SAP ERP software directly affect the complexity of an upgrade, because modifications are overwritten during upgrades and the ones that are still needed after the upgrade have to be adjusted. Therefore, a significant part of the upgrade resources is spent on solving customer modification issues, adjusting and testing them. Depending on the scope and approach of the upgrade project, the organization might have to provide training not only for end users but also for the project team to ensure the success of the entire project. If organization's internal resources are limited or the staff lacks the skills required for an upgrade, the organization should consider hiring external assistance to keep the project on time and minimize the risk.

SAP ERP upgrades can be complex and demanding, but an upgrade project is like all other projects —it's all about skills, commitment from the business, and good project planning. In Chapter 2.4 Managing ERP Upgrade Project, more essential considerations to SAP ERP upgrade were highlighted. The success of the upgrade project depends heavily on the project people and the way the project is organized. Project team should include members with several skill sets and different background. Especially in case of a functional upgrade, when upgrade has major impact on the business processes, strong involvement of the business is required. Preferably, the project manager should be experienced in managing complex ICT or business affecting projects. Moreover, an ERP upgrade project should always have the support from the top management. To facilitate the efficient execution of critical project tasks and overall progress of the project, the organizations should use available project management guidelines, standards for project documentation, checklists and templates. SAP recommends its own SAP Upgrade Road Map methodology to successfully complete an upgrade project. It provides best practises for basic project management and functional and technical aspects to upgrade a typical SAP ERP system landscape.

In Chapter 2.4.3 based on ERP upgrade literature review, some factors that could threaten the successful outcome of an upgrade project were identified. These very same factors also identify risks related to an upgrade project, because often terms “critical success factors” and ”risk factors” are used to convey the same concept. The following success factors were identified: release maturity, upgrade timing, top management support, business driver, project scope, change management procedures, adequate planning, project management, communication, modification reversal, external support, user involvement, training, adequate testing, and use of upgrade tools.

The Chapter 2.5 looked at the execution phase of an upgrade and discussed two key upgrade challenges, the system landscape management and downtime minimization during an SAP ERP upgrade. The SAP ERP system landscape design alters during the upgrade process and therefore good landscape management is required. Landscape management consists of a clear plan that tells exactly in what order the systems are upgraded and what the status is of a certain system during the upgrade period. In addition, outlines were provided for how to set up and manage a standard three-system landscape in an upgrade project in order to minimize upgrade risk and minimize the duration of the code freeze period. Downtime is a big challenge especially for the businesses whose system requires 24/7 availability. Already during the planning phase the maximum downtime available should be determined and precise upgrade tasks to be performed during the cutover should be specified. Even if an organization plans when the downtime takes place, what steps are required and estimates the time required, it should also be prepared for unexpected problems such as operating system failures or human errors.

The Chapter 2.6 provided insight into testing, one of the most important factors to consider during ERP upgrade. Adequate testing was already mentined as one of the upgrade success factors. In this chapter also the following research questions were answered: Is testing necessary after upgrading? Why? What kind of testing activities are related to upgrade projects? SAP ERP users are required to perform adequate testing before production instance is upgraded and upgrade adjustments transported there in order to minimize the risk of potential errors that may affect key application functionalities and disrupt critical business processes. Although the actual amount of testing required during upgrade depends on complexity of the individual ERP landscape, the level of modifications and the functional scope of the project, there is no getting around the fact that a big part of the total upgrade effort will involve testing. Even in a purely technical upgrade thorough testing is essential. Organizations should expect to spend 25% of the project schedule and 30% of the total project costs on testing. In general, testing activities are divided in two categories, functional and technical. Functional testing ensures proper functionality of the software, and therefore verifies correctness. Technical testing tries to identify bottlenecks that slow down the system, and therefore ensures that a business process works quickly. These can in turn be further divided into subtypes that differ in testing integration level. The integration level increases from unit testing level of single objects to end-to-end integration test level where all SAP ERP business areas as well as other integrated SAP and non-SAP systems in the system landscape are tested. Also testing the technical upgrade process repeatedly is important. Repeated upgrades help to prepare an upgrade ”script” (also known as “runbook”), which can be used perform the upgrade smoothly when the ”real upgrade”, production cutover, is reached. Traditionally testing has been carried out manually. Now, as ERP software landscapes are coming more complex and business processes more spread-out, even outside the company, automated testing tools have been developed to make testing faster and more effective. Regardless of the approach, manual or automated, for both it is very essential to create accurate test cases. Without meaningful set of good test cases, that provide adequate coverage of the areas to be tested, automated testing will be no more effective than manual testing. In other words, what is tested is more important than how much is tested.

4.1.2 Summary of the empirical part

Empirical testing part consisted total of six test cases – five course exercises and an assignment to perform and automate period closing. Testing was conducted during the summer 2010 and detailed test report was submitted to the responsible instructor in form of a result table as agreed. The test result tables for each test case can be found in the research appendices.

The primary objective of testing in the empirical part was to ensure that the required functionality in the SAP ERP system supports the business processes as defined in the test cases. Testing was not restricted only to the business processes functionality but involved also testing of master data, student authorization rights, performance, printing as well as usability and consistency of the course exercise instructions that were used as test cases.

Tests were executed in two iterations. If defects or problems were found in the system during the first testing round the tester tried to solve the failed test case steps during the second testing round. However, the tester was not responsible of solving these. If the tester was unable to find any solution, step failed again. The problem solving or all the failed steps in the second testing round, was left to the instructor, as agreed.

There were no major problems with testing. With the first testing round the common problem with nearly all test cases was that the reference master data from which new master data was supposed to be copied from, did not exist. This was due to the fact, that when Haaga-Helia's SAP ERP IDES system was upgraded, the new system client number was provided by UCC Magdeburg. In SAP ERP master data is client dependent and the client data from the 5.0 version was not transported to the data tables of the upgraded system. Therefore it did not exist. In the second testing round, the tester solved the missing master data problem either by creating corresponding reference material or by using master data that existed in the client data tables as pre-populated IDES master data. However, in the test case 3 the tester was unable to create a BOM correctly and MRP-run could not be executed correctly. The problem was most likely related to some production routing or MRP-settings of the BOM. The BOM was prerequisite for all the other subsequent steps, therefore all the other steps but the first one failed. Another problem that tester was also unable to solve during the second testing round was printing of invoice on paper.

4.1.3 Research assessment, limitations and future research

This research has several limitations. Firstly, before the researcher started with this bachelor thesis project her knowledge about SAP ERP software was limited and the knowledge was tied to the courses taken at Haaga-Helia University of Applied Sciences. In order to fulfil the objectives of this study, the researcher has used quite a lot of time to gather and study resource material related to ERP upgrade in general, testing in general and material that is specifically related to SAP ERP, its upgrade and testing. As far as the researcher knows, there has not been done any academic research on the SAP ERP upgrade and testing topic. Also surprisingly little research attention has been given to ERP software upgrade in general. No research was found on ERP upgrade testing. Only material found that was directly related to SAP ERP upgrade and testing was published by the vendor SAP itself or consulting companies offering SAP services. Secondly, since the SAP ERP used by the Haaga-Helia is provided by University Competence Center Magdeburg in Germany, it does not exactly correspond with typical real-life SAP ERP environments in organizations. And since the entire SAP ERP system landscape is operated by UCC Magdeburg, they also took care of the upgrading process, so the researcher could not taken part in it. Neither has the researcher taken part in any real-life upgrade projects. Moreover, now afterwards considered, for a bachelor thesis the scope of the research questions was too broad. Either there should have been fewer questions or they should have been better defined. Therefore, the length of the research exceeds the suggested standard length for a bachelor thesis at Haaga-Helia. Also the required work effort was more than average 400 working hours.

Despite the above limitations, this study tries to answers to all research questions and thus provides a good information package on SAP ERP upgrade and how the testing should be handled during the upgrade project. It also gives an idea of how big of a process it is to upgrade a new SAP ERP version and what numerous things should be considered when planning, managing and executing the upgrade. The objectives of the empirical testing part were also achieved and therefore the thesis was useful for the sponsor. All the test cases were tested within the agreed time and test results were approved. The study will not only benefit the sponsor but will hopefully give the insight about upgrading and testing SAP ERP to students, researchers, upgrading organizations and tester as well. Also the secondary objective, learning during the thesis process, was achieved. The researcher has achieved better knowledge about ERP software in general (e.g. lifecycle) and SAP ERP (e.g. about period closing) but especially regarding the upgrade and testing. Also the skills about how to conduct a research have been improved.

For the future study it would be interesting to know how the new technologies, such as SOA and webservices, affect the upgrading process and testing. It would also nice to see some empirical studies on automated testing. The researcher also believes that due to the importance of ERP systems to organizations, it is important to keep developing academic research about upgrade, trying to depict new trends and the challenges associated to them.

Bibliography

3gerp 2011. The 3gERP Project. Project description. URL: http://www.3gerp.org/. Quoted 23.2.2011.

Anderson G. W. & Nilson C. D. & Rhodes T. 2009. SAP Implementation Unleashed: A Business and Technical Roadmap to Deploying SAP. Pearson Education, Inc.

Boler Consulting Pte Ltd 23.3.2008. Blog. SAP BASIS: SAP Resource Requirements. URL: http://bolerconsulting.blogspot.com. Quoted 11.62011.

Bortoulus D. 3.9.2010. Upgrading ERP Systems. URL: http://www.ehow.com/how_6944809_upgrading-erp-systems.html. Quoted 11.6.2011.

CFO & Agresso 2009. The High Cost of Change for ERP. What Does It Cost to Keep Up to Date? A report prepared by CFO Research Services in collaboration with Agresso. CFO publishing corp.

Computerworld UK 2010. SAP extends support for R/3 4.6c ERP until March 2013 URL: http://www.computerworlduk.com/news/applications/3241794/sap-extends-support-for-r-3-46c-erp-until-march-2013/. Quoted 23.2.2011.

Encyclopedia of Small Business. 2007. Enterprise Resource Planning (ERP). Retrieved Encyclopedia.com: http://www.encyclopedia.com/doc/1G2-2687200235.html. Quoted 25.2.2011.

ERP 2011a. ERP Vendors Tapping Into SaaS or Software as a Service ERP Applications. URL: http://www.erp.com/erp-archive/324-erp-archive/5405-erp-vendors.html. Quoted 23.2.2011.

ERPDB 14.5.2009. SAP System and Instances. URL: http://www.erpdb.info/sap-system-and-instances/. Quoted 10.6.2011.

Escobar S. & Toma N. 2007. Is permanent external consulting necessary for post-implementation stage? Bachelor's thesis within Informatics. Internationella Handelshögskolan. Jönköping.

Esteves J. & Bohórquez V. 2007. An updated ERP systems annotated bibliography: 2001-2005. Communications of the AIS, 19, pp. 386-446.

Getman G. 2008. Defining ERP: Enterprise Resource Planning Principles, Strategies, Practices and Technology. Grant Getman's blog, Etszone. URL: http://www.etszone.com/blog/define-erp-enterprise-resource-planning-principles-strategy-practices-technology. Quoted 23.2.2011.

Gidlund A. & Ribbenhed N. 2006. Tailoring of ERP systems. Influencing factors on the choice of adjustments. Master's thesis within informatics. Jönköping International Business School. Jönköping.

Hecht S. 2011. An approach for the management of ERP maintenance based on maturity models. Workshops on Information Systems and Services Sciences (WISSS) No. 21. Technische Universität München, pp. 187-191. URL: http://vmkrcmar23.informatik.tu-muenchen.de/1863/1/7th_WISSS_Proceedings.pdf#page=196. Quoted 23.5.2011.

Hedman J. 2003. On enterprise systems artifacts: changes in information systems development and evaluation. Doctoral Dissertation. Department of Informatics. School of Economics and Management. Lund University. Lund. URL: http://www.lub.lu.se/luft/diss/soc_459/soc_459.pdf. Quoted 26.5.2011.

Holm A. & Mattsson S. 2008. A benchmarking model for Maintenance and Support assignments in an ERP environment. Master's thesis. Department of Applied Information Technology. IT University of Göteborg. Göteborg.

Ittoolbox 2011. What is erp? URL: http://infor.ittoolbox.com/documents/what-is-erp-10860. Quoted 23.2.2011.

Kankaanpää I. & Pekkola S. 2010. Timing the Information System Upgrade. 18th European Conference on Information Systems (ECIS 2010) Proceedings. Paper 108, pp.1-12. URL: http://web.up.ac.za/ecis/ECIS2010PR/ECIS2010/Content/Papers/0338.R1.pdf. Quoted 1.6.2011.

Kidd, J. B. 2001. Discovering inter-cultural perceptual differences in MNEs. Journal of Managerial Psychology, 16(2), 106-126.

Kimberling E. 26.7.2010. Tips for avoiding common pitfalls of ERP upgrades. URL: http://searchmanufacturingerp.techtarget.com/news/2240020512/Tips-for-avoiding-common-pitfalls-of-ERP-upgrades. Quoted 20.6.2011.

KKA 2010. Korkeakoulujen arviointineuvosto. Ammattikorkeakoulujen koulutuksen laatuyksikköarviointi vuonna 2010 alkavalle toiminta- ja tuloskaudelle. HAAGA-HELIA AMK, Degree Programme in Business Information Technology. URL: http://www.kka.fi/files/739/HAAGA-HELIA_business_information_technology.doc. Quoted: 1.6.2010.

Kouki R. 2009. Enterprise Resource Planning Post-Implementation Assimilation Challenge: An Integrative Framework for a Better Post-Implementation Assimilation. Dissertation. Faculté des Sciences de L'Administration, Université Laval, Québec.

Law C. &, Chen C. & Wu B. 2010. Managing the full ERP life-cycle: Considerations of maintenance and support requirements and IT governance practices integral elements of the formula for successful ERP adoption. Computers in Industry, 61, 3, pp. 297-308.

Martinsen H. 2010. Attaining important ERP upgrade factors. A case study of an off-shore company. Master of Science Thesis. Report No. NM-10/7. Department of Shipping and Marine Technology. Chalmers University of Technology. Gothenburg.

Moad J. 25.2.2011. ERP Upgrades: Who Needs Them? URL: http://www.manufacturing-executive.com/community/enterprise_resource_planning/blog/2011/01/25/erp-upgrades-who-needs-them. Quoted 30.5.2011.

Monk, E. & Wagner, B. 2009. Concepts in Enterprise Resource Planning. 3rd edition. Course Technology Cengage Learning. Boston.

Muir, N. & Kimbell, I. 2010. Discover SAP. 2nd edition. Galileo Press. Bonn.

Mukherji, N. & Rajagopalan B. and Tanniru M. 2006. A decision support model for optimal timing of investments in information technology upgrades. Decision Support Systems, 42, 1684-1696.

Myers, M. D. 2010. Qualitative Research in Information Systems. MIS Quarterly (21:2), June 1997, pp. 241-242. MISQ Discovery, archival version, June 1997 URL: http://www.misq.org/discovery/MISQD_isworld/. MISQ Discovery, updated version, last modified: September 23, 2010. URL: www.qual.auckland.ac.nz. Quoted: 2.6.2010.

Ng C.S.P. & Gable G. G. 2010. Maintaining ERP packaged software – a revelatory case study. Journal of Information Technology, 25, 1, pp. 65-90.

Ng C. S. P. & Gable G. & Chan T. 2003a. A Revelatory Case Study into the Adequacy of Standard Maintenance Models in an ERP Context. PACIS 2003 Proceedings. Paper

72. p. 1039-1054. URL: www.pacis-net.org/file/2003/papers/**erp**/208.pdf. Quoted 25.2.2011

Ng C. S. P. & Gable G. & Chan T. 2003b. An ERP Maintenance Model. HICSS '03 Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03), 8, 8, pp. 234–243.

Ng C.S.P. & Gable G. G. & Chan T. 2002. An ERP-client benefit-oriented maintenance taxonomy. Journal of Information Technology, 64, 1, pp. 87-109.

Ng C.S.P. & Chan T. 1999. Optimal Timing for ERP Upgrade: a Preliminary Model. 3rd Annual SAP Asia Pacific, SAPPHIRE '99, Singapore. Institute of Higher Learning Forum, pp. 102-114. URL: ftp://www.pathguy.com/pub/modula-2/gpm/ism/Papers/CNg99-2.pdf. Quoted 6.6.2011.

Nurmi J. 2009. SAP-järjestelmän jatkokehittämisen ja ylläpidon prosessien kehittäminen Kelassa [Improving the support and further development processes of SAP-Systems in the Social Insurance Institute of Finland]. Bachelor's thesis. Haaga-Helia University of Applied Sciences. Helsinki.

Orlikowski, W.J. & Baroudi, J.J. 1991. Studying Information Technology in Organizations: Research Approaches and Assumptions. Information Systems Research (2), pp. 1-28.

Otieno J. O. 2010. Enterprise Resource Planning Systems Implementation and Upgrade (A Kenya Study). Thesis. School of Engineering and Information Sciences. Middlesex University.

Panaya 2011a. Preparing for Your SAP Upgrade - A New Study from Panaya Reveals Common Issues and Best Practices. URL: http://www.panayainc.com/Articles/Preparing-for-Your-SAP-Upgrade-A-New-Study-from-Panaya-Reveals-Common-Issues-and-Best-Practices.html. Quoted 14.6.2011.

Panaya 2011b. 2011 SAP Testing Survey. URL: http://go.panayainc.com/2011-SAP-Testing-Survey.html. Quoted 26.6.2011.

Panaya 2010a. 2010 SAP Upgrade Benchmark Survey. Survey Results and Executive Summary. URL:
http://go.panayainc.com/rs/panayainc/images/2010_SAP_Upgrade_Survey.pdf. Quoted 28.5.2011.

Panaya 2010b. SAP Support Package Implementations: Which Policy Should You Choose? URL:
http://go.panayainc.com/rs/panayainc/images/SAP_Support_Packages.pdf. Quoted 11.6.2011.

Paul L. G. 19.1. 2009. ERP upgrade planning best practices demystified. URL: http://searchmanufacturingerp.techtarget.com/news/1345398/ERP-upgrade-planning-best-practices-demystified. Quoted 20.6.2011.

Peslak A. & Subramanian G. & Clayton G. 2008.The Phases of ERP Software Implementation and Maintenance: A Model for Predicting Preferred ERP Use. Journal of Computer Information Systems. Winter 2007-2008, 48, 2, pp. 25-33. URL: http://www.iacis.org/jcis/pdf/Peslak_Subramanian_Clayton_48_2.pdf. Quoted 18.5.2011.

Portougal V. & Sundaram D. 2006. Business Processes: Operational Solutions for SAP Implementation. IRM Press. Hershey.

Rashid M. A., Hossain, L. & Patrick, J. 2002. Enterprise Resource Planning: Global Opportunities and Challenges. Idea Group Publishing. Hershey PA.

Reed J. 2010. 49 Do’s, Don’ts, and Customer Lessons for SAP Upgrades

Going beyond the SAP Upgrade Clichés. Panaya. URL: http://go.panayainc.com/rs/panayainc/images/49_Dos_Donts_and_Customer_Lessons_for_SAP_Upgrades.pdf. Quoted 11.6.2011.

Riedel M. 2009. Managing SAP ERP 6.0 Upgrade Projects. Galileo Press. Boston.

Salmeron J. & Lopez C. 2010. A multicriteria approach for risks assessment in ERP maintenance. Journal of Systems and Software, 83, 10, pp. 1941-1953.

Salonen, M. 2006. Koulutuksen suunnittelu ja toteutus sekä muutoksenhallinta toiminnanohjausjärjestelmän käyttöönotossa. Diplomityö. Lappeenrannan teknillinen yliopisto, tuotantotalouden osasto. URL: https://www.doria.fi/bitstream/handle/10024/30082/TMP.objres.281.pdf?sequence=1. Quoted 24.2..2011.

SAP 2010a. SAP korkeakoulujen tukena. Press release. URL: http://www.sap.com/finland/about/pressroom/press.epx?pressid=12966. Quoted: 1.6.2010.

SAP 2010b. SAP homepage / About SAP. URL: http://www.sap.com/about/index.epx. Quoted: 1.6.2010.

SAP 2011. SAP ERP. SAP solution map. www.sap.com/finland/solutions/business-suite/erp/iratkaisukartta. URL: http://www.sap.com/global/scripts/jump_frame.epx?content=/businessmaps/82087E7A8EB94E1F885C5A1F62C789FB.htm&CloseLabel=. Quoted 23.2.2011

SAP 2007. How to Plan for an SAP ERP 6.0 Upgrade. Brochure 50 082 075 (09/07). URL: http://peningoblog.com/sap/downloads/whitepapers/sap_erp_upgrade.pdf. Quoted 11.6.2011.

SAP HELP 2010. SAP Help Portal. IDES – The SAP Model Company. URL: http://help.sap.com/saphelp_46c/helpdata/en/af/fc4f35dfe82578e10000009b38f839/frameset.htm. Quoted: 1.6.2010.

SAP SDN 2010. The University and Academic Competence Centers in Europe, the Middle East and Africa. Navigation: www.sdn.sap.com/ University Competence Centers. URL: http://www.sdn.sap.com/irj/uac/ucc. Quoted: 1.6.2010.

SAP SDN 2008. How to Design a SAP NetWeaver - based System Landscape (NW7.0). Brochure, Version 1.0 (03/08). URL: http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/50a9952d-15cc-2a10-84a9-fd9184f35366. Quoted 10.6.2011.

SAP Support Portal 2011a. SEPA Overview. URL http://service.sap.com/sepa. (Customer login required). Quoted 28.5.2011.

SAP Support Portal 2011b. SAP Business Suite: Getting Started with an Upgrade. Brochure 50 082 075 (03/11). URL: http://service.sap.com/upgrade. (Customer login required). Quoted 9.6.2011.

SAP Support Portal 2011c. Maintenance strategy - Maintenance phases. URL: http://service.sap.com/maintenance. Quoted 10.6.2011.

SAP Support Portal 2011d. SAP Maintenance Strategy (EN). Navigation: Maintenance strategy - Maintenance phases – Media Library – SAP Maintenance Strategy (EN). URL: http://service.sap.com/maintenance. Quoted 10.6.2011.

SAP Support Portal 2011e. Frequently Asked Questions (EN). Navigation: Maintenance Strategy - Frequently Asked Questions (EN). URL: http://service.sap.com/maintenance. (Customer login required). Quoted 10.6.2011.

SAP Support Portal 2011f. SAP'S Release Strategy For Large Enterprises. For all major shipments from 2005 onward Status: March 2011. Brochure 50 019 667 (03/11). URL: http://service.sap.com/releasestrategy. (Customer login required). Quoted 10.6.2011.

SAP Support Portal 2010a. SAP Enhancement Package 5 for SAP ERP. SAP Business Suite. Presentation Version 6.0 (20/12/10). Navigation: SAP Enhancement packages for SAP ERP/Media Library - SAP Enhancement packages /Enhancement packages EHP5 /Technology Facts EHP 5 for SAP ERP. URL: http://service.sap.com/erp-ehp. (Customer login required)Quoted 9.6.2011.

SAP Support Portal 2010b. Upgrading to SAP ERP - Frequently Asked Questions for Customers of SAP Software. Brochure 50 084 034 (11/10). URL: http://service.sap.com/upgrade-erp. (Customer login required). Quoted 8.6.2011.

SAP Support Portal 2010c. Transition and upgrade to SAP ERP. Presentation. Version 3.5 (22/05/10). Navigation: SAP ERP - Media Library – SAP ERP Overview – Upgrade – Transition and upgrade to SAP ERP. URL: http://service.sap.com/erp. (Customer login required). Quoted 8.6.2011.

SAP Support Portal 2008. SAP Standard for Test Management. Whitepaper. Version 1.0. Navigation: Application Life-Cycle Management / Processes /Test Management / Standard for Solution Operations. URL: http://service.sap.com/alm-processes. Quoted 21.6.2011.

SAP UA EMEA 2010a. SAP University Alliance EMEA Portal. Navigation: portal.ucc.uni-magdeburg.de/UA Concept. URL: https://portal.ucc.uni-magdeburg.de/irj/portal/anonymous?NavigationTarget=navurl://947dc385438972c8cb8dffce31075db9&windowId=WID1289486535488. Quoted: 1.6.2010.

SAP UA EMEA 2010b. SAP University Alliance EMEA Portal. Navigation: portal.ucc.uni-magdeburg.de/Services. https://portal.ucc.uni-magdeburg.de/irj/portal/anonymous?NavigationTarget=navurl://fddf2b45b4f53d48272f5b79970b5952&windowId=WID1289486148731. Quoted: 1.6.2010.

SAP UA Finland 2010. SAP korkeakouluyhteistyö Suomessa. URL: http://www.sap-uni-alliance.fi. Quoted: 1.6.2010.

SearchCRM 2011. What is implementation? URL: http://searchcrm.techtarget.com/definition/implementation. Quoted 25.5.2011.

Search Manufacturing ERP 3.8.2010. FAQ: ERP upgrade best practices. URL: http://searchmanufacturingerp.techtarget.com/news/2240020891/FAQ-ERP-upgrade-best-practices. Quoted 20.6.2011.

Selchert, M. 2004.Value Added Through Sap Best Practices. Empirical international study with users of SAP Best Practices. Ludwigshafen University of Applied Sciences, Ludwigshafen. URL: http://help.sap.com/bp_smv2007/SSM_US/Documentation/Ludwigshafen_sap_best_practices_study_summary.pdf. Quoted 24.2.2011.

Sens, M. 2008. Upgrading SAP. Infinity Science Press LLC. Hingham, Massachusetts.

Stackpole B. (17.12. 2008). Five signs it's time for an ERP system upgrade. URL: http://searchmanufacturingerp.techtarget.com/news/1342896/Five-signs-its-time-for-an-ERP-system-upgrade. Quoted 6.6.2011.

Stake, E. R. 1995. The Art of Case Study Research. Sage Publications. Thousand Oaks, California.

Storani, M. 2008. Storani’s blog. Enterprise resource planning (ERP) – Concepts, Methods and Frameworks. URL:

http://mauriziostorani.wordpress.com/2008/12/06/enterprise-resource-planning-erp-concepts-methods-and-frameworks/. Quoted 22.2.2011.

Sullivan L. 2009. Post-Implementation Success Factors for Enterprise Resource Planning (Erp) Student Administration Systems in Higher Education Institutions. Dissertation. College of Education at the University of Central Florida Orlando, Florida.

Sumner, M. 2005. Enterprise Resource Planning. Pearson Education Inc. Upper Saddle River, New Jersey.

Taylor S. & Lacy S. & MacFarlan I. 2007. ITIL Service Transition. The Stationery Office. United Kingdom.

Vaman J.N. 2008. ERP in practice: ERP strategies for steering organizational competence and competitive advantage. Tata McGraw-Hill, New Delhi.

Wenrich, K & Ahmad, N. 2009. Lessons learned During a Decade of ERP Experience: a Case Study. International Journal of Enterprise Information Systems, 5, 1, pp. 55-75.

Wikipedia 2011a, The Free Encycolopedia. Search word: Enterprise resource planning. URL: http://en.wikipedia.org/wiki/Enterprise_resource_planning. Quoted. 23.2.2011

Wikipedia 2011b, The Free Encyclopedia. Search word: Software testing. URL: http://en.wikipedia.org/wiki/Software_testing. Quoted 28.6.2011.

Worrell, J. L. 2008. Running the Erp Marathon: Enhancing Erp - Business Fit in the Post-Implementation Phase. Dissertation. Florida State University, College of Business.

Yin, R. K. 2002. Case Study Research, Design and Methods. 3rd ed. Sage Publications. Newbury Park.

Zhao, F. 2007. An Empirical Study of Enterprise System Upgrades. A Doctoral Dissertation. University of Nebraska. Lincoln, Nebraska.

Attribution Statement and Disclaimer

Attribution Statement

SAP®, SAP® R/3®, mySAP.com® mySAP™, SAP® R/3® Enterprise, SAP NetWeaver®, SAP® Business ByDesign™, ABAP™, SAP® Business Suite, SAP® Customer Relationship Management (SAP CRM), SAP® ERP, SAP® ERP Human Capital Management (SAP ERP HCM), SAP® Best Practices, SAP® Business One, SAP® Business All-in-One, SAP NetWeaver® Business Intelligence (SAP NetWeaver BI), SAP® Business Information Warehouse (SAP BW), SAP® Solution Manager are the trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Disclaimer

This publication contains references to the products of SAP AG. SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, Clear Enterprise, SAP BusinessObjects Explorer, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP France in the United States and in other countries.

SAP AG is neither the author nor the publisher of this publication and is not responsible for its content. SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

Appendices

Appendix 1. Test result table for the test case 1

I TESTING ROUND **Test case 1: SAP R/3 User interface exercise**				**Date:** **9.06.2010**
Steps	**Result (passed/ FAILED)**	**Notes**	**Output /outcome**	**Other remarks**
1				Old version R/3 mentioned →ECC 6.0
1.1	passed	-	Shp. Cond. = as soon as possible	
1.2	passed	-	Shp. point = Hamburg 1000	
2	passed	-	Account name: Trade receivables - domestic	
3				
3.1	passed	-	932	- Instructions not clear enough. - System Status on a wrong row -> to the next row /page.
3.2	passed	-	SAP ECC 6.0	Old version R/3 mentioned →ECC 6.0
3.3	passed	-	932	
3.4	passed	-	Oracle	
4				
4.1	passed	-	5000 customer entries found	
4.2	passed	-	260 customers in Berlin	
4.3	passed	-	Hender*: 2 entries	
4.4	passed	-	Customers with name Becker: 32 entries	
4.5	passed	-	Customers in Frankfurt: 215 entries	
5				
5.1	passed	-	Customer no.: 2300	
5.2	passed	-	VAT reg.no.: DE123456789	
5.3	passed	-	Contact person: Schäfer Robert	
5.4	passed	-	Birthday: 12.09.1958	

5.5	passed	-	No. of employees: 90	
6				- Old version R/3 mentioned →ECC 6.0
6.1	passed	-	Materials DP*: 163 entries	
6.2	passed	-	Materials *yellow*: 36 entries	
6.3	passed	-	Materials type “semi-finished product” & *25: 59 entries	
7				- Old version R/3 mentioned →ECC 6.0
7.1	passed	-	Trading goods	
7.2	passed	-	0,015	
7.3	passed	-	EUR	
7.4	passed	-	1.518	
8	FAILED	Sales order 13601 cannot be found in the database.	-	
8.1	FAILED	Sales order 13601 cannot be found in the database.		

II TESTING ROUND **Test case 1: SAP R/3 User interface exercise**				**Date: 15.06.2010**
Steps	**Result (passed/ FAILED)**	**Notes**	**Output /outcome**	**Other remarks**
8	passed	Solution: Display sales order 11663.		
8.1	passed		Delivery document:80014914 Invoice:90036014 *Delivery document* Created by:MAASSBERG Date & time:22.12.2005, 04:00:52 *Invoice* Created by:MAASSBERG Date & time: 27.12.2005, 04:03:34	

Appendix 2. Test result table for the test case 2

I TESTING ROUND Test case 2: SAP R/3 order-delivery process, Exercise RT 3a K2010 ver. 2.3.2010				Date: 10.06.2010
Steps	Result (passed/ FAILED)	Notes	Output /outcome	Other remarks
1	FAILED	Material KTA98 does not exist	-	
2	FAILED	Error: Customer 34999 has not been created	-	
3	FAILED	Requires step 1 be passed. Material needs to exist before price can be maintained.	-	
4	FAILED	Requires step 1 be passed. Material needs to exist before discount can be created.	-	
5	FAILED	Requires step 1 be passed. Material needs to exist before material can be maintained for MPR.	-	
6	FAILED	Error: Vendor 14399 has not been created.	-	
7	FAILED	Requires step 1 and 5 be passed. Vendor cannot be defined to be source of the material if they both don't exist.	-	
8	FAILED	Requires steps 1, 2, 3, 4, 5 to be passed. Sales order cannot be created without material and customer master data.	-	
9	FAILED	Requires steps 1, 5, 6, 7 to be passed. Purchase price cannot be defined without material and vendor master data.	-	
10	FAILED	Requires step 1 to be passed. Checking the materials amount in stock is not possible if material does not exist.	-	

11	FAILED	Requires step 1 and 5 to be passed. Material cannot be processed with MRP if material master data does not exist and if material is not maintained for MPR. Checking the materials amount in stock is not possible if material does not exist.	-	
12	FAILED	Requires step 1 and 11 to be passed. Checking the materials amount in stock is not possible if material does not exist and the result of MPR process cannot be seen in stock amounts.	-	
13	FAILED	Requires step 12 to pass. Purchase request cannot be converted to purchase order if MPR fails.	-	
14	FAILED	Requires step 13 to pass.	-	
15	FAILED	Requires steps 1, 11, 14 to pass.	-	
16	FAILED	No documents were created, because all the previous steps failed.	-	
17	FAILED	Requires at least step 8 to be passed. No delivery can be done, because no sales order could be created.	-	
18	FAILED	Requires step 1 to be passed. Checking the materials amount in stock is not possible if material does not exist (neither can the decrease in amount be seen after delivering the products to the customer).	-	
19	FAILED	No document flow because all the previous steps failed.	-	
20	FAILED	Billing the customer not possible, because all the previous steps	-	

		failed.		
21	FAILED	Requires step 20 to pass. Releasing the bill for accounting not possible, because bill could not be created.	-	
22	FAILED	No document flow because all the previous steps failed.	-	

II TESTING ROUND **Test case 2: SAP R/3 order-delivery process, Exercise RT 3a K2010 ver. 2.3.2010**				**Date: 16.06.2010 – 17.06.2010**
Steps	**Result (passed/ FAILED)**	**Notes**	**Output /outcome**	**Other remarks**
1	passed	Solution: a material **TS10** is created in ECC 6.0, exactly for same organization structure and with same information as the KTA98 in ECC 5.0, which is used as a copy material for new materials.	Material: **BITE10_TS**	
2	passed	Solution: customer **34999,Bite GmbH** exactly with same customer number, under same company code, sales organization, distribution channel and division was created as a base for students to copy their new customer for the exercise.	Customer: **44499, Meisterwerk GmbH**	- Gives a warning that “Reference account group is Payer” → enter - For Account group Sold-to party-0001 lower limit is 0000000001and upper limit is 0000099999 so in the exercise the given students customer number pattern needs to be changed., e.g. 4444nn→444nn
3	passed	-	-	What is the meaning of this step? Not very clear for the student.
4	passed	-	-	What is the meaning of this step? Not very clear for the student.
5	passed	-	Message: “The material BITE10_TS has been	What is the meaning of this step? Not very clear for the student.

			marked for the MRP run”	
6	passed	Solution: already in data base existing vendor **1005**, **PAQ Deutschland GmbH** under same company code and purch. organization was selected as a reference vendor when copying.	Vendor: **33399, Vendor TS**	
7	passed	-	Message “Source list changed”	
8	passed	-	Standard order: **12061** Printing: OK	Purch. order no → in system exact field name is PO Number
9	passed	-	Message: “Purchasing info record 5300005706 1000 1200 created”	
10	passed	-	Stock: -10	
11	passed	-	MRP carried out. A pur.rgs. 0010013563/00010 was created.	
12	passed	-	Available quantity 10.	
13				
13.1	passed	-	-	
13.2	passed		- Standard PO created under the number **4500017174** Printing: OK - Stock status Pch.ord: 10	This step is somewhat difficult to understand how to move the purchase request to the shopping basket if you are doing it the first time. It would be nice to at least change the word ‘move’ → ‘drag and drop’, or even better to add some screenshots.
14	passed	-	Material document **5000011981** posted	First when trying to save Error message: “Posting only possible in periods 2010/05 and

				2010/04 in company code 1000" appeared. Posting date was changed to 31.5.2010 after that saving was possible and material document was created. Solution: Posting period closed and new 2010/6 opened since the testing date was in June 2010. This was done according the test case 6 – Period opening and closing. After that saving was possible and material document was created.
15	passed	-	Stock available quantity 10.	
16	passed	-	Trading goods account number 310000, € 100,00	
17	passed	-	Availability check was carried out.	
17.1	passed		Delivery **80015175** Printing: OK	
17.2				
17.3	passed		Message: Delivery 80015175 has been saved	First when trying to do Post goods issue: "Posting only possible in periods 2010/05 and 2010/04 in company code 1000" appeared. Solution: Posting period closed and new 2010/6 opened since the testing date was in June 2010. This was done according the test case 6 – Period opening and closing. After that posting goods issue –document was possible
18	passed		Stock empty, availability 0.	
19	passed	-	-	

20	FAILED	Invoice could be created correctly, but printing on paper failed, displayed on screen ok..	Invoice **90036254** Printing: FAILED	
21	passed	-	Message: The document has already been passed on to accounting	Step 21 is unnecessary since the invoice is released automatically for accounting?
22	passed	-	-	

Appendix 3. Test result table for the test case 3

I TESTING ROUND Test case 3: Exercise RT 3b part1: Master Data & part2: Production				Date: 11.06.2010
Steps	**Result (passed/ FAILED)**	**Notes**	**Output /outcome**	**Other remarks**
1				
1.1	FAILED	Material KTA98 does not exist in the system.	-	
1.2	FAILED	Material KTC98 does not exist in the system.	-	
1.3	FAILED	Requires steps 1.1 and 1.2 to pass. BOM cannot be created because the material master data is missing.	--	
2	FAILED	Requires 1.2 to pass. Routing cannot be created for main product if it does not exist in the sys-tem.	-	
3				
3.1	FAILED	Requires 1.2 to pass. Sales order cannot be created for main product if it does not exist in the system.	-	
3.2	FAILED	Requires 1.2 and 3.1 to pass. MRP cannot be started and purchase order created because main product does not exist in the system and if it has not been included in any sales order.	-	
3.3	FAILED	Requires step 3.2 to pass. If no purchase order is created, the products	-	

		cannot be received to stock.		
3.4	FAILED	Requires step 3.3 to pass. The main product cannot not be manufactured if no components in stock.	-	
3.5	FAILED	Requires step 3.4 to pass. If no main products were manufactured, the amount cannot be confirmed in the system.	-	
3.6	FAILED	Requires all the previous steps to be passed. The main product cannot be delivered to a customer because it does not exist.	-	

II TESTING ROUND **Test case 3: Exercise RT 3b part1: Master Data & part2: Production**				**Date: 22.06.2010 /1.7.2010/10.8.2010**
Steps	**Result (passed / FAILED)**	**Notes**	**Output /outcome**	**Other remarks**
1				
1.1	PASSED(1.7)	Material KTA98 does not exist in the system. Solution: use a material **TS10**as an existing reference material. That was created in test case 2 (step 1) in ECC 6.0, exactly for same organization structure and with same information as the	Material: **WHEEL10_TS** Material: **BARROW10_TS**	Add menu path for the transaction mmh1.

		KTA98 in ECC 5.0.		
1.2	FAILED	Material KTC98 does not exist in the system. DID NOT WORK! Solution: tester created a new main material **TSC1** in ECC 6.0 by copying the material T-F1220 that is already in the IDES system existing BOM. (Includes all the BOM and routing information).	DID NOT WORK! Material: **WHEELBARROW_TS1**	TESTER'S NOTES FOR OWN USE, NOT PART OF THE RESULTS: NOTE FOR THIS POINT SEE https://forums.sdn.sap.com/thread.jspa?threadID=632125&tstart=51240' How system will come to know which BOM and Routing to be used. This is done thru Prod.Version. In production version we maintain the combination of BOM and routing. Go to MM02--->**MRP4** / Work scheduling view ---->Prod.Version. Enter the validity period and lot size. and the production version should be unlocked. After entering the reqd. routing no. and BOM alternative , carry a check. After getting the Green signals ,Continue. Thus you have saved the prod. version. (OLD Solution: a material **TSC10** is created in ECC 6.0, exactly for same organization structure and with same information as the KTC98 in ECC 5.0, which is used as a copy material for new materials. (Note: MRP4. tab 'Definerepetitive-manufacturing for this material … not sure if settings same than in old KTC)
1.3	FAILED	-	DID NOT WORK! Message: Creating BOM for material WHEELBARROW_TS1 - Messages (2 times, when defining purchase prices): Purchasing info record xxxxxxxxxx 1000 1200 created - Messages: The material WHEELBARROW_TS1/ WHEEL10_TS /BARROW10_TS	- 1.3 b) Maybe for clarification should be noted that the current date can be accepted in the **Valid from field.** **-** For clarification the step for maintaining material price could be marked d) and the step for maintaining products to MRP could be marked d). - When maintaining material prices the system gives a warning message: "Material WHEELBARROW_TS1 mainly procured internally (please check your input)". Just press Enter. ????

			has been marked for the MRP run	- The sentence 'Maintain material price of 600 euros (see RT_3a task 3) for …' is repeating what was said already above.
2	FAILED	-	DID NOT WORK! Message: Routing was saved with group 50001298 and material WHEELBARROW_TS1	
3				
3.1	FAILED		DID NOT WORK! Standard Order **12162** has been saved	During testing on 5.8.2010 when trying to create the sales order, the system error message related to Unicode appeared, therefore sales order could not be created. At later point on 10.8. when executing the test case 5 and creating a sales order there, a message appeared informing that Unicode conversion has been performed. Supposingly, they had performed Unicode conversion or some updated regarding to it at UCC Magdeburg inbetween. In any case the Unicode error message did not appear in this test case when creating the sales order again.
3.2	FAILED		DID NOT WORK! MRP carried out for WHEELBARROW_TS1 1200 1 200	
3.3	FAILED	Requires step 3.2 to pass. If no purchase order is created, the products cannot be received to stock.	-	
3.4	FAILED	Requires step 3.3 to pass. The main product cannot not be manufactured if no com-	-	

		ponents in stock.		
3.5	FAILED	Requires step 3.4 to pass. If no main products were manufactured, the amount cannot be confirmed in the system.	-	
3.6	FAILED	Requires all the previous steps to be passed. The main product cannot be delivered to a customer because it does not exist.	-	

Appendix 4. Test result table for the test case 4

I TESTING ROUND **Test case 4: SAP Order-delivery process exercise**			**Date: 11.06.2010**
Steps	**Result (passed/ FAILED)**	**Notes**	**Other remarks**
1. Master data			
1.1	FAILED	Reference material KTA98 does not exist in the system.	
1.2.	FAILED	Requires step 1. to pass. Material cannot be maintained in MRP if it does not exist in the system.	
1.3.	FAILED	Reference customer 34999 does not exist in the system.	
1.4.	FAILED	Reference vendor 14399 does not exist in the system.	
1.5.	FAILED	Requires steps 1 and 4 to pass. Vendor cannot be defines as source of a material if material and vendor don't exist in the system.	
2 . Pricing			
2.1.	FAILED	Requires steps 1.1 to pass. Material sales price cannot be maintained if the material doesn't exist in the system.	
2.2.	FAILED	Requires steps 1.1 and 1.3 to pass. Discount cannot be maintained if the material / customer doesn't exist in the system.	
2.3.	FAILED	Requires steps 1.1 and 1.4 to pass. Purchase price cannot be maintained if the material and vendor don't' exist in the system.	
3.Order-delivery process			
3.1.	FAILED	Requires steps 1.1 and 1.3 to pass. Sales order cannot be created if customer and material don't exist in the system.	
3.2.	FAILED	Requires steps 1.1 and 1.4 to pass. MRP run cannot be executed if the material does not exist and vendor for the material	

		needs to exist before purchase order can be created.	
3.3.	FAILED	Requires step 3.2 to pass. Material cannot be received because no purchase order was created.	
3.4.	FAILED	Requires steps 3.1, 3.2 and 3.3 to pass, otherwise sales order cannot be completed.	
3.5.	FAILED	Requires all the previous steps to be passed. The main product cannot be delivered to a customer because it does not exist.	
3.6.	FAILED	Requires step 3. 4 and 3.5 to pass. Billing document (sales invoice) cannot be created if no sales order exists and materials were never delivered.	
3.7.	FAILED	Requires step 3.6 to pass. Billing document cannot be released to the accounting because it was never created.	

II TESTING ROUND **Test case 4: SAP Order-delivery process exercise**				**Date: 23.08.2010**
Steps	**Result (passed/ FAILED)**	**Notes**	**Output/outcome**	**Other remarks**
1. Master data				
1.1	passed	Solution: use the material **TS10**that was created in the 2nd test case in ECC 6.0, exactly for same organization structure and with same information as the KTA98 in ECC 5.0, which is used as a copy material for new materials.	Material: **BITE05TS, Description BITE Tester**	Numbering of the whole exercise 4 steps should be revised so that it is easier to refer in case of questions: 1.1, 1.2 …3.7
1.2.	passed	-	The material BITE05TS has been marked for the MRP run	
1.3.	passed	-	Customer: **44599, BITE Tester**	- For Account group Sold-to party-0001 lower limit is 0000000001and upper limit is

				0000099999 so in the exercise the given students customer number pattern needs to be changed., e.g. 4445xx→445xx
1.4.	passed	Solution: use vendor 33399, Vendor TS created in the 2. exercise as the reference vendor when copying	Vendor: **33599**, **BITE Tester**	
1.5.	passed	-	Message “Source list changed”	
2 . Pricing				
2.1.	passed	-	Condition records saved	
2.2.	passed	-	Condition records saved	
2.3.	passed	-	Message: Purchasing info record 5300005716 1000 1200 created.	
3.Order-delivery process				
3.1.	passed	-	Standard Order **12261** has been saved. Printing:OK	
3.2.	passed		- MRP carried out for BITE05TS 1200 1200 - Standard PO created under the number **4500017176** Printing: OK	
3.3.	passed		Material document 5000012001 posted	
3.4.	passed		Delivery completed	
3.5.	passed		- Delivery 80015375 has been saved Printing: OK	
3.6.	FAILED		Invoice**90036454** has been saved. Printing: FAILED	
3.7.	passed		The document has already been passed on to accounting	

Appendix 5. Test result table for the test case 5

I TESTING ROUND **Test case 5: SAP IMG assignment**			**Date: 11.06.2010**
Steps	**Result (passed/ FAILED)**	**Notes**	**Other remarks**
1			- Page numbering - For consistency and learning purposes the path should be added, not just transaction number → Tools-Customizing – IMG - Execute Project – Click SAP Reference IMG
1.1	PASSED	No action needed. Just additional information for the exercise.	
1.2	FAILED	When trying to save a new sales organization a popup window “Prompt for Customizing request” appeared. There were no instructions in the exercise how to handle this situation.	- “’Go back to ‘Display IMG’ – view “ unnecessary → Delete the text. After the text ‘Choose button Next Entry (F8)’ it should be added: The system gives a following warning “WARNING: Changing the statistics currency causes data inconsistency”. Press Enter to continue. - Text “Address, country”: It’s unclear for the student if a new address information needs to be typed in or not → clarify. - Text save – Enter would be more consistent, because whenever there was a change to complete the action with ic✓ icon in prior exercises, the command enter was used. →Enter instead of save. - For making sure the student’s action was successful before continuing the following text could be added: “Check that your new sales organization is added to the sales organization list.”
1.3	FAILED	When trying to save a distribution channel a popup window “Prompt for Customizing request” appeared. There were no instructions in the exercise how to handle this situation.	

1.4	FAILED	When trying to save a sales office channel a popup window "Prompt for Customizing request" appeared. There were no instructions in the exercise how to handle this situation.	Instead of saying give name "Pasila" → give description "Pasila".
1.5	FAILED	When trying to save a sales group a popup window "Prompt for Customizing request" appeared. There were no instructions in the exercise how to handle this situation.	
2.	FAILED	Requires step 1.2 – 1.5 to pass. Assigning (defining) the sales organization structure cannot be done because no new sales organization / distribution channel/ sales office/ sales group could be created.	It would be convenient if Step 2. would be divided under substeps and numbered accordingly 2.1, 2.2 … helps orientating if a student wants to ask something about a specific step.
3.	FAILED	Requires steps 1.2 -2 to pass. No pricing procedures could be defined because the new sales organization could not be created in previous steps.	
4.		Requires steps 1.2 -2 to pass. G/L accounts for the new sales organization could not be assigned because the new sales organization could not be created in previous steps.	
5.	-		
5.1	FAILED	Requires steps 1.2 -4 to pass. An order-delivery process cannot be executed because the new sales organization could not be created in previous steps.	
5.2	FAILED	Requires steps 1.2 -5.1	

		to pass. An order-delivery process cannot be executed because the new sales organization could not be created in previous steps.	
5.3	FAILED	Requires steps 1.2 -5.2 to pass. An order-delivery process cannot be executed because the new sales organization could not be created in previous steps.	
5.4	FAILED	Requires steps 1.2 -5.3 to pass. An order-delivery process cannot be executed because the new sales organization could not be created in previous steps.	
5.5	FAILED	Requires steps 1.2 -5.1 and 5.4 to pass. An order-delivery process cannot be executed because the new sales organization could not be created in previous steps.	
5.6	FAILED	Requires steps 1.2 -5.1, 5.4, 5.5 to pass. An order-delivery process cannot be executed because the new sales organization could not be created in previous steps.	
5.7	FAILED	Requires steps 1.2 -5.6 to pass. An order-delivery process cannot be executed because the new sales organization could not be created in previous steps.	
5.8	FAILED	Requires steps 1.2 -5.7 to pass. An order-delivery process cannot be executed because the new sales	

		organization could not be created in previous steps.	
5.9	FAILED	Requires steps 1.2 -5.8 to pass. An order-delivery process cannot be executed because the new sales organization could not be created in previous steps.	

II TESTING ROUND **Test case 5: SAP IMG assignment**				**Date: 5.08.2010**
Steps	**Result (passed/ FAILED)**	**Notes**	**Output /outcome**	**Other remarks**
1				- Page numbering - For consistency and learning purposes the path should be added, not just transaction number → Tools-Customizing – IMG - Execute Project – Click SAP Reference IMG
1.1	PASSED already in I testing round	No action needed. Just additional information for the exercise.		
1.2	Passed	In the testing round I when trying to save a new sales organization a popup window “Prompt for Customizing request” appeared. Solution: **The student has to create his own customizing request that can be used later in this exercise also. Introduction has to be added.**	- A new customizing request created: **BI_test** - A new sales organization created: **BI31** “Helsinki”	- “’Go back to ‘Display IMG’ – view “ unnecessary → Delete the text . After the text ‘Choose button Next Entry (F8)’ it should be added: The system gives a following information “WARNING: Changing the statistics currency causes data inconsistency”. Press Enter to continue (see print screen 1 below. - Text “Address,

				country": It's unclear for the student if a new address information needs to be typed in or not → clarify e.g. enter address and country. - Text save – Enter would be more consistent, because whenever there was a change to complete the action with ic✓ icon in prior exercises, the command enter was used. →Enter instead of save. - For making sure the student's action was successful before continuing, the following text could be added: "Check that your new sales organization is displayed in the sales organization list."
1.3	passed	Solution: use the same customizing request than created in the step 1.2.	A new distribution channel created: **50, 'Fair sales'**	- The following advice should be added after the step Save: When the prompt for customizing request appears, use the request you created earlier in this exercise. - For making sure the student's action was successful before continuing, the following text could be added: "Check that your new distribution channel is displayed in the distribution channels list."
1.4	passed	Solution: use the same customizing request than created in the step 1.2.	A new sales office was created: **QQ0, 'Pasila'**	Instead of saying give name "Pasila" → give description "Pasila". -- The following advice should be added after the step Save: When the prompt for customizing request appears, use the re-

				quest you created earlier in this exercise. -- For making sure the student's action was successful before continuing, the following text could be added: "Check that your new sales office is displayed in sales office list."
1.5	passed	Solution: use the same customizing request than created in the step 1.2.		- The following advice should be added after the step Save: When the prompt for customizing request appears, use the request you created earlier in this exercise. - For making sure the student's action was successful before continuing, the following text could be added: "Check that your new sales group is displayed in the sales groups list."
2.	passed	**ASSIGNMENT OF ENTERPRISE STRUCTURE HAS BEEN CHANGED IN ERP 6.0** - **Assign sales organization to company code:** In the ECC 6.0 version the assigning function 'Assign sales organization to company code' has been changed -> updated print screens required. Also "Assign" (F2) is not valid any more ->When you select 'Assign sales organization to company code" a screen 'Change view "Assignment sales organization – company code" Overview' will open. Find your sales organization from the list and select company code		- It would be convenient if Step 2. would be divided under substeps and numbered accordingly 2.1, 2.2 … helps orientating if a student wants to ask something about a specific step.

		in the CoCd –field. Save. ALSO - The following advice should be added after Save: When the prompt for customizing request appears, use the request you created earlier in this exercise. - **Assign distribution channel to sales organization:** In the ECC 6.0 version the assigning function 'Assign distribution channel to sales organization' has been changed ->New instructions: Click **New entries** –button. Choose **your own** Sales organization, then choose **your own** distribution channel and **save**. When the prompt for customizing request appears, use the request you created earlier in this exercise. Check that your new enterprise structure is displayed in the list 'Change view: assignment sales organization – distribution channel'. **- Assign division to sales organization:** In the ECC 6.0 version the assigning function 'Assign distribution channel to sales organization' has been changed ->New instructions: Click **New entries** –button. Choose **your own** Sales organization, then choose division00 and **save**. When the prompt for customizing request appears, use the request you created earlier in this exercise. Check that your new enterprise structure is displayed in the list 'Change view: assignment sales		

		organization – division'. - **Set up sales area:** In the ECC 6.0 version the assigning function 'Set up sales area' has been changed ->New instructions: Click **New entries** –button. Choose **your own** Sales organization, then choose **your own** distribution channel and then choose division00 and **save**. When the prompt for customizing request appears, use the request you created earlier in this exercise. Check that your new enterprise structure is displayed in the list 'Change view: assignment sales org. –Distribution channel – Division'. - **Assign sales office to sales area:** In the ECC 6.0 version the assigning function 'Set up sales area' has been changed ->New instructions: Click **New entries** –button. Choose **your own** Sales organization, then choose **your ow**n distribution channel, then choose division00 and then choose **your own** sales office and **save**. When the prompt for customizing request appears, use the request you created earlier in this exercise. Check that your new enterprise structure is displayed in the list 'Change view: assignment sales office –Sales area'. - **Assign sales group to sales area:** In the ECC 6.0 version the assigning function 'Assign sales group to sales area' has been changed		

		->New instructions: Click **New entries** –button. Choose **your own** Sales office, then choose **your ow**n Sales group and **save**. When the prompt for customizing request appears, use the request you created earlier in this exercise. Check that your new enterprise structure is displayed in the list ‘Change view: assignment sales office –Sales groups’. - **Assign sales organization – distribution channel –plant:** In the ECC 6.0 version the assigning function ‘Assign sales organization – distribution channel –plant:’ has been changed ->New instructions: Click **New entries** –button. Choose **your own** Sales organization, choose **your own** distribution channel, then choose plant **1200** and **save**. When the prompt for customizing request appears, use the request you created earlier in this exercise. Check that your new enterprise structure is displayed in the list ‘Change view: Assignment Sales organization/Distribution Channel –Plant’. - **Define rules by sales area:** No changes in the ECC 6.0 in this function. Add though -> When the prompt for customizing request appears, use the request you created earlier in this exercise.		
3.	passed	When the prompt for customizing request appears, use the request you cre-		

		ated earlier in this exercise.		
4.	passed	When the prompt for customizing request appears, use the request you created earlier in this exercise.		Note! The account number should be 80000 not 800000.
5.	-			
5.1	passed		Existing material BITE10_TS and customer 44499 have been extended to the new sales area (Sales org. BI31, Dchl. 50, Division 00)	
5.2	passed		Sales price (100) maintained for the material BITE10_TS at the new sales area.	
5.3	passed	System was doing some updating by itself. See message screenshots.	Standard Order **12161** has been saved.	
5.4	passed		Purchase request was created for the material BITE10_TS, 10 pce. Message: MRP carried out for BITE10_TS 120 0 1200	
5.5	passed		Standard PO created under the number **4500017175**	
5.6	passed		Material document **5000011991** posted.	
5.7	passed		Delivery **80015275** has been saved.	
5.8	FAILED	Printing the invoice on paper failed.	(Billing) Document **90036354** has been saved. PRINT: FAILED	
5.9	passed		Document flow seems to be correct.	

Appendix 6. Test result table for the test case 6

I TESTING ROUND **Test case 6: Automating period closing**				**Date: 17.06.2010**
Steps	**Result (passed/ FAILED)**	**Notes**	**Output/outcome**	**Other remarks**
1. Opening current posting period for the company code				
1.1	passed	.	Status of opened booking periods for the company 1000 IDES: already opened till 12/2015, no need to create new periods	-
2 . Close past period				
2.1.	passed	-	Check current posting period: 5/2010, is one period in the past of the actual company code period	-
2.2.	passed	-	Current posting period altered: the period 5/2010 is closed and as result the 6/2010 is set as the current period	-
3. Automate scheduling				
3.1.	passed	-	Create program that takes care of the scheduling: UCCPERI created as variant of RMMMPERI program	-
3.2.	passed	-	Set attribute: First day of current month	-
3.3.	passed	-	Schedule program as background job: from 17.6.2010 (the testing date) till 16.6.9999	-