CAN WE SAY "AGILE" AND "DB2 FOR IBM Z/OS" IN THE SAME SENTENCE?

The

term “DevOps” is occupying a lot of space in articles, presentations, and blogs lately. But it doesn’t apply to the mainframe or Db2 on z/OS… or does it?

As Db2 DBAs, we’re too busy keeping the lights on to worry about a trend that affects only distributed and mobile apps. But the truth is that many of those apps access Db2 data on z/OS, and they’re deploying changes at an accelerating rate. Application development teams have learned to use a variety of tools to automate and speed up the process of deployment, but they don’t understand the mainframe and how to deploy to it. **If we don’t want to be seen as the barrier to effective deployment of applications, we need to be part of the solution**.

Our job as DBAs has always been maintaining the stability of the platform and applications. Stringent change control processes have served us well in the mainframe environment for decades, and we don’t want to abandon them in the name of expediency.

This blog will discuss four processes that are necessary to avoid risk, and how to speed them up without increasing risk:

- Validating SQL changes.
- Changing schemas.
- BINDing packages.
- Test environment provisioning

Before we can deploy an application SQL change into our production environment, we need to be sure it won't adversely affect performance, particularly if the SQL is accessing operational tables (the tables that contain the data on which we run our enterprise). We also want to be satisfied that the SQL to be deployed meets our coding standards. When most of us were young DBAs, we did this by examining each SQL statement before it was deployed. With a change control cycle of weeks, we could afford that luxury. **With today's change control cycles of days or even hours, we need to automate this process.** Rigorous integration and volume testing can help, but performance in the test environment doesn't guarantee performance in production.

We can meet both these demands by maintaining EXPLAIN history for every SQL statement in our application. This could be the subject of an entire blog by itself. The idea is that we can identify access path regressions by comparing optimizer costs for each SQL statement being changed, version to version. We can also identify SQL statements being added. The point of this automation is to reduce the amount of data to be checked manually. We can also highlight violations of our coding standards that result in inefficient access paths, for example. A tool such as BMC Performance for Db2 SQL and its Workload Access Path Compare feature is invaluable in this context. Don't forget that this depends on statistics being copied from production to test, and ideally we have set up a profile to simulate the buffer pools and processors in production as well.

Now that we are happy with the performance profile of our application change, the next concern for us as DBAs is changes to the schema. These must be identified and validated, and then we are concerned about any outages necessary, for ALTER commands, REORG utilities, BIND commands, or in the worst case, dropping and recreating one or more objects because of schema changes that cannot be accomplished with an ALTER. The schema changes must be coordinated with application code promotion. To accomplish this process, we need a tool like BMC Change Manager for Db2, to compare current and desired schemas, and generate scripts to implement the necessary changes. A critical item of communication here is the outage (BIND, Reorg, or DROP and CREATE?). DBAs can add value here by sharing that impact with the application teams. We hope that most changes will not demand a significant outage, but identifying those that demand extra planning and preparation is essential to streamlining deployment.

Whenever static SQL is changed, we must BIND the appropriate packages. A best practice here is to use versioned packages, so the BINDs can be run beforehand, for the cases where there are no schema changes. We need to remember to be careful not to FREE the current version until we're sure it won't be needed. Having prepared JCL to run the necessary BINDs is very useful; the application development team can submit the JCL from our agile deployment tool. When schema changes are involved, the BINDs must of course be submitted after the DDL has been run. For simplicity's sake, it makes sense to delay all the BINDs, and avoid trying to separate them into two groups, one to run before and one after the DDL.

"Test environment provisioning" refers to supplying or finding test data in a volume test or pre-production environment. Of course, test data must be provided at every stage in the process of promoting application changes that start in a developer's sandbox and end up in production, but this case is an issue for Db2 on z/OS. Development teams don't understand the mainframe or Db2, or how to find space, allocate tables, etc. Quite often, copies or near copies of production data already exist on a pre-production subsystem, and can be reused. The other major issue is that DDL changes must be made, SQL must be promoted, and necessary reorgs, drop/create jobs, and BINDs must be performed. But these are specifically the actions we've just been discussing! Test provisioning is an opportunity for us to build and test the scripts and JCL we will use to validate SQL, accomplish schema changes, and execute any necessary utilities or commands. This process will be much easier if we have pre-populated JCL streams that can be reused.

I hope this blog has helped you think about how you, as a DBA responsible for Db2 for z/OS, can contribute value in the brave new agile world. The key is identifying repetitive actions that can be automated, to reduce the time necessary to deploy application SQL changes in Db2.

We should get to know our application development teams. Let’s talk to them about how they see mainframe deployment working, and let’s share our challenges. We can work together to reduce the risk of changes to our Db2 applications at the same time as we speed up the process.

Talk to your BMC account team or a BMC salesperson about BMC products that can help you make the most of DevOps for mainframe processes. BMC provides full suites of performance, administration, and recovery tools for Db2 on z/OS.