Author:
Mariusz Debowski
Active Global Support

SAP Architecture Bluebook

Near Zero Downtime
Reduction of Business Downtime

Version 1.01
August 2010

Acknowledgement

The following colleagues contributed in various ways to make this or previous editions of the document possible:

Erwin Burkhardt, Greg Clavier, Klaus Daschakowsky, Oliver Kapaun, Steffen Meissner, Jochen Merkel, Zawisza Pierzchalski

1 Introduction

Over the past few years there has been significant progress made in extending the overall availability of SAP systems. The necessity of such demanding availability requirements is clearly visible for global companies. With the nature of business in a global economy, critical systems need to remain available around the clock. A *natural downtime* like nights or weekends no longer exists in business today.

With the growing demand of business solutions being able to operate 24x7, technologies were developed that improved system availability. The reliability of the systems improved significantly. *Unplanned downtimes* were reduced to a frequency and duration, which is acceptable by most customers. In this situation, *planned downtime* became a major importance regarding the system availability. The necessity to update the software on all layers of the vertical stack (starting from the hardware, through operating system level, database level up to application software) results in the disruption of system operations. Short system outages – 30-60 min – can be accepted by business users. However, there are many maintenance events which require longer outages of the system. This applies to the database maintenance and the application software. Currently, customers with high availability requirements are exposed to an accumulated downtime of 80-160 hours per year. Here, the costs of single downtimes increase strongly with the duration of the outage. Therefore, most customers would likely prefer a higher number of shorter downtimes than a single long downtime.

SAP is constantly improving the tools for software updates. On the other hand, the complexity of the application software, new functionalities, and the resulting adjustment of the existing data lead to business downtime requirements not being met.

Reduction of the business downtime requires minimization of the technical downtime (optimization of the software update tools) and minimization of the time spent on other tasks executed in the downtime – e.g. ramp-down, implementation of customers' transports, validation procedures.

A procedure that would reduce the overall downtime of SAP systems to less than 20 hours a year would satisfy most customers. This document details the Near Zero Downtime (NZDT) method, which provides the opportunity to meet this target.

Table of Contents

2 Technical Description

The Near Zero Downtime method was designed to minimize the downtime related to SAP software updates. It covers major release upgrades, implementation of Enhancement Packages and installation of Support Package Stacks. With NZDT also a Unicode conversion can be performed alone or in combination with the software update mentioned above. The major target was the minimization of business downtime in the areas not available for the standard methods. With the NZDT method the overall number of downtimes can be reduced through a combination of multiple maintenance events without having to extend the total downtime.

2.1 General Description

The general principle of the Near Zero Downtime method can be illustrated through the example of a major SAP release combined with Unicode conversion. In the described case, the start release is SAP R/3 4.6C non-Unicode (single code page) and the target release is SAP ERP 6.03 Unicode.

The procedure is shown as a 10-step process in Fig. 1.

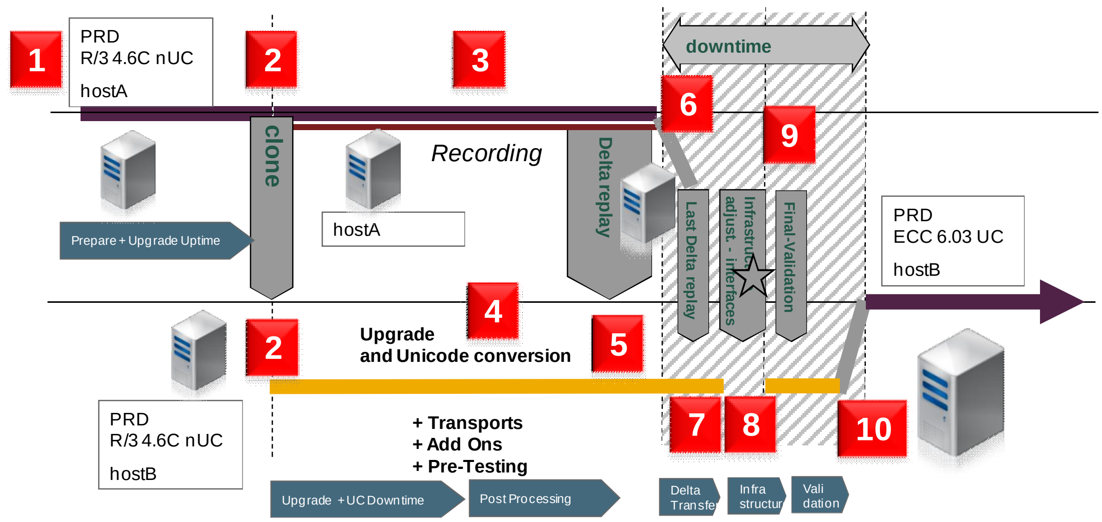

Fig. 1 Schematic description of the Near Zero Downtime procedure

Step 1: Production system is fully available to the users. The uptime preparation activities for the PRD system upgrade and Unicode conversion are executed.

Step 2: Recording of the transaction data is started – all changes on the database tables are being logged. No configuration changes (customizing) are allowed. The consistent clone is created and isolated. The clone keeps the SID of the production system.

Step 3: Recording of the transaction data continues on the PRD system.

Step 4: Upgrade procedure continues on the clone. After the completed upgrade, the clone is converted to Unicode with the standard method (export/import). The dual maintenance transports are imported into the clone. Most of the data conversions take place during Step 4 – XPRAs, After Import Methods, Conversion reports.

Step 5: After the upgrade and Unicode conversion of the clone are completed, the recorded data (Delta Records) is transferred from the production system to the clone – On-Line Data Replay. This transfer occurs during the operation of the production system. The users can work on the PRD system during the on-line data transfer.

Step 6: Real downtime starts. The production system is ramped-down. Users have to leave the system and batch jobs have to be stopped. The collected queues have to be processed and emptied. The final data transfer – Final Data Replay – is executed. All data is synchronized between the old production system and the clone (new production system).
Final data conversions are executed.

Step 7: Technical reconciliation of the transferred data is executed.

Step 8: The necessary infrastructure adjustments are performed.
Functional validation is performed.

Step 9: GO decision.

Step 10: The clone takes over the role of the production system and the PRD system is ramped up. User can start working on the upgraded and converted Unicode system.

The procedure is generally applicable to other maintenance events. Instead of the upgrade combined with Unicode conversion, another activity can be performed, e.g. Upgrade, implementation of an Enhancement Package, Customer's Release – mass transports, Support Package Stack implementation as well as a non-SAP event like database reorganization.

2.2 Cloning and Isolation

In order to perform the Near Zero Downtime procedure, a consistent hot clone needs to be created. This clone has to be isolated appropriately in order to avoid any data inconsistencies.

The clone of the production system is more considered as a clone of the production database than as a clone of the whole system. The cloned database requires just a Central Instance in order to start the SAP system on it.

Contrary to the common cloning procedure, the clone created here will take over a role of the production system. For this reason, this procedure has to be planned, tested and performed very thoroughly in order to ensure consistent data after the completion of the NZDT procedure. The size of the cloned database is the same as the size of the production system. The hardware used for the Central Instance of the clone has to be powerful enough to execute the upgrade, the Unicode conversion or other planned maintenance activity. The cloning of the Application Servers is not necessary. The Application Servers can be reconnected to the new system after the completion of the NZDT procedure.

2.2.1 Cloning Procedure

There are alternative possibilities to create a clone of the production database. Depending on the technology that is currently in use or easily available to the customer, the appropriate cloning technology can be chosen.

Examples of the cloning technologies:

- Hardware solutions, e.g. flash copy
- Operating system mirroring – splitting the mirror
- Standby or hot standby databases
- Point-In-Time recovery – backup/restore

In order to assure absolute data consistency within Near Zero Downtime, the point in time for the clone creation has to be defined taking into account when the recording of changes was started. The point in time is determined in such a way that all database transactions created during the activation of the recording mechanism are committed. Otherwise, the executed transaction would not be recorded, which may result in inconsistent data.

2.2.2 Isolation of the Clone

The created hot clone will eventually become the new production system. Therefore, data consistency on this system is of crucial importance.

The isolation procedure has to ensure that:

- No application documents are posted in the cloned system
- No inbound queues are processed in the cloned system
- No documents are sent out of the cloned system, i.e. no outbound queues are processed

On the other hand, the cloned system has to be accessible by administrators for the continuation of the upgrade or other maintenance, and for the NZDT tool for the replay of the recorded data.

The isolation procedure will cover two aspects of the system isolation:

2.2.2.1 External Isolation

The communication of the clone system (Central Instance started on the cloned database) with other systems has to be cut. The possible solutions include network isolation, firewalls, and SAP specific isolation, e.g. deactivation of the RFC connections (changing the RFC destinations to non-existing hosts on the db level), which will disable sapconnect.

2.2.2.2 Internal Isolation

After the start of the Central Instance on the clone, no application documents may be created automatically. Otherwise, inconsistencies after the data replay might occur, e.g. in the numbering of the documents. This can be achieved by deactivating the batch scheduler (internal as well as any external batch schedule tool, e.g., Redwood Cronacle, UC4, etc.). Furthermore, the update and RFC queue tables have to be cleaned up on the database level.

2.3 Table Classification

All tables on the systems must be classified whether changes are allowed or not. This activity is called table classification. Table classification ensures that no unwanted changes, which may cause conflicts between the PRD system and the clone, occur. This step is very important, because semantic conversions and potential conflicts are analyzed, solved and tested up front. Table classification ensures that these activities remain predictable during the productive cutover.

For each table there are essentially three process options:

- DELTA: Changes on this table are recorded and replayed. This process option is used for transaction data and master data
- FREEZE: No changes are allowed to this table. Any change causes an ABAP shortdump. This process option is chosen for most of the customizing tables
- RECORD_ONLY: This process option is used only for core basis tables like exchange tables or tables of the ABAP workbench. The content of these tables is not replayed, as a replay would cause a "downgrade."

2.4 Recording

The identification of the records, which have to be replayed, requires a reliable logging mechanism. This logging mechanism (recording) has to be able to capture each change relevant for the SAP system. For this reason, the recording is set on the database table level. The content of the database uniquely describes the SAP system. All kinds of inputs (User Interface, Batch Input, RFC calls, etc.) are reflected on the database level. Capturing the data on the database level ensures that all sources of data of the SAP system are correctly considered.

The database trigger technology was chosen to record the changes in the database. Each DML operation (INSERT, UPDATE, DELETE) on database tables results in the creation of an additional entry in the logging tables. Based on the information in the logging tables, the NZDT workbench identifies the records to be transferred from the production system to the clone (delta records).

The database trigger technology offers the necessary selectivity in the recording of changes. It allows to record changes on all kinds of tables used by SAP systems: transparent tables, cluster tables (containing compressed data, e.g. RFBLG) or pool tables (more SAP tables stored in a single database table, e.g. ATAB). Due to restrictions of NZDT (see details in Chapter *Table classification*), some tables will be set to read-only mode (freeze trigger). This is done also with the trigger technology.

In some cases, logical tables contained in one database table require a different kind of recording. For some of them, changes are allowed, but others are set to read-only.

Activation of the recording results in an additional resource usage on the production system. The IO consumption is enhanced during the recording phase. The impact on the production system can be kept on a negligible level if the recording is appropriately optimized. This optimization is a part of the NZDT implementation project.

2.5 Replay of Data

The bulk of the recorded data, typically more than 99.5%, is transferred to the clone during the uptime. In the downtime only the residual delta records are transferred to the clone.

2.5.1.1 On-line Delta Replay

After the completion of the upgrade or other maintenance task on the clone, the transfer of the recorded data can be started. The design of the data recording allows the transfer of the data on-line without the risk of a locking situation on the production system. However, the reading of the recorded data results in an additional load on the production system, especially on the database. This load has to be monitored and the parallelization level of the data transfer should be adjusted to the capacities of the production environment. The parallelization level can be adjusted dynamically according to the current load on the system.

2.5.1.2 Final Delta Replay

The final data synchronization and the transfer of the last recorded data can be performed once no new business activities are performed on the production system. The final delta replay is executed during the downtime.

Because the majority of the data was transferred during the uptime, only few records have to be transferred during the downtime. In addition, data from tables where the on-line replay was not possible is transferred during the downtime (e.g. for spool tables). Nevertheless, the duration of the final delta replay can be reduced to less than 30 minutes.

2.6 Resynchronization and Switch

2.7 Data Conversions

The release upgrade or other software updates sometimes require adjustments in the data model. Immediately after the implementation of the new software, some database tables have to be modified. It could be a structural change, e.g. a new field added to the existing table. This kind of change is automatically included in the NZDT workbench. Another type of changes is a semantic change, which is when the tables receive new or modified content. Those conversions are performed either by the execution of special reports, XPRA, or by execution of After Import Methods. Sometimes the changes are implemented through the execution of dedicated conversion reports.
Most of the conversions are included automatically in the upgrade/update execution. Some of them are referenced in the corresponding SAP Release Notes.

The data conversions usually apply to customizing tables. Those tables remain read-only during the NZDT procedure. Therefore, the conversion of those tables on the clone leads to the correct result for the new production system.

If semantic conversions apply to data which was transferred, the corresponding conversion methods (XPRA, AIM or data conversion report) have to be executed after the completed delta replay in the downtime. Alternatively, for large tables with long conversion runtimes, the conversion rules can be implemented in the delta replay. This however, requires a manual adjustment within the project.

2.8 Unicode Conversion

2.8.1.1 Near Zero Downtime Method and Unicode Conversion

If the current production system is installed as a non-Unicode system and the new production system will be a Unicode system, a Unicode conversion can be integrated in the NZDT process. In addition to the NZDT Upgrade, some additional steps are required if a Unicode conversion is also in scope. In this chapter, the additional steps of the Unicode conversion are described. Here, however, just the technical conversion aspects are covered. Topics like ABAP enablement or interface adjustment have to be treated within the projects according to the standard procedures.

In this scenario, the central system that contains the NZDT workbench has to be a non-Unicode system with the same codepage settings as the current production system.

2.8.1.2 Unicode Preprocessing

The Unicode Preprocessing will start in the production system before any trigger is activated. For Unicode Preprocessing the same steps have to be processed as those in a standard Unicode conversion (Transaction SPUMG).

2.8.1.3 Unicode Preprocessing During Recording Phase

Before the Unicode conversion on the clone system can start, the Unicode conversion settings must be transported to the central system so they can be used in the *Integrated Unicode Conversion* during the delta data transfer. The Unicode Conversion Settings are transferred into the central system via the *TwinTransfer* functionality provided by the Unicode conversion.

2.8.1.4 Standard Unicode Conversion

After all Unicode preprocessing steps have been done, a standard Unicode conversion can be executed on the clone system. This also includes the Unicode Postprocessing steps. When the Unicode conversion of the clone system is finished and all preparations of the Integrated Unicode conversion are completed, the On-line Delta Replay can start.

2.8.1.5 Integrated Unicode Conversion

In the On-line and Final Delta Replay, data from a non-Unicode System to a Unicode System is transferred. To enable a proper transfer of the delta data, a Unicode conversion of these records has to be done in between. This functionality is called Integrated Unicode Conversion.

This Integrated Unicode Conversion requires the Unicode conversion settings used on the clone system for the standard Unicode conversion. The Unicode conversion settings have to be transferred from the clone system to the central system before the standard Unicode Conversion can be executed.

2.9 Integration of Customer's Transports in NZDT

During an Upgrade project or in the case of larger software changes on the system – e.g. custom release – a large number of objects have to be transported into the system during the downtime. On one hand, these transports result from the necessary customizing on the new software release. On the other hand, a large number of transports result from the dual maintenance for the duration of the project. This number can easily reach 10,000-20,000 for very large systems. The duration of the implementation of those transports can exceed 15 hours even if tuned appropriately.

Therefore, it is very important to offer the possibility to integrate these transports into the NZDT procedure. The vast majority of the transports can be implemented during the uptime, before the on-line delta replay starts. Here, it is important to avoid collisions between the data imported with the transports and data transferred from the production system. Similar criteria as for the execution of XPRAs and AIMs apply here as well.

A safe and efficient implementation of the customer's transports during the uptime requires a thorough analysis of the content of these transports. The workbench transports (coding adjustments and extensions) can be applied without restrictions in most of cases. The transports, including data content or executing After

Import Methods, have to be evaluated in detail. Once a potential collision between the transported content and the data synchronized during delta replay is identified, the consequences have to be assessed. In most of cases, the identified transports (After Import Method) would be executed in the downtime after the final delta replay is completed. Usually, the impact on the downtime duration is in the area of 20-40 minutes.

2.10 Technical Reconciliation

After the final replay, a technical reconciliation is performed. This step verifies the technical correctness of the delta replay. As the reconciliation activities increase the downtime, the scope of the technical reconciliation has to be defined within the specific project together with the customer.

The technical reconciliation includes the following functions:

- Comparison of the row counts on the PRD and on the clone system.
- Comparison on the row and field level between PRD and the clone system.

2.11 Workbench

The whole NZDT procedure is controlled from the central system – NZDT workbench. The control system contains the NZDT tools that are executed during the entire project. Both the production system and the clone system have to be connected to the control system via an RFC connection. The control system acts as a central system and starts the required NZDT functionality on the production and the clone system. The NZDT software must be installed in all participating systems.

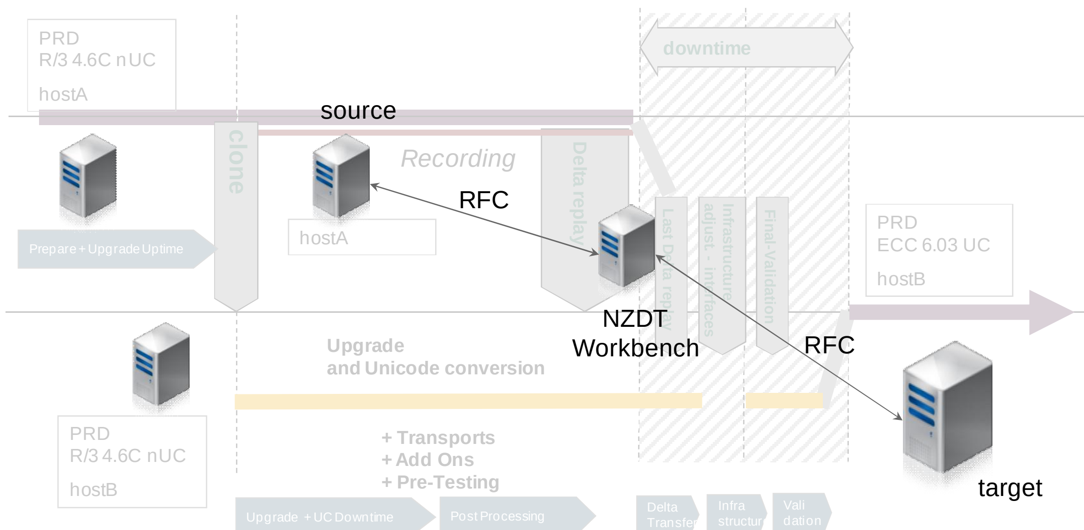

Fig. 2 Connections between the NZDT workbench and both systems (source and target)

2.11.1.1 Sizing of the Central System

The sizing of the central system is essential, since performance of critical jobs that are executed immediately affects the downtime. The major factors influencing the sizing of the workbench are the volume of the recorded (and thus transferred) data. The jobs for the technical reconciliations of the cluster tables are the other factor.

A minimum of four CPU cores and 20 available batch processes must be available in order to consider it for purposes other than testing. For a very large ERP system with high transactional load and a high data volume to be transferred and reconciled, 16 CPU cores are recommended.

The NZDT workbench does not have any persistence of the transferred records. Therefore, the file space size and performance are of minor importance.

2.11.1.2 Software Versions Requirements

The NZDT tools are based on the DMIS package. The most recent version of the DMIS add-on should be used. In addition, a set of tools is delivered as a transport. The DMIS add-on has to be implemented on the NZDT workbench as well as on the production system. The possible dependencies and minimum software level requirements have to be considered.

Production system Supported Basis Release: 4.6C or higher

NZDT workbench Minimum required Basis Release: 6.20 (WebAs 6.20) SP41

If a Unicode Conversion is in scope, a higher support package level is required

Basis Release	Support Package level
6.20	65
6.40	23
7.00	17
7.01	2
7.02	0

3 NZDT Project

Customers operating SAP solutions deal with downtime challenge regularly. SAP is constantly improving the standard optimization methods for the downtime minimization in the case of upgrades, Enhancement Package installations or Support Package installations. However, sometimes the business needs require methods that go beyond the available standard tools.

Near Zero Downtime is a sophisticated method that requires an implementation in the customer's environment. Typically, the implementation is performed in three major steps:

- Initial information
- NZDT Planning Workshop
- NZDT Project

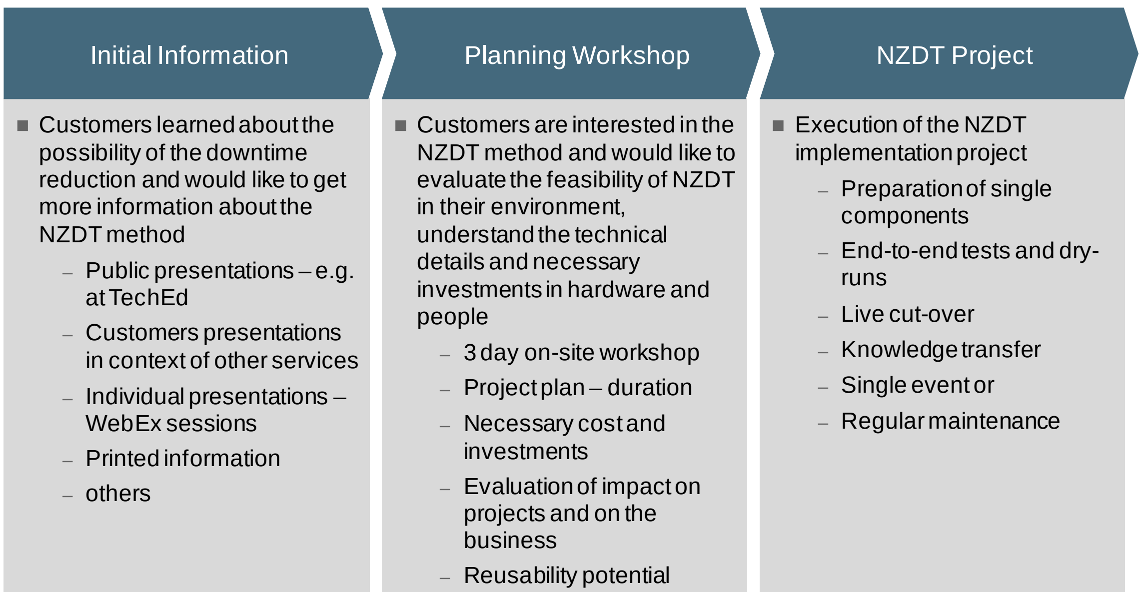

Fig. 3 NZDT cycle – from initial information to successful implementation

3.1 Initial Information

SAP constantly informs its ecosystem about new tools and innovations supporting the operations of SAP systems. In addition, the Near Zero Downtime method was presented on different occasions to the customers.

Once customers learned about the possibility to considerably reduce downtime, they requested more detailed information about the method. This detailed information is presented in public forums, e.g. TechEd, but also in direct meetings with customers. SAP also offers remote sessions, where the details of the NZDT are discussed. Based on this information, customers can evaluate whether the method can address their requirements. More details, including the specific hardware requirements, additional project tasks or the expected cost of the service, are clarified in the Planning Workshop.

3.2 Planning Workshop

The planning workshop allows for the projection of the generic NZDT method on the customer's environment. Here, starting with the discussion about the existing environment and business processes, the SAP team will individually adapt the standard implementation methodology to the specific requirements and constraints. Understanding the technical infrastructure, the IT landscape, the IT procedures and the hardware capabilities, the details of the NZDT implementations can be identified and used as the base for the planned NZDT project.

During the workshop, a high-level project plan will be created. This project plan has to be aligned with other activities planned for the specific maintenance event. It would specifically include the testing of individual items contributing to the project – e.g. cloning test, trigger test and validation, installing of the necessary software – as well as the testing cycles approaching the live cut-over. In some cases, the NZDT project has to be aligned with a project that is already in progress. The dependencies and necessary check-points will also be defined during the Planning Workshop.

As the NZDT method might result in some business restrictions, e.g. customizing changes are strongly limited in the period prior to the cut-over and during the tests including production data, it is important to discuss these restrictions in detail.

During the workshop the prerequisites and necessary tests will be discussed. Here, SAP can support customers in choosing the appropriate cloning technology and verifying hardware capacities to handle the NZDT method.

Finally, as the result of the workshop, a solid base for the detailed cost estimation will be established. This includes beside of the external service cost also the input for the necessary hardware investment or additional internal project effort in order to cover all necessary tasks in the NZDT project.

3.3 Near Zero Downtime Project

In the past projects SAP established an NZDT implementation methodology assuring the high confidence and minimizing the risks related to the usage of NZDT. In Fig. 4 the major steps of the NZDT project methodology are shown.

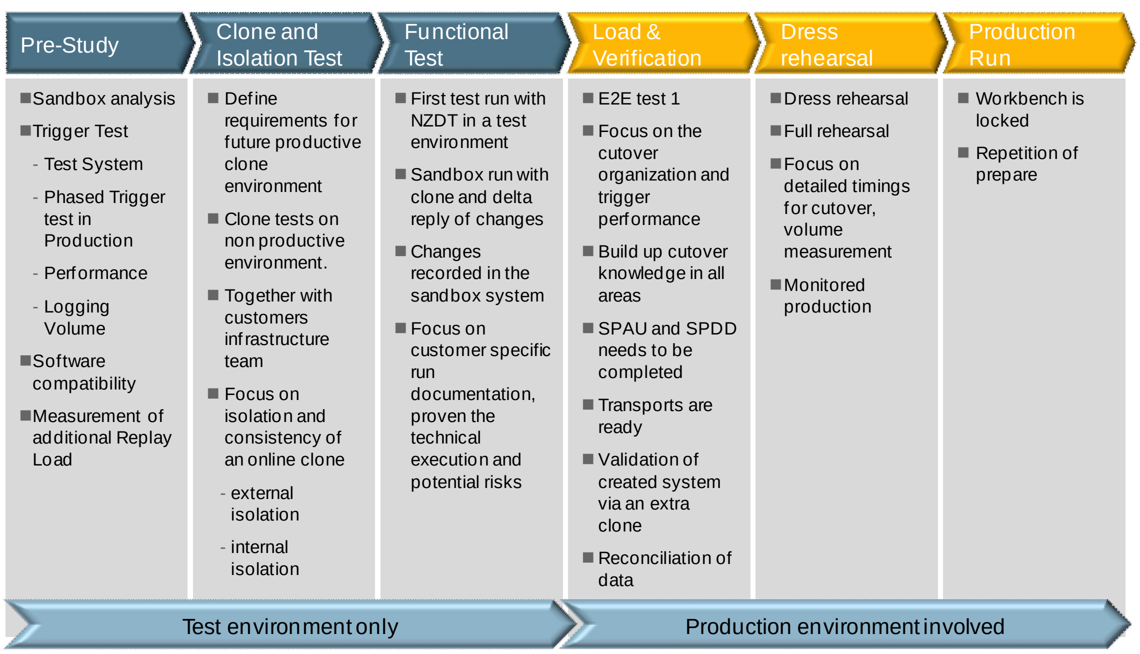

Fig. 4 Near Zero Downtime implementation project – project methodology

3.3.1 Pre-requisites

In the first phase of the project the single components of NZDT need to be provided. They include the necessary hardware and system provisioning, the installation of the required software components.

3.3.1.1 Workbench Installation

The whole procedure is being controlled and executed from the NZDT workbench. The technical details related to the workbench are described in Chapter *Technical Description - Workbench*. In the project planning, the lead time for ordering the hardware and the preparation of the system for the installation of the NZDT workbench has to be included.

This step has to be completed first, as all further steps are controlled directly or indirectly from the workbench.

3.3.1.2 Clone and Isolation Test

Based on the identified appropriate cloning method, the procedure should be established and tested. The major focus is a thorough check of the isolation measures and verification. If any postings have been created on the isolated system, for example, the recording technique can be used. The timing for the creation of the clone should be verified and, if necessary, optimized.

3.3.1.3 Ramp-down Test

In the live cut-over, the real production system will be ramped down under recording conditions. This means some tables will remain as *read-only*. The impact of these conditions on the seamless ramp down should be evaluated in this step.

3.3.1.4 Software Installation

The NZDT tools are installed on the NZDT workbench as well as on the production system. The detailed specification and dependencies are described in the Chapter *Technical Description - Workbench*.
The installation of the required add-on DMIS on the production system has to be included in the project planning. The software change management procedures require a corresponding step sequence in order to promote the DMIS software to the production environment.
The DMIS add-on is a non-modifying add-on, which can be applied on-line. Nevertheless, the possible impact on the existing software should be verified in the standard software change management process.

3.3.1.5 Recording – trigger validation

Before starting end-to-end tests with production data, the impact of the recording technique on the production system has to be verified. The triggers used for the recording of changes usually do not impact the system performance. On the systems with extremely high load the impact has to be verified in more detail. Ideally, the processes generating the highest load on the system should be tested with active triggers on a test environment. Here, the optimization measures can be established e.g. optimized layout of tables with the recorded data on the database.
On production systems, a gradual setting of triggers and the detailed monitoring of the system behavior would minimize the risk related to the triggers activated on the production system.

3.3.1.6 Table Classification

With the NZDT tool, SAP provides an initial classification of the tables. This table classification has to be fine tuned to the actual situation on the customer's system. The specific table classification can depend on the used software components, custom enhancements and active business processes on the system.

3.3.1.7 Data Conversions

Some software changes require adaption of the data model. The corresponding changes will be performed by the execution of XPRAs, After Import Methods or Conversion reports. The execution of the data conversion can usually be done on the clone during the uptime. Sometimes, it has to be executed in the downtime, as the tables with recorded changes are affected.
NZDT workbench offers the capability to perform data conversion during the data transfer. The specific pattern for the conversions has to be configured in the workbench individually, based on the actual project requirements.

3.3.1.8 Customer's Coding Analysis

The NZDT method can work together with custom modifications and enhancements. Nevertheless, there is a risk that the custom coding is using direct database commands which might result in data changes not being recorded by the triggers. For this reason, the custom coding will be searched for the direct database commands and the result will be verified if there is any risk of data inconsistency.

3.3.1.9 Analysis of Customer's Transports

In the case of upgrades or other major software changes, customers establish the dual maintenance setup. As a result, during the production cut-over a large number of transports have to be imported. The transports will be imported on the clone during the uptime of the production system. However, the influence of the transported content on the recorded changes has to be evaluated. If a transport triggers the execution of an After Import Method, the impact has to be analyzed carefully.
The set of transports, which will be imported in the live cut-over, is growing constantly. The analysis has to be repeated for the dry runs and for the cut-over.

3.3.1.10 NZDT Functional Test

Based on the single items prepared as described above, the first functional test can take place. This test includes all steps of the NZDT procedure. The test is performed on a sandbox environment. Here, during the upgrade (or other change) on the clone, some business activities should be executed on the source system. The resulting changes will be recorded and transferred to the clone.
Finally, the technical and functional reconciliation of the data will be done.
This test will confirm the feasibility of the method within the specific customer's environment. The major dependencies will be verified. It will also be a rehearsal for the first end-to-end test.

3.3.2 End-to-End Tests and Dry Runs

Based on the experience of completed projects, SAP recommends the customer perform at least two end-to-end tests of the NZDT method. The successful completion of those tests will guarantee a smooth, safe and predictable execution of the live cut-over.

End-to-end tests include all critical components of the NZDT method. They are executed in the sequence and involve the same teams and persons as during the live cut-over. The tests are based on the real production data. Therefore, they reflect the real situation very closely.

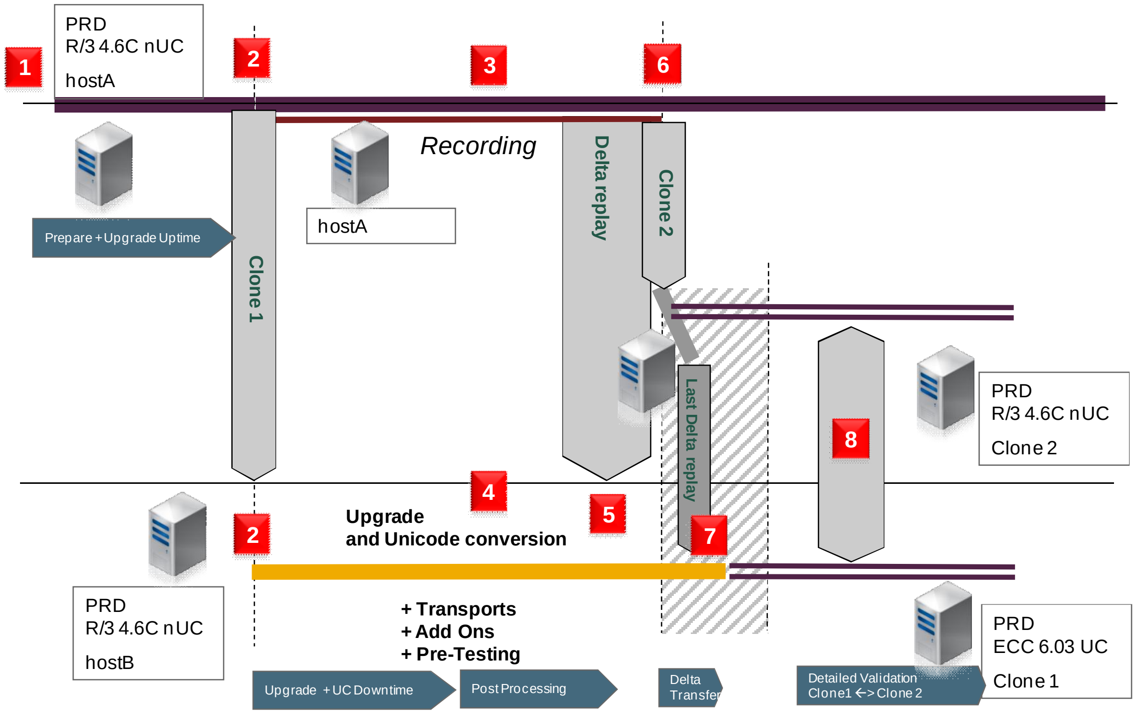

Fig. 5 End-to-end of the NZDT method – dry run

Similar to the NZDT procedure as shown in Fig. 1 Fig. 5 illustrates a step-by-step procedure for an end-to-end test of NZDT.

Steps 1-5 are identical to the live cut-over.

Step 1: Production system is fully available to the users. The uptime preparation activities for the PRD system upgrade and Unicode conversion are executed. Step 2: Recording of the transaction data is started and all changes on the database tables are logged. No configuration changes (customizing) are allowed. The consistent clone is created – Clone 1 – and isolated. Clone 1 keeps the SID of the production system.

Step 3: Recording of the transaction data continues on the PRD system.

Step 4: The upgrade procedure continues on the clone. After the upgrade has completed, the clone is converted to Unicode with the standard method (export/import). The dual maintenance transports are imported on the clone. Most of the data conversions take place during Step 4 – XPRAs, After Import Methods, Conversion reports.

Step 5: After the upgrade and Unicode conversion of the clone are completed, the recorded data (Delta Records) is transferred from the production system to the clone – On-Line Data Replay. This transfer occurs during the operation of the production system. The users can work on the PRD system during the on-line data transfer.

Instead of Step 6 from the live cut-over (*start downtime*), the downtime will be simulated in the end-to-end test.

Step 6: A second clone is created – Clone 2. This clone will also be isolated and ramped down. Because of the isolation, some limitations might occur here.

Step 7: Final replay of recorded data is performed from Clone 2 (simulated PRD) to Clone 1. At this stage, Clone 1 has the new SAP release and the data as of the time of the creation of Clone 2.

Step 8: A thorough validation and comparison between Clone 1 (new release) and Clone 2 (old release) is performed. This validation includes the technical and functional validation.

This procedure allows end-to-end testing of the whole NZDT process. Each component of the procedure will be verified. Additionally, the collaboration of the whole involved team will be trained. The procedure will be optimized and the duration of the process will be a base for the detailed timeline planning for the live cut-over.

The resulting system obtained during the end-to-end testing cycle (Clone 1) can be used as the system for User Acceptance Tests. In this case, the SID of the system has to be chosen appropriately.

3.3.3 Go-Live – Cut-Over – Business Downtime

Based on the results of the optimized end-to-end test, the exact planning of the live cut-over can be performed. This plan will include the components executed during the uptime of the production system and during the downtime.

The example discussed in detail will be based on the upgrade from SAP R/3 4.6C non-Unicode to SAP ERP 6.03 Unicode as seen in Fig. 1.

Fig. 6 shows the comparison of the downtime planning for a standard combined upgrade and Unicode conversion and NZDT procedure. Only the major steps are displayed in the picture.

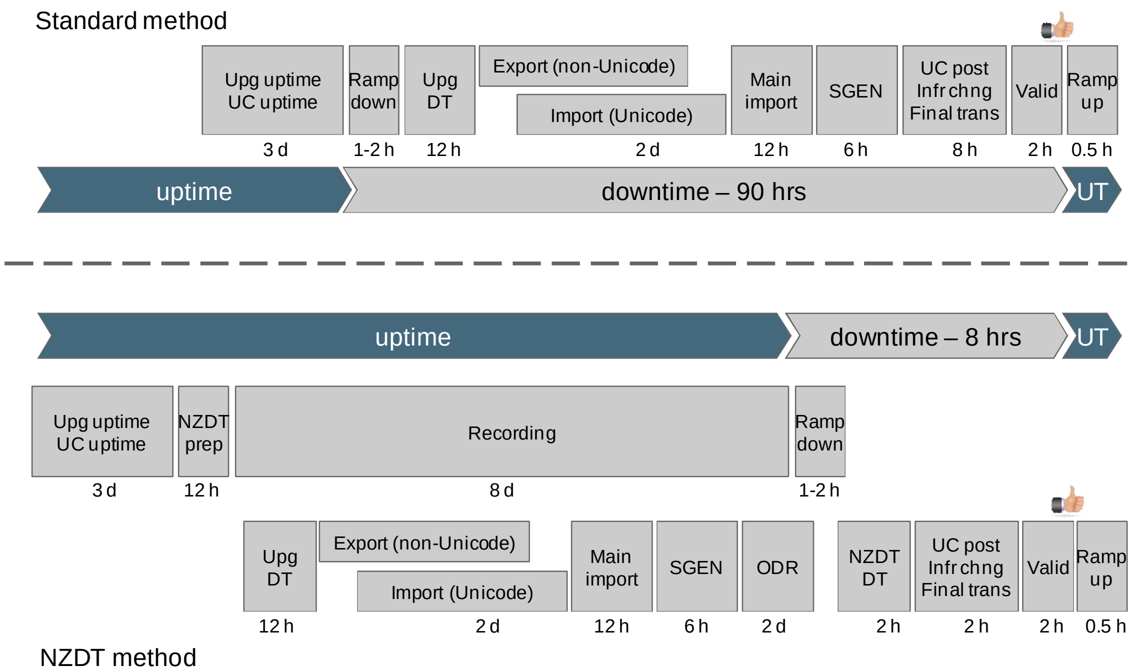

Fig.6 Comparison of the standard upgrade combined with Unicode conversion (upper part) and the NZDT upgrade and Unicode conversion (lower part)

A detailed explanation of the abbreviations used in Fig. 6 is listed below:

Upg uptime	Part of the upgrade executed during the system uptime – preparation
UC uptime	Unicode conversion – preparation
Upg DT	Upgrade – downtime part
UC post	Unicode conversion post-processing
Infr chng	Infrastructure changes – necessary adjustments of the hardware, interfaces, etc.
Final trans	Final transports not included in the main import phase
Valid	Validation of the final solution
NZDT prep	NZDT preparation steps – creation of logging tables, start recording
ODR	On-line delta replay
NZDT DT	NZDT downtime steps – final delta replay, downtime conversions, etc.

Due to the complexity of the upgrade combined with Unicode conversion, the achievable business downtime is 8 hours. For most of the maintenance events the business downtime can be optimized to about 6 hours. This downtime includes the business validation of the system. The end-to-end duration of the downtime is a compromise between the duration of the validation steps and the business requirements regarding system availability.

3.3.4 NZDT High-Level Project Plan

The total duration of the NZDT implementation project depends on the actual event for which the downtime would be minimized. For example, a major release upgrade combined with Unicode conversion will exceed seven months, whereas it is possible to implement an Enhancement Package in five months.

An example of a high-level project plan is shown in Fig. 7. In this plan the major phases with sample timelines are illustrated. In this example, the Load Verification tests (dry runs) are executed three times. This would allow further optimization of the cut-over process and increase confidence for the method.

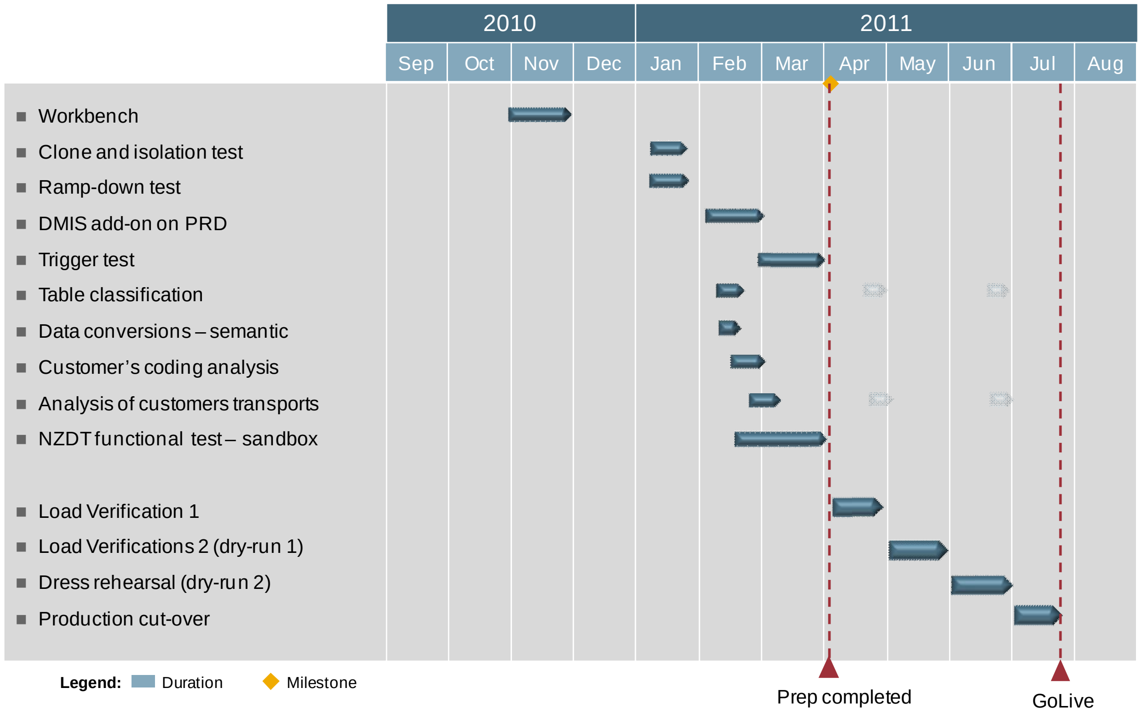

Fig. 7 Sample project plan for an NZDT implementation

The project timelines would be considerably shorter for subsequent executions of NZDT for a given environment. In this case most of the preparation steps can be omitted or strongly reduced. More details about the reusability of NZDT and the optimization potential for regular usage of NZDT are described in the Chapter *Applicability - Reusability*.

4 Impact on Business

According to SAP best practices, a dual maintenance transport landscape should be established for upgrade projects and other projects with a major software modification (EhP implementation, large Go-Live with customer's release, etc.). The production system will be fed with transports from systems on exactly the same software level as the production system. This approach implies that a development freeze phase will occur in the project.

Moreover, we can define further freeze phases where additional restrictions apply in a NZDT project.

4.1 Soft Freeze

The soft freeze phase starts with the first end-to-end test with production data (Load Verification). After this test the first setup of the NZDT method is verified and each change to this setup has to be considered in the validation of the method. Starting with the soft freeze, only high priority transports are allowed. The NZDT team has to verify the impact of the transported objects on the NZDT approach.

Dictionary (domains, data elements, structures, table types, tables, views, indexes, etc)	ABAP	Customizing	Roles
▪ Dictionary changes are allowed ▪ No new tables ▪ No dropping (deletion) of tables	▪ ABAP objects e.g. single report, user exist, etc. can be transported, but could have side effects ▪ NO native sql commands are allowed!	▪ Customizing which contains only data like exchange rates is not critical ▪ Customizing could end in a process change → side effect ▪ Customizing can create tables (Conditions) → side effect ▪ Customizing changes can cause table changes e.g. FI table BSEG) → side effect	▪ No restriction

4.2 Hard Freeze

During the recording of changes in the NZDT method some restrictions on the production system are applied. Whereas, most of the regular functional activities would be available, the configuration changes would be restricted (no customizing changes are allowed during the recording of data). Besides customizing and repository changes, which are not allowed in most customer systems, the following restrictions apply:

- No period closing – this makes changes in some customizing tables
- No creation of ad-hoc financial reports with report writer and report painter
- No creation of BW DataSources
- Planning versions and planning layouts in CO and CO-PA have to be created before the hard freeze phase.
- Archiving runs have to be inactive – reduction of additional data load on the database

For most of customers these restriction is acceptable for a limited period – e.g. for one week. In the case of very specific requirements where transportable changes are absolutely necessary for the daily business, a workaround can be established with the project work. This workaround might result in a longer downtime.

In special situations where emergency corrections would have to be implemented on the production system, the NZDT execution team has to evaluate the impact of such a change on the data transfer. As a worst case, if the implementation of the transport could not be avoided due to business requirements, the whole NZDT execution would have to be stopped and rescheduled.

Dictionary (domains, data elements, structures, table types, tables, views, indexes, etc)	ABAP	Customizing	Roles
▪ No transports are allowed ▪ Workbench is locked ▪ No changes are allowed	▪ No transports are allowed ▪ Workbench is locked ▪ No changes allowed	▪ No customizing change is possible ▪ The critical customizing relevant for daily adjustments should be specified upfront in order to be specially treated within NZDT	▪ Emergency changes of roles must be available → roles will have a separate naming convention ▪ SAP will provide list of SAP system users required for the execution of the NZDT upgrade ▪ SAP will provide a list of technical users required by NZDT workbench for the communication between the systems ▪ Important to consider both, the execution of the delta-replay and the reconciliations

5 Risk Estimation

From a risk perspective, the NZDT method generates additional technical risks on but also reduces the risks related to the standard method for upgrades or other maintenance.

5.1 Risk Resulting from NZDT

The potential risks resulting from the usage of Near Zero Downtime and the proposed mitigation measures are summarized below.

5.1.1 Risk During Recording

The major risk related to the NZDT is the performance degradation during the recording or during the replay of the recorded data in the uptime (on-line delta replay). Although the recording procedure is already optimized, the specific customer's environment can result in additional performance overhead during recording. This can happen with very frequently accessed tables (hot-spot tables). This risk can be mitigated by testing the recording mechanism first in a test environment using a test case including the functions generating the heaviest database load. During this test the unexpected resource consumption or excessive runtime increase (more than 20%) can be observed and the appropriate tuning can be applied. Usually, satisfactory results can be obtained by adjusting the tablespaces containing the logging tables.

The resulting tuned procedure will be verified in the production environment. The NZDT method can remove the triggers from the database in an emergency case.

5.1.2 Risk During On-line Delta Replay

Reading the data from the production system and transferring it to the clone consumes the hardware resources on both sides. However, this effect is critical only on the source side. The resources support the regular production activities at the same time.

The delta replay process is technically optimized in order to minimize the resource consumption. However, in many cases a large volume of the data has to be read and transferred. The NZDT workbench offers the possibility to adjust the load to the current situation. For example, during the peak time of usage of the production system, the number of jobs transferring the data can be reduced. When more production hardware recourses are available, this load can be increased again. The load balancing is performed with the corresponding logon groups for the access from the workbench to the production system.

The important measure here is the allocation of the sufficient time interval for the on-line delta replay including contingency.

5.1.3 Restricted Availability of Some Functions

As described in detail in the Chapter *Table Classification*, some database tables are set to a *read-only* mode during the execution of NZDT and during the end-to-end tests. Although the selection of the *read-only* tables is performed very thoroughly, some functions could remain unavailable during the mentioned periods. This applies especially to the changes of the configuration. Typical examples are: changes of the structures in the areas of FI-SL or CO-PA, generation of new BW extractors.

The risk can be minimized by reducing the recording time to a minimum period (optimization of the procedure executed on the clone). Further measure is the early information exchange with the affected functional areas in order to verify the severity of this impact and identification of a workaround if necessary.

5.2 Risk Reduction Through NZDT

The Near Zero Downtime reduces other risks resulting from the maintenance activity. It offers additional contingency and increases the confidence and reliability of the downtime.

In the NZDT procedure, the most critical steps are performed on the clone and are completed during the uptime. The risk related to these tasks is removed from the downtime. If any unexpected incident occurs, the issue can be resolved on the clone without impacting the downtime.

In the situation where the upgraded system cannot be approved because of severe errors, the downtime can be cancelled. If this is identified first in the downtime, the *old* production system will still be available and can be reconnected and provided to the end users with minimal effort.

6 Applicability

SAP is constantly improving the offering for the minimization of the downtimes resulting from different maintenance activities. The Near Zero Downtime method, as described above, was initially developed for ABAP based SAP ERP systems.

6.1 Near Zero Downtime – Use Cases

The Near Zero Downtime method is a generic approach that can be used for the minimization of downtimes resulting from numerous maintenance events. In the operations of complex IT landscapes, the challenge is to keep the systems up-to-date by applying the Support Packages regularly, applying the legal changes according to the regulations, optimizing the operational costs by optimizing the location of IT services, and allowing access to innovation through the usage of the modern version of the software. On the other hand, all these tasks lead to interruptions on the production systems.

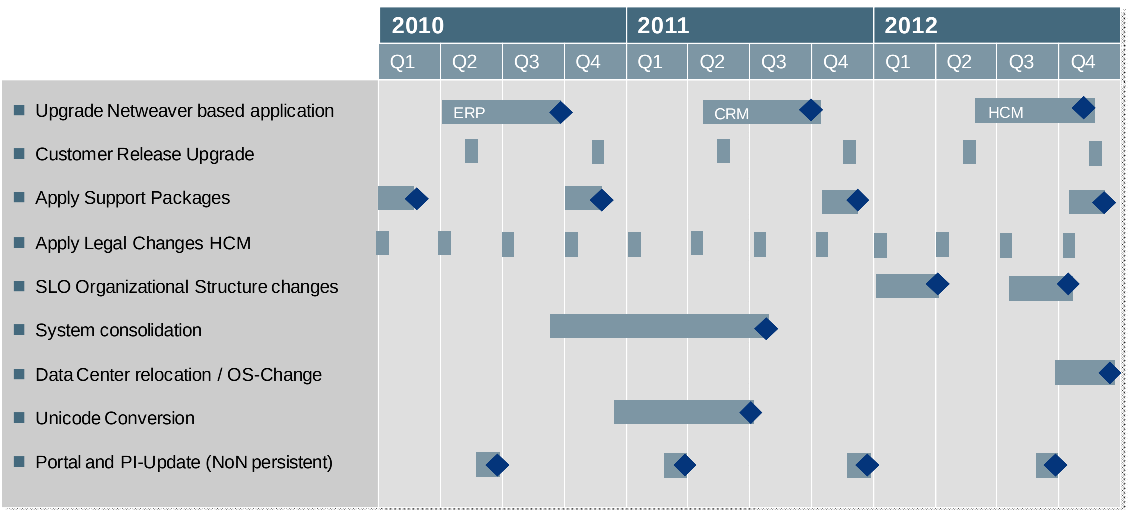

Fig. 8 An overview of different maintenance events within an IT landscape and their frequent occurrence

With the Near Zero Downtime method, the downtime resulting from most of the events mentioned above can be considerably reduced. The architecture of NZDT allows application of this method especially to:

- Major release upgrades
- Installation of Support Package Stacks
- Installation of Enhancement Packages
- Unicode conversions
- OS/DB migrations
- Data Center relocations
- Custom Development Releases – Mass transports
- Combinations of the events above

The NZDT method offers the opportunity to combine multiple maintenance events within a single downtime. This can be done without extension of the system outage. The risk mitigation resulting from the NZDT method applies to all events combined within the downtime.
This is a big advantage for the customers using NZDT for the minimizing of their system downtimes.

6.2 Near Zero Downtime – Reusability

The implementation of the NZDT method requires some project effort and results in some hardware investment. This investment can be used in repeating events during the regular operations of the system for the standard maintenance tasks if they cause a considerable outage of the system. The single components developed for the implementation of NZDT can be reused in subsequent executions of the method.

The table below summarizes the tasks around the implementation of the NZDT method and provides an estimate on the reusability level of each task. The scale of reusability spans the range between *very low* – this task has to be repeated for each execution of the method and *very high* – this task has to be developed and tested once and can be later reused without modifications.

	Upgrade and Unicode Conv	Upgrade	EhP installation	Support Pack	Mass trans-ports	Non-SAP event eg. DB-maint.	Reusability level
Clone creation	Once	Once	Once	Once	Once	Once	Very high
Clone isolation	Once	Once	Once	Once	Once	Once	Very high
Trigger tests and validation	Once	Once	Once	Once	Once	Once	Very high
Setup of the NZDT tools	Once	Once	Once	Once	Once	Once	High
Identification/classification of logging tables	Yes	Yes	Yes	Yes	Yes	No	Low
XPRA-like changes NZDT tool configuration	Yes	Yes	Yes	Yes – low effort	Yes – low effort	No	Very low
Cut-over test	Yes	Yes	Yes	Yes	Yes	Yes	Very low
Additional Hardware	Once	Once	Once	Once	Once	Once	Very high

The table illustrates the potential of the effort reduction if the NZDT method were to be applied to a single system a number of times. The improvement results from the re-usage of the packages established during the first execution. With each execution, SAP will transfer more tasks to the customer's operations team. Therefore, the external effort would be further reduced.

The resulting effort per execution for the multiple usages of NZDT is illustrated in Fig. 9.

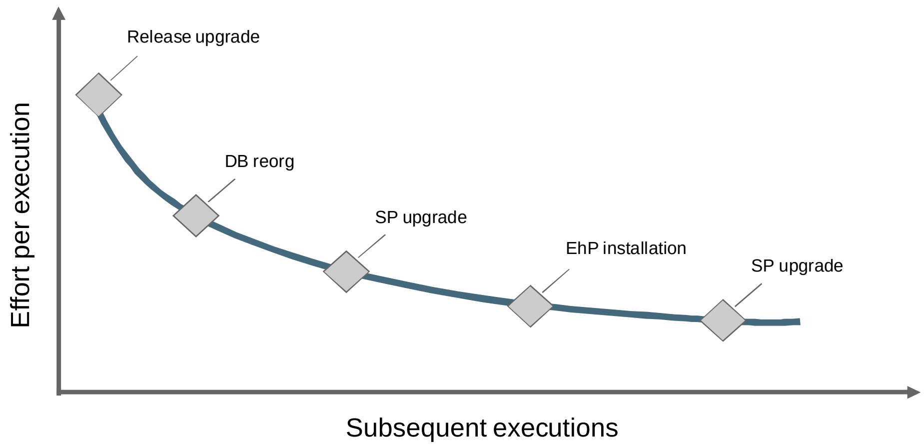

Fig. 9 The effort for the implementation of NZDT is decreasing with each subsequent usage of the method. The exact figures can be obtained based on the specific scenarios as used by customers.

7 Near Zero Downtime Methods for Further SAP Products

The interest in the Near Zero Downtime method refers to other SAP systems than ERP. Even if ERP systems usually have the highest criticality in regard to system availability, in many cases the critical business processes in customers' landscapes include additional SAP software components. For these components, SAP is providing solutions for the downtime minimization (Process Integration (PI) or Enterprise Portal (EP)), or is planning to offer the solution in the near future. See the Chapters below for details. Currently, the solutions for CRM, SCM, GTS and Banking Services are considered to be most critical from the availability perspective.

7.1 Near Zero Downtime Maintenance for PI (Pilot)

The NZDT method that has been transferred to PI is called Near Zero Downtime Maintenance PI and allows the application of Support Package Stacks on the PI system with the business downtime reduced to one system restart. This holds true only for updating the SAP software – OS or DB updates are not included. The update is performed on a mirror system while the production system remains available until the final switch to the new SP level is initiated by a restart. During the SP update all standard PI scenarios are fully available, including Synchronous/asynchronous messages, mapping and third party adapters.

The Fig. 10 depicts the general procedure of the method. The production remains fully functional while a copy of the production system is utilized for the Support Package update.

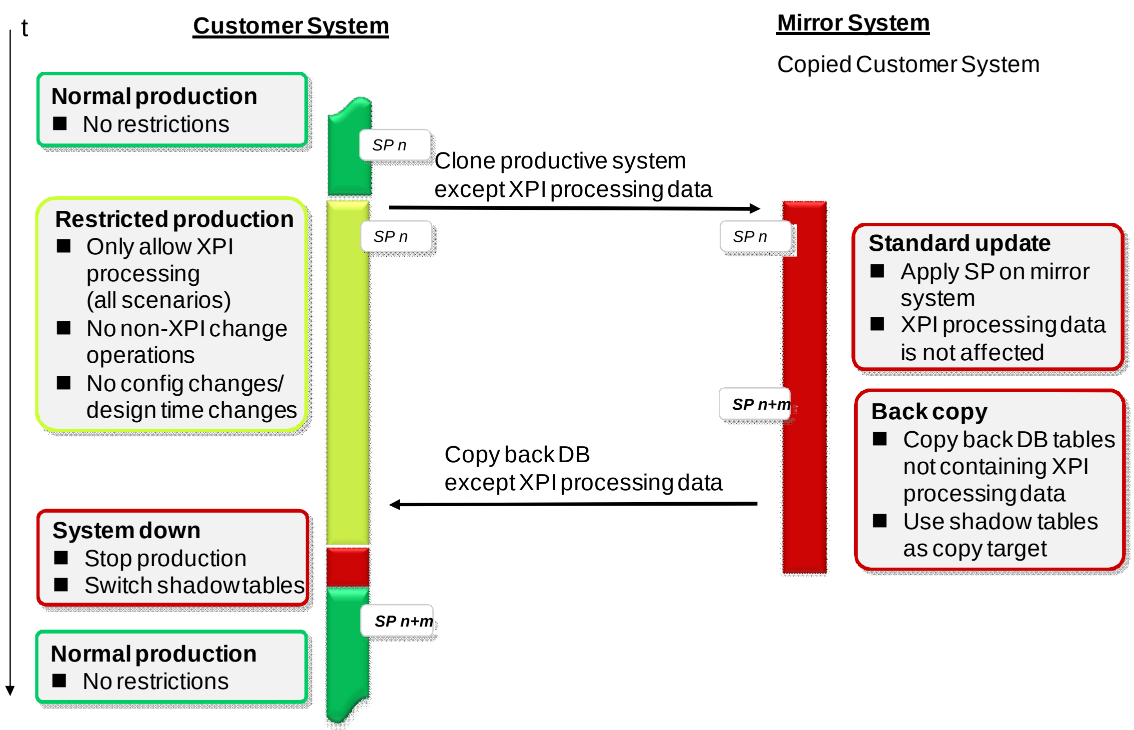

Fig. 10 Near Zero Downtime Maintenance for PI: Mirror update with back copy

After the Support Package(s) is/are updated on the mirror, the corresponding changes are copied back to production.

During the update process, certain restrictions to production apply. These include: configuration and design time changes, transports/deployments, and additional hardware that is required for the mirror system.

The overall duration of the procedure is not reduced, additional steps are needed compared to the classic SP procedure. The procedure occurs during piloting and is limited to SAP NW Release 7.1 and certain DB/OS combinations.

The customer efforts can be derived from the additional hardware needed along with additional disk space in the DB and file system. Named contacts for the preparation and update phase are estimated to take ~3-5 PDs per system.

The customer should run a scenario test on the QA system to prove the successful SP update, but implementing patches on the OS and DB might be necessary.

The technical requirements:

- NW Release 7.1 family
- Supported DB systems are Oracle 10 and MaxDB 7.7
- DB6 and MSS is planned for Q1/2011
- Supported OS systems are Linux, AIX, HP-UX, Solaris and Windows Server

Additional hardware requirements are:

- Separate update host of same OS system version as Clone (Copy) of PI system with local 100 GB disk space
- Temporary additional 30 GB space in PI database
- Temporary additional 20 GB local disk space on Clone (Copy) of PI system

The usage for production systems has to be confirmed by SAP.

7.2 Near-Zero Downtime Maintenance for Portal (Pilot)

The NZD for Portal allows the application of Support Package Stacks here on the Portal system with business downtime reduced to one system restart. This holds true only for updating the SAP software – OS or DB updates are not included. The update is performed on a mirror system while the production system remains available until the final switch to the new SP level is initiated by a restart. The main portal remains accessible all the time. Fig 11 depicts the general procedure of the method. The production remains fully functional while a copy of the production system is utilized for the Support Package update.

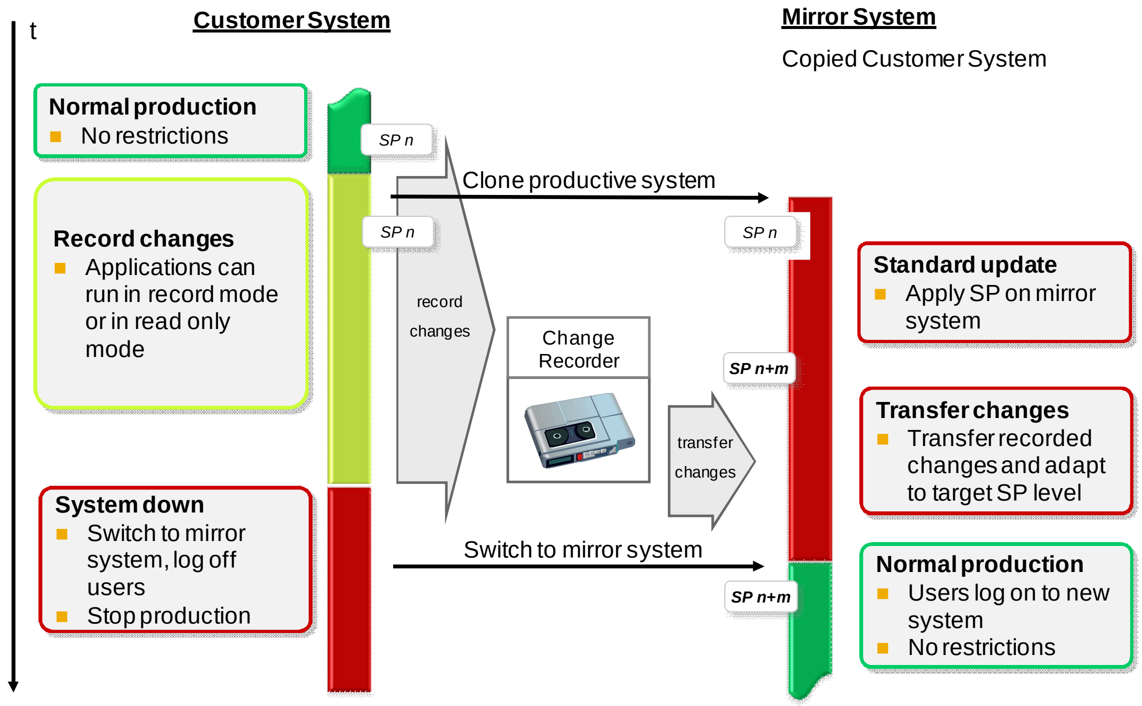

Fig. 11 Near Zero Downtime Maintenance for Portal: Mirror Update with Switchover

Copyright