



A Short Guide: Quality in SAP Change Management

The 5-system landscape as a key factor to success



Too often, the measure of Basis administration is simply how fast necessary changes can be imported into the production systems of the enterprise-wide SAP landscapes. This may be true from a business perspective, but who asks about the consequences that may arise from the poor quality of the individual SAP change processes?

Fact: In the face of the growing complexity of SAP environments, SAP Basis administrators today are mainly dealing with one question: How do I need to organize my SAP change management to ensure its quality and avoid system failures?

One way to create a solid quality foundation for the entire SAP change management process is to implement a 5-system landscape that allows for the parallel development of ad-hoc changes and scheduled releases.

This white paper provides a short guide on how companies can establish and successfully run such a landscape.



1. Quality trumps quantity: ITIL-compliant change management

As opposed to the well-known saying "never change a running system" many SAP organizations operate under the principle "always change a running system". When it comes to their SAP change and transport management, SAP Basis administrators are judged by how quickly new developments and enhancements can be transported into the production systems. But is performance really all that matters?

We believe that it is not. Given the central role of enterprise IT systems in value creation, it is particularly important to focus on the quality of new releases and to ensure that all IT systems are available and error free. While the issue of "high availability" has been appropriately addressed early on at the hardware level, the risk of system failures due to constant changes to the software has been considered acceptable for a long time. Some companies even take greater care in handling their training systems than they do their production systems.

Now that ITIL is becoming the de facto standard for change management processes, change managers need to start thinking outside their box!

Changes are no longer implemented and moved to production on-demand. They are now governed by a predefined process that, among other things, involves a precise change request, a detailed specification of the change, approvals and scheduling combined with the assignment of the change to a specific release date. Apart from urgent emergency changes, all changes are generally scheduled for go-live on a few select dates.

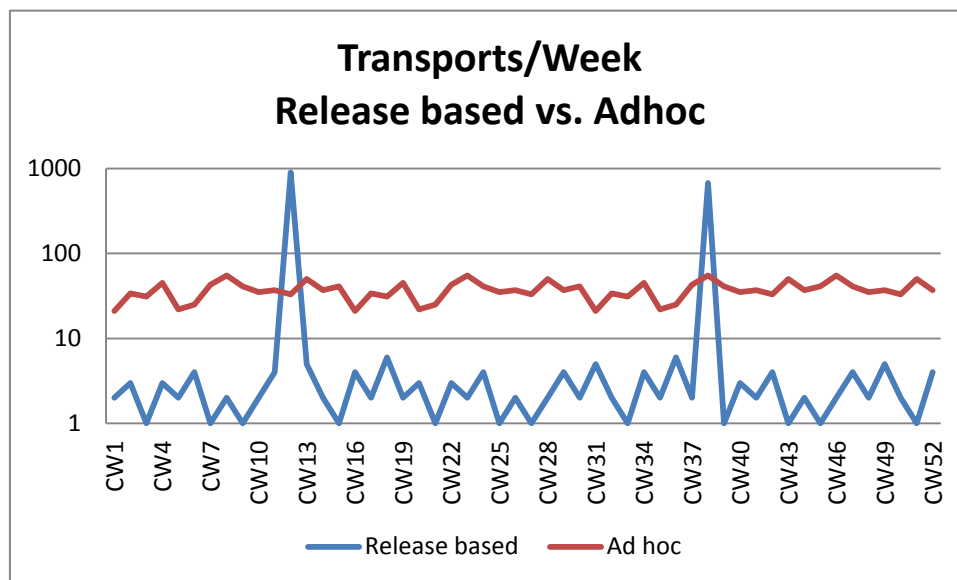


Figure 1: Transport profile of released-based development vs. ad-hoc development



2. Creating a solid foundation: moving from a 3- to a 5-system landscape

The benefits of ITIL-compliant change management are obvious:

- Improved stability of production systems
- Testing can be planned ahead of time
- Testing is performed on a defined number of object versions
- Resources for follow-up work and adjustments can be accurately planned and made available

But experience from many projects shows that conventional SAP 3-system landscapes simply do not measure up to the requirements that are associated with this type of high-quality release management.

Any company that uses the same development system for its emergency changes and its release development will eventually run into conflicts when dealing with emergency changes and their speedy delivery.

Example: Developer A has to make changes to the 'ZFI_ANALYSE_ACCOUNTS' report to fix a critical error. The developer is able to quickly identify and resolve the problem. Just before releasing the report, however, the developer realizes that significant enhancements have been made to the report for a scheduled release six months from now. And even though the business department is eagerly waiting for the changes to be implemented, the developer is unsure as to how to proceed.

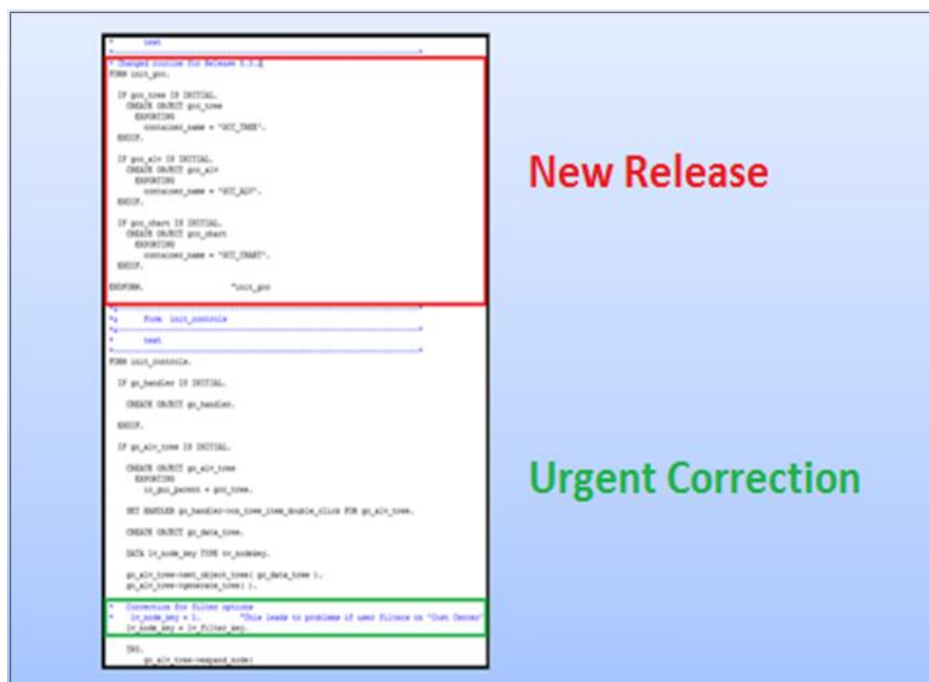


Figure 2: Conflict between an emergency change and the planned release schedule in a 3-system landscape



The developer is basically left with two options: To take back the changes for the new release or to analyze the new source code to find out whether it would even be active in the production system.

If the developer decides on option two, he or she will be faced with a number of risks that need to be taken into consideration:

During a source code analysis, it is easy to overlook dependencies with other (new or also modified) objects. Sub-transports that are essential to the new feature run the risk of not being imported into the production system, which in turn, depending on the criticality of the missing objects, could cause massive system failures.

Conversely, there is also the risk of a premature go-live of individual sub-transports that, according to the release schedule, were to be implemented at a later date. This will eventually "water down" the release level of the production system and, over time, widen the gap between the system release levels of the quality assurance system and the production system. The result: Emergency transports that were successfully imported into the quality assurance system could potentially cause serious errors in the production system that jeopardize the availability of business applications.

There is a solution to all of these problems: the **SAP 5-system landscape**.

The introduction of an additional development system and integration testing system allows IT organizations to completely decouple the development of new releases from the maintenance of existing systems.

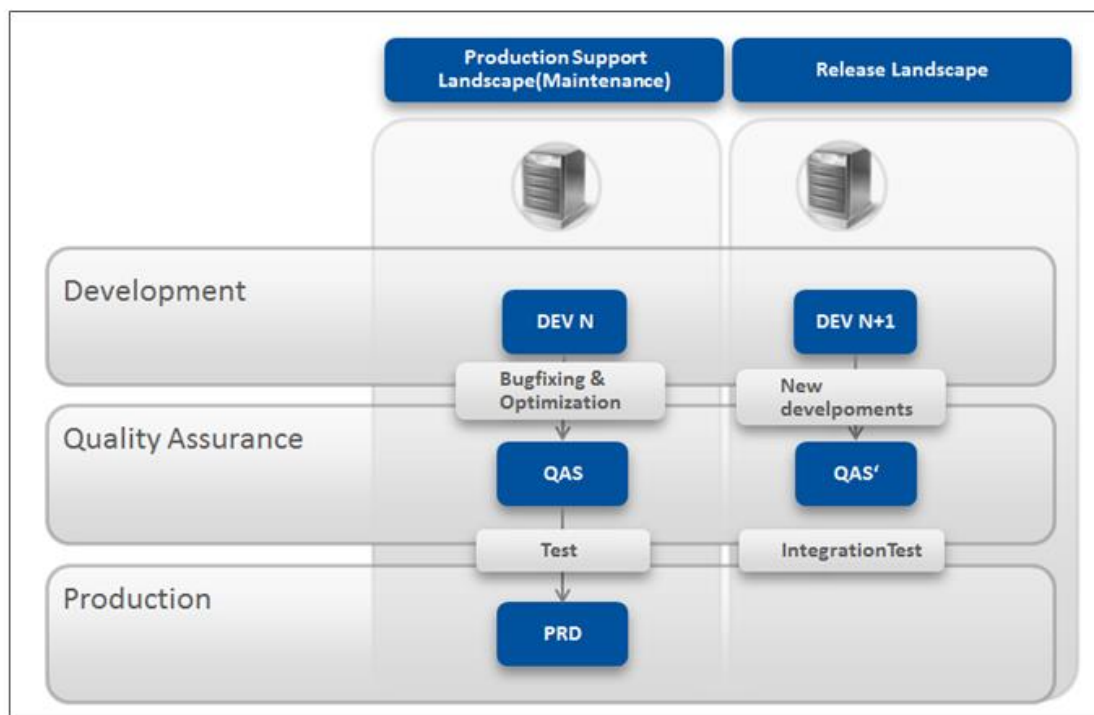


Figure 3: The 5-system landscape



Advantages:

- The strict separation between emergency changes and the development of new releases increases the elimination of errors
- The release levels of the QA system and the production system are almost identical (error corrections are usually characterized by short "processing times").
- The production system remains entirely unaffected by any developments for new releases until the go-live.
- If a higher EHP level is intended for the new release, the two additional systems can be upgraded to the new level while the remaining three systems are still kept at the old level.



3. Overcoming new challenges: automated system synchronization

All of these advantages enable SAP Basis administrators to dramatically improve the quality of their SAP change management process. But in order to make the most of the advantages of a 5-system landscape, they have to address the new challenges that go along with it.

All changes made in the development system of the maintenance landscape must be synchronized in the second development system. This requires significant manual effort and has SAP customers shying away from implementing a 5-system landscape despite its obvious benefits.

But there is another way: Modern change management solutions like **REALTECH's theGuard! SmartChange** allow SAP administrators to automatically synchronize their systems. The "Synchronization Management" module in theGuard! SmartChange takes the hassle out of managing developments. Developers will know at all times what synchronization tasks remain to be completed. And the release coordinator will never again lose track of the overall synchronization status.



Figure 4: Overview of synchronization progress with theGuard! SmartChange



The automatic object analysis shows which objects and/or customizing settings are causing conflicts immediately after a request has been exported from the development system of the maintenance landscape. Objects that are not causing any conflicts can be directly moved to the release development system using a transport. This is done either by using the original transport request or by creating a new request specifically for the synchronization.

Objects	Ver. N	Obl. T	Obj. Type	Status	Hot	Mod/Creat	Time	Owner	Request	Adj	Stat	Resp. User
IVFORGE/COOL_GRP			Function Group	EQUAL		14.05.2013	12:55:17	LINGSCHIED	C01K901581		Z	
ZCHECK			Function Group	EQUAL		03.06.2013	15:37:27	FRITTSCH	C01K901688		Z	
ZDEHOC			Function Group	EQUAL		26.04.2013	08:28:19	FRITTSCH	C01K901554		Z	
ZDEHOD			Function Group	EQUAL		26.04.2013	10:38:36	FRITTSCH	C01K901565		Z	
ZDEMO_BULL01			Function Group	EQUAL		27.05.2013	13:38:14	HALD	C01K901638		Z	
ZDEMO_PROX_001			Function Group	EQUAL		03.04.2013	10:07:12	FRITTSCH	C01K901451		Z	
ZDEMO_TARGET_BULLSEYE01			Function Group	NOT ON DEV SYSTEM		23.06.2013	22:15:21	KUBSCA	C01K901618		Z	
ZFIN_1			Function Group	EQUAL		04.06.2013	10:09:47	FRITTSCH	C01K901696		Z	
ZLAT1			Function Group	MODIFIED		05.02.2013	09:07:22	FRITTSCH	C01K901247		Z	
Z_LAT1	2	FUNC	Function Module	MODIFIED MANUALLY		13.02.2013	14:40:53	FRITTSCH	C01K901352		Z	
Z_LAT2	31	FUNC	Function Module	EQUAL		24.04.2013	16:32:42	FRITTSCH	C01K901535		Z	
LZLAT1TOP	1	REPS	Report Source ...	EQUAL		05.02.2013	09:08:35	FRITTSCH	C01K901247		Z	
LZLAT1U00X	1	REPS	Report Source ...	EQUAL		05.02.2013	09:08:35	FRITTSCH	C01K901247		Z	
SAPLZLAT1	1	REPS	Report Texts	EQUAL		05.02.2013	09:08:35	FRITTSCH	C01K901247		Z	
LZLAT1TOP	1	REPT	Report Texts	EQUAL		05.02.2013	09:08:35	FRITTSCH	C01K901247		Z	
LZLAT1U00X	1	REPT	Report Texts	EQUAL		05.02.2013	09:08:35	FRITTSCH	C01K901247		Z	
SAPLZLAT1	1	REPT	Report Texts	EQUAL		05.02.2013	09:08:35	FRITTSCH	C01K901247		Z	
ZSC_CBERT2013			Function Group	EQUAL		27.06.2013	14:48:21	FRITTSCH	C01K901803		Z	
ZTRAIN1			Function Group	MODIFIED		04.03.2013	11:06:22	FRITTSCH	C01K901368		Z	
Z_TRAIN_1	5	FUNC	Function Module	MODIFIED MANUALLY		04.03.2013	16:15:36	FRITTSCH	C01K901380		Z	
Z_TRAIN_2	4	FUNC	Function Module	MODIFIED MANUALLY		04.03.2013	16:15:36	FRITTSCH	C01K901380		Z	
LZTRAIN1TOP	1	REPS	Report Source ...	EQUAL		04.03.2013	11:07:20	FRITTSCH	C01K901368		Z	
LZTRAIN1U00X	1	REPS	Report Source ...	EQUAL		04.03.2013	11:07:20	FRITTSCH	C01K901368		Z	
SAPLZTRAIN1	1	REPS	Report Source ...	EQUAL		04.03.2013	11:07:20	FRITTSCH	C01K901368		Z	
LZTRAIN1TOP	1	REPT	Report Texts	EQUAL		04.03.2013	11:07:20	FRITTSCH	C01K901368		Z	
LZTRAIN1U00X	1	REPT	Report Texts	EQUAL		04.03.2013	11:07:20	FRITTSCH	C01K901368		Z	
SAPLZTRAIN1	1	REPT	Report Texts	EQUAL		04.03.2013	11:07:20	FRITTSCH	C01K901368		Z	
ZTRANSPORT_WIZARD_IMPORT_F			Function Group	NOT ON DEV SYSTEM		16.01.2013	13:37:14	HORFMANN	S07K902751		Z	

Figure 5: Status overview for all objects and the recommended synchronization methods

For the remaining objects, manual synchronization is supported by enabling the direct navigation to the object-specific delta monitor.

Function Module: Z_TRAIN_2 Active version	Function Module: Z_TRAIN_2 Version 00004
Local System: C14	Remote system: C01
Request: C14K900067	Request: C01K901380
Text: package	Text: Bugfix again and again

```

1 FUNCTION Z_TRAIN_2.
2 *
3 ***Local Interface:
4 *
5
6
7
8 *{ INSERT C14K900063 1
9
10 * Neues Release coding
11
12 * nochmal
13
14
15
16
17
18 *} INSERT
19
20
21 ENDFUNCTION.
    
```

```

1 FUNCTION Z_TRAIN_2.
2 *
3 ***Local Interface:
4 *
5
6
7 * Bugfix
8
9
10 ENDFUNCTION.
    
```

Figure 6: Direct navigation from the object to the object-specific delta monitor

**Conclusion:**

In order to establish a well-defined change and release management process in SAP, IT organizations need to expand their conventional 3-system landscapes to create 5-system landscapes. This will allow them to strictly separate their maintenance development from their release development, which will not only drastically improve the quality of their developments, but also ensure the continuous availability of their business-critical SAP applications. Given the tremendous impact these advantages have on the success of companies today, it is important not to shy away from moving to a 5-system landscape just because it involves a certain amount of synchronization effort. Professional change management solutions like theGuard! SmartChange automate the synchronization of development systems, keeping the effort required to a minimum and making the 5-system landscape a key factor to success.