Office Of the CTO

Authors:
Gordon Mühl
Wolfram Kleis

Contributors:
Peter Conrad
Roger Gutbrod
Nicolai Jordt
Torsten Kamenz
Andreas Kemmler
Carsten Schmitz
Tobias Stein

Stable Core And Enhancement Packages

Based on work by ERP Architecture group

Version 1.5

September 2007

Table of Contents

Changes and updates:

Version	Date	Name	Change/Enhancement
1.01	08.06.2007	Wolfram Kleis	Minor changes, adjusted release names and wording (for example SAP ERP 2005 -> SAP ERP 6.0)
1.5	DKOM 07	Gordon Muehl, Roger Gutbrod, Andreas Kemmler, Torsten Kamenz, Carsten Schmitz	NetWeaver Enhancement Packages, Reuse Layer, Details for Switching in Non-Abap Environments

1 Motivation, Goals and Basic Rules

This document explains the motivation, goals and basic rules of the development and delivery model for the SAP Business Suite 2008. SAP Business Suite 2008 comprises the following components[1]: SAP CRM 2008, SAP ERP 6.0, SAP PLM 2008, SAP SCM 2008, SAP SRM 2008[2].

The new delivery model replaces the traditional practice of shipping release upgrades every 1 or 2 years. The new model is based on the concept of a **stable core** of the product that is extended by optional **enhancement packages**. SAP ERP is already working with this new model.

1.1 Motivation and Goals

The Challenge

In the past the high effort and costs of applying release upgrades often caused customers to stay with old releases. Customers who decided not to upgrade could not benefit from new functionality. If they decided to do the upgrade they could not be sure that the existing functions were not affected. Often they had to run a full regression test of all functions that were relevant for their business. In addition, time was wasted by upgrading installed components that were never used by the customer and by installing new components that the customer did not intend to use.
At the same time SAP has to offer new products or innovation within existing products to make their products attractive. Customers only buy new software if they expect reasonable innovation with added value for their business.

The Goal

The goal of the new delivery model is to make innovations accessible to customers more quickly while minimizing the risk of affecting existing functionality used by the customer. The customer should be able to benefit from new functionality without being forced to apply a release update. In addition, the customers should be able to select only those new functions that are required for their business.

That means that the offering and delivery should be two-fold. On the one hand the core functionality must be stable. On the other hand innovation within the product must be allowed.

The Solution

The practice of shipping new releases every 1-2 years is replaced by a new delivery model. SAP Business Suite 2008 will be shipped as the base release that will remain stable for the following years. New functionality will be shipped as optional enhancement packages that can be used by customers selectively if required for their business. The enhancement packages introduce new functions on top of the SAP Business Suite 2008 release with minimum impact on existing functionality.

The customers can rely on the rule that the core remains stable for the lifetime of the product. Customers who do not install and activate enhancement packages get support packages for the core that are guaranteed to contain no new functionality. Enhancement packages introduce only documented changes with a clearly defined functional benefit. Applying an enhancement package

[1] The names for all components except SAP ERP 6.0 are „working“ names that should be used as a placeholder for the actual name that will be used when the product is shipped.

[2] The exact list of components that belong to SAP Business Suite 2008 will be compiled in the suite excellence project.

must be easy and fast compared to traditional upgrades. The following sections explain what “stable” means by listing the restrictions for developing enhancement packages.

For ERP the stable core is SAP ERP 6.0 release. For the other components, the stable core is the release that is shipped as SAP Business Suite 2008. The business functions available with SAP ERP 6.0 are kept stable with SAP Business Suite 2008. The other products of the business suite may introduce disruptive changes until the first delivery of SAP Business Suite 2008. Architectural changes are critical in enhancement packages. Therefore planned architecture changes, like relocations between software components, should be done until the release of the stable core.

1.2 Basic Rules For Enhancement Packages

1.2.1 Enhancement Package Technology

Technically an enhancement package is a set of software component versions that are delivered together and which can be optionally installed. For applying an enhancement package the existing release upgrade technology is used.

1.2.2 Basic Rules For Developing Enhancement Packages

To achieve the goals described in 1.1 the following rules must be obeyed:

1.2.2.1 Single Enhancement Package for Whole Suite

For the whole Business Suite 2008 one enhancement package will be shipped at one time (even if it technically consists of different parts). However, SAP cannot expect that customers apply enhancement packages to all application systems simultaneously. It shall be possible to run a customer landscape with business suite applications that have different enhancement package versions installed. This implies that interoperability between the applications of the suite must not be affected by installing enhancement packages (see also 5.3).

1.2.2.2 Enhancement Packages Must Be Strictly Optional

Development that is done for an enhancement package version must not influence the customers who are not installing this enhancement package. This implies that it is not allowed to change core objects in a way that introduces dependencies to enhancement packages. It also implies that enhancements must not be developed in support packages. Support packages contain fixes of the core and they must be applicable for all customers and must not have dependencies on enhancement packages. Patches for enhancement packages are shipped with separate enhancement support packages.

1.2.2.3 Enhancements Must Be Switchable

Customers who install enhancement packages shall not be affected by enhancement package content they don’t want to activate. A customer shall not be forced to do any regression tests or end-user trainings if no new functionality is activated. Therefore, all changes in the enhancement packages have to be switchable. This can be done by customizing or – in the ABAP case – based on the switch framework technology (other development environments must use or develop appropriate technologies). Only after actively switching on the new functionality the changes shall be noticeable.

In particular, with all enhancements switched off, there must be no noticeable changes of

- user interface (This must be obeyed strictly. It is expected that even small changes in the menus or a single additional screen would lead to customer effort for user training.)
- business processes, functionality of transactions
- customizing
- system performance

- system stability
- resource consumption
- running deployment scenarios

Guidelines for using the ABAP switch framework in enhancement packages are presented in section 2.1.1.

1.2.2.4 Compatibility

Changes that are made as part of any enhancement package version must be strictly compatible with the base release of Business Suite 2008. This is a prerequisite for making software component versions of an enhancement package optionally installable. If software component SWC1 provides elements that are used by other software components, it is not allowed to change these elements in an incompatible way. If compatibility is achieved, SWC1 can be upgraded to an enhanced version without forcing the customer to upgrade the dependent software components as well. Incompatibilities are regarded as errors that must be fixed by the component that made the incompatible change. Exceptions may be defined by the different areas[3]. The explicit definition of reuse contracts (for example based on ABAP package concept, Java development component model) is a recommended preparation for managing the compatibility.

1.2.2.5 Cumulative Enhancement Packages

All enhancement packages must be installable directly on the Business Suite 2008 base release (without previous enhancement packages installed). New versions of an enhancement package contain the whole functionality instead of only the delta compared to the previous version.

1.2.2.6 Minimize Customer Effort For Applying Enhancement Packages

Customers shall experience minimal downtime, minimal upgrade processing time and minimal modification adjustment effort when applying an enhancement package.

- Application specific upgrade or migration programs (XPRA, after import methods or similar mechanisms in non-ABAP technology) are not allowed in enhancement packages.
- Software components don't have to deliver a new version with each enhancement package version. If there are no enhancements in that software component, there is no need to include it into the enhancement package.
- Customer test activities shall be supported by delivering external test case templates[4], lists of changed transactions with detailed description and a bill of material with all needed components.

1.2.2.7 NetWeaver Dependencies

The SAP Business Suite 2008 and all enhancement packages must run on SAP NetWeaver 7.0. Content shall be developed using SAP NetWeaver 7.0 environments only. This applies to all content, regardless of the technology and development environment that is used (for example to ABAP, Java, BI, XI or portal content). Unless downported, SAP NetWeaver 7.1 features are generally not available for SAP Business Suite 2008.

New enhancement package versions may require a new NetWeaver support package stack. Support packages of enhancement packages shall not require a new NetWeaver support package stack.

[3] ERP has defined industry extensions that are heavily coupled via enhancement options as one exception. See the „ERP 2005 Enhancement Package" guide for details.

[4] See ERP Guidelines [1]

1.2.2.8 Train The Development Teams

The SAP development organizations have to ensure that the development teams know the technologies needed for enhancement packages. Training should cover the switch framework and how to use the enhancement framework for stabilizing mature parts.

1.2.2.9 Quality Assurance

Additional test cases have to be provided and executed to prove that new enhancement packages comply with the above rules. Test cases for switches have to ensure both that the deactivated functions do not affect the existing system and that the new functionality works correctly when the switches are activated. Compatibility of enhancement package versions of software components with the stable core version of other software components must be tested as well.

2 Technology Specific Rules

The technology specific rules for ERP enhancement packages in [1] apply to the whole business suite (exceptions and deviations have to be described in the application specific guidelines).

Enhancement package specific rules for object models are covered in the architecture guidelines of the different areas.

2.1 ABAP Development

General ABAP specific rules for enhancement packages are described in the ERP guidelines in [1].

2.1.1 Rules And Guidelines For Switches

The use of switches is fundamental for the enhancement package concept. Therefore we summarize in this section the most important rules for using the switch framework in enhancement package development. See [1] for further details.

2.1.1.1 Reasons for Switches

The main reasons to use switches in the enhancement packages are to keep the UI and the business processes stable, to lower regression test effort and to have the possibility to activate only the functions which the customer wants.

2.1.1.2 Types of Switches

There are different kinds of switches to be considered:

- Switch framework switches which are turned on via business functions
- Customizing switches
- Master data settings

For the purpose of activating functionality the switch framework shall be used wherever possible. Customizing switches and master data settings are always only second choice. For switching coding using the switch framework BAdIs shall be preferred to enhancement points or sections.

The switch framework shall not be used to configure business logic. It should only be used to activate new processes and functions. The installation of an enhancement package and the activation of switches must not affect the correct execution of business processes that were already started before the switch was set. In some situations the activation of the switch will change the behavior of an already started business process, but this must happen in a predictable and well documented way. A process that was initiated before the switch was activated, must continue correctly after the activation of the switch.

2.1.1.3 Business Functions

The creation of business functions is mandatory for all enhancement package developments. This is true even for completely new development that is not switchable. The business function serves as a link to the documentation and test cases relevant for the functionality and represents the first entry point for the customer to get information about the new functionality.

The switches that are defined with the ABAP switch framework are assigned to business functions. In that case, business functions are used as collections of switches that are switched together. The switches cannot be turned on directly but only via business functions. For the customer the business function represents a self-contained functionality that can be switched on separately.

2.1.1.4 Granularity Of Business Functions

Business functions shall be defined on a level of bigger functional blocks. Usually it will be something in between software components and single functionality (the industry extensions for example have two to six business functions defined).
The overall number of business functions should be kept small. This helps to keep the number of business functions manageable for the customer and to limit the number of required test clients and the testing effort.

However, the goal to keep the number of business functions small should not lead to a design that bundles unrelated switches into one big business function. The granularity of business functions should be driven by the semantics: A business function shall reflect a logically coherent and self-contained functionality which is meaningful for the customer.

The granularity of switches for reusable parts should reflect the granularity of reusable assets. If a software component exposes two switchable objects that can be reused separately, it should be possible to switch these two objects separately.

New business functions shall be used for switches introduced for an enhancement package. Do not assign new switches to existing business functions because this way it would be impossible to document and to switch the enhancement separately.

2.1.1.5 UI Switches vs Coding Switches

A coding switch is a switch that is assigned to enhancement framework enhancements or BAdI implementations. A UI switch is a switch that activates UI objects or new UI functionality like dynpro fields, menus (CUADs), IMG, easy access menu, switch BC sets, domain fixed values, search help, maintenance view fields.

It shall be possible to assign coding and UI enhancements to business functions separately. This implies that coding and UI enhancements shall not be assigned to the same switch. Normally the switches for coding and UI changes are assigned to the same business function. But in some cases it is required to assign the switches to additional business functions, for example if reused by another software component. To support these cases it shall be possible to assign UI and coding switches to business functions independently.

The number of coding switches should be kept small. Otherwise the code would become more and more complex and hard to understand and maintain. The effort for testing and for adopting in industry extensions would increase heavily. More precise rules and guidelines for using code switches have to be defined until the development of the first enhancement packs for Business Suite 2008 starts. This will be done based on future experience with the ERP enhancement packs.

2.1.1.6 Dependencies Between Business Functions

Business functions should be self-contained units that can be activated separately. Therefore dependencies to other business functions shall be avoided as a general rule. There is one exception: Consider a new version of a enhancement package that adds functionality to an enhancement shipped in a previous enhancement package. In this case a new business function is introduced that has a “required” dependency to the existing business function. This is allowed only if both business functions belong to the same software component. If dependencies between business functions are introduced, they must be declared correctly.

2.1.1.7 Technical Types Of Business Functions

Business functions that are not assigned to business function sets shall be of type “Enterprise Business Function”. New business functions in industry business function sets shall be of type “Industry Business Function”. The business function type ”Add On” shall not be used any further.

2.1.1.8 Switches and Package Design

The switching of development objects can be done in two ways: some objects have the switch assigned directly like UI elements and IMG paths. Other objects are switched when they are located in a package

which is assigned to a switch. This section gives some guidelines for the second alternative, i.e. for designing packages with switches.

The definition of business functions, switches, and packages is part of your overall architecture and has to be planned and designed in advance.

New packages shall be created that include the switchable objects. This makes sure that nothing existing is switched by accident.

Only some DDIC objects like domain values can be switched. The switchable DDIC objects shall be located in a separate package that has a separate switch which is marked as “DDIC-relevant”.

If a package is assigned to a switch one may easily get the impression that the whole package can be switched. To avoid misunderstandings such a package should not contain development objects of a type that cannot be switched. See the SAP help portal for a list of switchable objects [7].

2.2 Non-ABAP and Content

2.2.1 Java Development

For the development of enhancements in Java applications (for example Web Dynpro Java) separate Java development landscapes are required. Corrections to the stable core version are made in the standard support package landscape. Bug fixes are transported from the support package landscape to the enhancement package landscape and have to be merged manually with the changes made there.

The backend interfaces used from Java applications have to stay downward compatible so that Java applications of the stable core are running even if the backend component is upgraded to an enhancement package version. Therefore the corresponding backend interfaces (RFC interfaces, web services) should be locked for changes and only be opened via an approval process.

If the customer wants to use new Java functionality provided by an enhancement package it may not be sufficient to install the corresponding Java software component versions from the enhancement package. It may be required to install corresponding enhancement package versions of portal content, to install enhancement package versions of backend components and to activate the relevant switches in the backend. If such dependencies exist they need to be documented.

The rule that the installation of an enhancement package must not disrupt the existing system applies to Java development as well. To enable the selective installation of Java software component versions from enhancement packages, the enhancement package versions of Java software components must be binary compatible to the version in the base release of the stable core. Binary compatible means that the classes and interfaces exposed by the software component must not change in a way that requires recompiling or even changing the using components. Installed versions of using software components must still work after installing a new version of the used software component.

Currently there is no switch concept for Java that is similar to the ABAP switch framework. It is still an open question how this will be addressed (see 5).

2.2.2 Portal Content

New versions of the business packages will be delivered with the enhancement packages.

Compatibility of the portal content and the ABAP backend has to be ensured so that portal content of the stable core version (SAP Business Suite 2008/SAP ERP 6.0) still runs when an enhancement package version of the corresponding backend component is installed and activated.

Backend applications which are incorporated into a portal role, like Web Dynpro ABAP applications, have to be downward compatible. Only additional, optional application parameters are allowed. If the enhanced version of the ABAP Web Dynpro contains portal navigation which requires newer portal content than the

stable core version, the navigation must be put behind a switch. In addition the portal context version must be evaluated at runtime before the new navigation is executed.

The backend application has to be switched to ensure that the portal role shipped with the stable core runs as before if the switch is not activated. When the customer decides to switch on the enhancements in the back end the corresponding enhancement package version of the portal role must be installed as well.

The documentation of the backend switch for the application as well as the documentation of the new portal content version have to describe the dependency between installing and activating the backend application and installing the portal content.

2.2.3 BI Content

There is one BI content development system landscape from which BI content feature packs are delivered. The BI content feature packs contain regular content updates together with BI content for the enhancement packages. That means that a SAP Business Suite 2008/SAP ERP 6.0 customer can see BI content from enhancement packages which possibly has no corresponding content in the backend. An enhancement package customer may see BI content from newer enhancement packages which possibly has no corresponding content in the backend.

To avoid activation problems or data load conflicts for the customer the following rules must be obeyed

- A naming convention for BI info areas ensures that BI content for enhancement packages can be identified in the BI content system. The names for BI info areas are derived from the name of the corresponding business function and the enhancement package version.

- Changes must be done carefully and in a compatible way. If in doubt, copy the object and change the copy. Exception for this guideline: InfoObjects semantically represent characteristics and key figures. Therefore it is in general not desired to copy InfoObjects, if the semantics have not changed. It is allowed to add new attributes to existing InfoObjects.

- It is not allowed to migrate objects from BI 3.x technology to SAP NetWeaver 7.0 technology. This ensures that the BI 3.x objects stay in the maintenance cycle.

2.2.4 XI Content/Service Enabling

XI content is delivered with XI content software components that are the XI counterparts of corresponding software components in the backend. Each XI content software component relates to exactly one backend software component. This pattern is not enhancement package specific, but is shall be applied to enhancement packages as well: For each backend software component version that is created for a new enhancement package version a corresponding XI content software component version has to be created.

Example: For the backend software component “EA-HR” the backend software component versions “EA-HR 602”, “EA-HR 603” and “EA-HR 604” are shipped as parts of enhancement packages. Corresponding XI content is shipped in the same enhancement packages with the XI content software component versions “XI Content EA-HR 602”, “XI Content EA-HR 603” and “XI Content EA-HR 604”.

For the creation of content in enhancement packages the standard eSOA rules and eSOA methodologies have to be applied [6].

3 Enhancement Packages for SAP NetWeaver (Preliminary)

3.1 Technology Innovation Strategy

In the future, delivery of corrections and delivery of technology innovation will be clearly separated. Corrections will be delivered through support packages whereas technology innovation will be delivered through new SAP NetWeaver releases or through enhancement packages from SAP NetWeaver – which are similar in spirit to enhancement packages introduced with SAP ERP 6.0. Enhancement packages for SAP NetWeaver are discussed in more details in Section 3.2.

Due to the fact that the core applications within the SAP Business Suite 7.0 are running on SAP NetWeaver Application Server 7.0 at least until 2012, this results in the following channels for delivery of technology innovation for large enterprise customers in the near future:

- ***Through feature packages for SAP NetWeaver 7.0.*** *SP14 (F) will be the last feature package. Its code line will be maintained through support packages providing only corrections. The next support package for this code line will be named SP15 and is supposed to be released in March or April 2008. The formerly planned feature packages SP15 (F) and SP16 (F) will be replace by Enhancement Package 1 (see next bullet).*
- ***Through enhancement packages for SAP NetWeaver 7.0.*** *SP14 (F) will serve as the baseline for Enhancement Package 1 providing features required by SAP Business Suite 7.0 (including SAP ERP 6.0 EhP4).*
- ***Through SAP NetWeaver 7.1*** *for composition and as integration platform. These capabilities will be delivered, respectively, through the SAP NetWeaver Composition Environment (see Chapter* ***Error! Reference source not found.****) on the one hand, and in form of SAP NetWeaver 7.1 "hubs" on the other hand (see Sections* ***Error! Reference source not found.*** *and* ***Error! Reference source not found.*** *below).*

3.2 Enhancement Packages for SAP NetWeaver (Preliminary)

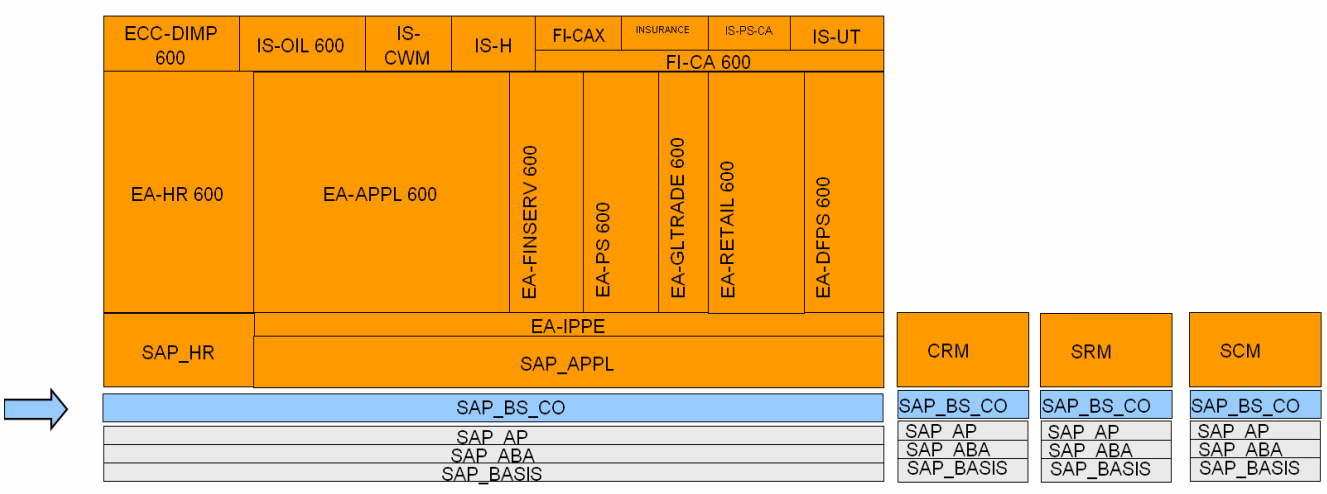
Enhancement Packages for SAP NetWeaver are the same approach as for the application will be followed. Only one aspect is different: NetWeaver Enhancement Packages will not use the ABAP switch framework technology to activate new functionality. On a more detailed level, NW EhPs content cannot be selectively switched on, or you could say there is only 1 switch in NW EhPs which is always on. Otherwise the EhP rules apply: there is a separate Support package line per EhP.

SAP will minimize the number of EhPs for NW, per main release there will not be more than 3. Every EhP needs explicit approval by the PTC to keep the number under control. So far only the first EhP for NW 7.0 has been approved.

- *Each EhP (Application) runs on a defined NetWeaver code line (NetWeaver release or EhP NetWeaver)*
- *Maintenance of EhP (NetWeaver) follows maintenance of the underlying release (5-1-2 maintenance strategy)*
- *Initial release and EhP (NetWeaver) are separately maintained through support packages*
- *Each additional NW Code line needs again PTC approval (innovation after BS 2008)*
- *Further BS2008 applications are expected to rely on EhP1 NetWeaver*

4 Reuse Layer For The Business Suite

For the Business Suite 2008 there are primarily two reuse layers available. The software component SAP_ABA and new the software component SAP_BS_CO.



Reuse rule 1: SAP_ABA shall be only used for objects which already exist there. For all new reuse objects SAP_BS_CO is the appropriate layer.

This new layer SAP_BS_CO will be synchronized with the suite releases. Hence, reuse development will be possible till short before development close of the suite applications. Also this reuse layer will obey the stable core rules described in this document. That means new function in reusable objects must use switches in the same way as the other application components.

Reuse rule 2: No moves from other software components to SAP_BS_CO are allowed. Only new reuse objects or a copy to a new namespace are allowed.

For the sake of stability of our customers' landscapes no relocations between software components are allowed.

A governance process for all reusable objects in this new reuse layer and a dedicated reuse team responsible for the governance process will be set up soon. The team and governance process have to ensure that all objects in SAP_BS_CO obey both the stable core rules and reuse guides.

5 Open Issues

5.1 Rules and Guidelines For Coding Switches

As mentioned in section 2.1.1.5 detailed rules and guidelines have to be defined for using coding switches. Without clear rules and governance the uncontrolled use of coding switches may lead to code that is hard to understand, maintain and test. The semantics of a method or a function module may be hard to define if the implementation is switchable by multiple independent switches. The rules should define which type of coding switched are allowed and on which granularity (for example by allowing only one switch per method or function module etc).

5.2 Switches And Non-ABAP Technologies

5.2.1 Switches In Non-ABAP Technologies

It has to be investigated how a switch concept similar to the ABAP switch framework can be implemented for non-ABAP technologies like Java. This includes a switch concept for model driven and generator based technologies like for example Web Dynpro Java. This investigation will be done in the Suite Excellence project and shall provide precise rules and guidelines until the development of the first enhancement packs for Business Suite 2008 starts. This will be done based on future experience with the ERP enhancement packs.

5.2.2 Managing Switches Across Technologies

Assuming that the problem in 5.2.1 has been solved it still has to be answered how functionality shall be switched that is implemented across multiple technology stacks. How are business functions technically defined in such a case, how are switches assigned and how are the business functions activated? This investigation will be done in the Suite Excellence project and shall provide precise rules and guidelines until the development of the first enhancement packs for Business Suite 2008 starts. This will be done based on future experience with the ERP enhancement packs.

5.3 Interoperability

5.3.1 Interoperability Between Different Enhancement Package Versions

It has to be defined how interoperability can be ensured between applications of the business suite that run on different enhancement package versions. In 1.2.2 we stated that it shall be allowed to run a landscape with business suite applications based on different enhancement package versions. For example: ERP could be on enhancement package n, CRM on enhancement package n+1 and SRM could have no enhancement package installed at all. While different suite applications can have installed different enhancement package versions only business functions can be activated for which the required enhancement package versions for ALL suite applications are met. This has to be managed centrally for the whole suite landscape.

An interface is public when it can be consumed from outside its software component (e.g. public parts of an ABAP package interface, XI messages, BAPIs, public RFCs,.; if those public interface can be consumed by SAP customers or partners (e.g. B2B messages, BAPIs, external BAdIs,..) they can only be changed in a compatible way.

Every public interface of a software component that is changed in an enhancement package must fulfill one of the following criteria:

- The change is compatible to the previous interface (e.g. optional parameter added to a function module; an optional attribute is added to an XI interface). Compatibility means that consumers of the interface are not required to do any changes.
- The change is incompatible but all consumers of the interface can be adjusted accordingly by the same switchable business function (e.g. change of an internal BAdI and simultaneous change of all BAdI implementations; simultaneous change of an A2A eSOA message, the provider and all its consumers

5.3.2 Interoperability Between Add-ons and Enhancement Packages

For every SAP delivered Add-Ons we have to ensure, that a version of the Add-On exists for every Enhancement Package and released Business Function setting.

Therefore all application based add-ons have to be analyzed whether the add-on is affected by EhPs. There it is important whether the add-on is independent or dependent.

An add-on is independent:

- Add-on is non-modifying and
- Add-on only uses BAPIs and released function modules and implements BADIs.

An add on is dependent:

- Add-on is modifying or
- Add-on uses other interfaces as BAPIs, BADIs or released function modules

As consequence of the analysis existing independent add-ons can be released for new EhPs. Dependend add-ons have to be adapted for each EhP. That means a new add-on release for each EhP have to be built. Exceptional for dependent add-on a limitation of availability can be decided (not available for EhP x).

It is currently in discussion, how we document the dependency . Also the rules above are currently subject of discussion with the legal department.

5.3.3 Retrofit of Add-ons to Enhancement Packages

Overall a retrofit should be targeted to be included in the Go-To Release. Independent Add-Ons can further coexist and be included in the EhP delivery stack, if they want to apply to the same release cycles and suite restrictions.

Dependent Add-Ons cannot be retrofit by default, exceptions need to be defined.

5.3.4 Interoperability Between Systems With Switched Functionality

How can interoperability be ensured between business suite applications if single functions can be activated or deactivated in the different systems? Are enhancements that affect interoperability managed with switches like it is done for UI enhancements? Are global switches (cross system) required? How are cross-system dependencies between switches described and managed?

5.3.5 Compatibility and Interoperability between Enhancement Packages for Applications and Enhancement Packages for SAP Netweaver

The definition and rules for this compatibility and interoperability is currently under investigation. They will be documented and release in December 2007 in this document, after approval in the PTC meeting in November 2007.

5.4 Governance Process And Exception Handling

A process is required for ensuring that the rules for enhancement package development are obeyed. It is also required to define how exceptions are requested, approved and documented.

6 Further Reading

[1] ERP 2005 Enhancement Packages, Downport/Development Rules
https://portal.wdf.sap.corp/irj/go/km/docs/corporate_portal/WS%20PTU%20Applications/Projects_1_/ERP%20Program/ERP_Enhancement_Packages/Development%20Documents/EHP%20Development%20Guidelines.doc

[2] ERP Enhancement Package pages in SAP Corporate Portal,
https://portal.wdf.sap.corp/wcm/Infocenters/WS%20PTU%20Applications/Applications/Projects/ERP%20Enhancement%20Packages

[3] ERP EhP Development Guidelines Overview Slide Deck
https://portal.wdf.sap.corp/irj/go/km/docs/corporate_portal/WS%20PTU%20Applications/Projects_1_/ERP%20Program/ERP_Enhancement_Packages/Development%20Documents/EhP_Development_Guideline_Overview.ppt

[4] EHP Rules for Table Entries and Switch BC Sets (Developer Version)
https://portal.wdf.sap.corp/irj/go/km/docs/corporate_portal/WS%20PTU%20Applications/Projects_1_/ERP%20Program/ERP_Enhancement_Packages/Development%20Documents/EHP_Table_Entry_Switch_BC_Sets_developer_version.ppt

[5] Rules for downward compatible development
https://portal.wdf.sap.corp/irj/go/km/docs/room_project/cm_stores/documents/workspaces/815b2262-1516-2910-e9a1-f6811174dfc0/EHP%20rollout/Guidelines_Rules%20for%20Ehp/Rules%20for%20downward%20compatible%20development.doc

[6] Wiki@SAP enterprise SOA Home
https://wiki.pal.sap.corp:8443/display/PTGSOA/enterprise+SOA+Home

[7] Switchable Objects, SAP NetWeaver 7.0 documentation in the SAP help portal
http://help.sap.com/saphelp_nw70/helpdata/en/e6/c487410643010de10000000a1550b0/frameset.htm

