Umstellung von CA AutoSys nach IWS

Jürgen Schimpke, Beta Systems DCI Software AG

Inhalt

1. Rahmenbedingungen
2. CA AutoSys auf einen Blick
3. Vorabanalysen
4. Umsetzung der Migration der Meta File Transfer (FT) Anwendung
5. Umsetzung der Migration der Data Warehouse (DW) Anwendung
6. Fazit

Kapitel 1: Rahmenbedingungen

Kapitel 1: Rahmenbedingungen -1-

Gründe für die Migration

- Eingebunden in ein großes Projekt der RZ-Verlagerung an einen anderen Standort
- Software Konsolidierung
- IWS als Scheduler und Beta92 sind gesetzt

Projektbeteiligte

- Projektleitung, Teilprojektleiter, externe Projektmitarbeiter
- Mitarbeiter der Anwendungen beim Kunden
- Mitarbeiter der zugehörigen Fachabteilungen
- Weitere Mitarbeiter aus verschiedenen Bereichen des Kunden
- Beta Systems Mitarbeiter

Kapitel 1: Rahmenbedingungen -2-

Mengengerüst

- Anwendung 1: „META FTP“ Anwendung mit circa 5.000 Jobs pro Stage (QS und P)
- Anwendung 2: Data Warehouse Anwendung mit circa 5.000 Jobs pro Stage (QS und P)
- die QS Umgebung soll ein Abbild der Produktion sein
- bis zu 150 taggleiche Änderungen an Datenbank und Plan möglich

Aufgaben

- für die FT Anwendung: Aufbau eines Konverters zur 1:1 Übernahme der Jobs ins IWS
- für DW: Aufbau eines eigenen Migrationsprogrammes zur Übernahme nach Neudesign der Anwendung

Kapitel 1: Rahmenbedingungen -3-

Vorgegebener Zeitrahmen

- Erstellung des Konvertierungsprogrammes für die FT Anwendung in Q4 2016
- Umstellung der FT Anwendung von Q1 2017 bis Ende Q3 2017
- Aufbau eines Release Managements in Q4 2018
- DW Anwendung von Q2 2017 bis Juni 201.
- Abschaltung des Rechners Ende Juni 2018

Weitere Bedingungen

- Nur 1 IWS zur Verfügung, kein Test IWS
- Skepsis bei Mitarbeitern und Anwendungsbetreuern
- Hoher Zeitdruck für die FT Anwendung, da Produktivnahme bevorstand
- Mitarbeiter zum damaligen Zeitpunkt mit dem Mainframe nicht vertraut

Kapitel 2: CA AutoSys auf einen Blick

Kapitel 2: CA AutoSys auf einen Blick -1-

CA AutoSys

- besitzt ein Repository, kennt aber keine Pläne.
- besitzt ein mehrstufiges Konzept von Boxes und Jobs - eine Box kann Boxes enthalten
- Eine Box enthält keine ausführbaren Kommandos.
- Bis zu 1.000 Jobs können sich in einer Box befinden.
- kennt keine Begrenzung auf 8 oder 16 Stellen für Boxes und Jobs
- unterstützt zwei Terminatoren.
  - Box Terminator: Wenn ein Job in Error geht oder terminiert wird („gekillt“), geht die ganze Box in den Status „Terminiert“.
  - Job Terminator: Wenn die Box, in der sich ein Job befindet, terminiert wird, muss der Job auch „gekillt“ werden.

Kapitel 2: CA AutoSys auf einen Blick -2-

CA AutoSys

- kennt u.a. folgende Status:
  - Activated, Running, Started
  - Success, Failure, Terminated
  - On Hold, On Ice
- Job Abhängigkeiten können definiert werden für Success, Failure, *Done* , *Not Running*
- Job Abhängigkeiten können in logischen Ausdrücken (Boolsche Algebra) aufgebaut werden
- Änderungen sind nur auf dem Repository möglich
- CA AutoSys benutzt zur Definition eine eigene strukturelle Sprache (JIL)
- man kann sowohl die gesamte als auch Teile der Anwendung über Filter im JIL Format exportieren

Kapitel 2: CA AutoSys auf einen Blick -3-

Für CA AutoSys gelten folgende Regeln:

- Ein Job darf nur in einer Box definiert sein.
- Ein Job kann in einer Box nur einmal laufen.
- Ein Job in einer Box kann nicht anlaufen, wenn die Box nicht angelaufen ist.
- Eine Box bekommt den Status „Success“ nur dann, wenn alle Jobs der Box beendet worden sind und den Status „Success“ haben. Eine Box geht in den Status „Failure“, wenn alle Jobs zu Ende gekommen sind und mindestens einer den Status „Failure“ hat.
- Eine Box geht nie zu Ende, wenn die Box nicht den Status „Success“ oder „Failure“ bekommt. Das bedeutet, dass diese Box auch nicht erneut zur Ausführung kommen kann.

Kapitel 2: CA AutoSys auf einen Blickv -4-

Beispiel JIL

```
/* ------ Test_Box ---------------- */
insert_job: test_Box job_type: b
owner: AV
permission: mx,me
date_conditions: 1
days_of_week: mo,tu,we,th,fr
start_times: "3:02"
description: "Test Box "
term_run_time: 120
```

```
/* ------ Test_Box_1 ------------ */
insert_job: test_box_1
job_type: cmd
box_name: test-Box
command: sleep 10
machine: unixa
```

```
/* ------ Test_Box_2 ------------- */
insert_job: test_box_2
job_type: cmd
box_name: test_Box
condition: success(test_box_1)
command: ls -l
machine: unixa
```

Kapitel 3: Vorabanalysen

Kapitel 3: Vorabanalysen -1-

Für beide Anwendungen mussten folgende Themen vorab analysiert bzw. durchgeführt werden:

- Export der JIL Statements für die entsprechende Anwendung und Analyse, welche Parameter finden sich in der JIL wieder
- Aufbau einer Crossreferenz zur Umsetzung der benötigten CA AutoSys Parameter in IWS Logik
- Wie gehen wir mit Parametern um, die sich nicht (einfach) umstellen lassen?
- Aufbau eines Konzeptes zum Abbau der Box in Box … Problematik in Applikationen und Jobs
- Aufteilen von Boxes, falls sich mehr als n Jobs in einer Box befinden
- Aufbau einer Namenskonvention für Jobnamen, Applikationsnamen, Ressourcen … pro Anwendung

Kapitel 3: Ergebnisse aus den Vorabanalysen

Folgende Entscheidungen wurden im Rahmen von Workshops und Absprachen getroffen:

- Ausbildung der Mitarbeiter in der z/OS Umgebung, die bisher nur in Unix und Windows aktiv waren
- Im Rahmen des Projektes erfolgte ein „training on the job“.
- Neudesign der Anwendung, um die Boxes in Boxes Problematik aufzulösen (DW).
- Entscheidung, dass für die FT Applikation nur ein Job in einer Applikation gelten soll.
- Aufteilen von Boxes mit mehr als 100 – 150 Jobs in kleinere Applikationen.
- Nutzen von Recovery Statements, um den Box Terminator umzusetzen.
- der Job Terminator wird nicht berücksichtigt, obwohl eine Lösung vorhanden gewesen wäre.
- nicht unterstützte Parameter wurden durch die Entwicklung als Beta92 EJM Parameter implementiert.

Kapitel 3: Ergebnisse aus den Vorabanalysen -2-

Folgende Software wurde im Rahmen der Migration eingesetzt:

- Scheduler IWS mit Beta92 EJM
- TDWC
- Beta92 und Dscovery WebEnabler
- BatchAD und BatchCP
- XINFO
- IWS Graph
- Es wird eine zu programmierende Konvertierungsroutine von JIL nach IWS und Beta 92 EJM erstellt
- Es wird eine zu programmierende ISPF Anwendung für die DW Anwendung erstellt

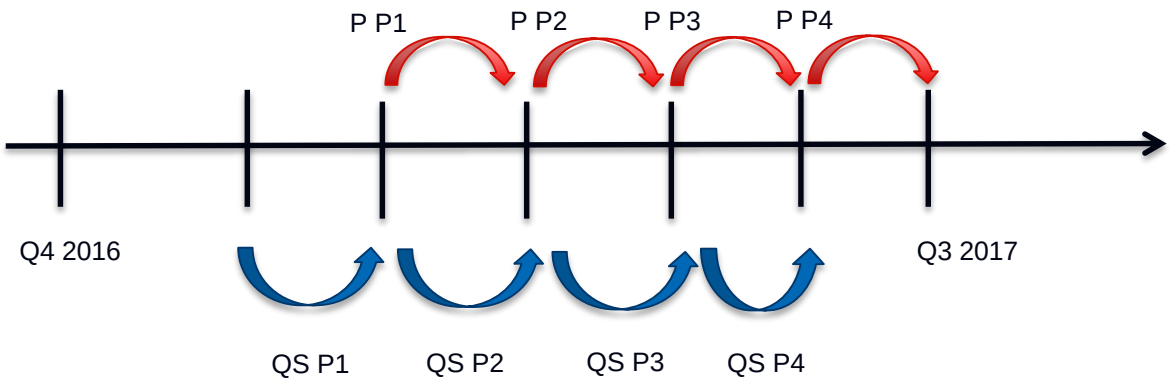
Kapitel 4: Umsetzung der Migration der FT Anwendung

Kapitel 4: Skizze des Projektverlaufs der FT Anwendung -1-

FT Anwendung

- Namenskonventionen absprechen
- Analyse der genutzten CA AutoSys Parameter und ihre Umsetzung in IWS und Beta92 EJM Spezifikationserstellung
- Erstellung Konvertierung
- Begleitung bei der ersten Übernahme

Kapitel 4: Skizze des Projektverlaufs der FT Anwendung -2-



Vorgehensmodell:

- In Q1 2017 begann die erste QS Phase für eine Teilmigration von Jobs nach IWS. Diese wurde dann abgeschlossen mit der Produktivnahme der entsprechenden Jobs aus der QS Phase. Parallel schloss sich dann die QS Phase 2 an.
- Mit jeder Produktivnahme einer Phase schloss sich die nächste QS Phase an. Mit der Produktivnahme der Phase 4 waren alle Jobs von CA AutoSys nach IWS und EJM migriert.
- Im Q4 2016 wurden die Konvertierungsprogramme entwickelt, welche die Jobs und Applikationen aus der JIL erstellt haben.

Kapitel 4: Release Management für FT Anwendung -1-

- Aus der Erfahrung mit der DW Applikation wurde ein Release Management Verfahren beauftragt, um die hohe Zahl der täglichen Anpassungen zu bewältigen.
- Die Basis sind ausgelieferte CSV Sätze der Fachabteilung, die in BatchAD Statements konvertiert werden.
- Um sicherzustellen, dass nur korrekte Daten in einem Release vorzufinden sind, wurde vorher die TMF Anwendung auf Unkorrektheiten analysiert. Diese Fehler müssen vor Einsatz des Release Managements behoben werden.
- Die Fachabteilung bekommt die Daten der FT Anwendung als CSV Satz zurückgemeldet.

Kapitel 4: Release Management für FT Anwendung -2-

```
Umgebung für die folgenden Optionen sehen Sie oben links. Change mit Option 0.

0  User Profile Variablen: Umgebung, Ausgabe-Datei-Benennung, etc.
1  Funktionen für Release Management. Generierung von Jobs und Applikationen
2  Funktionen für Release Management. Rücknahme fehlerhafte Updates
3  TMFXMIGA ausführen: Analyse bestehender Jobs und Apps und Modifikation
4  TMFRUECK-Job für Test erstellen: Ausgabe CSV-Daten aus TWS-AD
5  Restore der gesicherten Apps bei Problemen mit den geänderten Apps aus Opt.3
6  Restore der gesicherten Jobs bei Problemen mit den geänderten Jobs aus Opt.3

P  VIEW für Parameter-Member für alle Programme: Umgebung über Option 0.
A  Administrator-Funktionen: nur auf dem TSO von Kunde
E  Entwickler-Funktionen: nur auf dem TSO von Beta Systems
```

Kapitel 5: Umsetzung der Migration der DW Anwendung

Kapitel 5: Anforderungen -1-

Die Kollegen der DW Anwendung und der zugehörigen Fachabteilung stellten folgende Anforderungen im Rahmen von gemeinsam abgehaltenen Workshops und Requirements:

- Aufbau eines dialoggesteuerten Release Managements mit Austausch der gesamten Hauptbox
  - auf der Datenbank und im Plan
- Bei Fehlern im Release: Rücknahme der Änderungen und Setzen auf den vorhergehenden Zustand
- Aufbau eines dialoggesteuerten Verfahrens zum Einplanen von Ad Hoc Aufträgen aus Tickets und Changes
- Umsetzung des Box Terminators, der Status „On Hold“ und „On Ice“
- Dialoggesteuertes Löschen einer kompletten Hauptbox aus dem Tagesplan
- Dialoggesteuertes Hinzufügen einer kompletten Hauptbox in den Plan
- Aufbau eines Rückmelde Verfahrens der gesamten DW Anwendung an den Fachbereich als CSV Satz

Kapitel 5: Skizze des Projektverlauf der DW Anwendung -1-

Der eigentliche Projektverlauf lief wie nachfolgend beschrieben ab:

- Lange Testphase mit häufigem Löschen und erneutem Laden (Kunde, Beta Systems)
- Einarbeitung von Requirements (Beta Systems)
- Übernahme in Produktion und Begleitung (Beta Systems)

Das Projekt wurde im April 2017 begonnen und endete im Juni 2018.

Kapitel 5: Anforderungen -2-

- Aufbau eines Verfahrens zur Rückmeldung der Start und Endzeiten für alle gelaufenen Jobs bestimmter Hauptboxen
- Parameter gesteuerte Einführung eines automatisierten Überhangverfahrens falls eine Hauptbox nicht beendet ist:
  - plant sich der tagesaktuelle Ablauf selbst aus
  - wird der aktuelle Ablauf hinter abhängig auf den Vortag gemacht
- Verfahren zum Löschen der Applikationen einer Hauptbox und der zugehörigen Jobs aus der IWS Datenbank und der Job Bibliotheken
- Dialoggesteuerte (Massen)updates gegen den CP (MH, MR, SR, ...)
- Aufbau eines Aktionslogs zur Nachvollziehbarkeit der Änderungen durch Benutzer

Kapitel 5: Skizze des Projektverlauf der DW Anwendung -1-

Folgende Meilensteine wurden im Rahmen der Migration der DW Anwendung nach IWS / Beta92 EJM im Projekt erreicht:

- Absprache von Namenskonventionen (Kunde, Beta Systems)
- Neudesign der Anwendung für die Übernahme ins IWS (Kunde)
- Abstimmung zur Programmierung einer Schnittstelle, damit aus der Anforderung des Fachbereiches sowohl die Jobs als auch die BatchAD und BatchCP Statements erstellt werden können, um die Daten in die Datenbank des IWS und den Plan zu laden (Kunde, Beta Systems)
- Erstellung der Spezifikation (Beta Systems)
- Erstellung Migrationsprogramme (Beta Systems)

Kapitel 5: Skizze des Projektverlauf der DW Anwendung -2-

Der eigentliche Projektverlauf lief wie nachfolgend beschrieben ab:

- Lange Testphase mit häufigem Löschen und erneutem Laden (Kunde, Beta Systems)
- Einarbeitung von Requirements (Beta Systems)
- Übernahme in Produktion und Begleitung (Beta Systems)

Das Projekt wurde im April 2017 begonnen und endete im Juni 2018.

Kapitel 5: Generelle Umsetzung Primary Panel

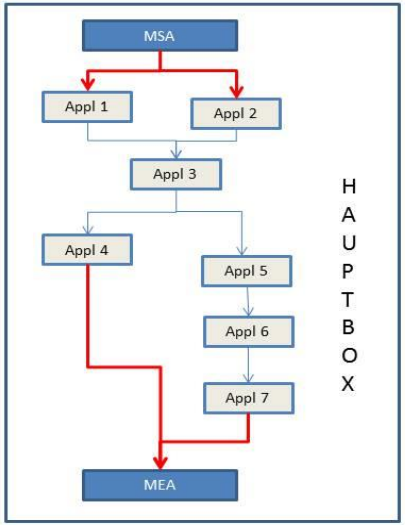
```
Umgebung Q   DWH Hauptauswahl
Command ===>

Umgebung für die folgenden Optionen sehen Sie oben links. Change mit Option 0.

0   User Profile Variablen: Umgebung, Ausgabe-Datei-Benennung, etc.      NT+NA
1   Funktionen für Release Management. Generierung Jobs.                 nur NT
2   Funktionen für Release Management. Rücknahme fehlerhafte Updates     nur NT
3   Box-Add (Detail-Angaben in Folge-Panel)                              nur NT
4   Box-Delete (ähnliches Verfahren für TWS Automatik Recovery)          nur NT
5   Box-Add/-Del: Falls "neue" Box aktiv war: Reset Oper-Status MH > MR  nur NT
6   Box-Add/-Del: NETZ_Ressource Available setzen nach CP-Kontrolle      nur NT
7   TWSKPA-Job für Test erstellen: Box-Angabe im Folgepanel.           nur Test
8   BSKRUECK-Job für Test erstellen                                    nur Test
9   NA-Verarbeitung generieren: schrittweise oder als Gesamtjob          nur NA
DNA Delete All, NA Objekte (AD,CP, Jobs): Auswahl im Folgepanel          nur NA
DNT Delete All, NT Objekte (AD, Jobs): Auswahl im Folgepanel             nur NT
RNT Restore NT Applikationen aus einem früheren Save: DSN im Folgepanel  nur NT
MH  Operationen im CP auf MH setzen (Manual Hold).                       NT+NA
MR  Operationen im CP wieder freigeben (Manual Release).                 NT+NA
SR  Special Ressource Status ändern: Available Y/N.        NT+NA
```

Kapitel 5: Umsetzung des Box Konzeptes

- Für jede Hauptbox wird immer eine Vorläufer (MSA) und eine Ende (MEA) Dummy Applikation erstellt.
- Alle Applikationen ohne Vorläufer werden abhängig gemacht auf die MSA, alle ohne Nachläufer auf die MEA.
- Für alle Applikationen wird eine DUMY_001, DUMY_002 und DUMY_255 Workstation hinzugefügt.
- Die DUMY_002 enthält als „extended field“ eine Beschreibung der Applikation und das Datum der letzten Änderung im Release.
- Externe Abhängigkeiten immer zwischen DUMY_255 auf DUMY_001.
- Alle Jobs ohne Abhängigkeiten werden immer abhängig auf DUMY_002, alle Applikationen ohne Nachläufer werden Vorläufer für DUMY_255.
- Operationsnummern werden aufsteigend in 1er Schritten vergeben. Durch Löschungen freigegebene Nummern werden beim nächsten Release wieder aufgefüllt.
- Planzeit für alle DW Applikationen ist 06.10 Uhr statt 06.00 Uhr.
- Die MSA vererbt die Runcycle Angaben an alle Applikationen der Hauptbox.



Kapitel 5: Release Management TAB

Applikationsblock

```
erstellen
Applikation
Kurzname:DIBSIP93NTDA1B05
Beschreibung: DWH_P_DAILY_1_XX_YYYY
```

Operationsblock

```
erstellen
Operation
Kurzname:EAA1FB05
Kurztext: KONSORTIAL_VEREIN_BCA_I
Langname:DWH_P_DAILY_1_XX_YYYY_KONSORTIAL_VEREIN_BCA_I_FW (Objekttyp: FW
Applikation:DIBSIP93NTDA1B05
Computer:CPU04711
Benutzer:cdi04711
Kommando:/sleep 10
Joblast:2
Wiederholungen:3
...
```

Abhängigkeitsblock

```
erstellen
Typ: intern
Applikation: DIBSIP93NTDA1B05
Job_Name:EAA1LB05
Vorlaeufer:EAA1FB05
```

Aus der TAB werden die Jobs und die Applikationen erstellt.

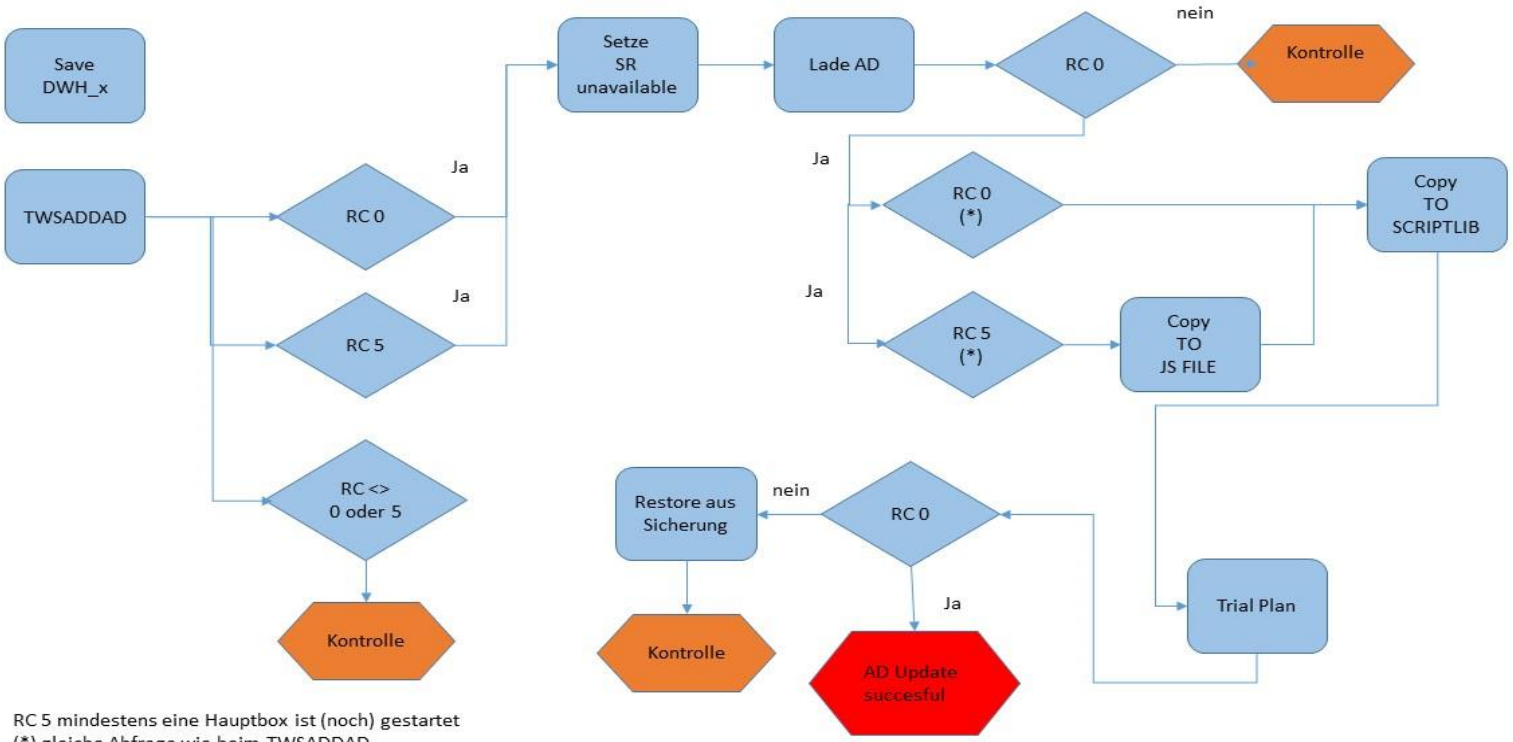
Im Release Management Prozess werden die Plausibilitätsprüfungen vorgenommen.

Es werden die BatchAD und BatchCP Statements für die Objekte erstellt und auch die Rücknahme-Dateien aufgebaut.

Das Release Management durchläuft die folgenden Aktionen:

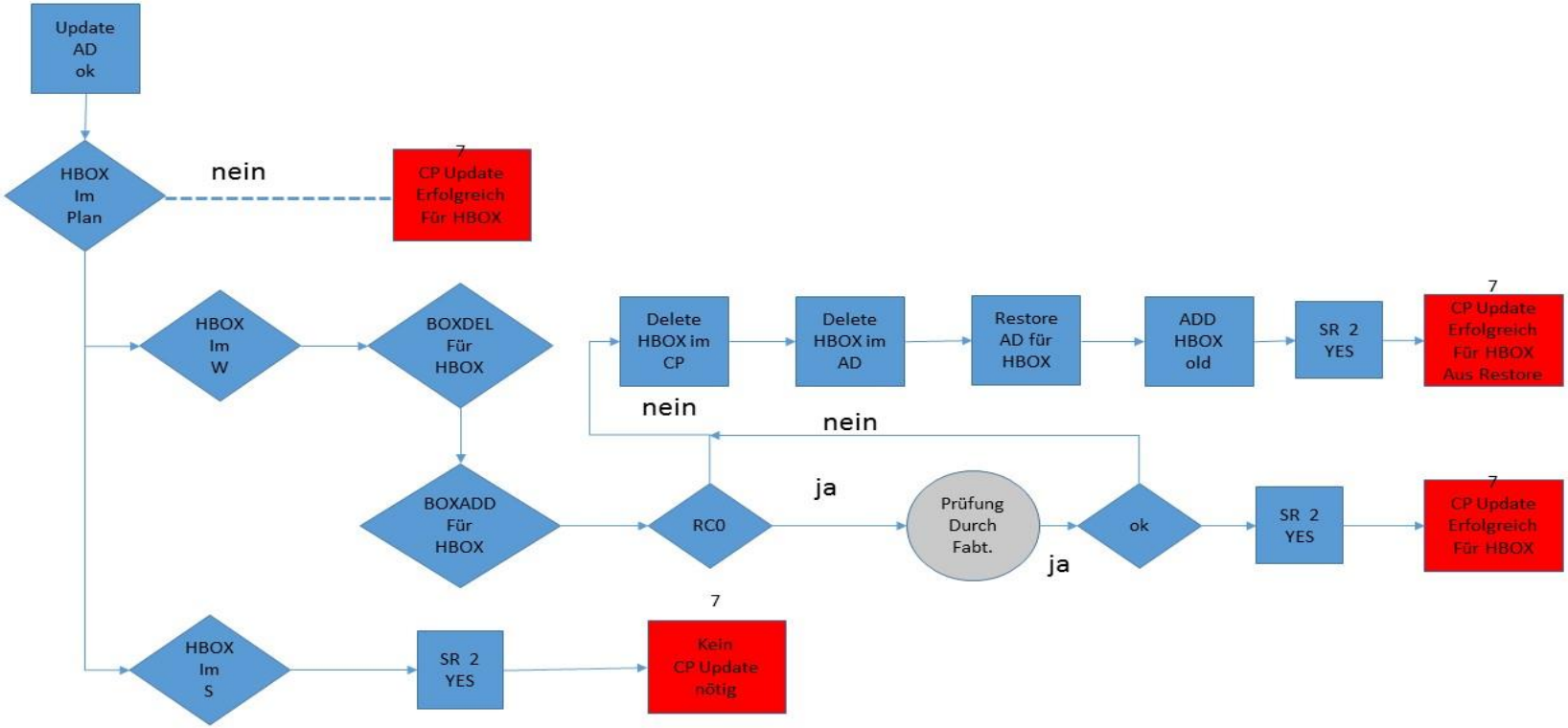
- Aufbau Batch AD Statements
- „Unavailable“ Setzen der Hauptbox SR
- Laden Batch AD Statements ins IWS
- Laden der JCL von zu löschenden Jobs ins JS File
- Alte Jobs löschen
- Neue Jobs kopieren
- Erstellen BatchCP Statements
- Ausführung CP Update
- „Available““ Setzen der Hauptbox SR

Kapitel 5: Release Management Teil 1 Update AD



RC 5 mindestens eine Hauptbox ist (noch) gestartet
(*) gleiche Abfrage wie beim TWSADDAD

Kapitel 5: Release Management Teil 2 Update CP



Kapitel 6: Fazit

Kapitel 6: Fazit

Positiv:

- Eine Übernahme von CA AutoSys nach IWS ist realisierbar.
- Die DW Übernahme lief sehr gut.
- Den Mitarbeitern macht das Arbeiten im z/OS Umfeld Spaß und sie haben Kenntnisse auf allen Plattformen.

Lessons Learned:

- Bei der Übernahme der FT Applikation holperte es etwas.
- Von der Strategie „1 Job pro Applikation“ ist abzuraten.

Neutral:

- Man benötigt ein Neudesign der Anwendungen wegen der Box in Box Problematik und der Begrenzung auf 255 Operationen im TWS.
- Es ist ausreichend Zeit für das Projekt einzuplanen.
- Die Trennung von Test und Produktion erfordert zwei IWS Instanzen.

Haben Sie noch Fragen ?

Vielen Dank für Ihre Aufmerksamkeit!