Db2 for z/OS, The Next Frontier

Daniel L Luksetich, DanL Database Consulting
Paul Bartak, Rocket Software

Presenters

Daniel Luksetich – Independent Consultant

- 36 years in information technology
- 31 years with Db2 (z/OS, LUW, iSeries)
- Certified Cicerone
- Punk rocker

Paul Bartak – Distinguished Engineer, Rocket Software

- 36 years in information technology
- 35 years with Db2
- Grandpa, Triathlete, Pilot, Sound Engineer

Presentation Concepts

- Today's Db2 is not your grandfathers DB2, but neither is the environment in which it operates
- Today's competitive landscape has changed dramatically
- Technology needs to keep up, or it will be left behind
- Multi-speed IT begs the question, "Do you have a seat at your company's innovation table?"
- Db2 for z/OS fast release cycle adaptation exists in Database-as-a-Service (and this isn't slapping a REST API on a legacy tool)

Agenda

1. Discuss the evolution of database administration
2. Identify challenges for DBAs and developers
3. Discuss which adaptations can be made
4. Identify issues associated with adapting
5. Identify and define the components of modern application development
6. Present IBM DevOps Experience as a solution to the issues

Definitions

Systems of Record

- This is the authoritative for a given element or piece of information
- Primary data store for an enterprise
- Often the primary database supporting transaction processing
- Typically centralized

Systems of Engagement

- People-focused applications
- Enables customers, partners, and employees to interact with the business
- Could be decentralized

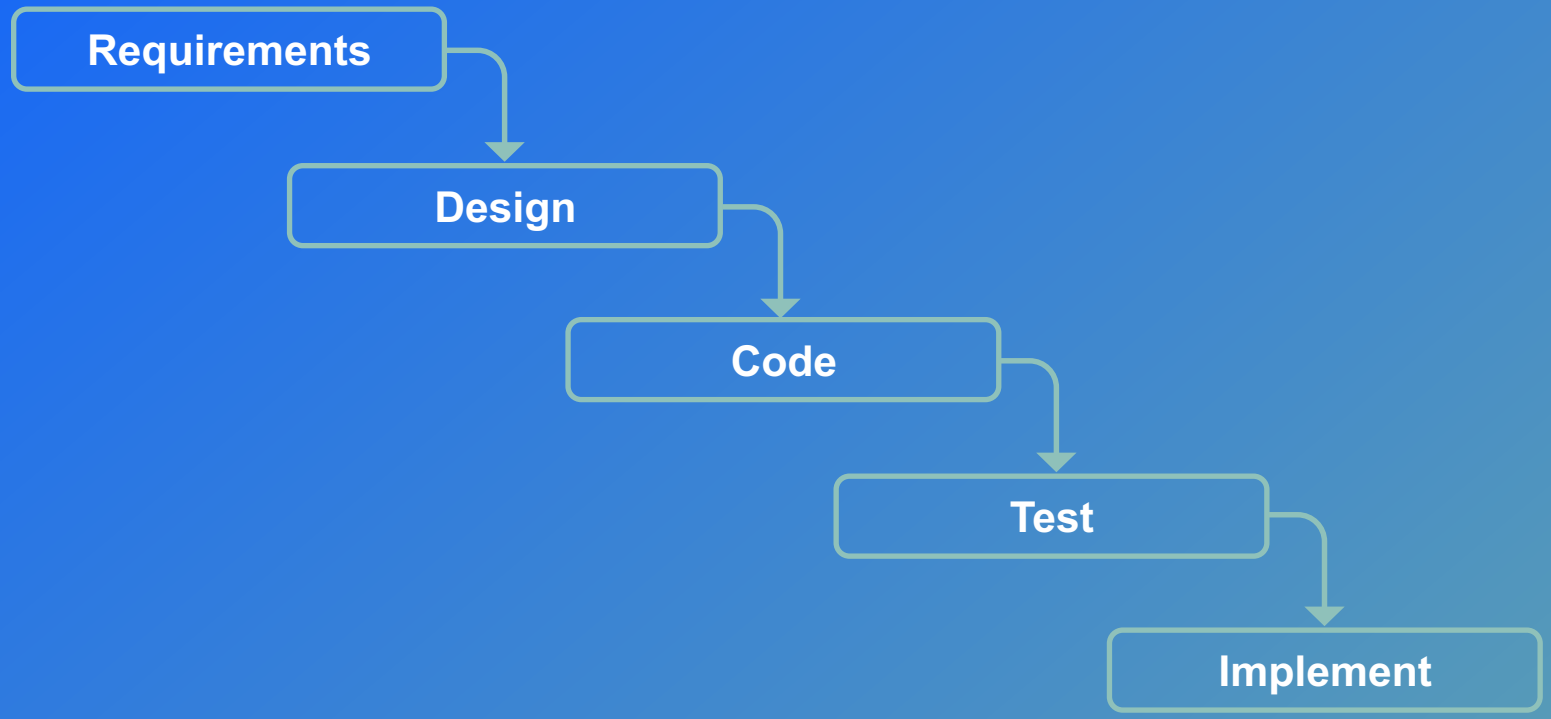
System of Insight/Innovation

- Technology that aims to improve the customer experience
- Evolving technology that drives new ways of doing business

Traditional Db2 for z/OS Development

Waterfall Development

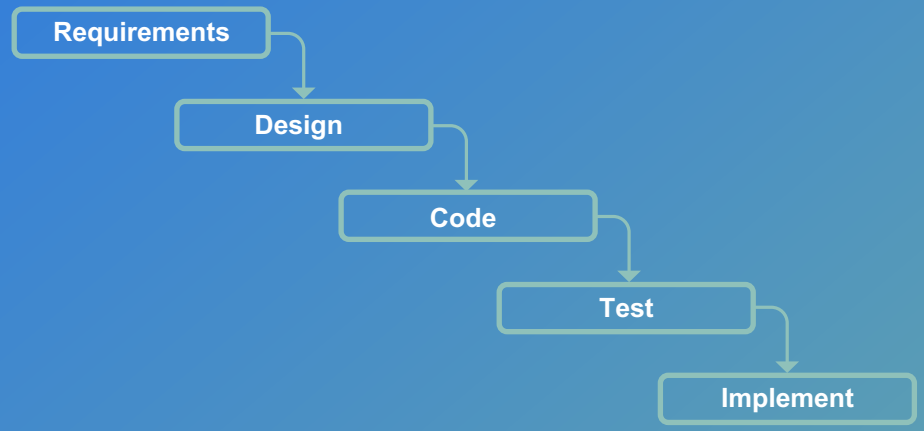
- Traditional application development methodology for many years
- Each phase of the project didn't begin until late in the previous phase



Waterfall Development and Db2 for z/OS

Data Administrators and Business Analysts were responsible for database and process design

- Building logical models and perhaps process models
- Established naming conventions
- Documented data and business rules



Database Administrators

- Converted logical models to physical models
- Developed and deployed databases
- Managed database storage
- Monitored and managed performance

Waterfall Development Challenges

"Big Bang" implementation presented challenges

- 11th hour design changes and design creep
- Long project duration, scope creep, and staff turnover
- Project delays result in limited testing
- Fire-fighting performance issues upon implementation

Database maintenance post-implementation

- Monitoring storage
- Monitoring performance
- Database REORGs
- Column and table additions/modifications

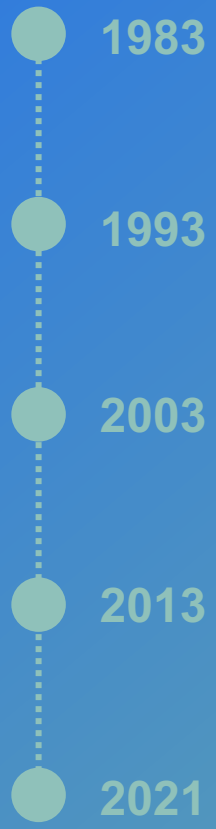
The Evolution of Db2 for z/OS

Db2 for z/OS of the past involved

- Tuning of SQL
- Frequent RUNSTATS and REORGs
- Majority of application were mainframe based

Db2 for z/OS of now includes

- More self-management features
  - Self tuning/monitoring
  - Better storage management
- More availability
- More remote applications accessing
  - More controls (e.g. profiles)
- z/OS improvements in DASD response and CPU



Db2 for z/OS is still a relevant data server for “systems of record”

The Evolution of Db2 Database Administration

DBAs are still performance focused

- A Lot of that focus is directed at web-based applications

DBAs still create/modify/deploy DDL

The management of database administration is changing

- More database responsibility and fewer DBAs
- Aging talent pool
- Old ways/traditions hard to break
- Younger talent views z/OS as a deal breaker
- Lack of appropriate training

Stressed out DBAs awaiting retirement

Modern Database Development

The Db2 for z/OS Development Conundrum

Db2 seems to hold its place as a data store for “systems of record”

- But NOT for “systems of engagement” or “systems of innovation”

A lack of training and tools for Db2 for z/OS DBAs

- Leads to resistance among the legacy DBA staff to adapt to new technologies and methodologies
- This resistance leads application developers to other databases and platforms for newer applications
- This results in an increase in data replication from Db2 for z/OS to other data servers

There is also a perceived lack of support for modern development using Db2 for z/OS

- No tools for the non-mainframer to access z/OS subsystems?

Modern Development

Enterprises need to remain relevant and competitive

Information technology drives the enterprise

The most successful enterprises recognize this

- Must outpace the competition
- Deliver new applications quickly
- Large enterprises need to outpace their smaller competitors

Your enterprise is no exception

- Example: Uber versus the taxi industry
- Example: Airbnb versus the hotel industry

Agile Development and DevOps Processes

- Agile development and DevOps is the answer to the challenges of waterfall development
- Application developers are empowered to develop and deploy quickly
- Automation assists with the development, testing, and implementation of application components

Agile Development

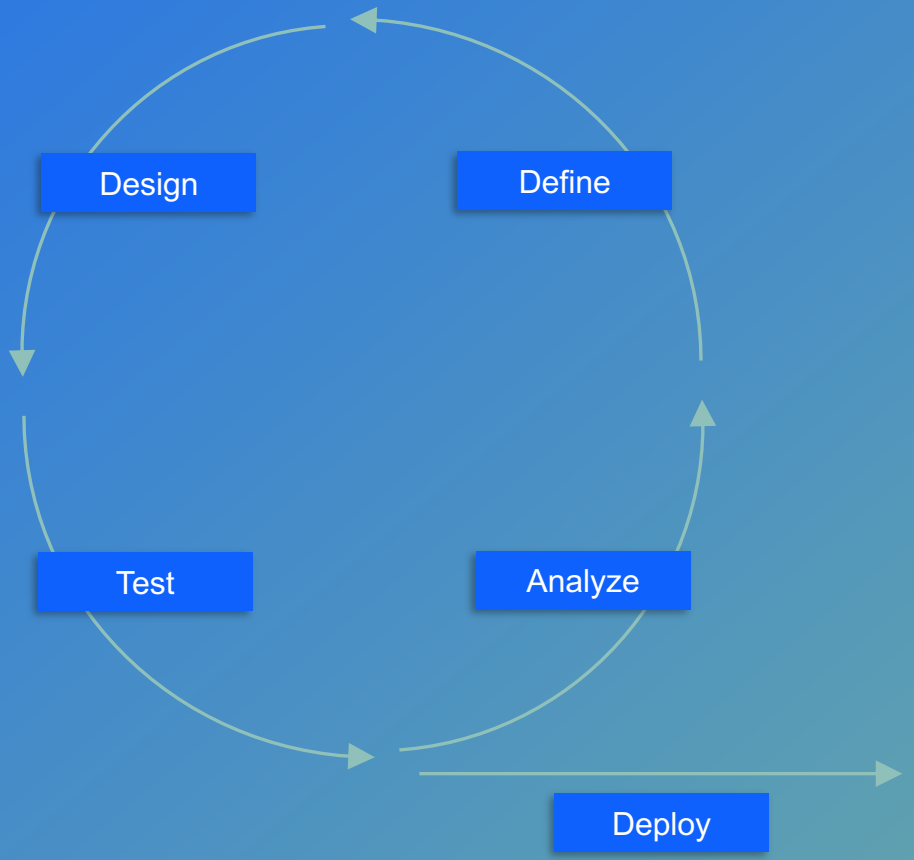
Iterative development cycle

Smaller more numerous development teams

- Scrum
- Typical might be 1 to 15 developers

Development broken up into smaller pieces

- Sprint
- Typical might be two weeks



DevOps Processes

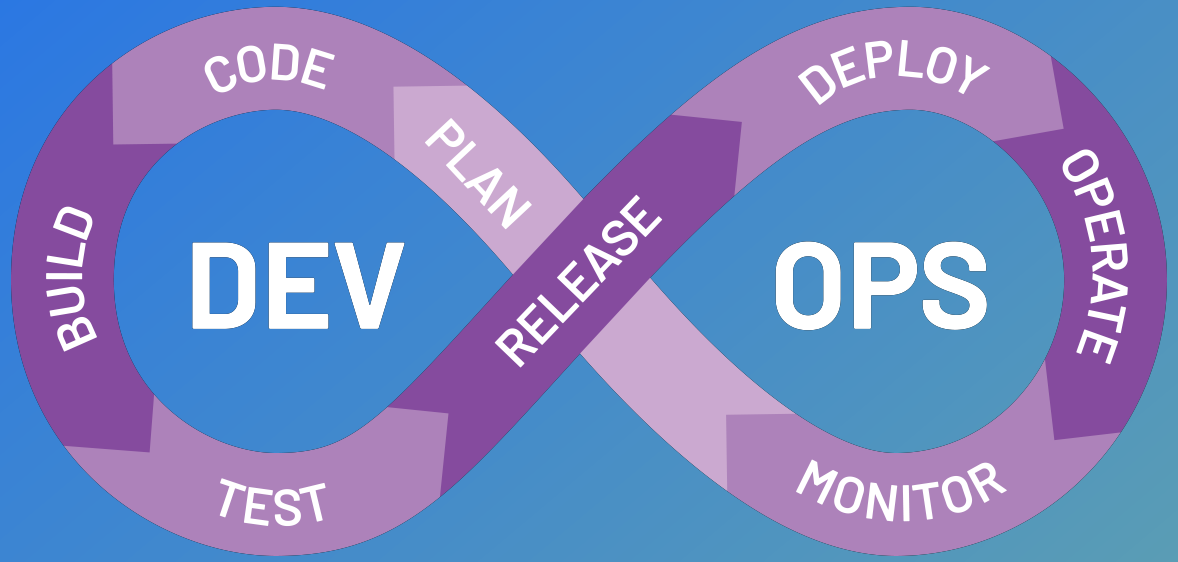
Builds on Agile principals

Introduces

- Continuous delivery
- Continuous integration
- Continuous testing
- Continuous collaboration

Automation is a key component of the DevOps process

- Code management
- Implementation
- Testing



DBA Responsibilities with Modern Development

Increased volume of development and test schemas

- More development teams means more individual schemas

Increased pace of database schema modifications

- More developments teams and faster pace means more changes
- Keeping track of changes and avoiding collisions becomes a serious challenge

Increased demand for backup/restore of test data

Rapid deployment schedule

Production duties remain and perhaps are more intense

- Faster rate of change perhaps means more instability

DevOps Resistance and Challenges

To effectively enable DevOps there needs to be a redistribution of responsibilities

- Application Developers need more control
- Provision schemas for changes and testing
- Make schema changes

Resistance

- z/OS Administrators don’t want to relinquish control
- Data Administrators don’t want to lose control
- Application Developers are not interested in TSO/ISPF

DORA Statement

*The DevOps Research and Assessment LLC (DORA)**

- *Database changes as a major source of risk and delay when performing deployments.*
- *Mitigating these risks is establishing good communication and comprehensive change management practices*
- *Integrating database work into the software delivery process helps contribute to continuous delivery*
- *Keeping all database changes in version control*

*Source: https://cloud.google.com/architecture/devops/devops-tech-database-change-management

"Data Friction" in DevOps is the primary driver that sends developers to other platforms

- Leaving Db2 for z/OS out of systems of engagement and innovation

Evolution of Db2 Development

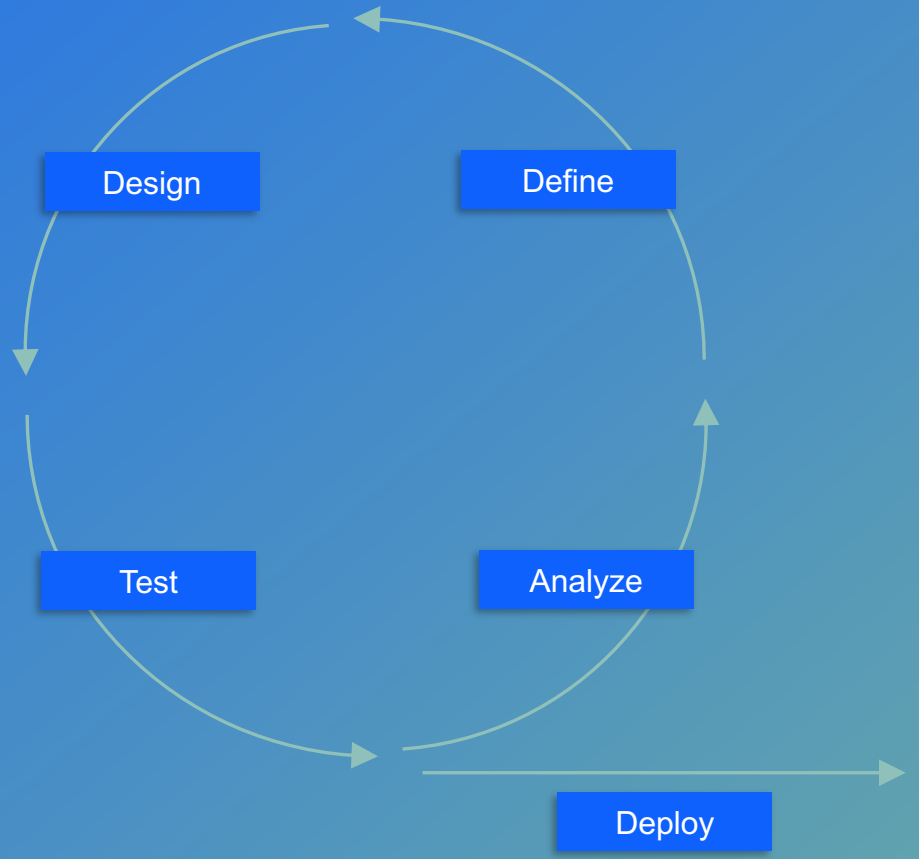
A Change in the Support Model

The Application Developer needs more control

- Provisioning and modifying database resources
  - Example: adding a column to a table
- Control is still necessary

Certain enterprise functions need to evolve

- Systems administration
- Data administration
- Database administration



Systems Administration

Database authorities will have to be reviewed

- Do you give Application Developers authority to create and modify database objects
- Which environments does this apply to?
- Which subsystems? Data sharing groups?
- Will storage allocations get out of control?

The choices include

- Enhanced controls
- OR increased workload

Data Administration

Data Administrators and Business Analysts

- Still retain responsibility for major database designs
- Still retain responsibility for major business process designs
- Still document business and data rules

The number one Application Developer change is column addition

- Administrators need not be involved
- Rules should be in place to retain control
- DevOps processes can help

Database Administration

Responsibility expands for Database Administrators

- Environment management
- Schema management
- Database authorities
- Rule enforcement

Adopting DevOps processes a definite plus here!

Database Administration

Environment Management

- Multiple development and test environments managed
  - Development
  - Testing
  - Integration
- Targeting appropriate Db2 subsystems

Database authorities

- Do we allow Application Developers to make changes?
- This can be a major bottleneck

Database Administration

Schema management and assignment

- Development teams identified
- Unique schemas created and assigned
- Database changes
  - Tracked
  - Approved
  - Integrated into other schemas
- Unused schemas deprovisioned

Adopt strategy of database as a service (DBaaS)

- DDL is code and must be maintained
- A tool is critical in this regard

The IBM DevOps Experience

Provision *on-demand/self-service* to reduce developer wait time

Provide *data sources as code* for fast response to the business

Honor IT/Admin standards and mandates with *codified control*

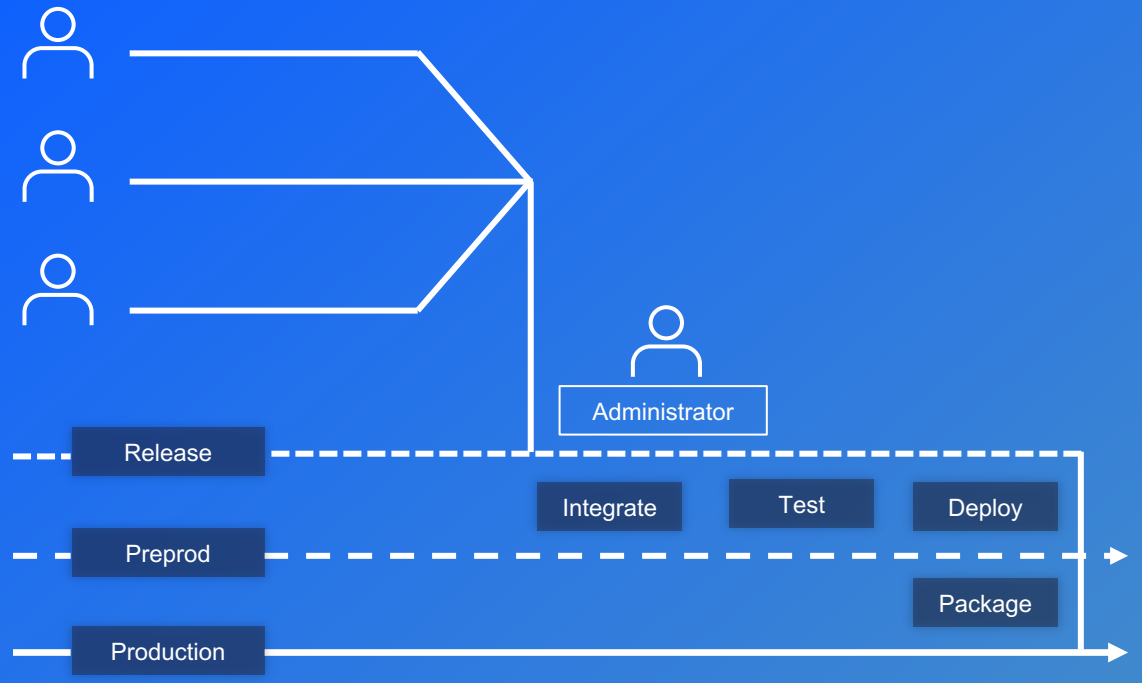
*Bring Db2 applications to market **FASTER** with lower costs and less risk*

Moving towards CI/CD

Features

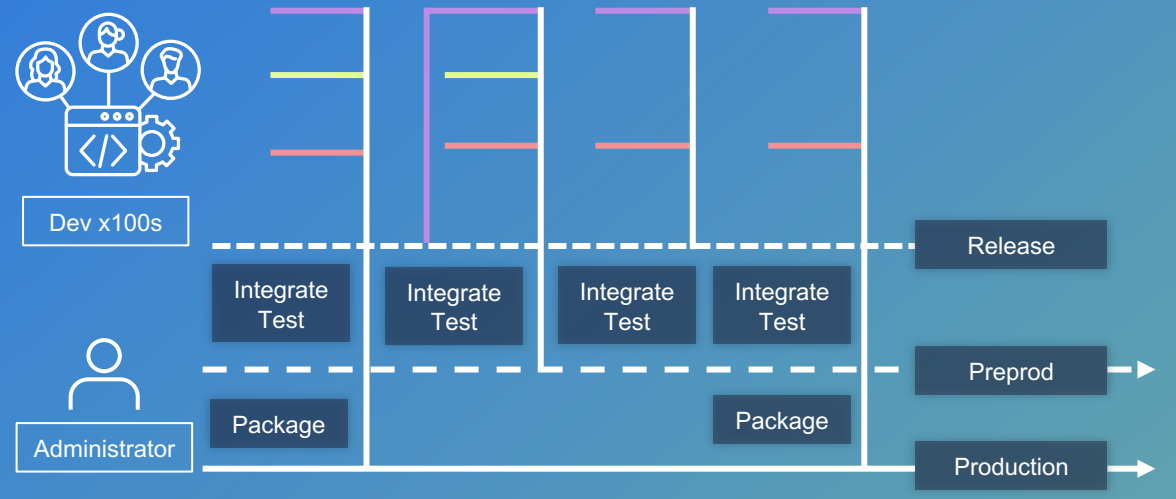
Long release cycles

Large release cycles | Slow delivery | Expensive | Disruptive | Large scope & slow remediation



Fast release cycles

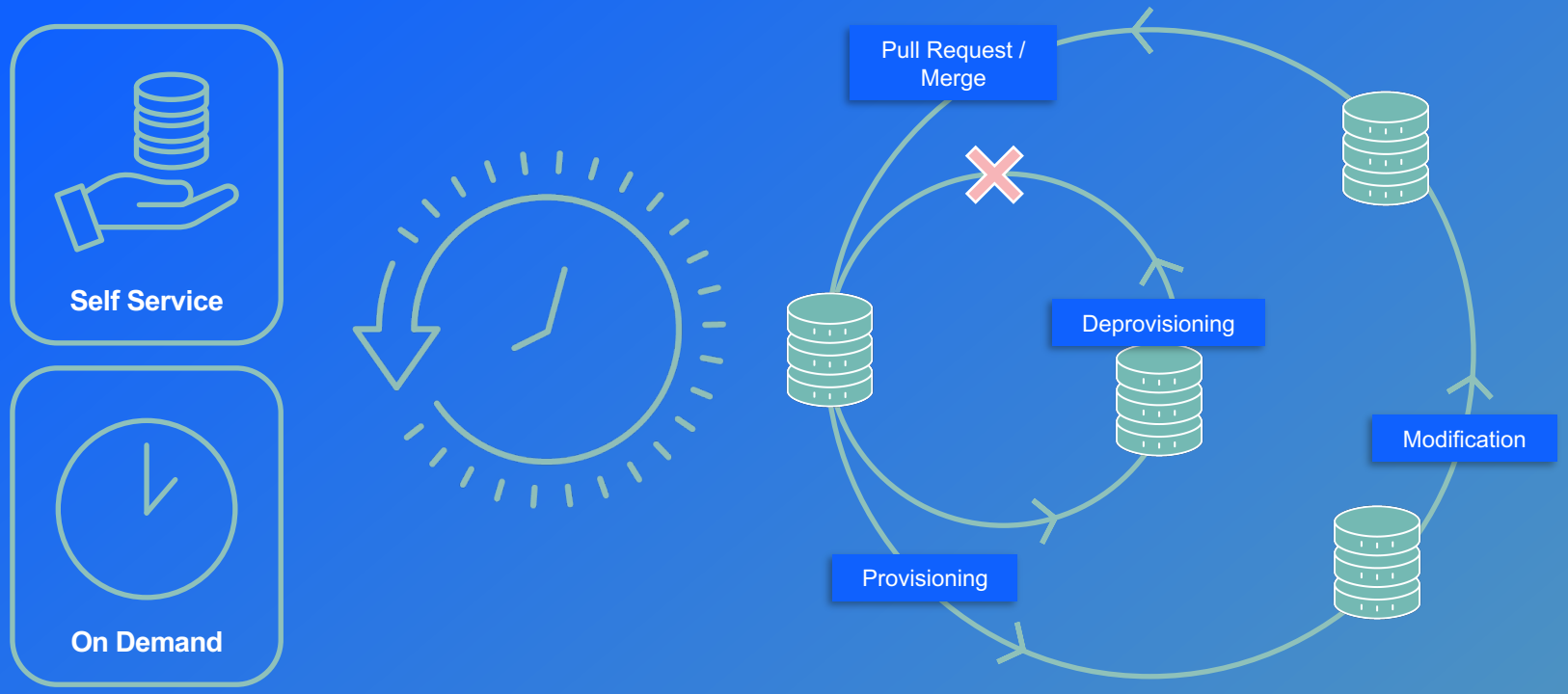
Short cycles baked in | Fast delivery | Measured & manageable | Small scope & quick remediation



- DB2 FOR Z/OS OPS, DATABASE AS A SERVICE

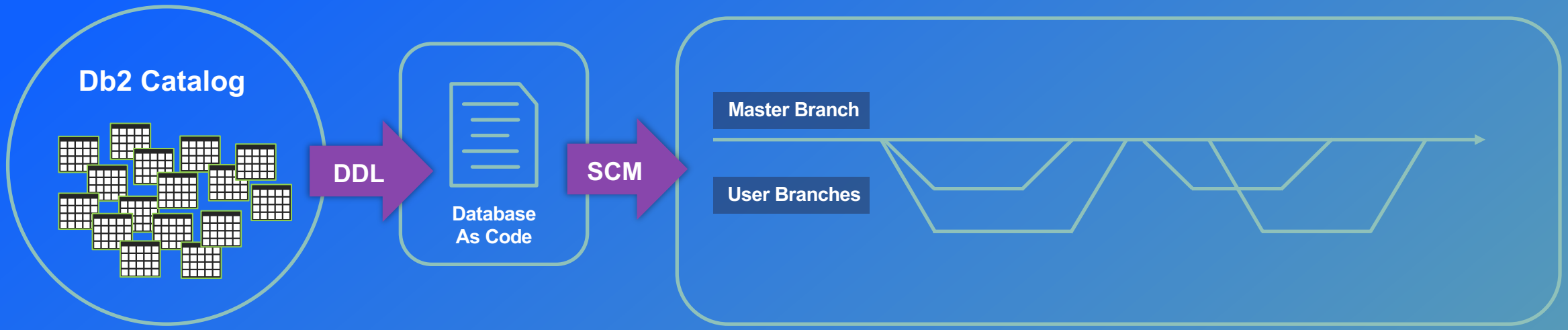
IBM Db2 DevOps Experience for z/OS

Provision on-demand/self service to reduce developer wait time



- Meet developer cadence
- Fail fast, deprovision, try again
- Deploy changes as needed
- Promote to master branch

Provide data sources as code for fast response to the business



Versioned DDL for logical groupings of Db2 objects

Unites with application version control and infrastructure as code

Fuels provisioning request and change deployments

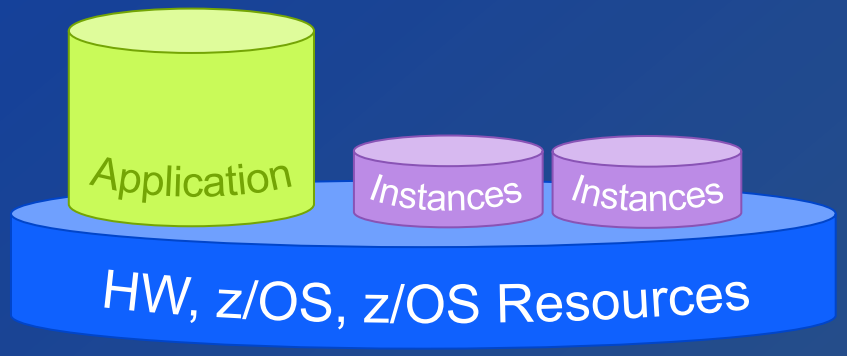
Honor IT/Admin standards and mandates with codified control

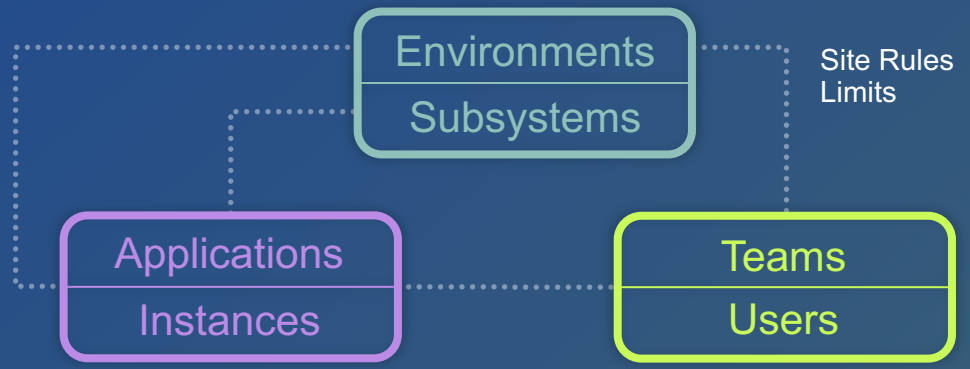
Overall management of constructs

Instance limits | Governance over DevOps | Rules | DevOps at your pace

Flexible rules for fluctuating change

DevOps in a shared environment





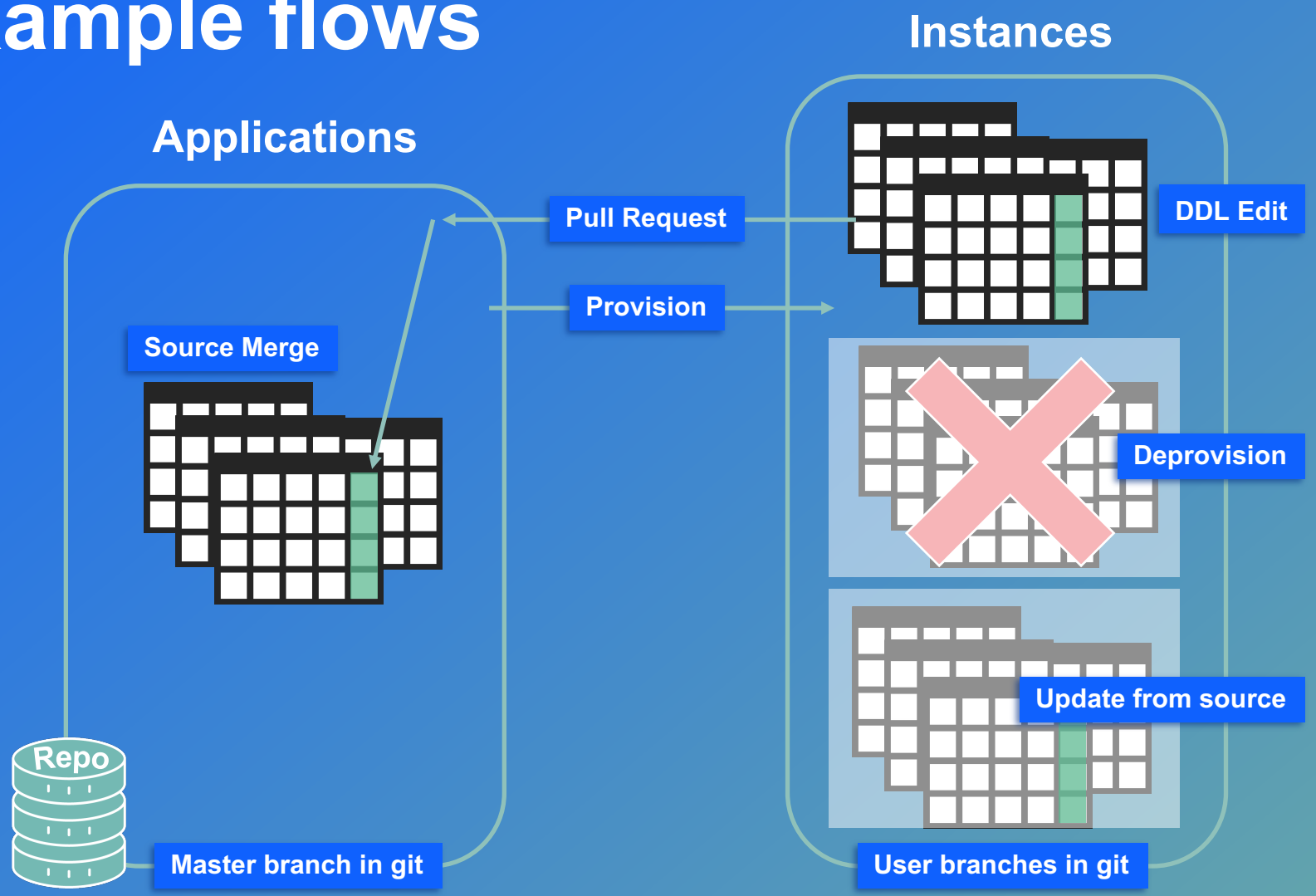
Registration of Db2s & Db2 objects

Control provisioning activities

Expanded, fenced authorities	Rules for naming, placement, definitions	Easy visibility to rules, metrics, etc.
Namespace for instance separation	Storage monitoring	Give developers their own environment

Db2 DevOps example flows

- Subsystems are registered
- Users, teams set up and assigned environments
- Site rules defined



Transition from workflows to tooling of your choice

Dev Engineer

Bamboo

Admin

CHEF

puppet

ANSIBLE

Developer

Maven™

Gradle

Metadata & DBaaS

DevOps adoption

DBaaS object definitions

- What is available & How is it being used
- Are the constructs easy for a developer to consume

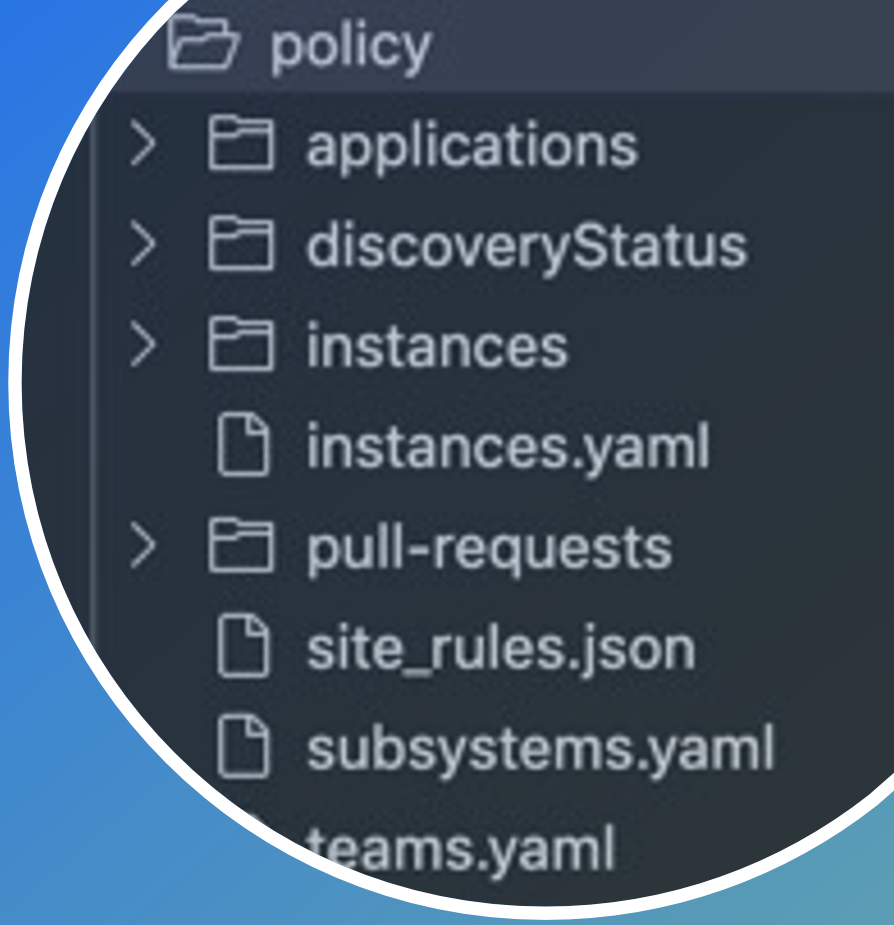
Ownership & editability

Approval cycle

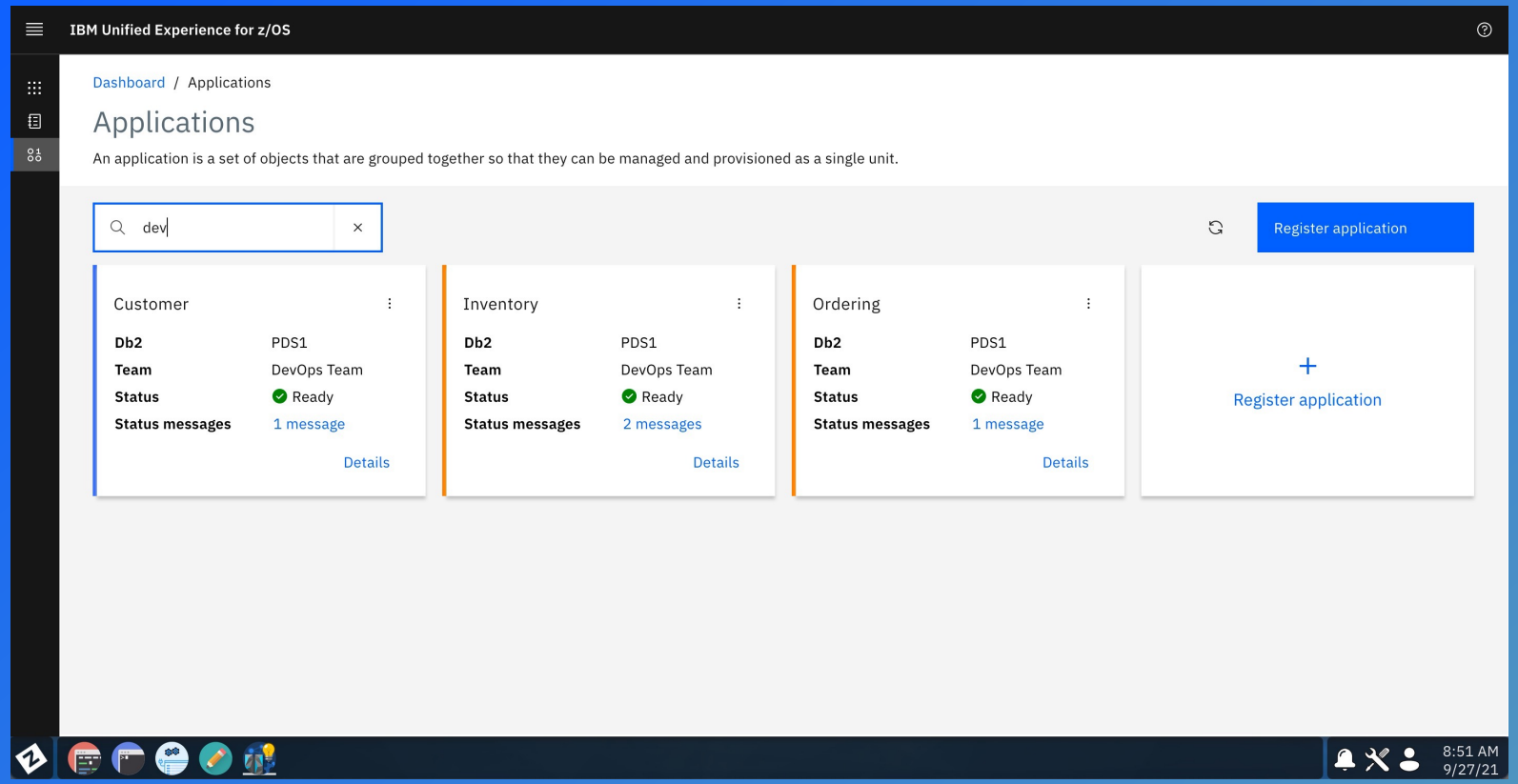
- Review and integration
- Drive toward schema synchronization

Monitoring & usage

- Control for administrators



Thank you!



Check out these blogs on Db2 for z/OS and modern DevOps processes:

Blog 1 of 4
Blog 2 of 4
Blog 3 of 4
Blog 4 of 4
What is DOE?