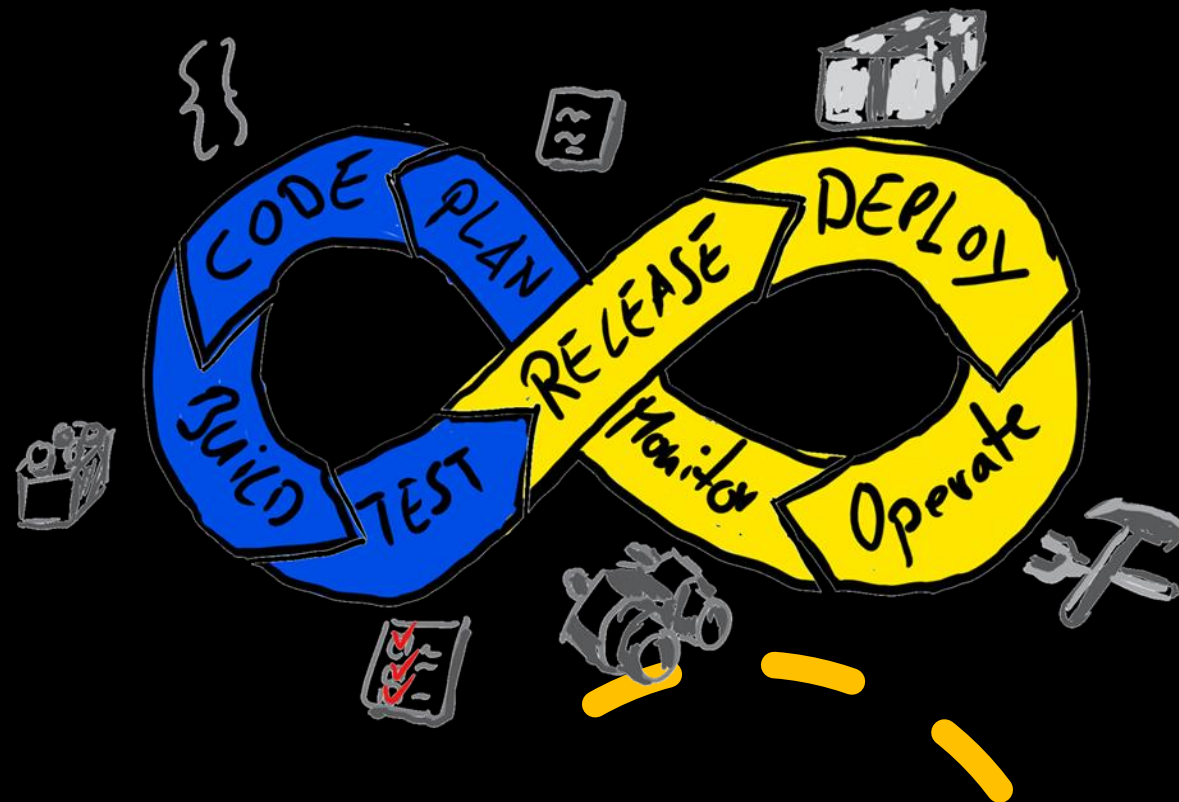


IBA

GROUP

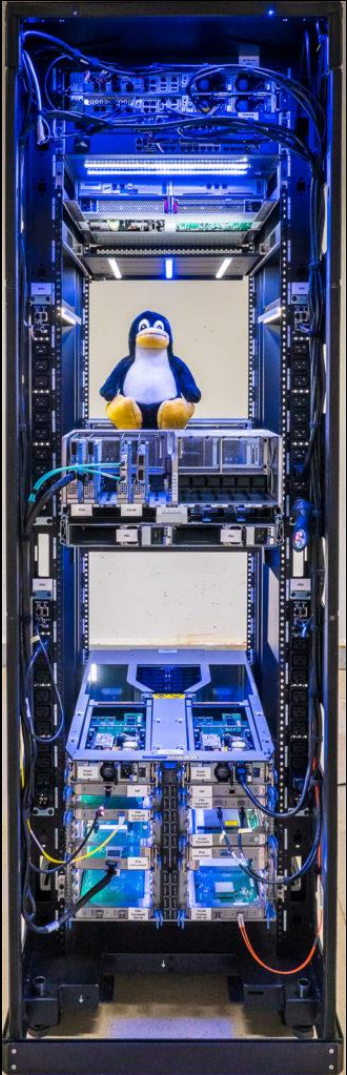
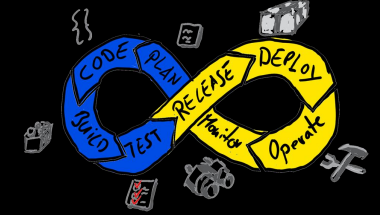


DevOps for Mainframe

Yuliya Varonina | Tatsiana Ihnashchenka

December 2020

Mainframe position in technology world



Mainframes are computers used primarily by **large organizations** for **critical applications**;

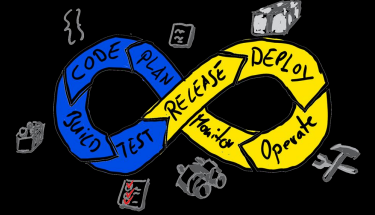
bulk data processing, such as census, industry and consumer statistics; enterprise resource planning; and transaction processing.

Spheres: military industry, banks and insurance companies, air carriers and tour operators, logistics and transport business

Development companies: IBM, BMC, BROADCOM, Rocket Software, Micro Focus ...

They are larger and have more processing power than some other classes of computers: minicomputers, servers, workstations, and personal computers.

What is DevOps?



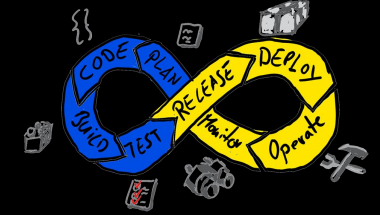
DevOps is a culture of automation and collaboration within organization, the set of best engineering practice together with mindset changes.

Mainframe DevOps?



DevOps is a culture of automation and collaboration within organization, the set of best engineering practice together with mindset changes.

Technology **stack** for mainframe DevOps



z/OS
Db2
MQ
CICS
C
PLI
Cobol
ASM
TWS

IBM Urban Code Deploy

Jira

Git

Shell

Groovy

Java

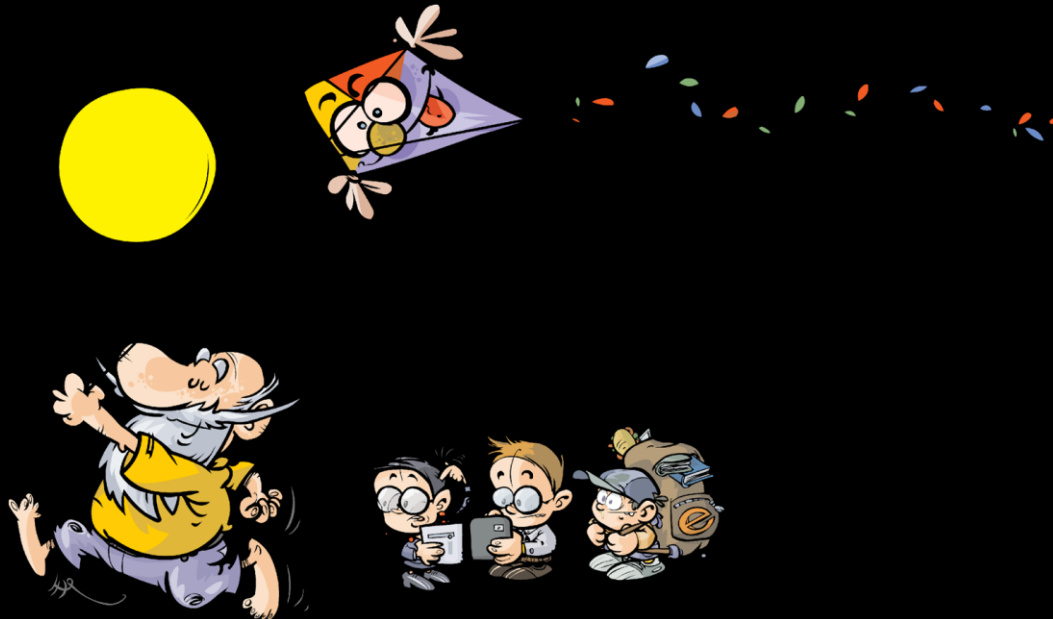
zUnix

IBM Infosphere

Docker

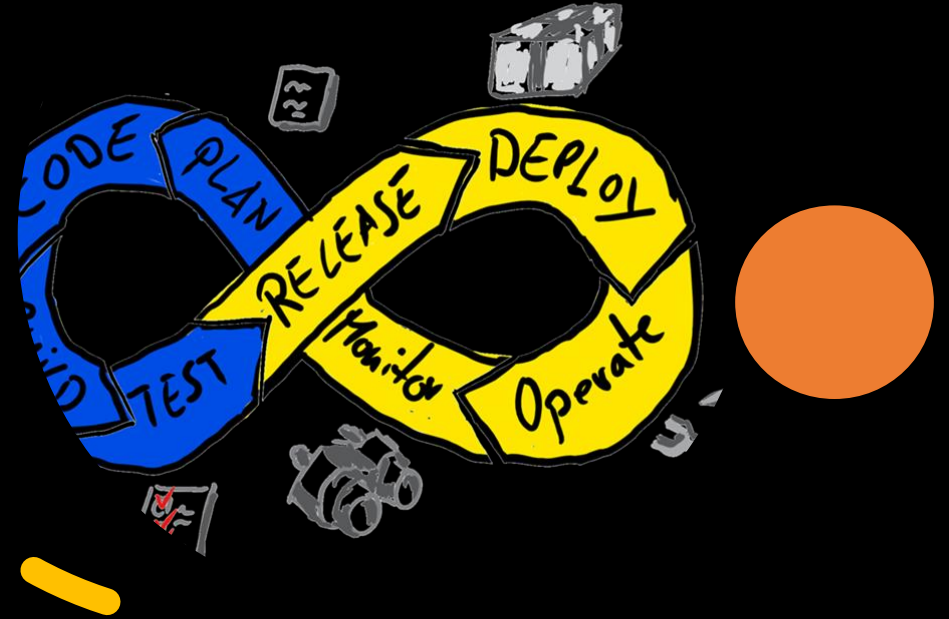
IBM Cloud

**DevOps pipeline to deploy mainframe
and distributed code as one solution**

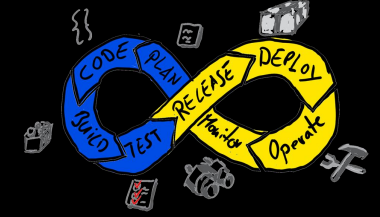


Pain of Mainframe-Based Apps

- Manual operations (build, environments setup and customization, installation, all levels of testing ...)
- Long deployment cycle ~ 1 week to 1 month
- Version control
- Limited set of automation tools
- Poor visibility and control at all stages



Solution



IBM Urban Code
mainframe/non-mainframe CI/CD

Unified solution

Automatic Code Review

Scalable pipeline constructor

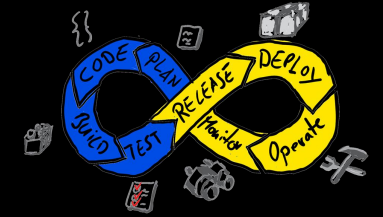
Quality improvement by reducing manual efforts
(always prone to mistakes).

Delivering **software quickly,**
reliably, and safely is at the heart of technology transformation
and performance.

Dev teams became more open to innovate.

Existing **toolset harmonization and standardization** to
provide common
code base and deployment processes, using the best engineering
practices.

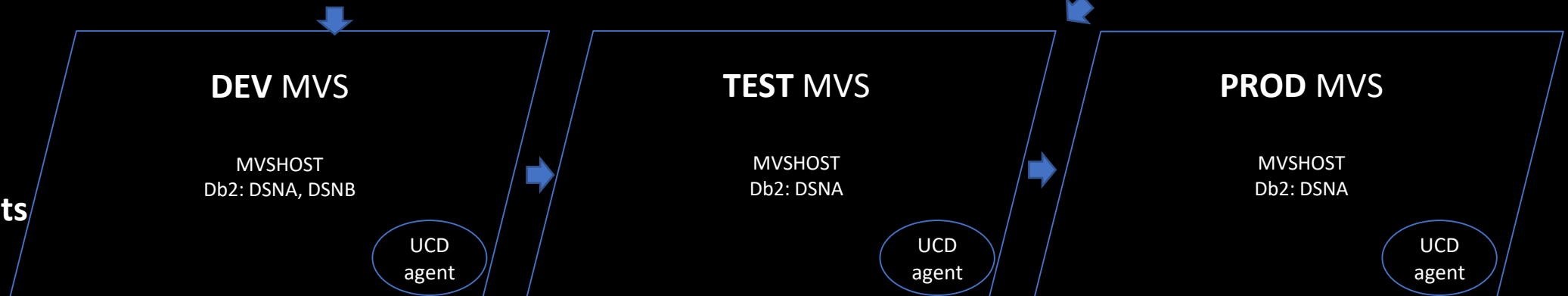
Infrastructure



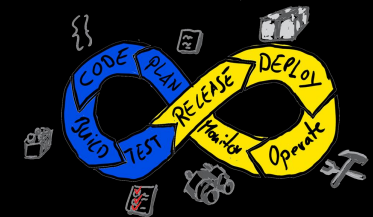
Data Stage deployments



MVS deployments



Servers architecture



IBM Urban Code Deploy
(client-server)

UCD
z/OS
plugin

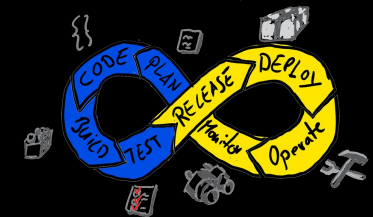
Mainframe
USS
UCD agent

Windows
UCD agent

Linux
UCD agent

- Allocate Data Set
- Allocate Data Set From Existing
- Allocate SMS Managed Data Set
- Copy Artifacts
- Copy Data Set
- Deploy Data Sets
- FTP Artifacts
- Generate Artifact Information
- Remove All Versions
- Remove Redundant Versions
- Replace Tokens MVS
- Rollback Data Sets
- Run MVS Command
- Run TSO or ISPF Command
- Submit Job
- Wait For Job
- zOSMF / Create Workflow
- zOSMF / Run Published Software Service Template
- zOSMF / Software Services Instance Discovery
- zOSMF / Start Workflow
- zOSMF / zOSMF Cloud Provisioning Discovery

Architecture



Dev Environment



IBM Db2

TWS / MVS

PL/I / REXX

Test Environment



Prod Environment



Dev initiates
build
package
process on
mainframe

UrbanCode CLI
Shell
Buztool



slack

Code
Station

Items handling

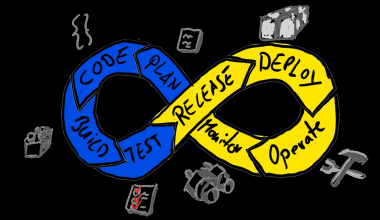


UA

T

slack

Buztool



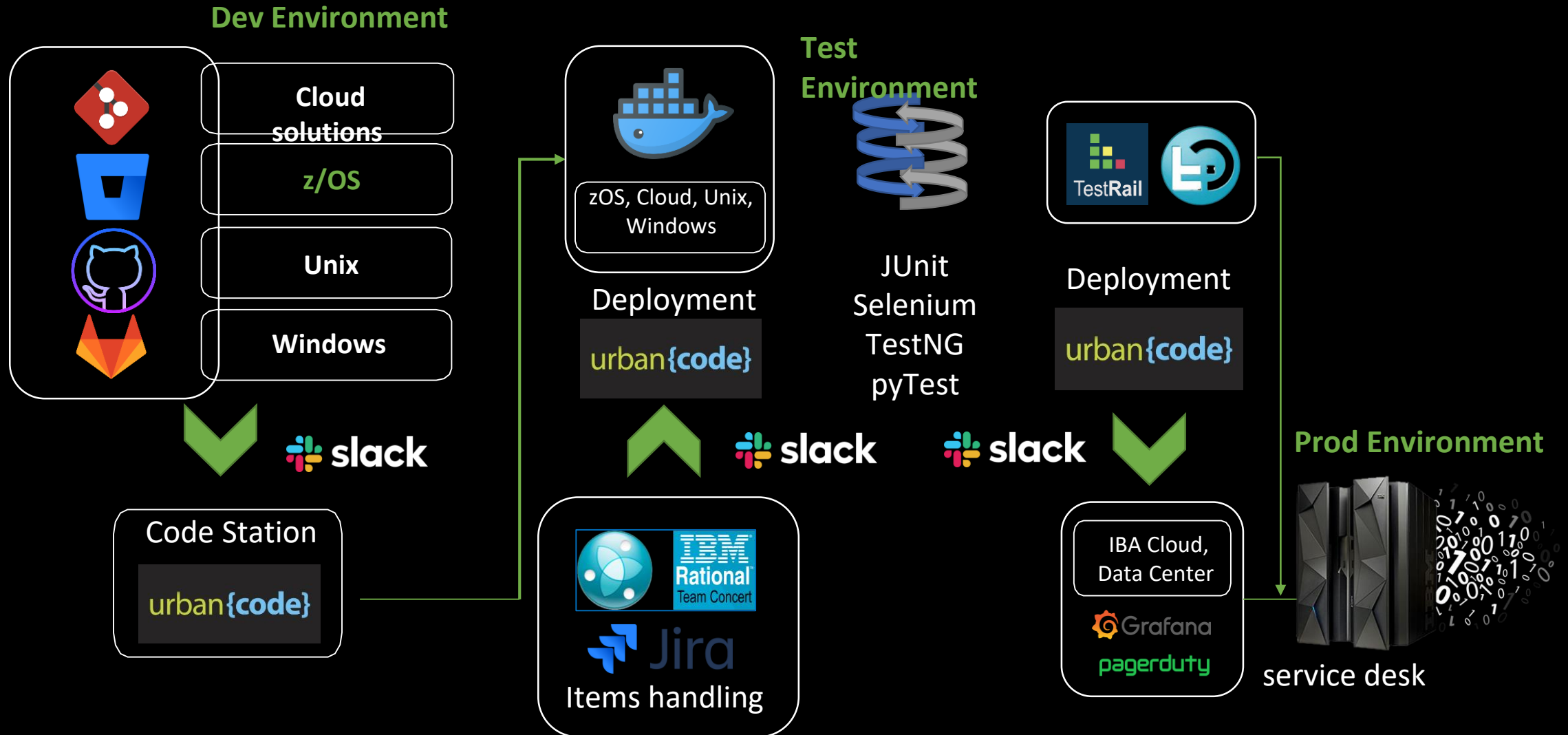
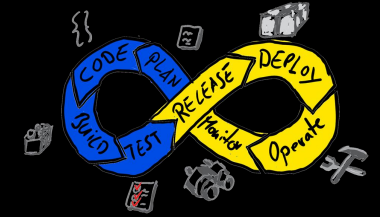
```
*****
==MSG> -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
000001 <?xml version="1.0" encoding="CP037"?>
000002 <manifest type="MANIFEST_SHIPLIST">
000003   <container name="APPL.APPLPACK.LOAD" type="PDS">
000004     <resource name="*" type="PDSMember" deployType="COPY"/>
000005   </container>
000006   <container name="APPL.APPLPACK.DBRM" type="PDS">
000007     <resource name="*" type="PDSMember" deployType="COPY"/>
000008   </container>
000009   <container name="APPL.APPLPACK.BIND" type="PDS">
000010     <resource name="*" type="PDSMember" deployType="COPY"/>
000011   </container>
000012   <container name="APPL.APPLPACK.CNTL" type="PDS">
000013     <resource name="*" type="PDSMember" deployType="COPY"/>
000014   </container>
000015   <container name="APPL.APPLPACK.SAMPLE" type="PDS">
000016     <resource name="*" type="PDSMember" deployType="COPY"/>
000017   </container>
000018   <container name="APPL.APPLPACK.CLISTD" type="PDS">
000019     <resource name="*" type="PDSMember" deployType="COPY"/>
000020   </container>
000021   <container name="0000001" type="GENERIC" deployType="RTC">
000022   </container>
000023   <container name="1234567" type="GENERIC" deployType="RTC">
000024   </container>
000025 </manifest>
*****
***** Top of Data *****
***** Bottom of Data *****
```

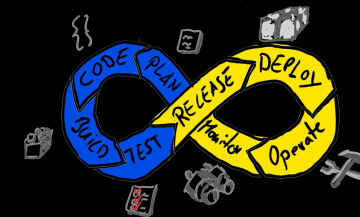
XML ship list stored
on UNIX part on
mainframe

JCL to run Buztool
utility

```
000012 //*
000013 //ST1      EXEC PGM=BPXBATCH
000014 //SYSPRINT DD SYSOUT=*
000015 //STDOUT   DD SYSOUT=*
000016 //STDERR   DD SYSOUT=*
000017 //STDPARM  DD *
000018 SH /SYS/ucd/agent/bin/buztool.sh "createzosversion"
000019   "-c" "ZosComponent"
000020   "-v" "VersionDemo"
000021   "-s" "/u/userHome/packageFolder/shiplist.xml";
000022 /*
*****
***** Bottom of Data *****
```

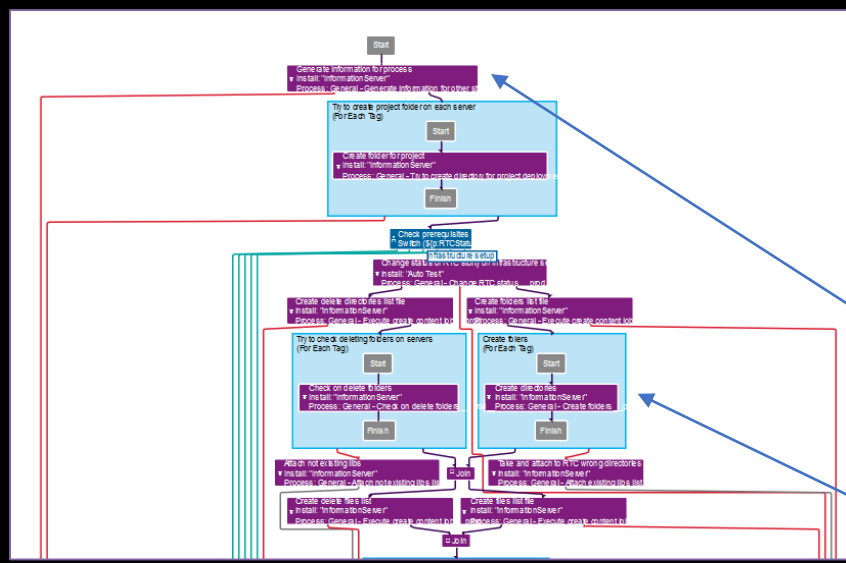
DevOps constructor



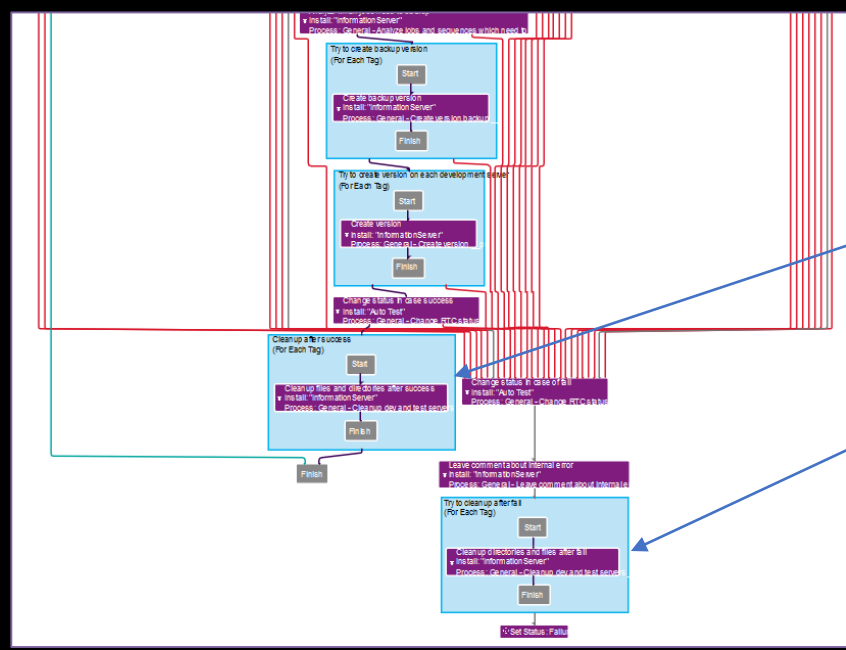


Process to build version

Process included more than 60ty component processes:



...



Step 1: Check Jira status, generate deployment information

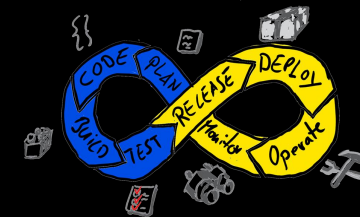
Step 2: Create infrastructure backup version

Step 3: Infrastructure deployment

Step 4: Create app backup version

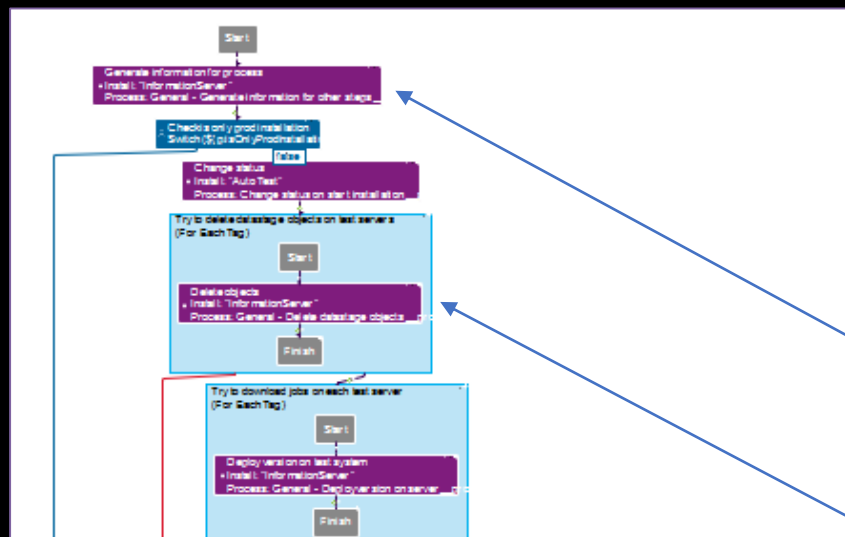
Step 5: Create app version in Urban Code code station

Step 6: Cleanup of infrastructure

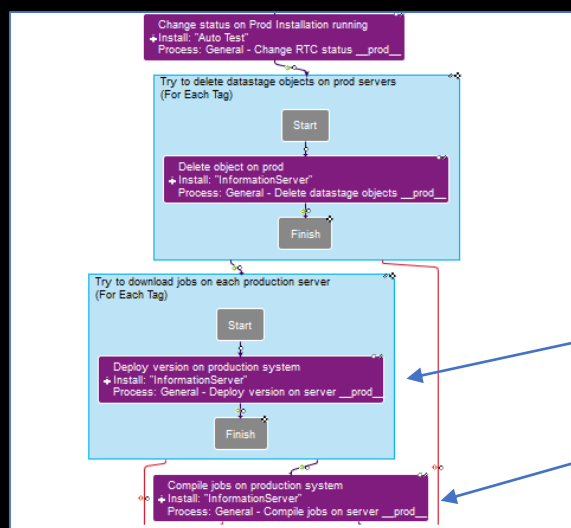


Process to deploy version

Process included more than 20ty component processes:



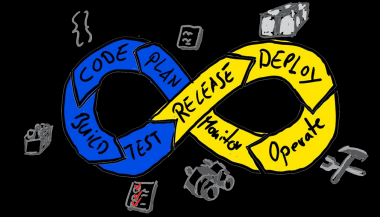
...



...

- Step 1** Check Jira statuses, generate deployment information
- Step 2:** Code customization
- Step 3:** Deploy app to TEST system
- Step 4:** Execute code review
- Step 5:** Execute test automation suite
- Step 6:** Deploy app to PROD system
- Step 7:** Notification

Developing integrations



Rest API calls

```
curl -L --post301 -D- -u %user%:%pwd% -X POST -  
H "X-Atlassian-Token: nocheck" -F  
"file=@%pathToFile%"  
http://host/rest/api/2/issue/%issueID%/attachments
```

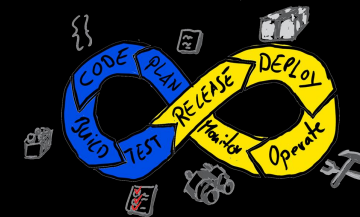
Java, Groovy, bash

```
java -jar codereview.jar APPLICATION MODE
```

JCL, Rexx

We need custom integrations when:

- there are no plug-in for a tool
- need to send reports of custom code back to the pipeline
- create a pipelines trigger outside the CI/CD tool
- need to analyze and modify legacy code



DevOps in large organizations

Questions

