Db2 12 Continuous Delivery

–

New challenges for deployment

Roy Boxwell

SEGUS & SOFTWARE ENGINEERING

Session code: <V4>

Monday, October 2nd - 02:00 PM-03:00 PM   Platform: <Db2 z/OS>

Agenda

- Agile, Continuous Delivery, DevOps

- Db2 Code, Catalog, Function and Application Levels

- Activation/Deactivation of new code and how to fallback and when you can't

- Different flavors of (pro-active) CD-screening and how it can be automated
  - Anomaly alerting based on Incompatibility Change Indicators (ICIs)
  - Dyn./Stat. access path change detection e.g. via Plan Management
  - Clone based code change pre-apply exploiting Backup System
  - Workload-KPI verification using SQL replay and KPI comparison

- Conclusion

Traditional vs Agile software development

- Traditional software development is a sequential process:
  - requirement definition – careful and complete
  - planning – visualization of the architecture
  - building – coding
  - testing – careful and complete
  - deployment – delivery and roll out
    - → The key phases are the definition and planning, because it's a hierarchical process.

- Agile software development is a diversified
dynamic evolution:
  - adoption
  - coding
  - (constant) compatibility testing

3

Traditional vs Agile software development

- Both methodologies have their pro's and con's

- Traditional:
  - *Isolated*, serialized steps
  - Requires clear definition of requirements <u>in advance</u>
  - Less flexible when it comes to changes, fixes, enhancements
  - More solid with regards to quality
  - Stringent documentation and deployment

Traditional vs Agile software development

- Both methodologies have their pro's and con's
- Agile:
  - Less rigorous process
  - Smaller chunks, but highly interacting pieces
  - More (iterative) phases of revision and improvement
  - Requires a close collaboration between user and developer
  - Problems and improvements can be addressed easier and less disruptively
  - Time to market is usually shorter, but the overall effort may increase

Traditional vs Agile software development

- Both methodologies affect wide aspects of a business
  - User, Business Analyst and Developer have to collaborate differently depending on the chosen method
  - With Agile, the role of the developer changes due to the extensive interaction with the user
  - Agile is a more granular approach leading to more flexibility, but increases demands for interaction/compatibility of the chunks

- The decision is *either or*

  Either go for the traditional way of software development that deals with bulkier, unilateral systems with better ease and forgo the other advantages, or compartmentalize the large system into smaller agile developed processes and play a compatibility gamble.

IDUG
Leading the DB2 User
Community since 1988

IDUG DB2 EMEA Tech Conference
Lisbon Portugal | October 2017

#IDUGDB2

Continuous Delivery and DevOps

Agile development requires near-time delivery

- Continuous Delivery (CD) is an approach to produce software in short cycles
- CD ensures that changes can be released at any time, considering building, testing and releasing faster and more frequent
- Key is a focus on more incremental updates
- CD requires a straightforward and repeatable deployment

Of course an enterprise can not just roll out a new code drop, but developers, operations, quality assurance, management and so forth have to work closer together.

DevOps (software **DEV**elopment and information technology **OP**eration**S**) aims to establish a culture and environment where building, testing, and releasing software can happen rapidly, frequently, and more reliably.

Agile, CD and our role in DevOPs

- Considering that Agile requires a close collaboration between development and the user, we are likely to be going closer to the labs – or are we already? Think about *IBMs vote for enhancements*, or *Sponsor Users.*

- Considering that Agile doesn't produce new versions, but an ongoing delivery stream, we are likely going to get new features, functions and fixes much quicker.

- Considering that Agile requires a total change in software development, we'll likely see "radical internal changes" at IBM

- Considering that Agile and CD requires a
different deployment methodology as well,
we'll very likely have to change our deployment.

Db2 levels

- Starting with Db2 12 GA, IBM is going forward to the CD model

- Only single phase migrations, no more CM, ENFM, NFM

- Enhancements may or may not require CATMAINT

- A new level is identified by
  - Version ID
  - Release number
  - Modification identifier

- New Enhancements and capabilities delivered as part of the maintenance stream as Function levels

- Function levels enabled via –ACTIVATE command

New!

Db2 levels

- Db2 customers will see a significantly high volume of CD items in a single maintenance stream

- Entirely new releases, or versions will be very rare and only for major changes, like
  - Extending control structures
  - Enabling an architecture level set
  - Adopting a new compiler

- Recommended Service Upgrades (RSUs) are still available on a monthly/quarterly interval and may, or may not include new features

- Although a code drop may include fixes as well as new/changed features and functions, Db2 users keep control over what they activate – really?!?

Db2 levels

To be able to separate fixes from new/changed features, applying maintenance now consists of up to four different, hierarchical levels:

1. **Db2 Code level** – [Also known as a Maintenance level]
   Db2 libraries via SMP/E. "Most features are disabled"

2. **Db2 Catalog level** – Db2 Catalog changes via CATMAINT

3. **Db2 Function level** – general available functions
   via Db2 command (-ACTIVATE FUNCTION LEVEL)
   • Function levels may be skipped

4. **Application level** – individual application functions
   via Db2 BIND/REBIND (default via ZPARM: SQLLEVEL
   APPLCOMPAT, or BIND with APPLCOMPAT VvvRrMmmm)

11

Db2 levels

Levels available for use can be set in a few different ways:

- SQLLEVEL – DSNHDECP for precompile or coprocessor (no more NEWFUN)
- SET CURRENT APPLCOMPAT may be used for prior levels
- Client property – clientApplCompat may be used for prior levels

Db2 levels

Db2 comes with a variety of sources to determine the levels

- The new Db2 Catalog table SYSIBM.SYSLEVELUPDATES maintains a history of the different levels
    - FUNCTION_LVL
    - PREV_FUNCTION_LVL
    - HIGH_FUNCTION_LVL
    - CATALOG_ LVL

- ...and additional columns:
    - OPERATION_TYPE
        - "C" for catalog change – CATMAINT
        - "F" for function level – ACTIVATE
        - "M" for maintenance update
    - Timestamps and LRSN/RBA

Db2 levels

Db2 comes with a variety of sources to determine the levels

- Additional Catalog tables have a new column FUNCTION_LVL

  - SYSIBM.SYSDYNQRY
  - SYSIBM.SYSPACKCOPY
  - SYSIBM.SYSPACKAGE
  - SYSIBM.SYSQUERY

Db2 levels

Db2 comes with a variety of sources to determine the levels

- New Global Variables deliver further insights into the current levels
    - SYSIBM.PRODUCTID_EXT: pppvvrrmmm
    - SYSIBM.CATALOG_LEVEL: VvvRrMmmm
    - SYSIBM.DEFAULT_SQLLEVEL (DECPSQLL): **VvvRr**Mmmm
- Db2 Command -DISPLAY GROUP shows the different levels
    - CATALOG LEVEL
    - CURRENT FUNCTION LEVEL
    - HIGHEST ACTIVATED FUNCTION LEVEL
    - HIGHEST POSSIBILE FUNCTION LEVEL
- Db2 Command -DISPLAY LOCATION shows the Function Level of each connected system

Db2 levels

Db2 comes with a variety of sources to determine the levels

- RIBRELX – available via multiple interfaces after an applications "CONNECT"
  - The format was extended from VV.R.M to VV.R.MMM
- SQLCA and GET DIAGNOSTIC do not include the levels
  - SQLERRP: pppvvrrm : Note the missing mm
- CLI comes with a new infotype
  - SQLGetInfo API returns SQL_DBMS_FUNCTIONLVL showing the function level/buildlevel
- JCC introduces
  - getDatabaseFunctionalLevel() API driver returning the function level/buildlevel
- DDF Location Statistics shows the FL of connected system
- DDF accounting header show the FL of connected system

Activation of levels

1. SMP/E apply of PTFs, or RSUs may increase the Db2 Code level [ML]

    – on the member level of a DS group

    → Verify the system to be stable

2. CATMAINT (if needed) increases Db2 Catalog level

    → Code level fallback is not possible after CATMAINT

3. ACTIVATE FUNCTION LEVEL (if needed/available) increases the Db2 Function Level

    – on the group level of a DS group

    → New functions/behavior that aren't related to SQL DML, DDL and DCL available

    → REBINDs pick up Db2 Optimizer enhancements/changes

    → Dynamic SQL (non-stabilized) affected accordingly

4. BIND/REBIND APPLCOMPAT and/or ZPARM APPLCOMPAT (if needed/available)

    increases the Db2 Function Level of an individual package

    → New functions/behavior related to SQL DML, DDL and DCL available

    → SET CURRENT APPLCOMPAT may be used for prior levels

    → clientApplCompat may be used for prior levels

Deactivation of levels

- Db2 Code level fallback is only possible when no CATMAINT has been executed and the Function level is still supported

- Db2 Catalog level fallback is impossible

- FUNCTION LEVEL fallback is possible via ACTIVATE FUNCTION LEVEL, but
  - the Db2 Code level has to remain on the highest level

- BIND/REBIND APPLCOMPAT and/or ZPARM APPLCOMPAT can be used for Db2 Function level fallback of an individual package
  → old functions/behavior related to SQL DML, DDL and DCL

- SET CURRENT APPLCOMPAT may be used for prior levels

- clientApplCompat may be used for prior levels

- Unless you REBIND, packages bound with higher Function levels remain functional…

18

Skipping levels

Db2 Function levels may be skipped, but always include all new/changed features/behavior of prior Function levels

CD Screening

- Now – is there a problem with Continuous Delivery?
- It depends … it's a matter of control & verification
  - → Thus assure proper testing

APPLCOMPAT results before continuous delivery:

- The way some of the most popular scalar BiFs worked started changing with Db2 10 for z/OS. For example, CHAR:

```
SELECT CHAR(DEC_COL) FROM SYSIBM.SYSDUMMY1;
```

	CHAR(000.1)	CHAR(1000.)	CHAR(1.1)
V10+	'.1'	'1000'	'1.1'
V9	' 000.1'	' 1000.'	' 1.1'

CD-Screening

- Incompatible changes as of Db2 11:
- 1: V9 version of CHAR(DEC) executed
- 2: V9 version of VARCHAR(DEC) executed
- 3: Unsupported character string representation of a TIMESTAMP
- 4: V10 default SQL path used instead of V11
- 7: SQLCODE -301 from a Db2 11 server
- 8: Stored procedure data types
- 9: TIMESTAMP TIMEZONE from DRDA
- 10: V9 version of LTRIM, RTRIM, STRIP executed
- 11: SELECT INTO with UNION (-109 problem)
- 1101: INSERT into XML column w/o XMLDOCUMENT function
- 1102: XPATH evaluation resulted in error
- 1103: Dynamic SQL ASUTIME limit RLF issue

- 1104: CLIENT_ACCTNG longer than supported length pre-V11
- 1105: CLIENT_APPLNAME longer than supported length pre-V11
- 1106: CLIENT_USERID longer than supported length pre-V11
- 1107: CLIENT_WORKSTNNAME longer than supported length pre-V11
- 1108: CLIENT register longer than supported used for RLF
- 1109: CAST(STRING AS TIMESTAMP) using invalid string lengths as of V11
- 1110: Argument for SPACE function greater than 32764
- 1111: Optional integer argument of VARCHAR greater than 32764
- 1112: Empty XML element

21

CD-Screening

… and the first one announced for 12:

- 1201: changed SQLCODE for POWER results out of range

- Out-of-range result from POWER built-in function with DOUBLE arguments in Db2 12

  - *Starting in Db2 12, when the POWER built-in function is invoked with DOUBLE arguments, and the result is out of range, warning SQLCODE +802 is returned, instead of error SQLCODE -802*

  - *Actions to take:*
    In Db2 12, before function level 500 or higher is activated, identify applications with this incompatibility by starting a trace for IFCID 0376, and then running the applications. Review the trace output for incompatible changes with the identifier 1201. Adjust error processing to account for the change in the returned SQLCODE from an error to a warning

22

CD-Screening

The variations of levels and their dependencies/restrictions demand for an adjusted maintenance/migration

- We need to be aware of applications being affected by changed behavior
  - Since Db2 10 we can use IFCID 366 and from Db2 11 IFCID 376 to catch potential incompatibilities (Incompatibility Change Indicator – ICI)

- We need to be able to prescreen access path changes resulting from code changes
  - Since Db2 9 we can use PLAN MANAGEMENT to fallback to prior runtime structures
  - Since Db2 12 we have stabilized dynamic queries to freeze runtime structures
  - Since Db2 collections we have the opportunity to REBIND (distributed) packages into a separate collection to compare and switch

CD-Screening

The variations of levels and their dependencies/restrictions demand for an adjusted maintenance/migration strategy

- We need a reliable, solid and automated way to test the different levels
- The checks can't take longer than a couple of hours, or a few days, otherwise we'll never catch up
- We need to be flexible in choosing the scope of testing
  - ICI Detection
  - Access Path Precheck and Protection
  - Instant Cloning
  - Workload Capture/Replay

CDDC for Db2 z/OS

SOFTWARE ENGINEERING/SEGUS provides a powerful package that exploits years of experience in QA, AP Precheck, Cloning and Db2 Workload capturing

Continuous Delivery Deployment Check for Db2 z/OS

provides four levels of automated alerting and testing:

- ICI based anomaly alerting
- Static & Dynamic SQL Access Path Pre- and/or Post-Check
- Clone based code level checks exploiting Instant Copy (e.g. Flashcopy)
- Workload Capture/Replay with KPI verification

CDDC for Db2 z/OS let's you decide the scope of testing, from incompatibility alerting up to a full system test including workload execution

CD-Screening

ICI Detection - ICI based anomaly alerting:

REBIND application static packages with higher APPLCOMPAT to exploit DDL/DML new functions/behaviors

→ ICI Detection covers Function Level changes and pinpoints the origin of a statement (stmt id, program, package, collection, statement text)

- Incompatibility Change Indicators play a major role in Function/Application level handling
- Db2 itself tells us via IFCID 376 if potential incompatibilities exist, e.g.
  - Changes to the behavior of Built In Functions (BIFs)
  - Changes of SQLCODEs
  - Changes of SQL results
  - Deprecated support e.g. for TIMESTAMP formats
  - (...)

CD-Screening

ICI Detection - ICI based anomaly alerting :

BIF/ICI Detection-2017-03-02 16:14:17.846

DB2 for z/OS 1

Package	Reason	WLX DB2 SSI...	Statement ID	Section number ...	Plan name	Statement Timestamp	Transaction name	End User ID	Workstation name
IQADBACP	DB2 9 VARCHAR Usage	DC10	180,221	114	IQAP0610	2016-08-30-10.15.04.581836			
IQADBACP	DB2 9 VARCHAR Usage	DC10	2,361	11	IQAP0610	2016-10-11-10.10.44.348841			
IQADBACP	DB2 9 VARCHAR Usage	DC10	180,221	114	IQAP0610	2016-08-30-10.15.04.581836			
IQADBACP	DB2 9 VARCHAR Usage	DC10	180,221	114	IQAP0610	2016-10-11-10.12.59.837638			
IQADBACP	DB2 9 VARCHAR Usage	DC10	180,722	114	IQAP0610	0001-01-01-00.00.00.000000	UNKNOWN	UNKNOWN	UNKNOWN
IQADBACP	DB2 9 VARCHAR Usage	DC10	1,260	11	IQAP0610	2017-01-25-09.42.51.340956	WLXPRVST	HOPPE	DB2CALL
SYSLH200	DB2 9 VARCHAR Usage	DC10	310	23	DISTSERV	2016-07-20-11.00.42.585936	INSERTRS	CHRISTO	DB2CALL
SYSLH200	DB2 9 VARCHAR Usage	DC10	310	9	DISTSERV	2016-07-20-11.00.42.585936	INSERTRS	CHRISTO	DB2CALL
SYSLH200	DB2 9 VARCHAR Usage	DC10	13,235	15	DISTSERV	2016-07-29-09.03.54.863931	EXECDDL	BOXWELL	DB2CALL
SYSLH200	DB2 9 VARCHAR Usage	DC10	13,239	18	DISTSERV	2016-07-29-09.04.15.676355	EXECDDL	BOXWELL	DB2CALL
SYSLH200	DB2 9 VARCHAR Usage	DC10	13,256	21	DISTSERV	2016-07-29-09.06.42.988871	EXECDDL	BOXWELL	DB2CALL
SYSLH200	DB2 9 VARCHAR Usage	DC10	13,262	36	DISTSERV	2016-07-29-09.12.39.004960	EXECDDL	BOXWELL	DB2CALL
SYSLH200	DB2 9 VARCHAR Usage	DC10	232	6	DISTSERV	2016-09-14-16.11.51.858847	DC10RS00	BOXWELL	UTILITY
SYSLH200	DB2 9 VARCHAR Usage	DC10	267	16	DISTSERV	2016-09-20-09.53.32.287974	DC10RS00	BOXWELL	UTILITY
SYSLH200	DB2 9 VARCHAR Usage	DC10	1,215	15	DISTSERV	2016-08-31-16.47.40.688532			
SYSLH200	DB2 9 VARCHAR Usage	DC10	1,215	15	DISTSERV	2016-09-29-13.09.25.096878	DC10RS00	BOXWELL	UTILITY
SYSLH200	DB2 9 VARCHAR Usage	DC10	2,321	176	DISTSERV	2016-10-10-09.32.49.278291			

CD-Screening

Access Path Check - Static & Dynamic SQL Access Path Pre- and/or Post-Check:

REBIND of packages with any APPLCOMPAT would pick up optimizer enhancements. Non-stabilized dynamic SQL would pick up optimizer enhancements.

→Access Path check covers Function Level Changes for both static AND dynamic SQL.

→Access Path check runs proactive (instead of PLANMGMT) and can be executed in your production environment (without harming applications), or in a virtualized test environment (simulates all optimizer relevant factors.)

→Access Path check clearly categorizes improvements and degradations (instead of PLAN STABILITY).

→ Access Path check opens up improvements and suppresses degradations.

CD-Screening

Access Path Check - Static & Dynamic SQL Access Path Pre- and/or Post-Check:

CD-Screening

Access Path Check - Static & Dynamic SQL
Access Path Pre- and/or Post-Check:

STMT No.	Section Number	DSC STMT ID	Impact	Serviceunits Old	Serviceunits New	Millisec			
2547	13	2547	DEGRADED	85	1				
796	2	796	DEGRADED	152	1				
804	3	804	DEGRADED	152	1				
812	4	812	DEGRADED	152	1				
820	5	820	DEGRADED	152	1				
671	2	671	DEGRADED	152	1				
679	3	679	DEGRADED	152	1				
687	4	687	DEGRADED	152	1				
695	5	695	DEGRADED	152	1				
3	3	3	DEGRADED	130	1				
4	4	4	DEGRADED	95	2				
5	5	5	DEGRADED	82	1				
6	6	6	DEGRADED	149	1				
968	7	968	IMPROVED	171	16	60	6	8	5
1167	9	1167	IMPROVED	174	10	62	4	10	5
235	1	235	IMPROVED	178	10	63	4	8	6
1194	2	1194	IMPROVED	150	191	53	67	12	1
2409	7	2409	IMPROVED	21	23	8	9	5	1
1194	2	1194	IMPROVED	175	191	62	67	12	1

Summary

REBIND Analysis

Error progs: 0
Worsened AP progs: 14
Improved AP progs: 5
Changed AP progs: 6
Unchanged AP progs: 31

● Unchanged AP progs ● Changed AP progs ● Improved AP progs ● Worsened AP progs ● Error progs
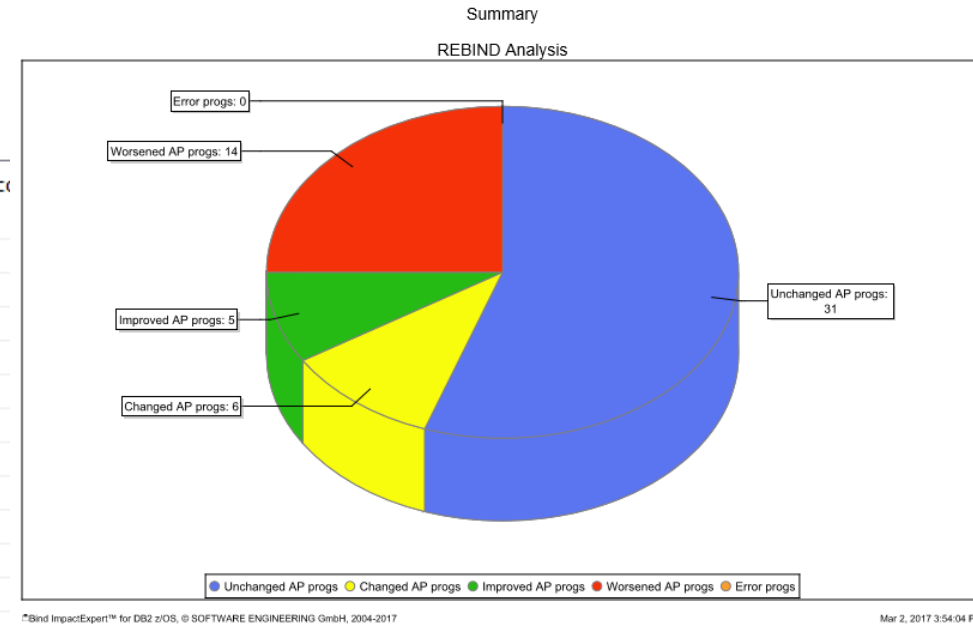
□Bind ImpactExpert™ for DB2 z/OS, © SOFTWARE ENGINEERING GmbH, 2004-2017   Mar 2, 2017 3:54:04 PM

IDUG

Leading the DB2 User
Community since 1988

IDUG DB2 EMEA Tech Conference
Lisbon Portugal | October 2017

#IDUGDB2

CD-Screening

Instant Cloning - Clone based code level checks:

PTFs (RSUs…) are applied that may increase the Code Level (CL) of a Db2 system. After system is stable on maintenance, execute (If Any) catmaint.

→ Instant Cloning covers Code Level, Catalog Level and Function Level testing and allows the most comprehensive scope.

- For a full check, an entire Db2 system can be cloned in minutes. Object level cloning allows being more granular.
- FTP support transfers required data to different LPARs.
- Scheduling Environment and Routing support handles system setups with different environments.
- The entire process is guided and controlled.

CD-Screening

- Instant Cloning - Clone based code level checks:

- Scope of Cloning: Subsystem level
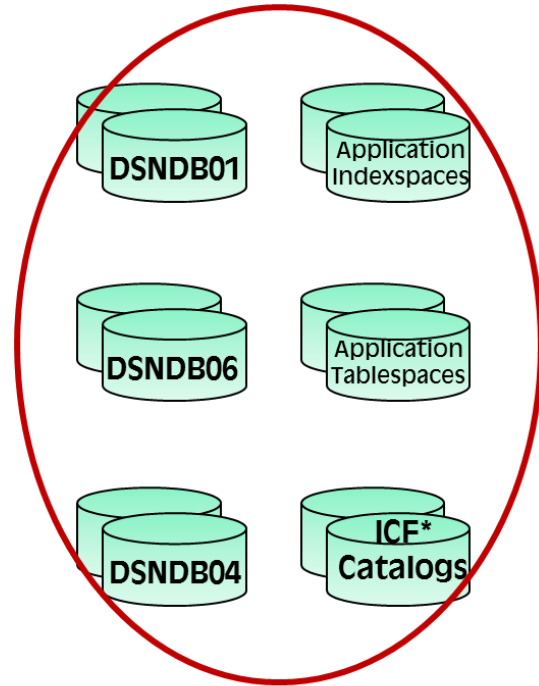
- Fully exploiting instant copy technology (e.g. Flashcopy)

- Supports DS$\leftarrow\rightarrow$NDS, as well as cross-version cloning

- Highly customizable and fully automated via XML scenario scheme
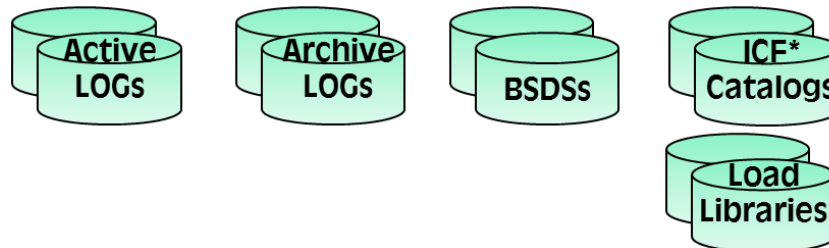
CD-Screening

- Instant Cloning - Clone based code level checks:
- Scope of Cloning: Object level

Application Indexspaces

Application Indexspaces

Application Tablespaces

Application Tablespaces

-Tables   -Sequences
-Synonyms   -Databases
-Views   -Trusted
-Aliases   Context
-Triggers   -Distinct
-Grants   Data
-Stogroups   -MQT
-Permissions   -Masks
-RI   -Roles
-Routines   -Variables
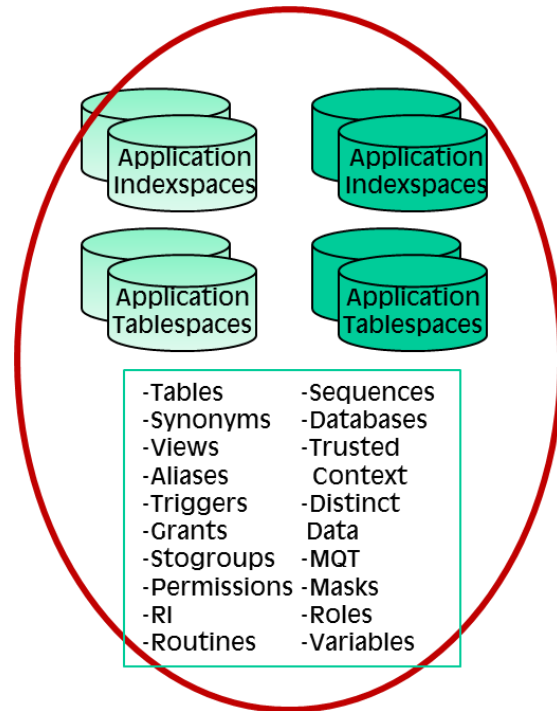
- Fully exploiting instant copy technology (e.g. Flashcopy)

- Supports DDL and/or data cloning

- Flexible include/exclude of dependent objects

- Powerful renaming capabilities

CD-Screening

Instant Cloning - Clone based code level checks:

CD-Screening

Workload Capture/Replay with KPI verification:

After Function Level is considered stable – allow new application feature rollout.

→ Workload Capture/Replay adds Application Level testing and completes the most comprehensive scope by executing sets of captured workload.

- Highly efficient IFCID (OPx) based capturing technology catches SQL for automated execution in the isolated cloned environment.
- Workload sets can be saved to represent quarters end, years end and other specific workload patterns.
- Tested workload is compared on a KPI level (e.g. # of getpages, rows returned, rows processed…) to report anomalies only.

CD-Screening

Workload Capture/Replay with KPI verification:
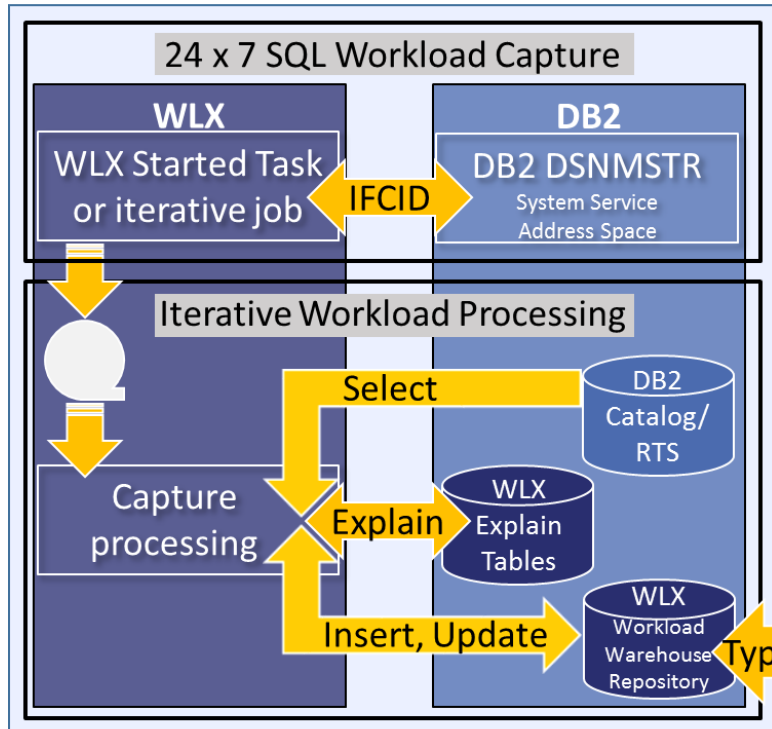
- Workload sets stored in a Workload Warehouse Repository
- Pre-filtered to remove any unwanted/unsupported SQL
- Transferred to the clone environment
- Executed and KPI compared to automatically find anomalies

CD-Screening

Workload Capture/Replay with KPI verification:

Selection 1

Primary Authorization ID	Package	Collection ID	Number of Statements	Sum of CPU Time	A
n/a	CMEMAPP	PTFCOLL008	1	0	
n/a	COISEAR	PTFCOLL008	1	4	
n/a	COQAPTF	PTFCOLL008	1	0	
n/a	DSN§EP2L	DSNTEP2	3	0	
n/a	DSNREXX	DSNREXX	1	0	
n/a	DSNTIAP	DSNTIAP	2	0	
n/a	FILLPROD	PTFCOLL008	1	0	
n/a	IADMPTF	PTFCOLL008	1	0	
n/a	IMEMB	PTFCOLL008	1	0	

Selection 2

Primary Authorization ID	Package	Collection ID	Number of Statements	Sum of CPU Time	A
n/a	CMEMAPP	PTFCOLL008	1	0	
n/a	COISEAR	PTFCOLL008	1	4	
n/a	COQAPTF	PTFCOLL008	1	0	
n/a	DSN§EP2L	DSNTEP2	3	0	
n/a	DSNREXX	DSNREXX	1	0	
n/a	DSNTIAP	DSNTIAP	2	0	
n/a	FILLPROD	PTFCOLL008	1	0	
n/a	IADMPTF	PTFCOLL008	1	0	
n/a	IMEMB	PTFCOLL008	1	0	

Selection details

Column name	Selection value 1	Selection value 2
Highest CPU Time	0.062996	0.062996
Sum of Elapsed Time	0	0
Average Elapsed Time	0.002149	0.002149
Highest Elapsed Time	0.079543	0.079543
Sum of Executions	37	37
Sum of GETPAGES	74	74
Average GETPAGES	2	2
Highest GETPAGES	74	74
Sum of Synchronous Buffer Reads	2	2
Sum of Synchronous Buffer Writes	0	0

Value 1   Value 2

CD-Screening

All tests can be combined and included in an automated execution stream to make the entire testing of a new Db2 code drop a standardized process.

A typical setup is for example:

- Have ICI Detection continuously govern the production system (overhead minimal)

- Make Access Path Check a vital part of REBIND processing as well as DSC check to keep control of dynamic SQL

- Before PTF/RSU apply:
    1. Clone into an isolated environment for CD-screening, run Workload Replay
    2. Apply the PTF/RSU and if needed CATMAINT
    3. Run Workload Replay
    4. Check system behavior and KPI report for anomalies

38

CD-Screening
The scope and the environment can be chosen flexibly

	Code Level	Catalog Level	Function Level	Application Level
ICI Detection	X	X	Test and/or Production	Test and/or Production
AP Check	X	X	Test* and/or Production	Test* and/or Production
Instant Cloning	Test	Test	Test	Test
Capture/Replay	Test	Test	Test	Test

Conclusion

- Compared to prior versions the path into the Agile world and their influence on existing behavior/compatibility started already in Db2 10 with BIF Compatibility

- APPLCOMPAT was first introduced in Db2 11

- The traditional three yearly release cycle doesn't fit to the technological development these days

- Retrofitting enhancements burns too many resources

- The idea of Function Levels should make the continuous delivery stream manageable

Conclusion

IBM is investing in new functionality with the key variable TIME, we must invest in TEST if we want to keep our level of stability.
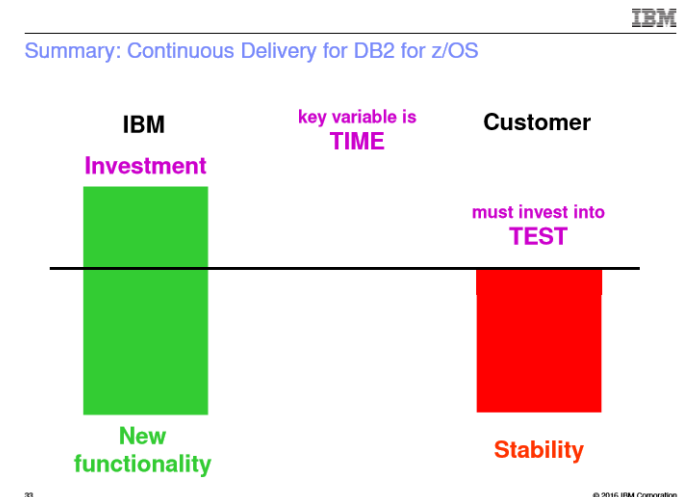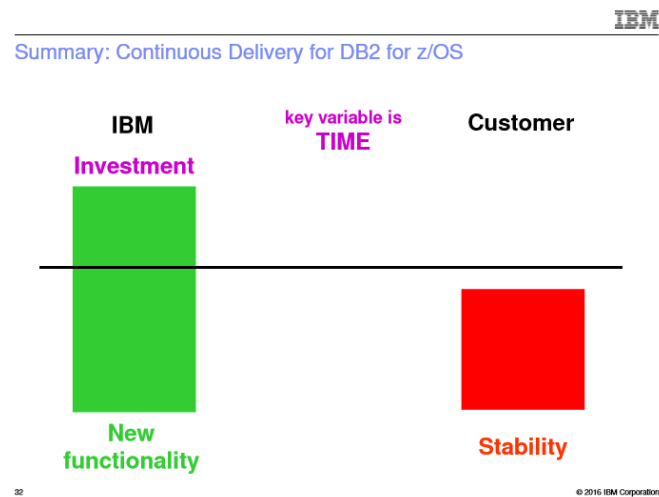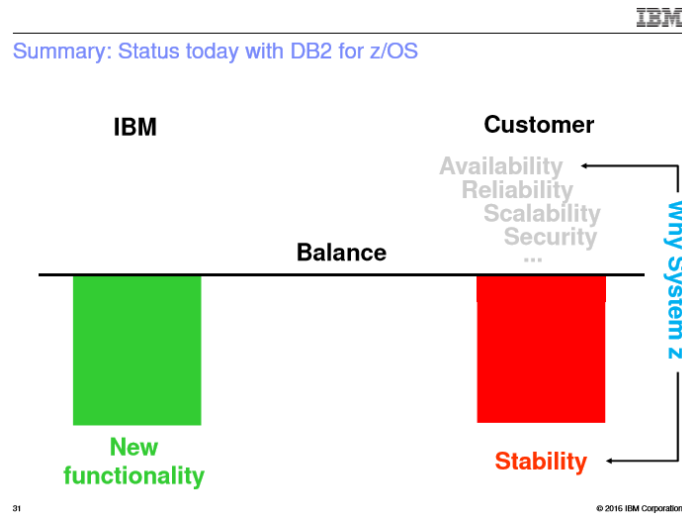
- Typically customers spent months to test new code, but that wont fit the continuous delivery stream.

- A multilevel, highly automated test framework is a reliable, robust solution to exploit and benefit from the new directions.

Recommendations

Understand Continuous Delivery

- Get familiar with the various levels and what they affect
- Verify and adjust your migration procedures
- Check the changes that affect a migration
- Resolve application incompatibilities before setting APPLCOMPAT to a new level
- Use the TEST option before activating a new level

Further information:

- Redbook: IBM Db2 12 for z/OS Technical Overview
- IBM Knowledge Center

Conclusion

Continuous Delivery Deployment Check for Db2 z/OS

- Comes with multicomponent quality assurance

- Depending on the importance of the system to maintain, tests can be extended, or reduced

- A focus on automation helps to test complex Db2 applications

- Exploiting efficient storage technology keep costs low

- Automated cloning scenarios clone an entire production system in minutes, ready for testing

- Workload Capture/Replay combined with KPI comparison assures that changes in Db2 behavior are detected quickly and efficiently

Freeware

BIF/ICI HealthCheck for Db2 z/OS

- A handy batch job, or started task that reports BIF incompatibilities in Db2 10, Db2 11, or Db2 12

- Both IFCID 366 and 376 can be captured as long as desired

- At the end of the capture process, or during -if desired- a report shows any affected dynamic, or static statement along with it's
  - PLAN
  - COLLECTION
  - PACKAGE
  - Number of occurrences (366)

Freeware

BIF/ICI HealthCheck for Db2 z/OS

```
SQL WorkloadExpert for Db2 z/OS Version 2.2   2015-09-24 at 15:15:58  Page    1
(C) 2013-2015 - SOFTWARE ENGINEERING GmbH


BIF Health Report for group GSA12A13 started at 2015-09-24 15:15:58.48
Type Plan      Collection          Package   ICI Count
Dyn   DSNESPUR DSNESPUR            DSNESM68   1    5
Dyn   DSNESPUR DSNESPUR            DSNESM68   5    4
Stat BIFTEST   BIFTEST_TEST        O2TESTB    1    2


SQL WorkloadExpert for Db2 z/OS Version 2.2   2015-09-25 at 10:48:17  Page    1
(C) 2013-2015 - SOFTWARE ENGINEERING GmbH


BIF Health Report for subsystem DB10 started at 2015-09-25 10:48:17.23
Type Plan      Collection          Package   ICI Count
Dyn   DSNESPUR DSNESPUR            DSNESM68   1    2
Stat BIFTEST   BIFTEST_TEST        O2TESTB    1    2
```
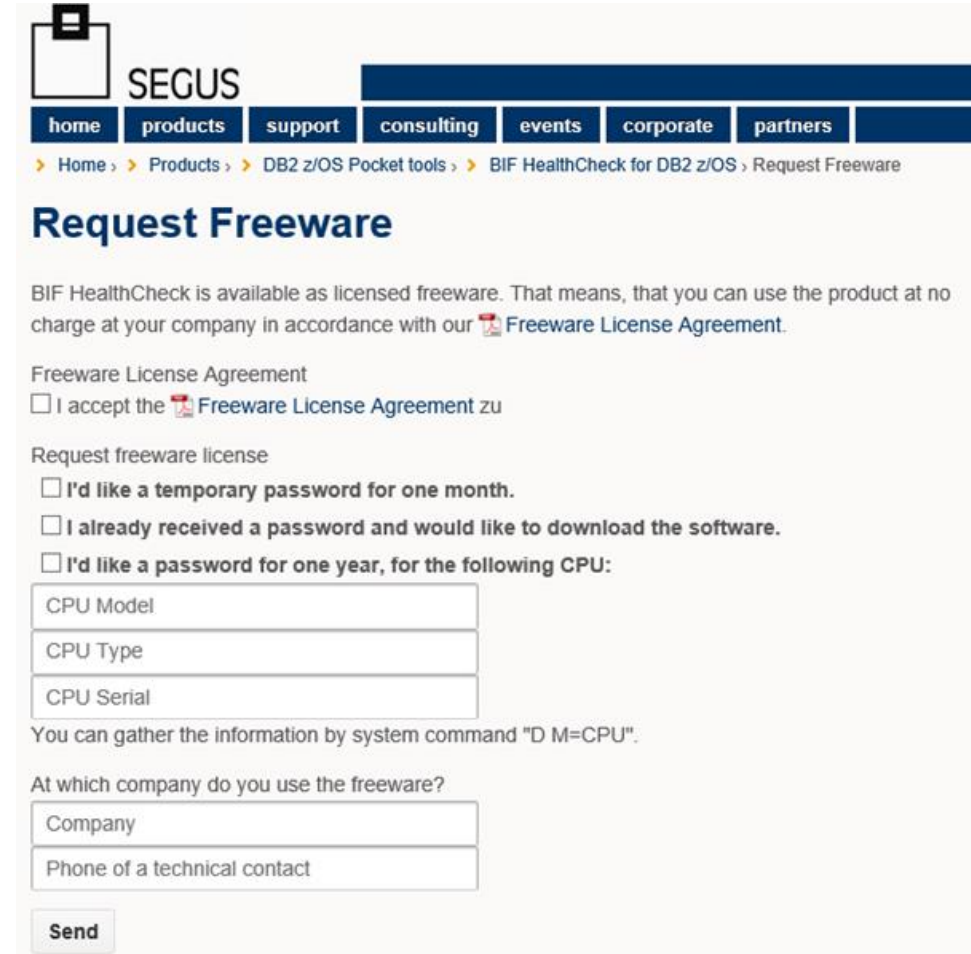
BIF/ICI HealthCheck for Db2 z/OS

Download* the Freeware at

https://www.segus.com/bif

*requires a free account at segus.com