TECHWELL™

STAR EAST

A TECHWELL EVENT

# T3

Test Automation
Thursday, May 3rd, 2018
9:45 AM

# DevOps Tools for Winning Agility

**Presented by:**

## Kellyn Pot'Vin-Gorman

**Delphix**

**Brought to you by:**

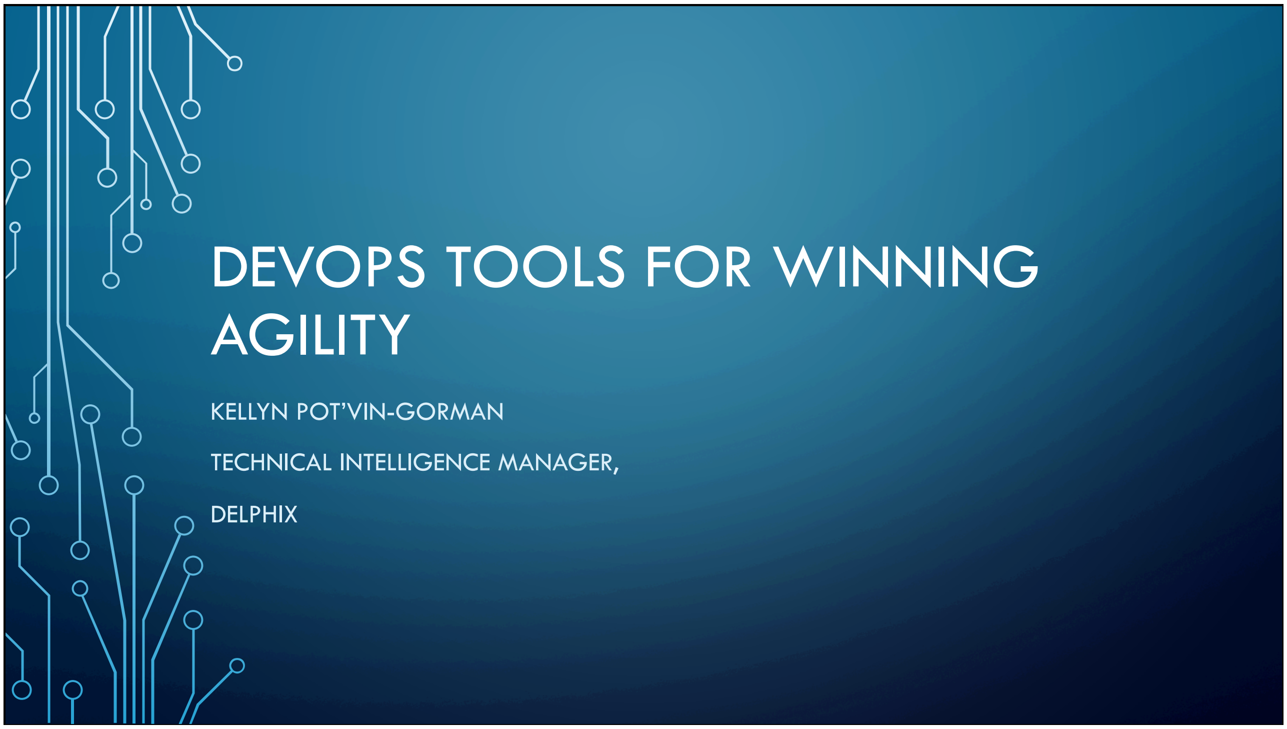
TECHWELL™

350 Corporate Way, Suite 400, Orange Park, FL 32073
888-268-8770 - 904-278-0524 - info@techwell.com - http://www.stareast.techwell.com/

# Kellyn Pot'Vin-Gorman

## Delphix

Kellyn Pot’Vin-Gorman is the technical intelligence manager for the office of CTO at Delphix, a company recognized for its virtualization and data masking environment capabilities. Kellyn is known for her extensive work with multidatabase platforms, cloud migrations, virtualization, command line interface, environment optimization tuning, automation, and architecture design. She is a member of the OakTable network and an Oracle ACE director alumnus. Kellyn’s blog and social media activity under her handle DBAKevlar are well respected for her insight and content.



# DEVOPS TOOLS FOR WINNING AGILITY

KELLYN POT'VIN-GORMAN

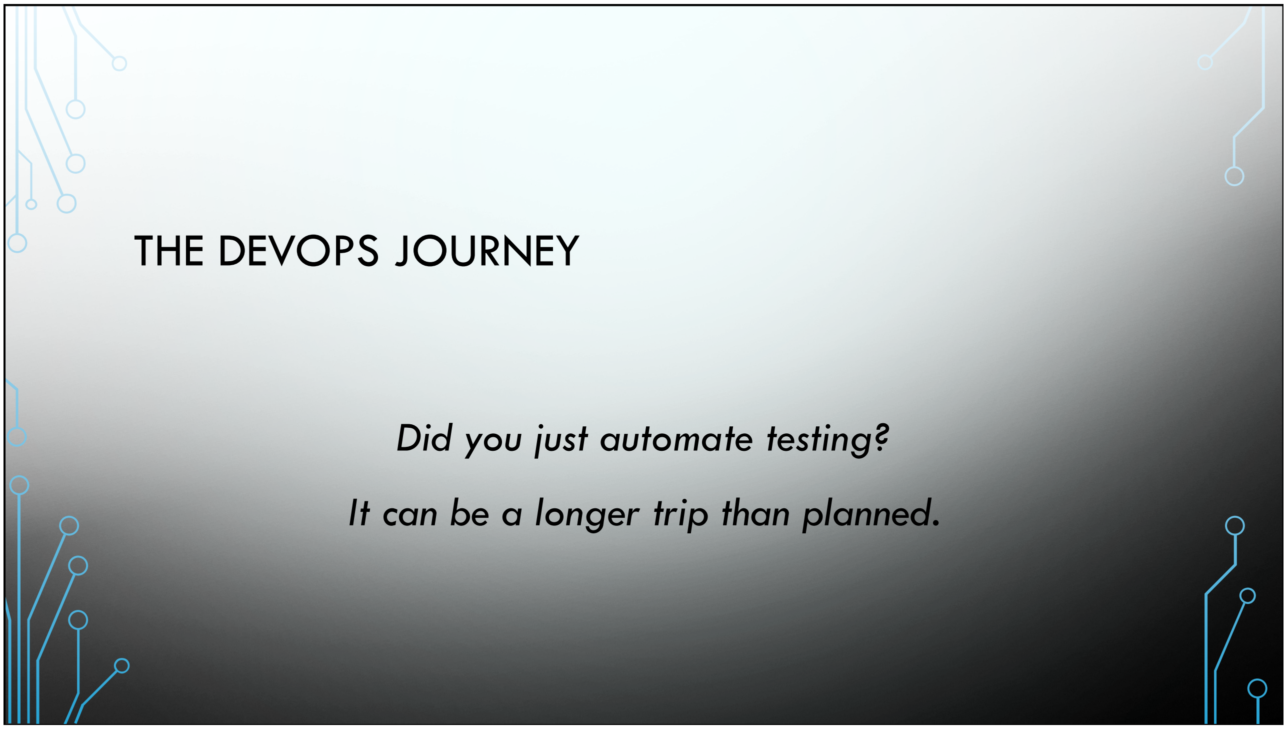
TECHNICAL INTELLIGENCE MANAGER,

DELPHIX

**Kellyn Pot'Vin-Gorman**

**Technical Intelligence Manager, Delphix**

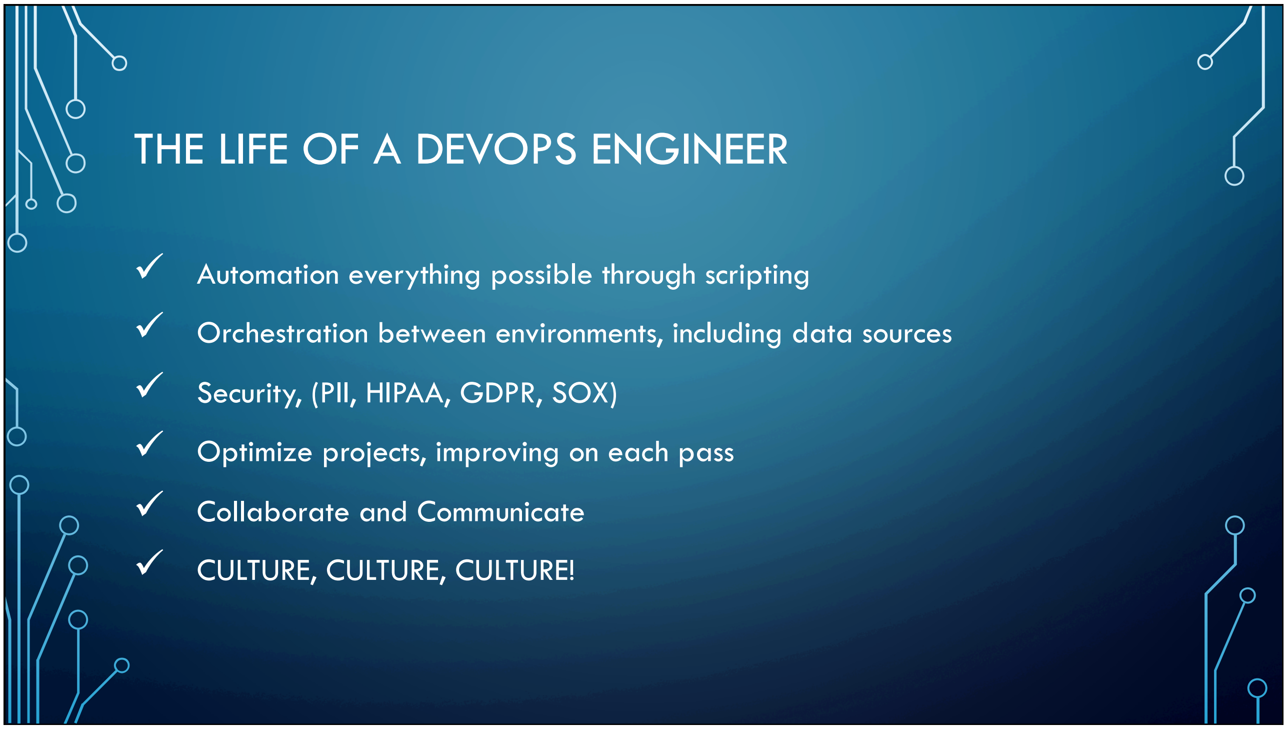
- Multi-platform DBA, (Oracle, MSSQL, MySQL, Sybase, PostgreSQL, Informix…)
- Oracle ACE Director, (Alumni)
- Oak Table Network Member
- Idera ACE 2018
- APEX Women in Technology Award, CTA
- STEM education with Raspberry Pi and Python, including DevOxx4Kids, Oracle Education Foundation and Girl Geek Dinners
- President, Rocky Mtn Oracle User Group
- President, Denver SQL Server User Group
- Technical author, instructor and presenter.
- Author, blogger, (http://dbakevlar.com)



## THE DEVOPS JOURNEY

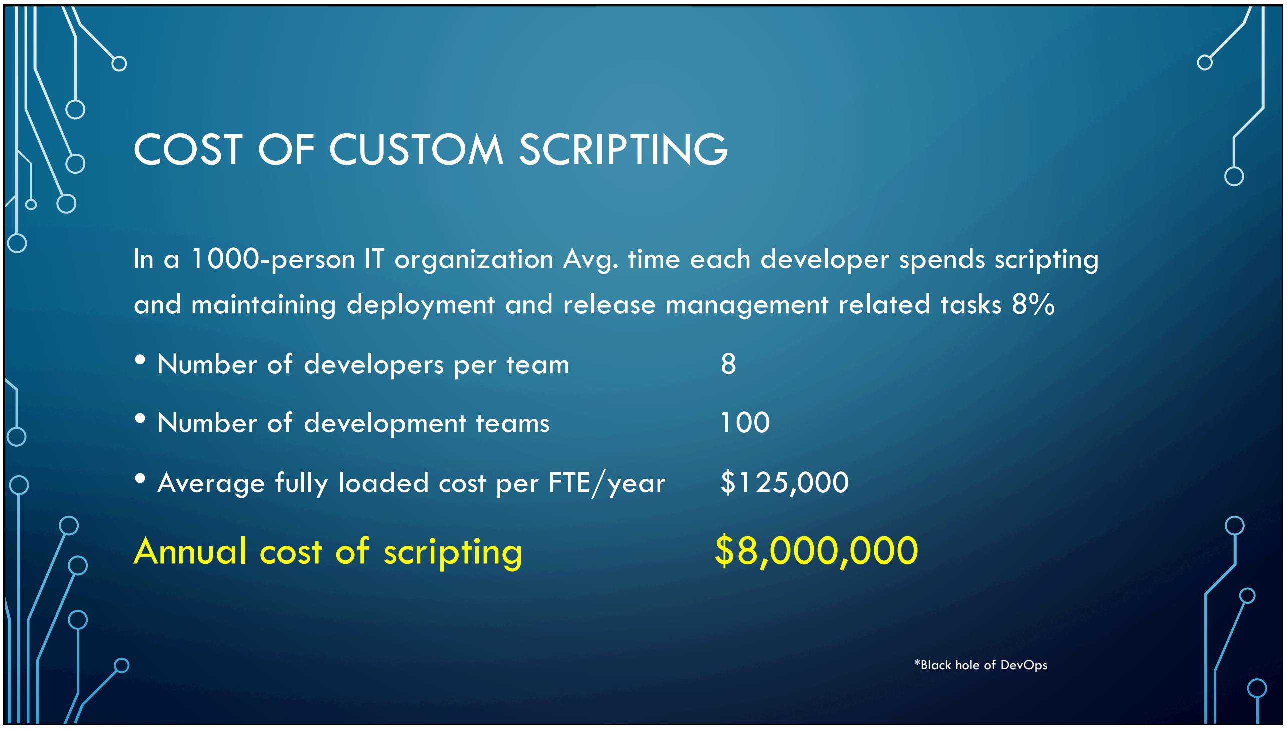
*Did you just automate testing?*

*It can be a longer trip than planned.*



## THE LIFE OF A DEVOPS ENGINEER

- ✓ Automation everything possible through scripting
- ✓ Orchestration between environments, including data sources
- ✓ Security, (PII, HIPAA, GDPR, SOX)
- ✓ Optimize projects, improving on each pass
- ✓ Collaborate and Communicate
- ✓ CULTURE, CULTURE, CULTURE!

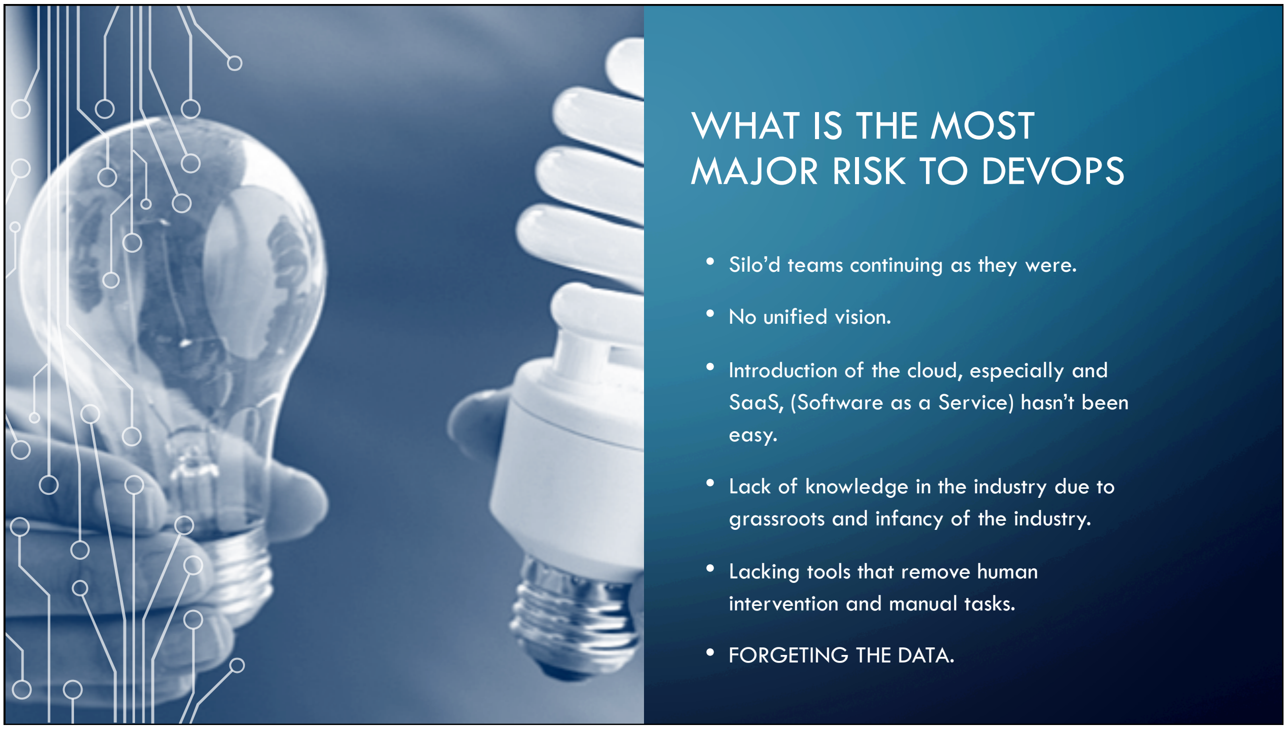


## COST OF CUSTOM SCRIPTING

In a 1000-person IT organization Avg. time each developer spends scripting and maintaining deployment and release management related tasks 8%

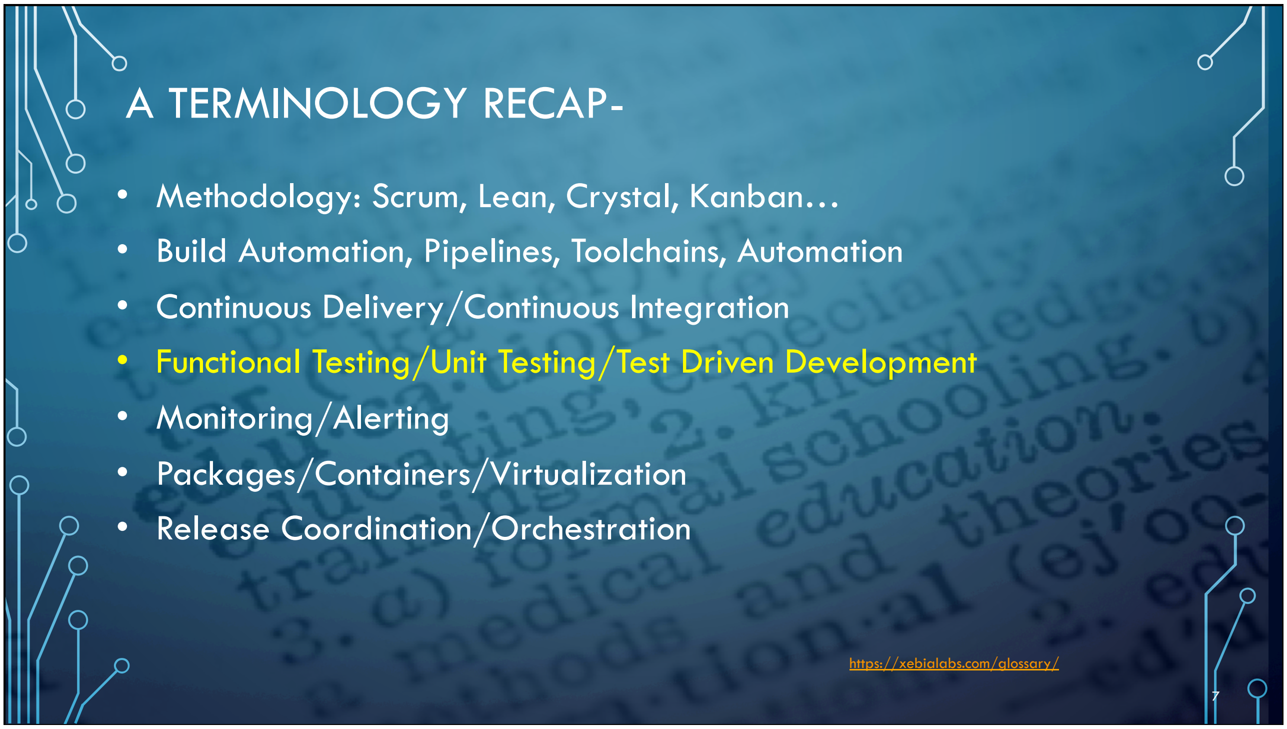
| | |
|---|---|
| • Number of developers per team | 8 |
| • Number of development teams | 100 |
| • Average fully loaded cost per FTE/year | $125,000 |
| Annual cost of scripting | $8,000,000 |

*Black hole of DevOps



## WHAT IS THE MOST MAJOR RISK TO DEVOPS

- Silo'd teams continuing as they were.
- No unified vision.
- Introduction of the cloud, especially and SaaS, (Software as a Service) hasn't been easy.
- Lack of knowledge in the industry due to grassroots and infancy of the industry.
- Lacking tools that remove human intervention and manual tasks.
- FORGETING THE DATA.



## A TERMINOLOGY RECAP-

- Methodology: Scrum, Lean, Crystal, Kanban…
- Build Automation, Pipelines, Toolchains, Automation
- Continuous Delivery/Continuous Integration
- Functional Testing/Unit Testing/Test Driven Development
- Monitoring/Alerting
- Packages/Containers/Virtualization
- Release Coordination/Orchestration

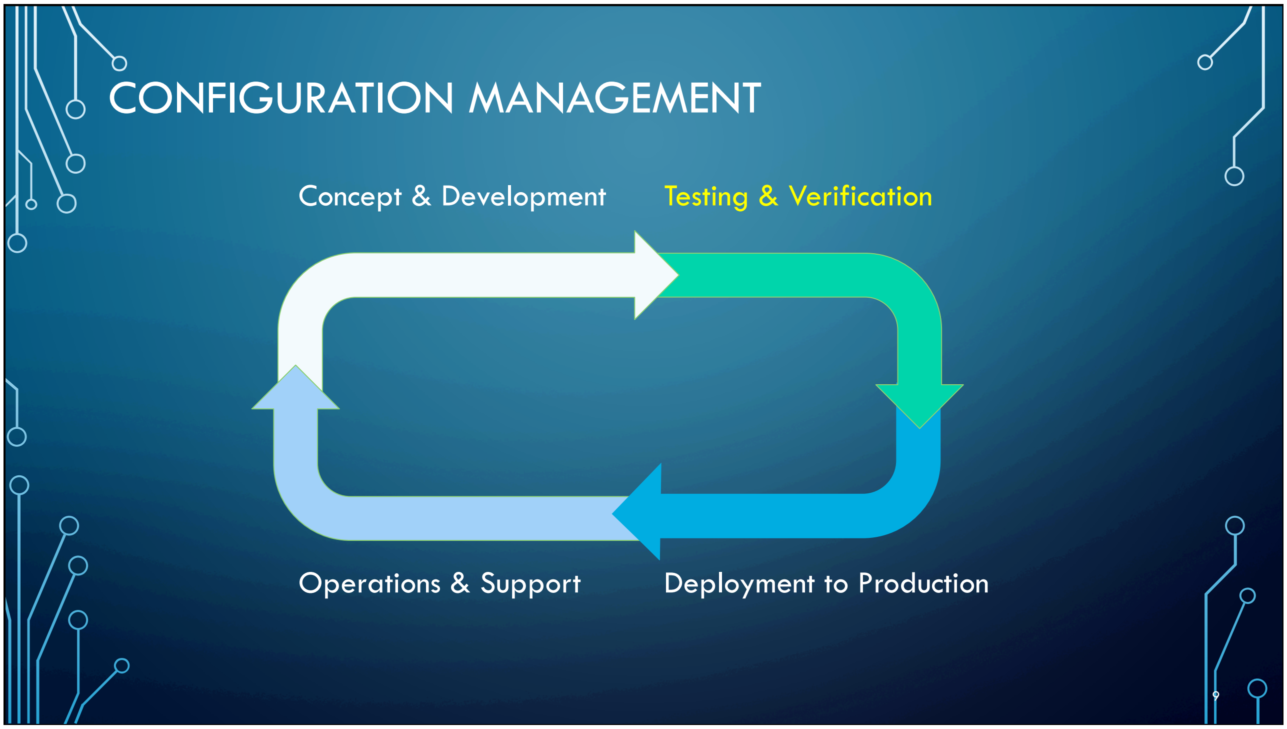
https://xebialabs.com/glossary/

7

## BUILD AUTOMATION

Requires scripting and automation experience for development, testing, releasing, monitoring and alerting.

8



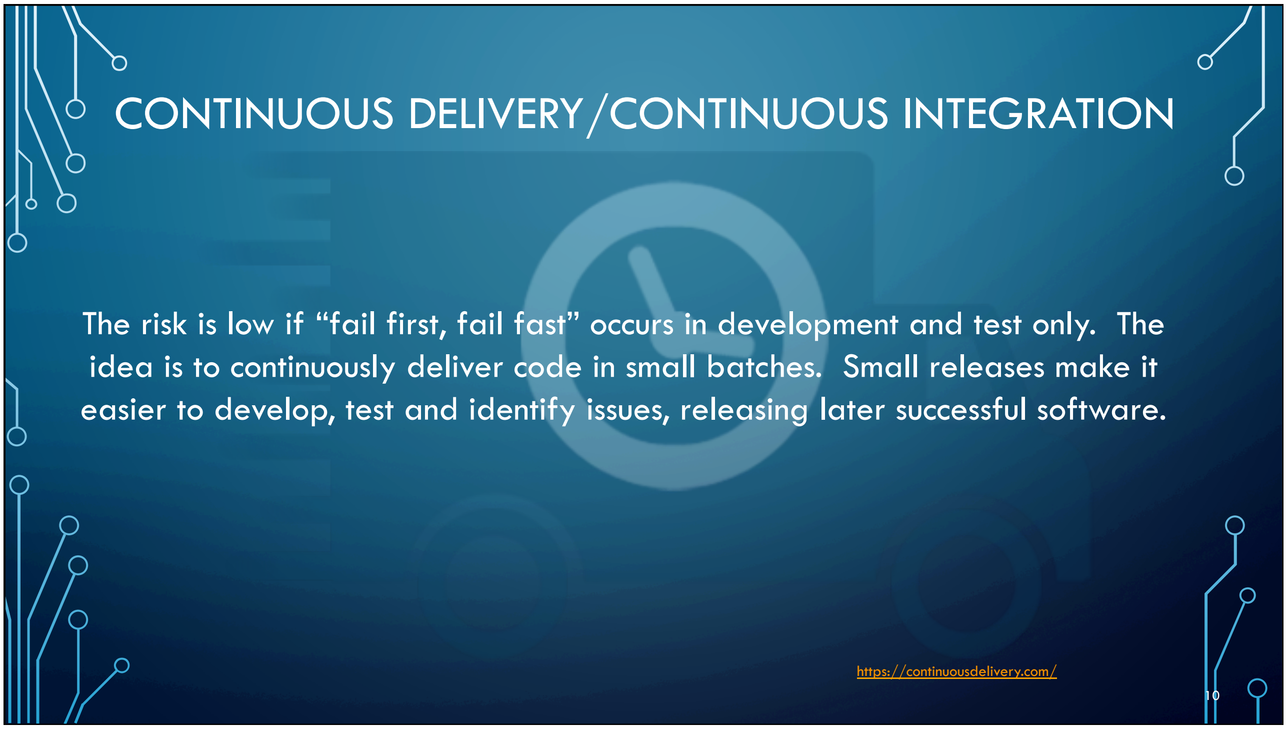
# CONFIGURATION MANAGEMENT

Concept & Development

Testing & Verification

Operations & Support

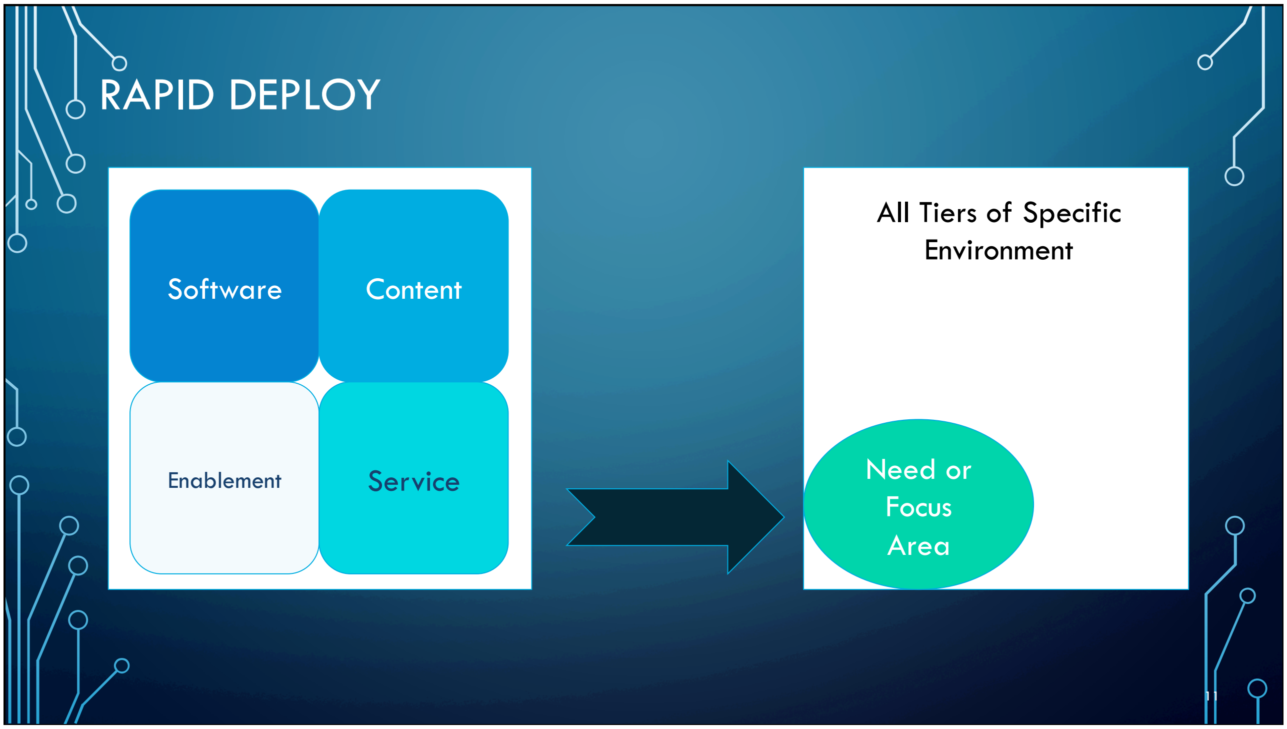
Deployment to Production



# CONTINUOUS DELIVERY/CONTINUOUS INTEGRATION

The risk is low if “fail first, fail fast” occurs in development and test only. The idea is to continuously deliver code in small batches. Small releases make it easier to develop, test and identify issues, releasing later successful software.

https://continuousdelivery.com/



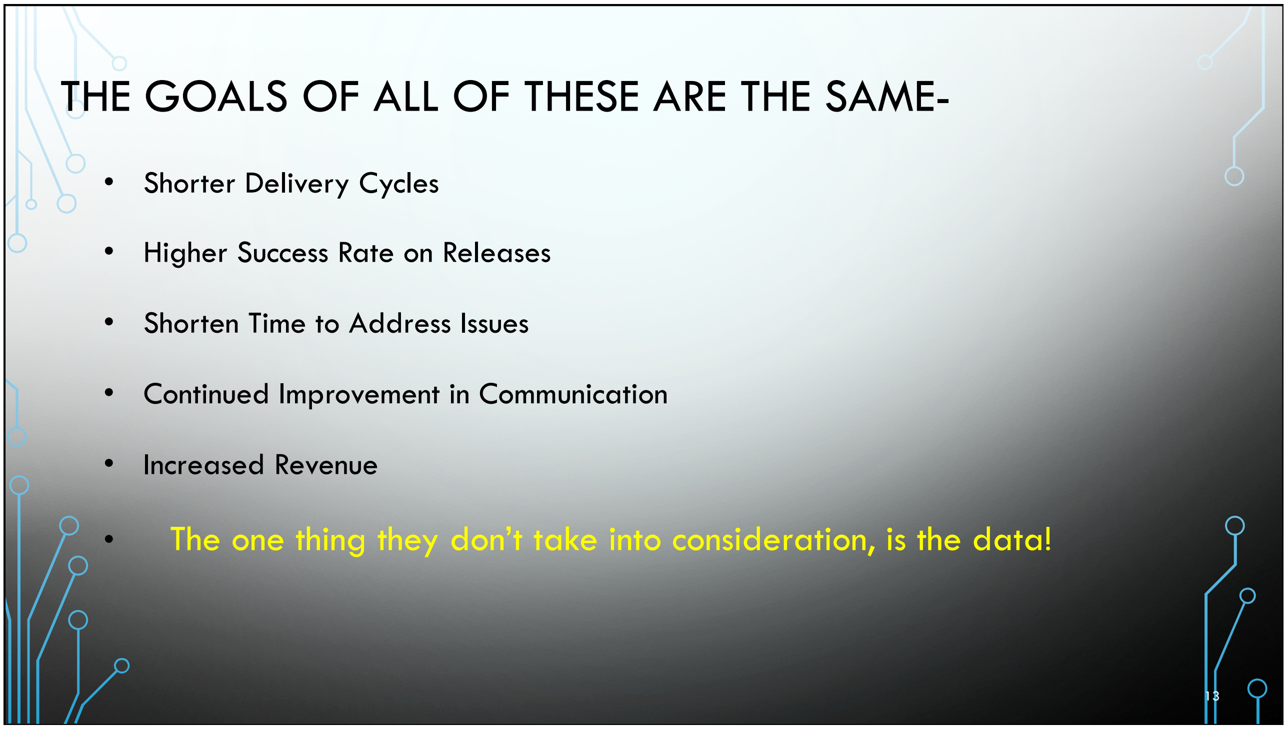
## RAPID DEPLOY

11

## RELEASE COORDINATION/ORCHESTRATION

Release coordination and orchestration is like a symphony. If the coordination of the instruments are done well, the music will be released in a melodious and successful manner. If it's not, then it can be disastrous.

12



## THE GOALS OF ALL OF THESE ARE THE SAME-

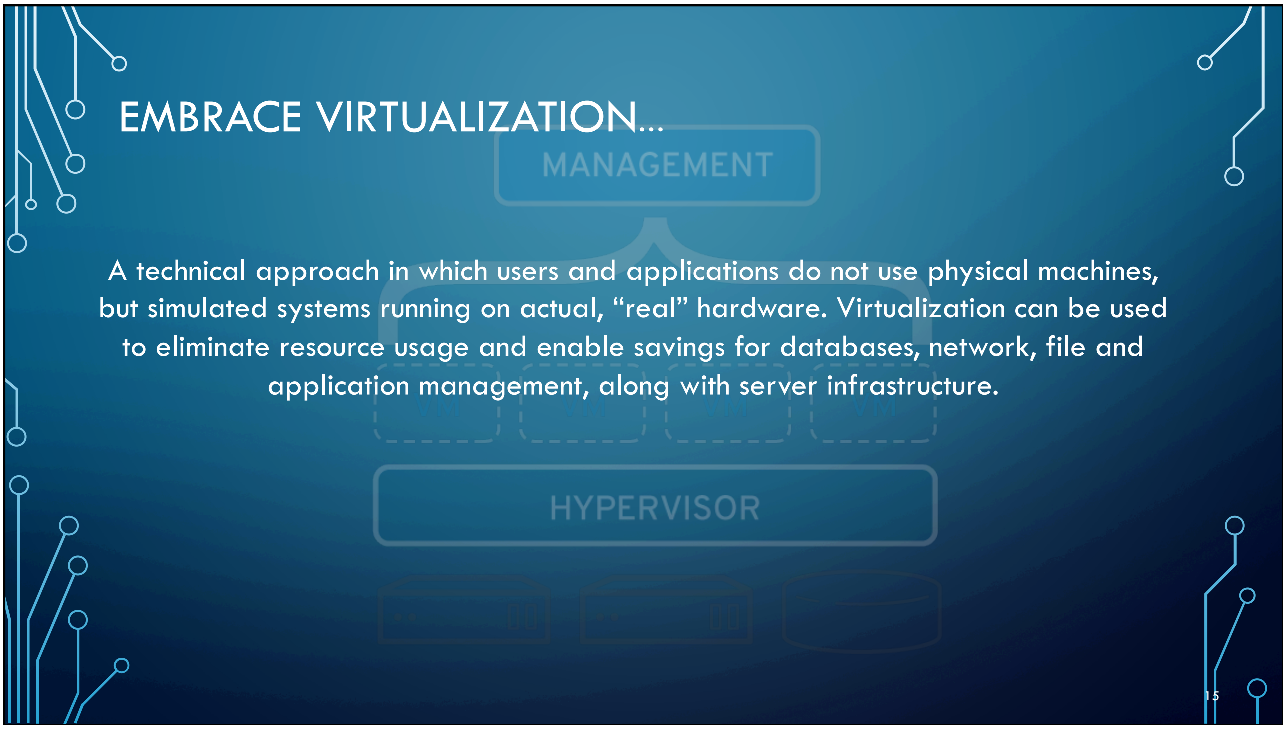
- Shorter Delivery Cycles
- Higher Success Rate on Releases
- Shorten Time to Address Issues
- Continued Improvement in Communication
- Increased Revenue
- The one thing they don't take into consideration, is the data!

13

## LET'S TALK DATA GRAVITY AND IT'S IMPACT

- Data is HEAVY.
- Relational databases, (or any data sources) are heavily weighted and can be impactful to DevOps implementations.

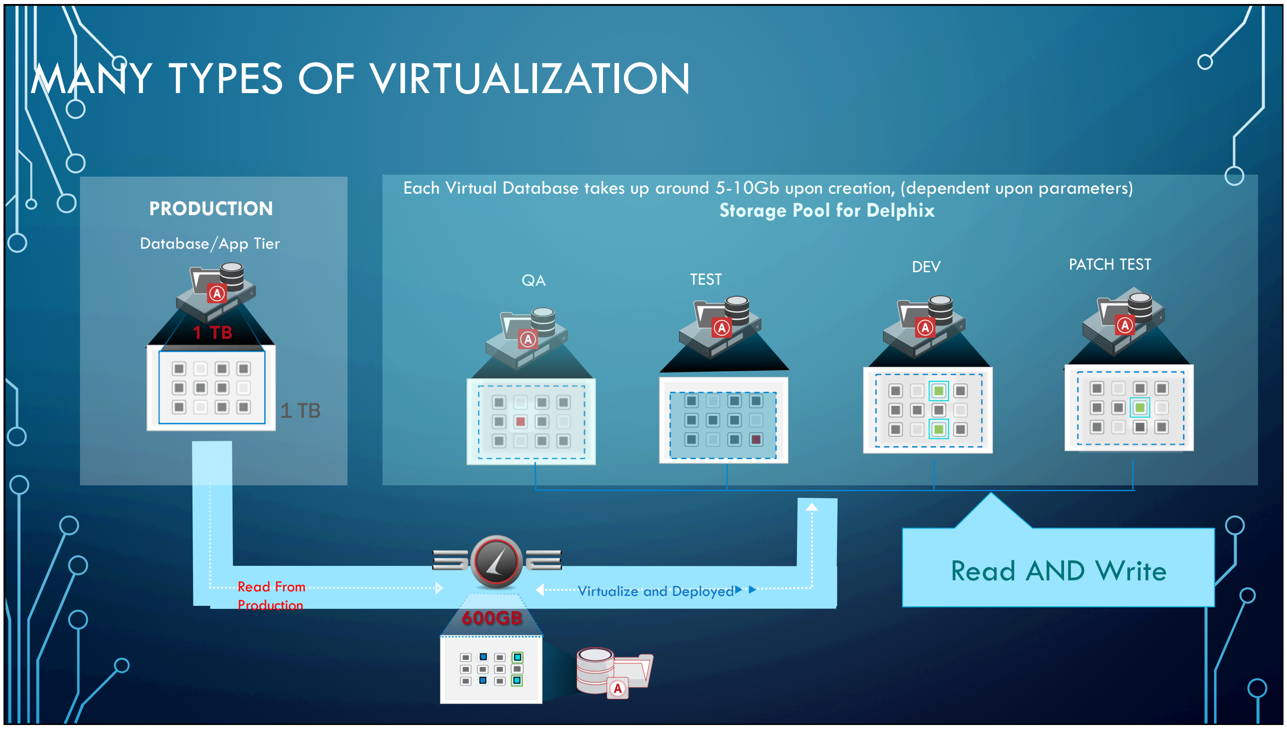
14



## EMBRACE VIRTUALIZATION…

A technical approach in which users and applications do not use physical machines, but simulated systems running on actual, "real" hardware. Virtualization can be used to eliminate resource usage and enable savings for databases, network, file and application management, along with server infrastructure.

MANAGEMENT
HYPERVISOR
15



## MANY TYPES OF VIRTUALIZATION

**PRODUCTION**

Database/App Tier

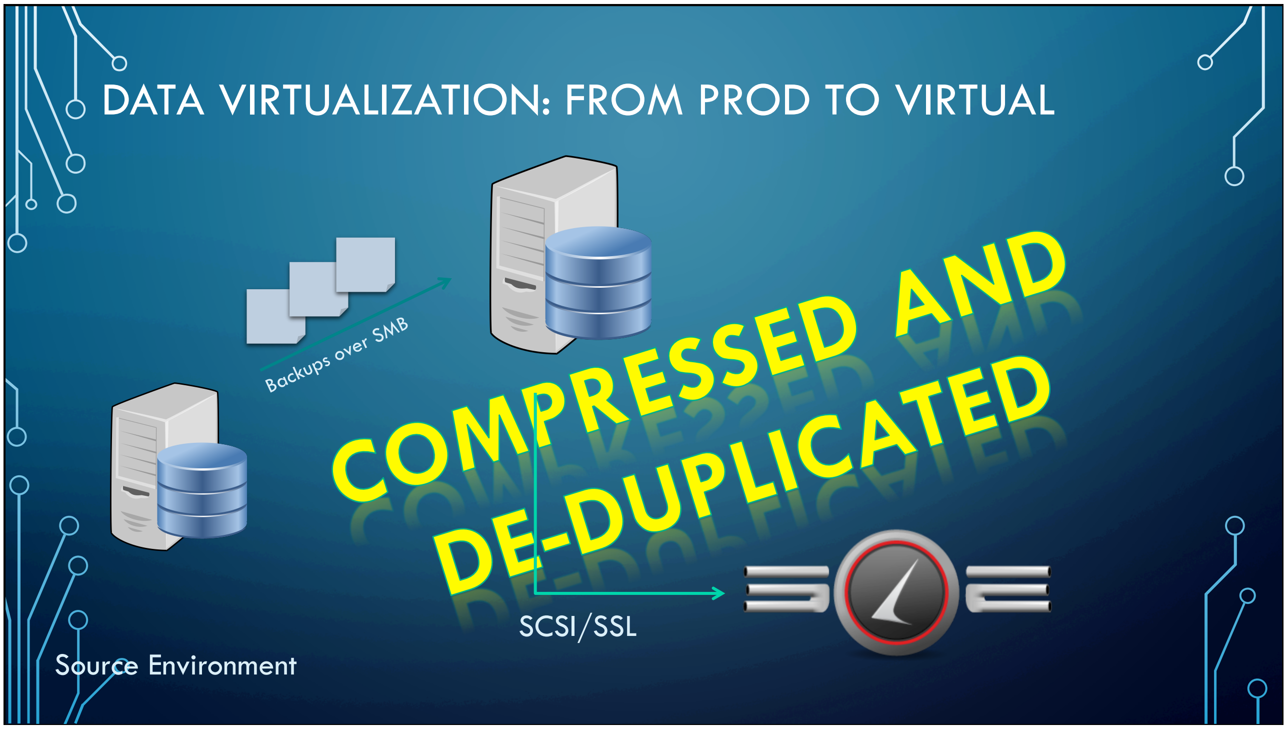
1 TB

1 TB

Each Virtual Database takes up around 5-10Gb upon creation, (dependent upon parameters)

**Storage Pool for Delphix**

Read AND Write



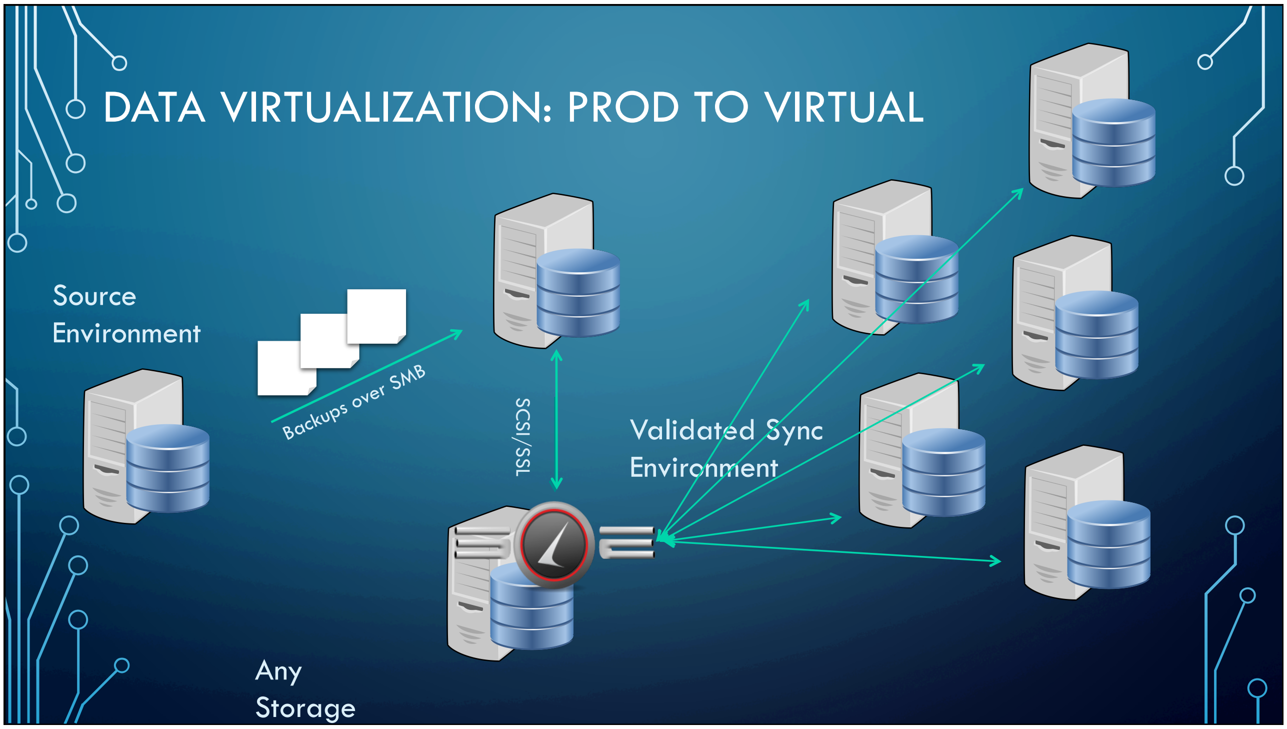
## DATA VIRTUALIZATION: FROM PROD TO VIRTUAL

Backups over SMB

COMPRESSED AND DE-DUPLICATED

SCSI/SSL

Source Environment



## DATA VIRTUALIZATION: PROD TO VIRTUAL

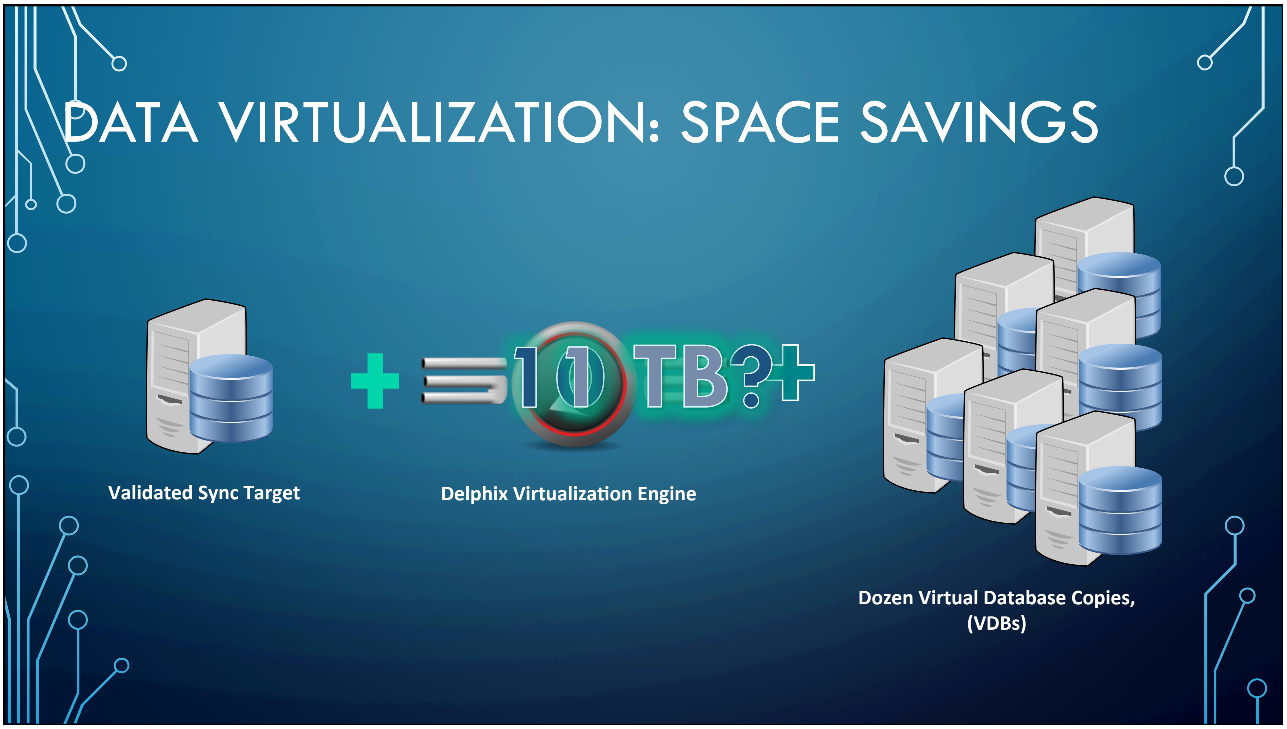
Source Environment

Backups over SMB

SCSI/SSL

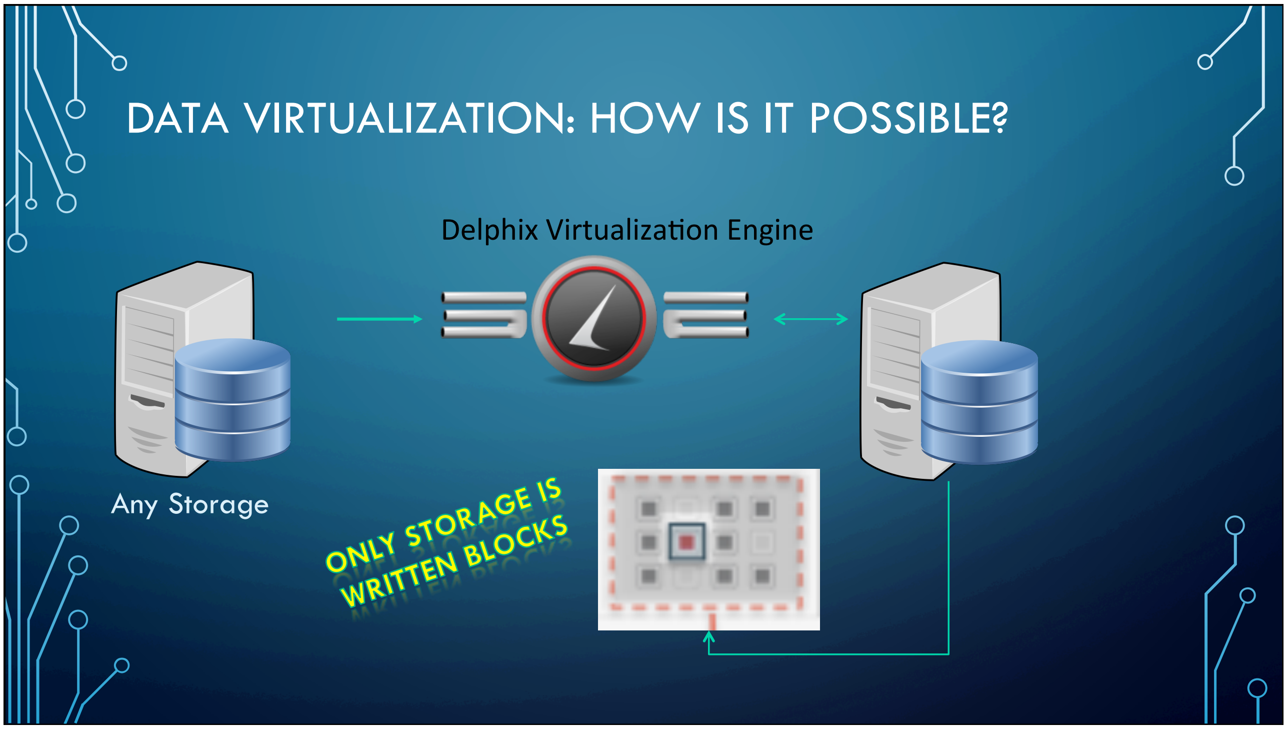
Validated Sync Environment

Any Storage

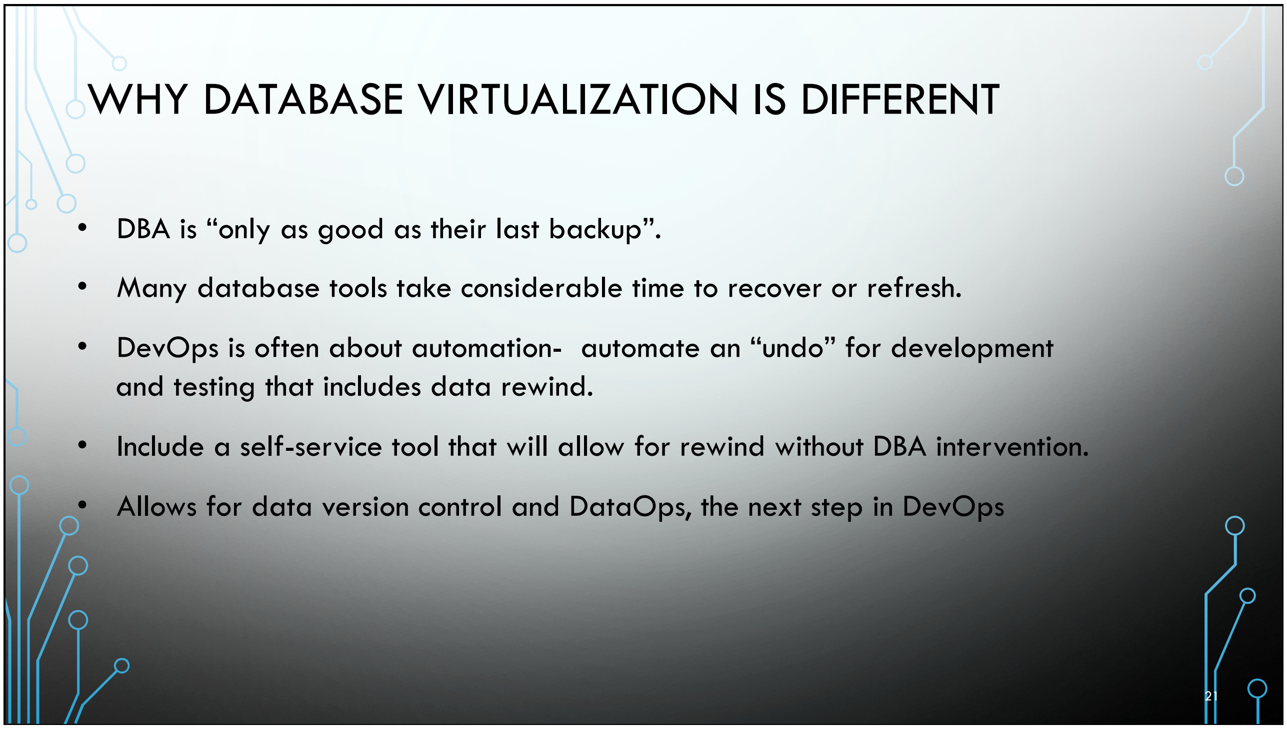


## DATA VIRTUALIZATION: SPACE SAVINGS

+
10 TB?+
Validated Sync Target
Delphix Virtualization Engine
Dozen Virtual Database Copies, (VDBs)



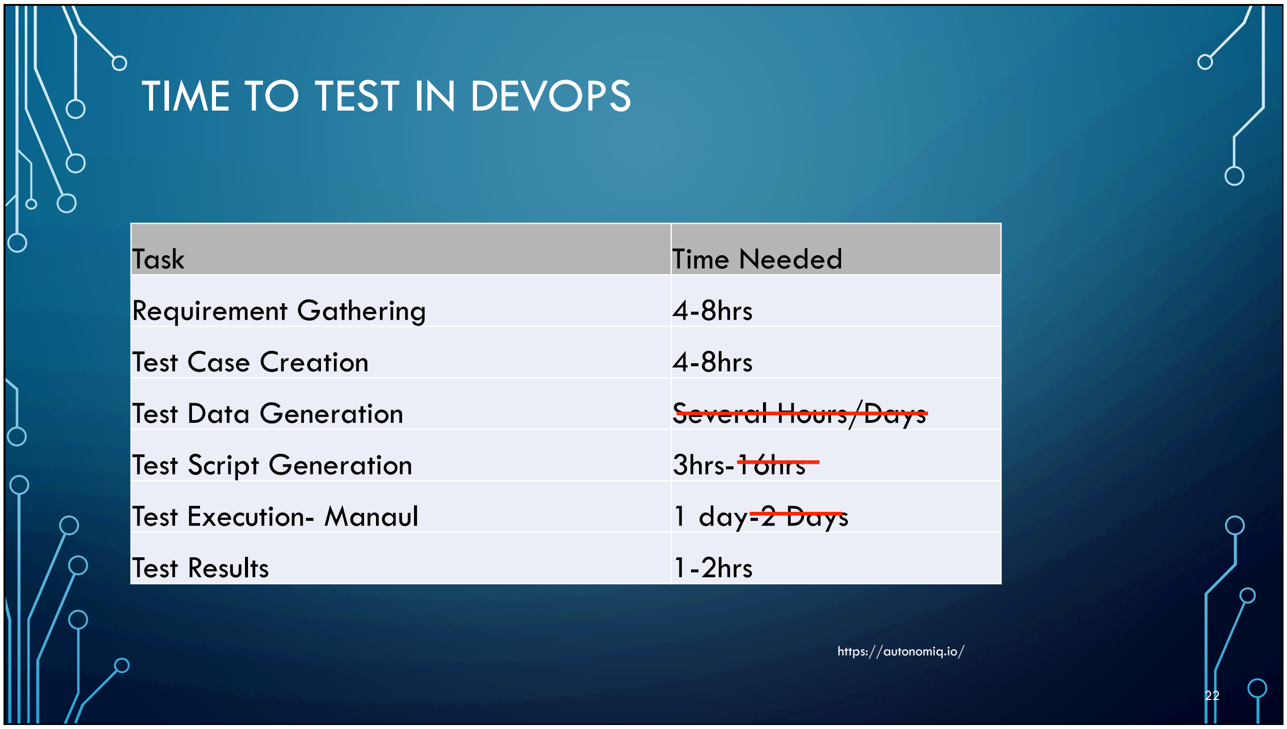
## DATA VIRTUALIZATION: HOW IS IT POSSIBLE?



## WHY DATABASE VIRTUALIZATION IS DIFFERENT

- DBA is “only as good as their last backup”.
- Many database tools take considerable time to recover or refresh.
- DevOps is often about automation-  automate an “undo” for development and testing that includes data rewind.
- Include a self-service tool that will allow for rewind without DBA intervention.
- Allows for data version control and DataOps, the next step in DevOps

21

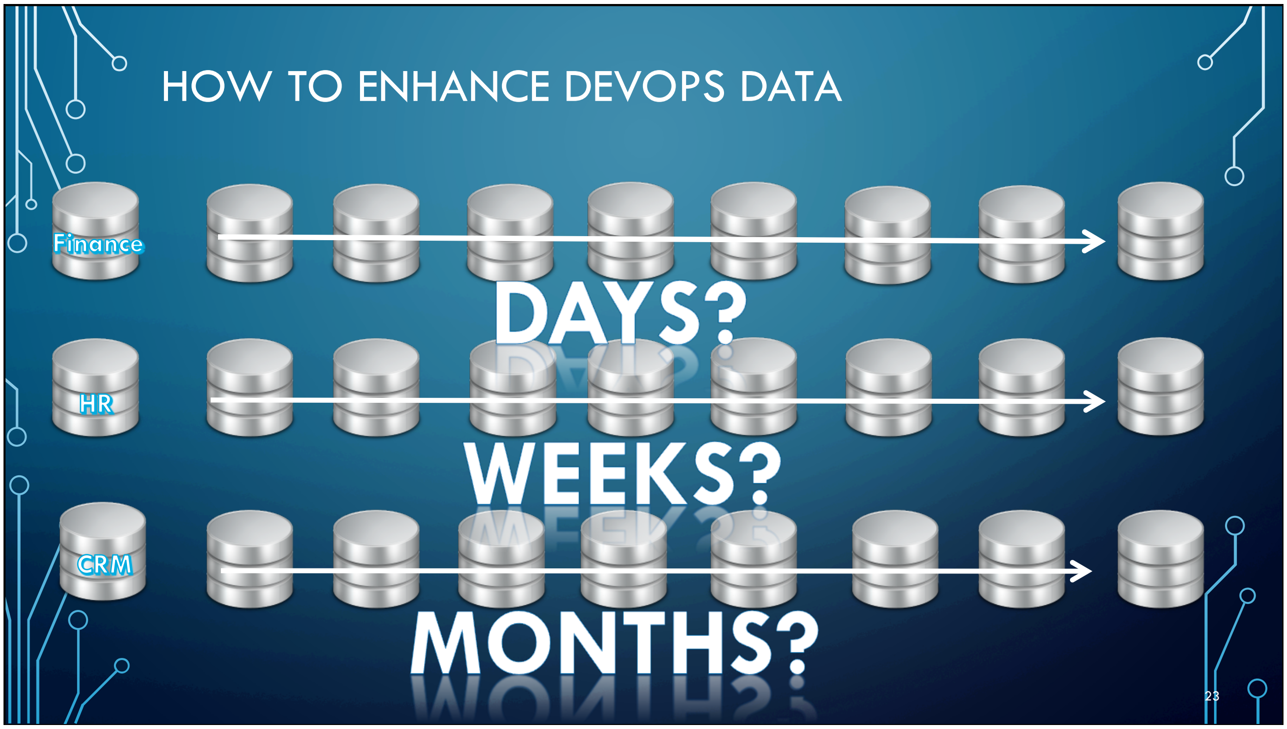


## TIME TO TEST IN DEVOPS

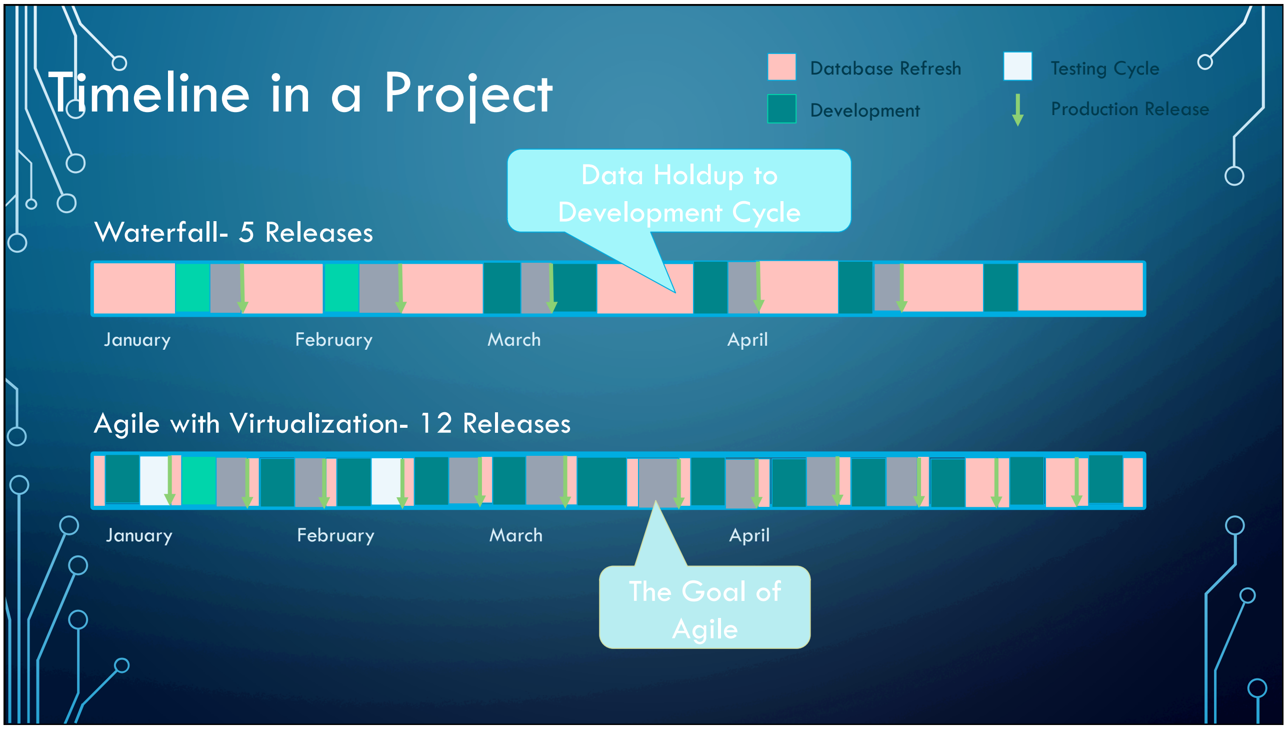
| Task | Time Needed |
|---|---|
| Requirement Gathering | 4-8hrs |
| Test Case Creation | 4-8hrs |
| Test Data Generation | ~~Several Hours/Days~~ |
| Test Script Generation | 3hrs-~~16hrs~~ |
| Test Execution- Manaul | 1 day-~~2 Days~~ |
| Test Results | 1-2hrs |

https://autonomiq.io/

22



## HOW TO ENHANCE DEVOPS DATA



## Timeline in a Project

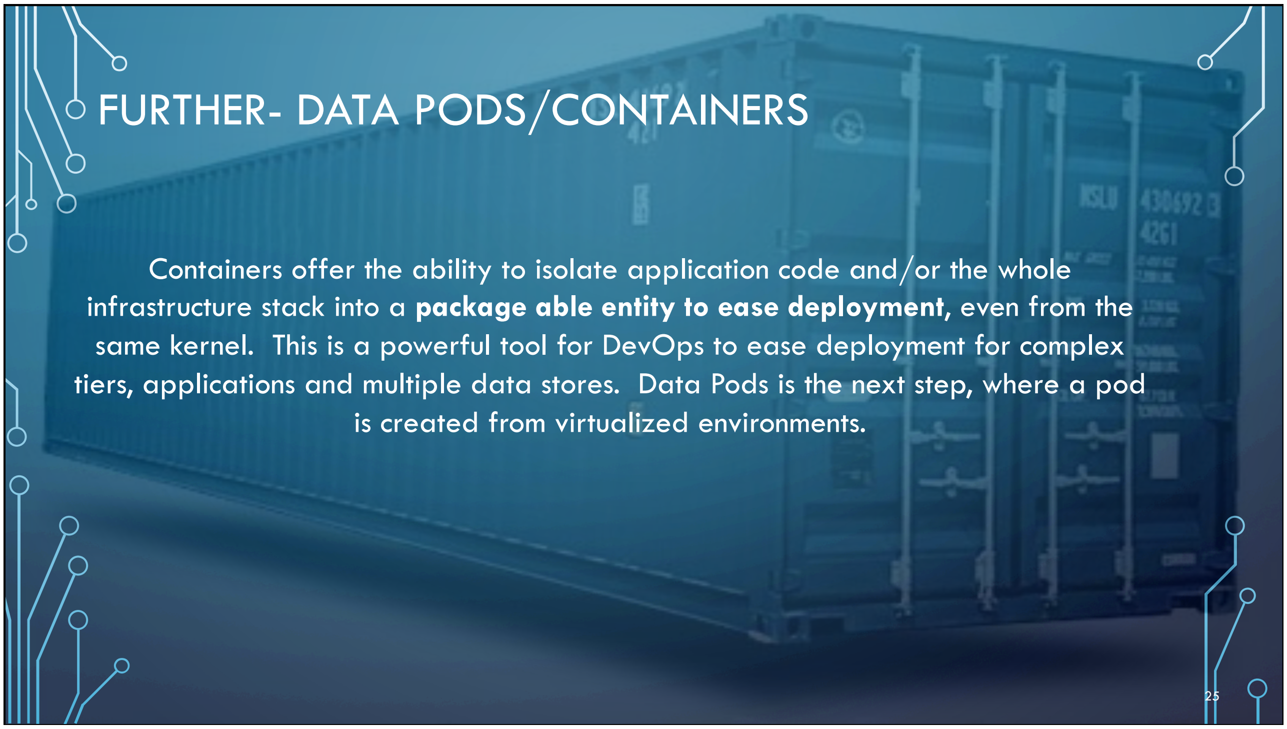
🟥 Database Refresh
🟩 Development
□ Testing Cycle
↓ Production Release

Waterfall- 5 Releases

Data Holdup to Development Cycle

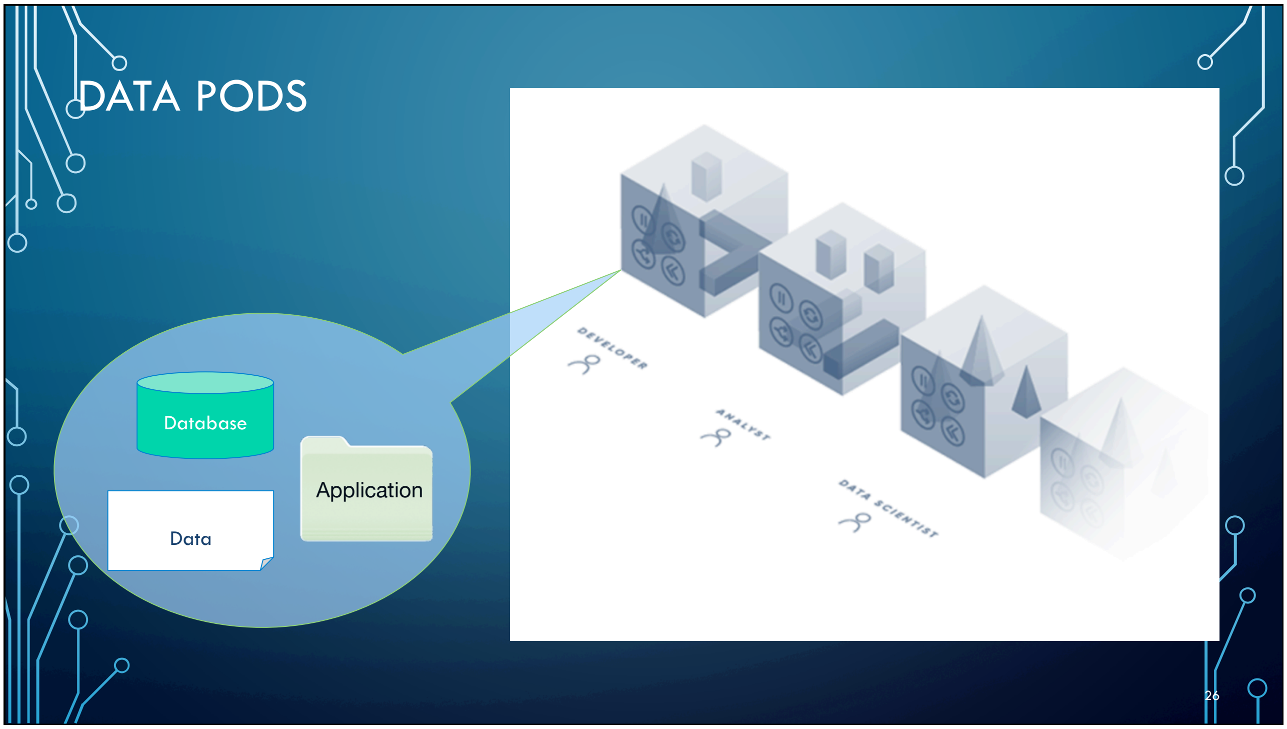
Agile with Virtualization- 12 Releases

The Goal of Agile

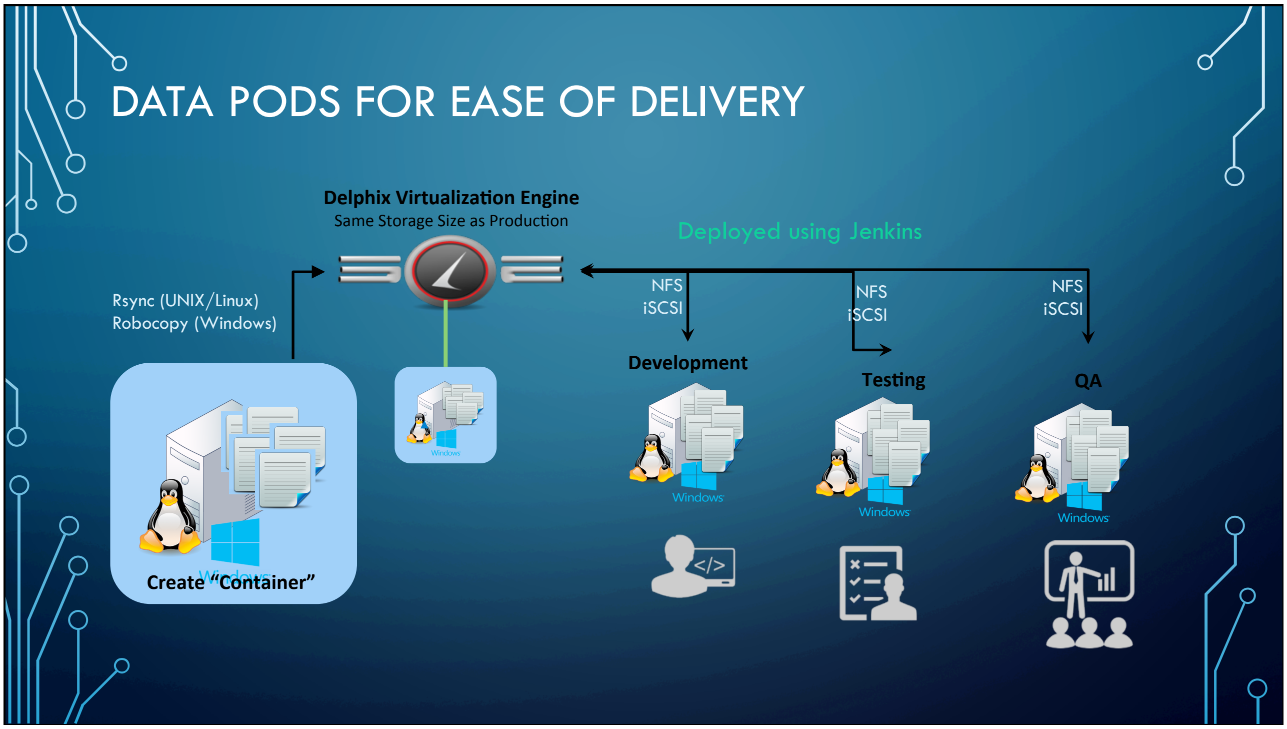


## FURTHER- DATA PODS/CONTAINERS

Containers offer the ability to isolate application code and/or the whole infrastructure stack into a **package able entity to ease deployment,** even from the same kernel.  This is a powerful tool for DevOps to ease deployment for complex tiers, applications and multiple data stores.  Data Pods is the next step, where a pod is created from virtualized environments.



## DATA PODS



# DATA PODS FOR EASE OF DELIVERY

Delphix Virtualization Engine
Same Storage Size as Production

Deployed using Jenkins

Rsync (UNIX/Linux)
Robocopy (Windows)

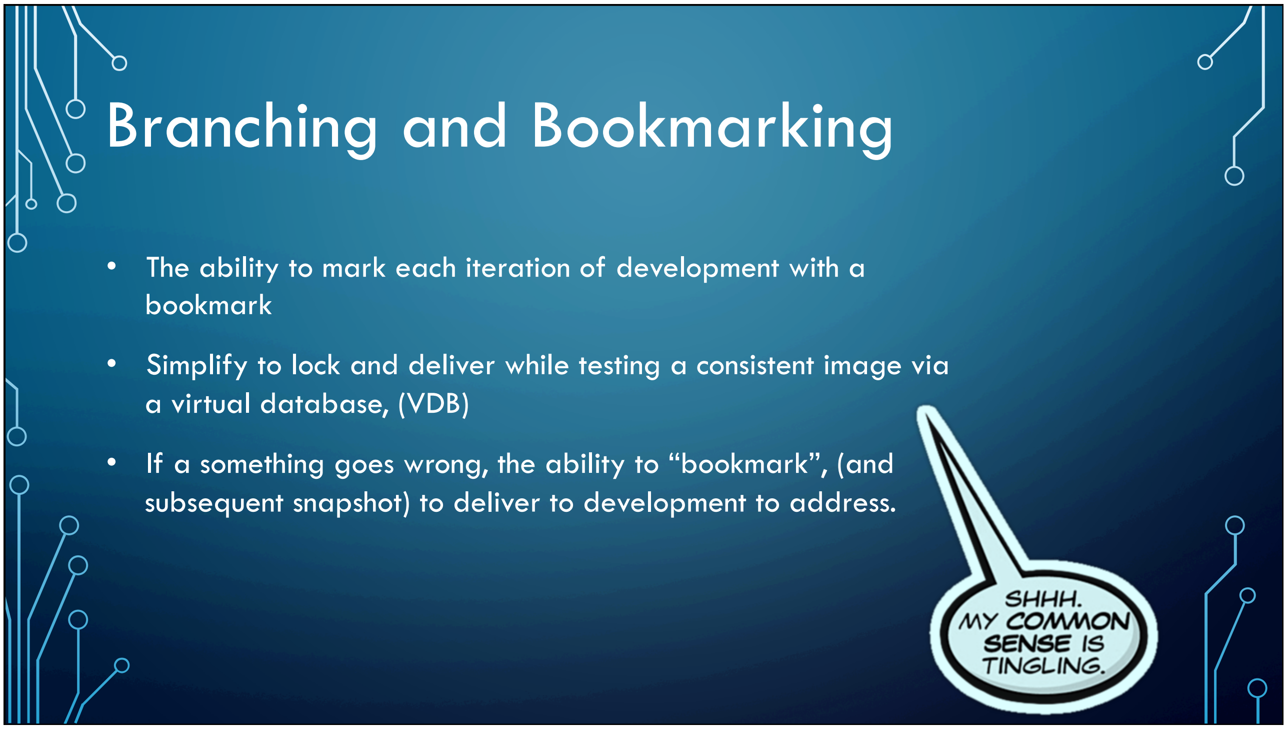
NFS
iSCSI

Development

Testing

QA

Create "Container"

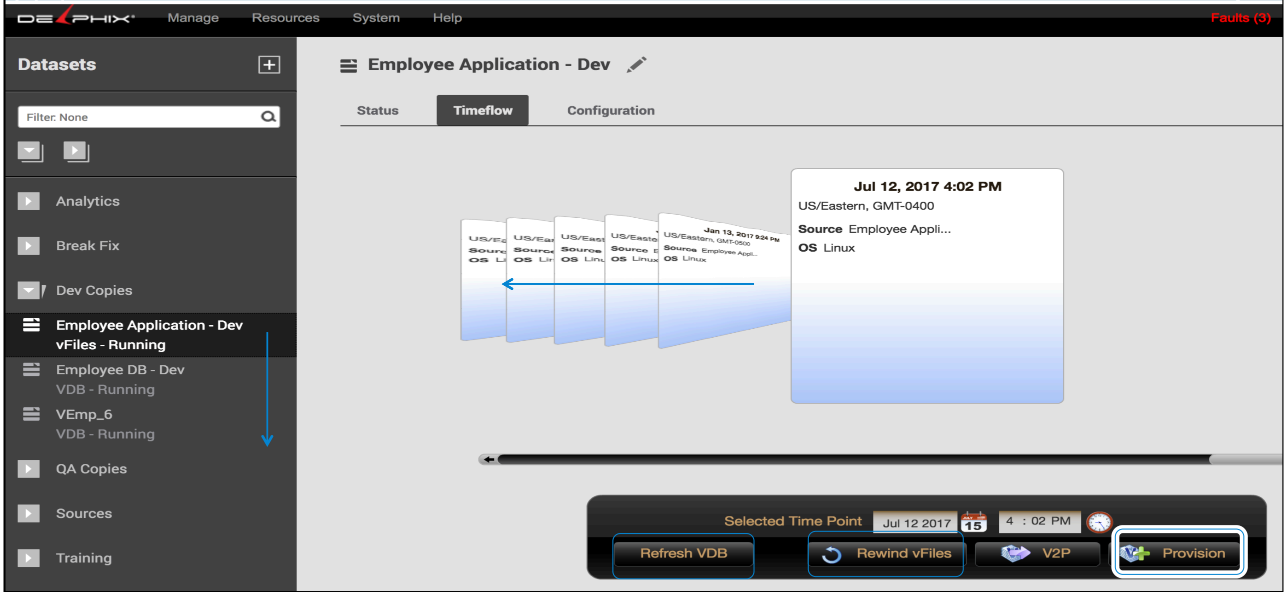


# Branching and Bookmarking

- The ability to mark each iteration of development with a bookmark
- Simplify to lock and deliver while testing a consistent image via a virtual database, (VDB)
- If a something goes wrong, the ability to "bookmark", (and subsequent snapshot) to deliver to development to address.

SHHH. MY COMMON SENSE IS TINGLING.

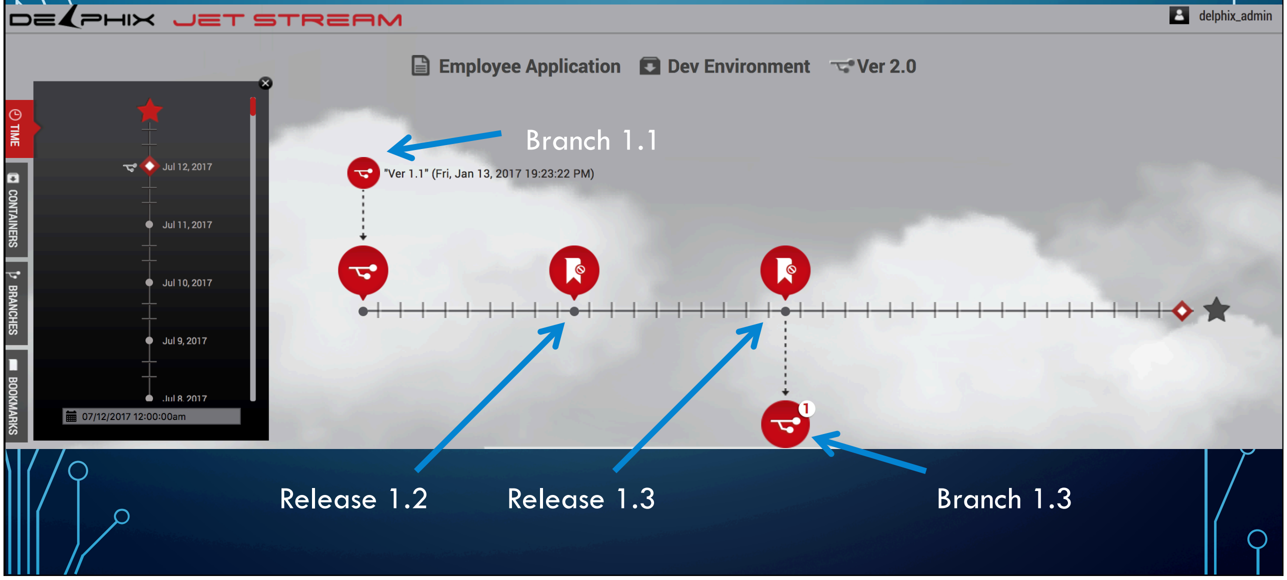
# Administration Ease



Jan 13, 2017 9:24 PM
US/Eastern, GMT-0500
**Source** Employee Appli...
**OS** Linux

Jul 12, 2017 4:02 PM
US/Eastern, GMT-0400
**Source** Employee Appli...
**OS** Linux

# Self-Service



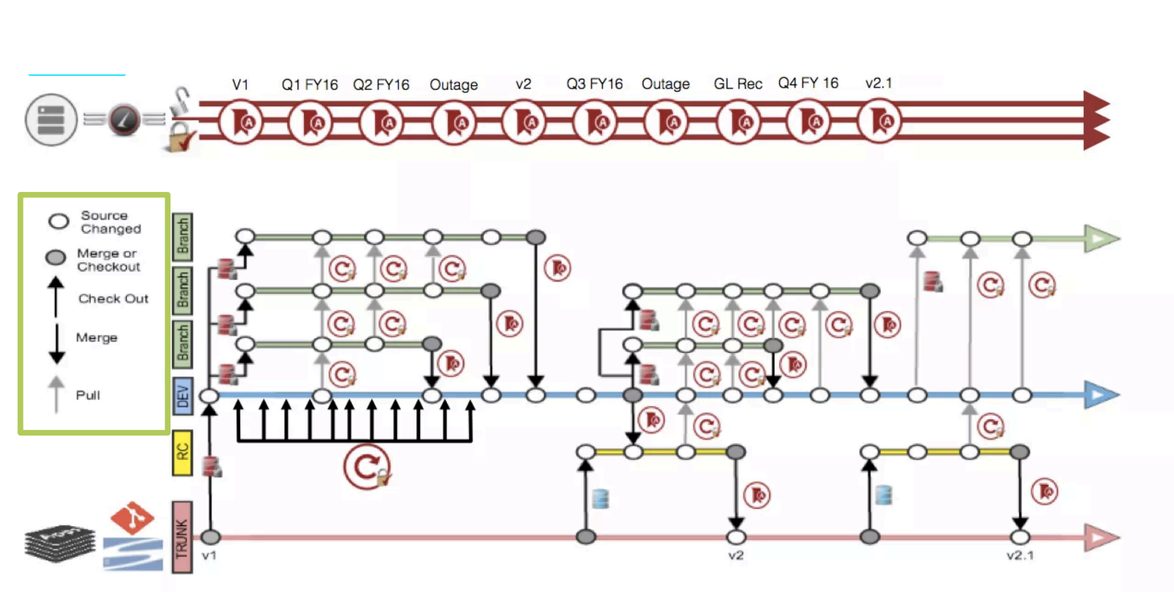
Branch 1.1

Release 1.2

Release 1.3

Branch 1.3

# DevOps and Source Control



V1 Q1 FY16 Q2 FY16 Outage v2 Q3 FY16 Outage GL Rec Q4 FY 16 v2.1

Source Changed

Merge or Checkout

Check Out

Merge

Pull

Branch

Branch

Branch

DEV

RC

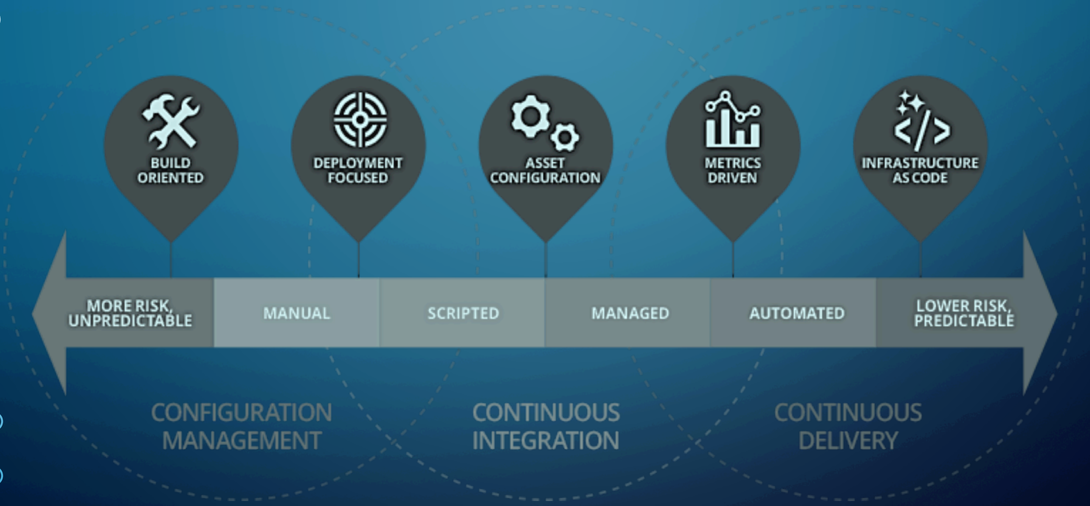
TRUNK

v1

v2

v2.1

# MATURITY



BUILD ORIENTED

DEPLOYMENT FOCUSED

ASSET CONFIGURATION

METRICS DRIVEN

INFRASTRUCTURE AS CODE

MORE RISK, UNPREDICTABLE

MANUAL

SCRIPTED

MANAGED

AUTOMATED

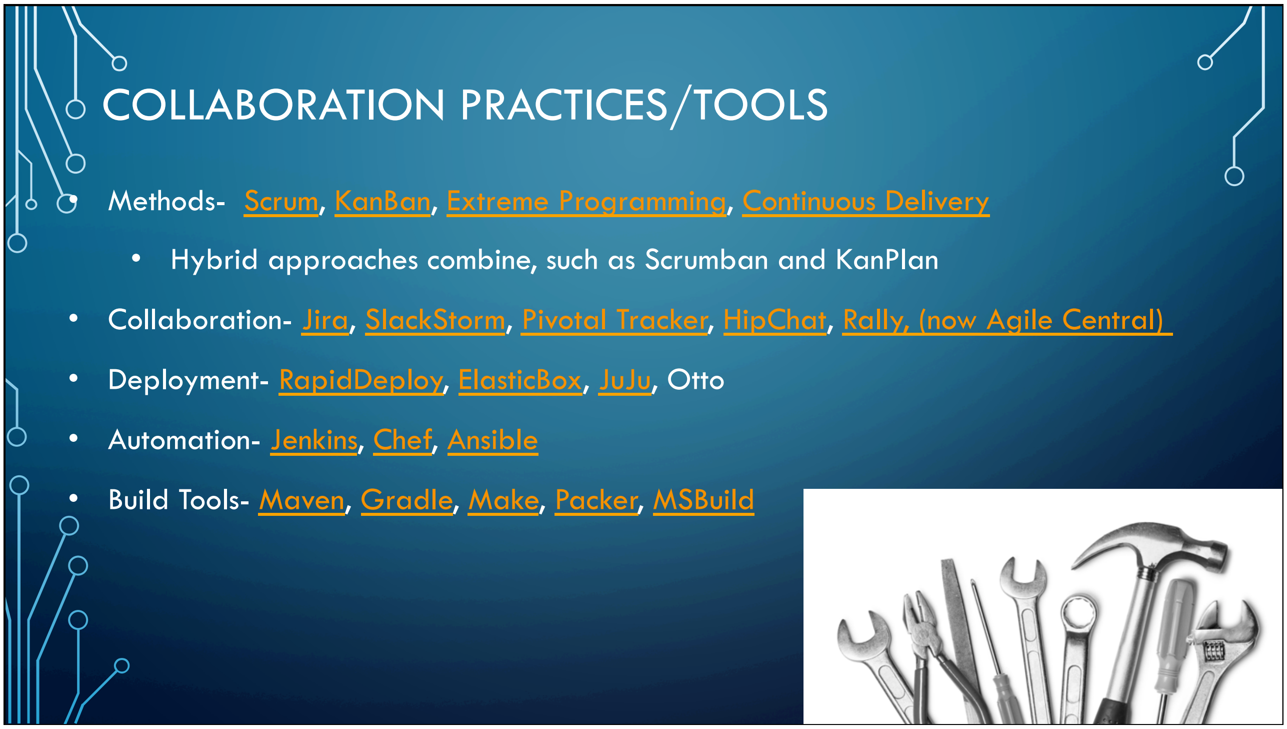
LOWER RISK, PREDICTABLE

CONFIGURATION MANAGEMENT

CONTINUOUS INTEGRATION

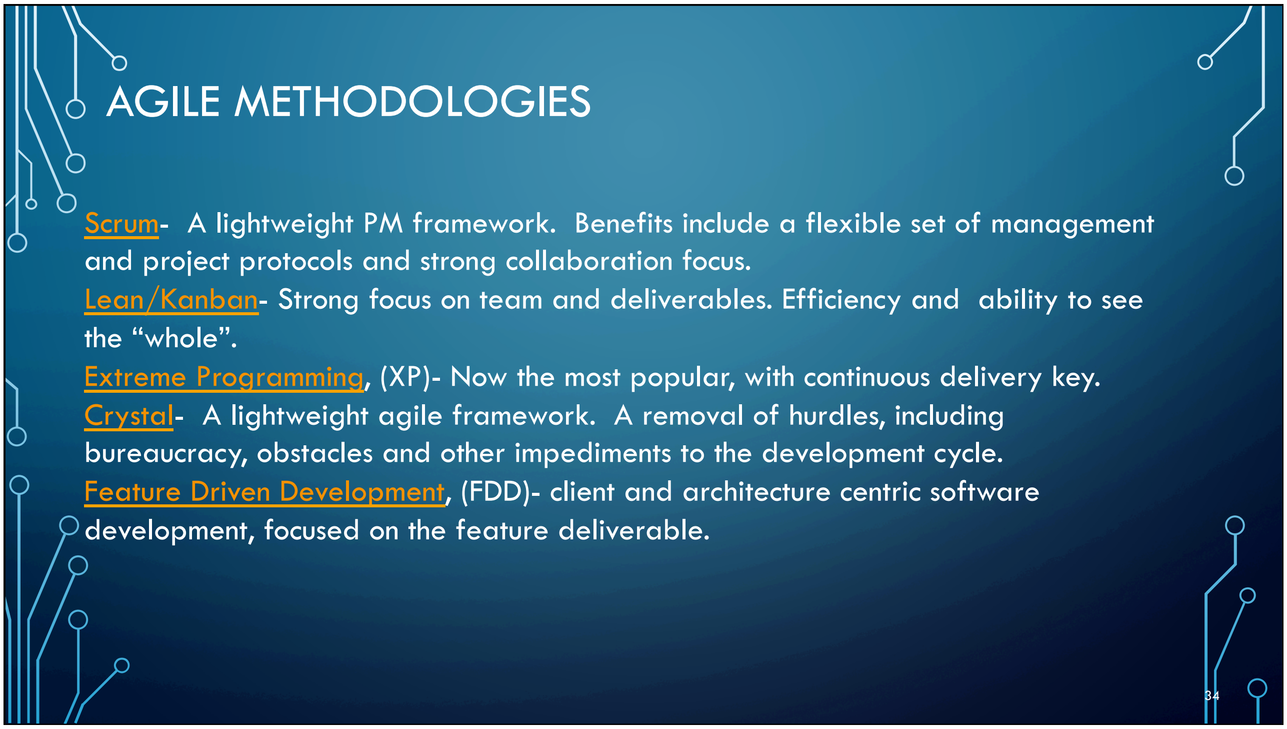
CONTINUOUS DELIVERY

32



# COLLABORATION PRACTICES/TOOLS

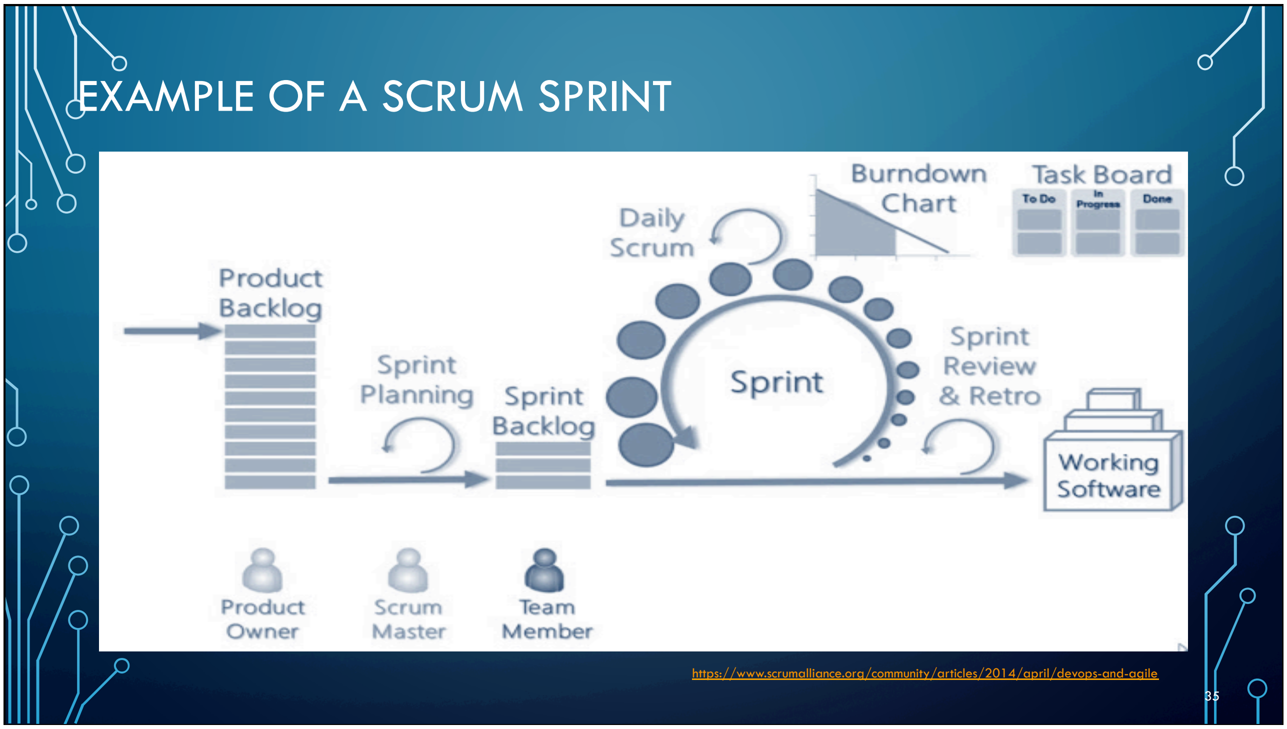
- Methods- Scrum, KanBan, Extreme Programming, Continuous Delivery
  - Hybrid approaches combine, such as Scrumban and KanPlan
- Collaboration- Jira, SlackStorm, Pivotal Tracker, HipChat, Rally, (now Agile Central)
- Deployment- RapidDeploy, ElasticBox, JuJu, Otto
- Automation- Jenkins, Chef, Ansible
- Build Tools- Maven, Gradle, Make, Packer, MSBuild



# AGILE METHODOLOGIES

Scrum- A lightweight PM framework. Benefits include a flexible set of management and project protocols and strong collaboration focus.
Lean/Kanban- Strong focus on team and deliverables. Efficiency and ability to see the “whole”.
Extreme Programming, (XP)- Now the most popular, with continuous delivery key.
Crystal- A lightweight agile framework. A removal of hurdles, including bureaucracy, obstacles and other impediments to the development cycle.
Feature Driven Development, (FDD)- client and architecture centric software development, focused on the feature deliverable.

34



## EXAMPLE OF A SCRUM SPRINT

https://www.scrumalliance.org/community/articles/2014/april/devops-and-agile

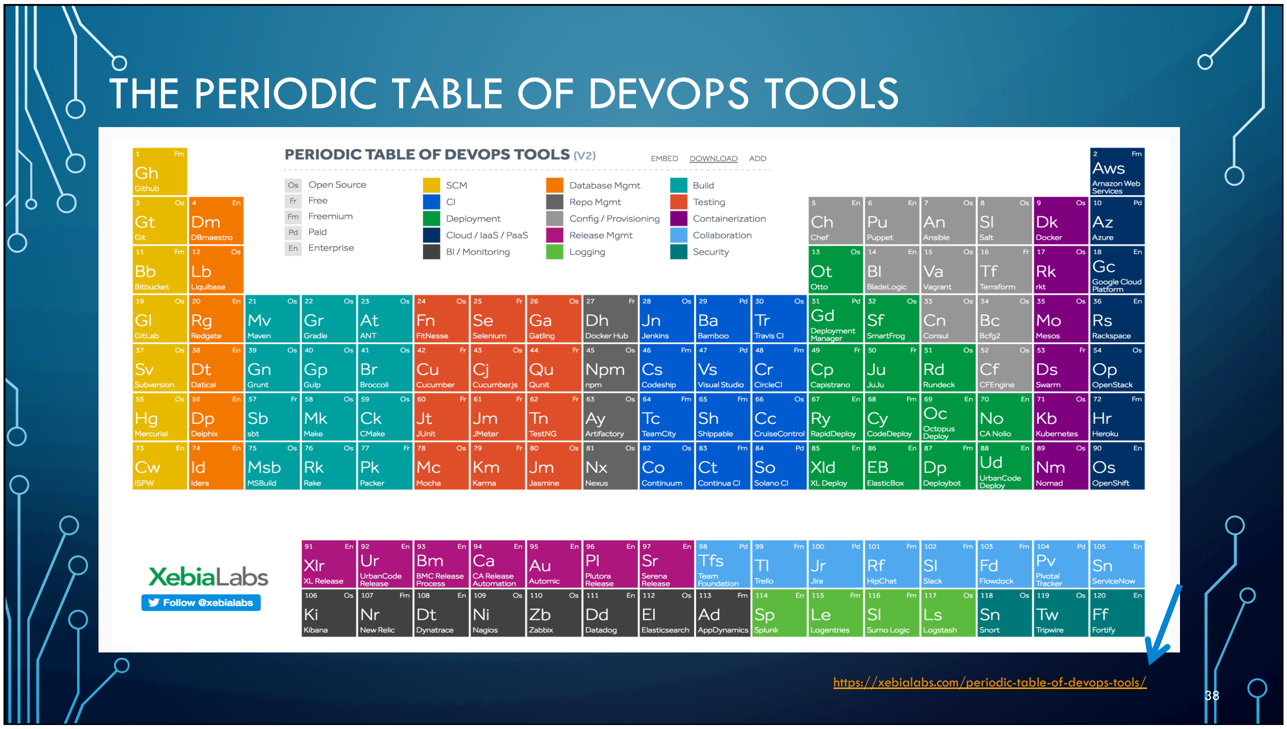
35

## DEVOPS TOOLS

CHEF
JIRA
ZAPTEST
Jenkins
kubernetes
amazon web services
ANSIBLE
Puppet
GitHub
DELPHIX
HashiCorp Terraform

36

# Use the Right Tool for the Job



# THE PERIODIC TABLE OF DEVOPS TOOLS

## PERIODIC TABLE OF DEVOPS TOOLS (V2)

EMBED DOWNLOAD ADD

| | | | | |
|---|---|---|---|---|
| Os | Open Source | SCM | Database Mgmt | Build |
| Fr | Free | CI | Repo Mgmt | Testing |
| Fm | Freemium | Deployment | Config / Provisioning | Containerization |
| Pd | Paid | Cloud / IaaS / PaaS | Release Mgmt | Collaboration |
| En | Enterprise | BI / Monitoring | Logging | Security |

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 Fm<br>Gh<br>Github | | | | | | | | | | | | | | | | | 2 Fm<br>Aws<br>Amazon Web Services |
| 3 Os<br>Gt<br>Git | 4 En<br>Dm<br>DBmaestro | | | | | | | | | | | 5 En<br>Ch<br>Chef | 6 En<br>Pu<br>Puppet | 7 Os<br>An<br>Ansible | 8 Os<br>Sl<br>Salt | 9 Os<br>Dk<br>Docker | 10 Pd<br>Az<br>Azure |
| 11 Fm<br>Bb<br>Bitbucket | 12 Os<br>Lb<br>Liquibase | | | | | | | | | | | 13 Os<br>Ot<br>Otto | 14 En<br>Bl<br>BladeLogic | 15 Os<br>Va<br>Vagrant | 16 Fr<br>Tf<br>Terraform | 17 Os<br>Rk<br>rkt | 18 En<br>Gc<br>Google Cloud Platform |
| 19 Os<br>Gl<br>GitLab | 20 En<br>Rg<br>Redgate | 21 Os<br>Mv<br>Maven | 22 Os<br>Gr<br>Gradle | 23 Os<br>At<br>ANT | 24 Os<br>Fn<br>FitNesse | 25 Fr<br>Se<br>Selenium | 26 Os<br>Ga<br>Gatling | 27 Fr<br>Dh<br>Docker Hub | 28 Os<br>Jn<br>Jenkins | 29 Pd<br>Ba<br>Bamboo | 30 Os<br>Tr<br>Travis CI | 31 Pd<br>Gd<br>Deployment Manager | 32 Os<br>Sf<br>SmartFrog | 33 Os<br>Cn<br>Consul | 34 Os<br>Bc<br>Bcfg2 | 35 Os<br>Mo<br>Mesos | 36 En<br>Rs<br>Rackspace |
| 37 Os<br>Sv<br>Subversion | 38 En<br>Dt<br>Datical | 39 Os<br>Gn<br>Grunt | 40 Os<br>Gp<br>Gulp | 41 Os<br>Br<br>Broccoli | 42 Fr<br>Cu<br>Cucumber | 43 Os<br>Cj<br>Cucumber.js | 44 Fr<br>Qu<br>Qunit | 45 Os<br>Npm<br>npm | 46 Fm<br>Cs<br>Codeship | 47 Pd<br>Vs<br>Visual Studio | 48 Fm<br>Cr<br>CircleCI | 49 Fr<br>Cp<br>Capistrano | 50 Fr<br>Ju<br>JuJu | 51 Os<br>Rd<br>Rundeck | 52 Os<br>Cf<br>CFEngine | 53 Fr<br>Ds<br>Swarm | 54 Os<br>Op<br>OpenStack |
| 55 Os<br>Hg<br>Mercurial | 56 En<br>Dp<br>Delphix | 57 Fr<br>Sb<br>sbt | 58 Os<br>Mk<br>Make | 59 Os<br>Ck<br>CMake | 60 Fr<br>Jt<br>JUnit | 61 Fr<br>Jm<br>JMeter | 62 Fr<br>Tn<br>TestNG | 63 Os<br>Ay<br>Artifactory | 64 Fm<br>Tc<br>TeamCity | 65 Fm<br>Sh<br>Shippable | 66 Os<br>Cc<br>CruiseControl | 67 En<br>Ry<br>RapidDeploy | 68 Fm<br>Cy<br>CodeDeploy | 69 En<br>Oc<br>Octopus Deploy | 70 En<br>No<br>CA Nolio | 71 Os<br>Kb<br>Kubernetes | 72 Fm<br>Hr<br>Heroku |
| 73 En<br>Cw<br>ISPW | 74 En<br>Id<br>Idera | 75 Os<br>Msb<br>MSBuild | 76 Os<br>Rk<br>Rake | 77 Fr<br>Pk<br>Packer | 78 Os<br>Mc<br>Mocha | 79 Fr<br>Km<br>Karma | 80 Os<br>Jm<br>Jasmine | 81 Os<br>Nx<br>Nexus | 82 Os<br>Co<br>Continuum | 83 Fm<br>Ct<br>Continua CI | 84 Pd<br>So<br>Solano CI | 85 En<br>Xld<br>XL Deploy | 86 En<br>EB<br>ElasticBox | 87 Fm<br>Dp<br>Deploybot | 88 En<br>Ud<br>UrbanCode Deploy | 89 Os<br>Nm<br>Nomad | 90 En<br>Os<br>OpenShift |

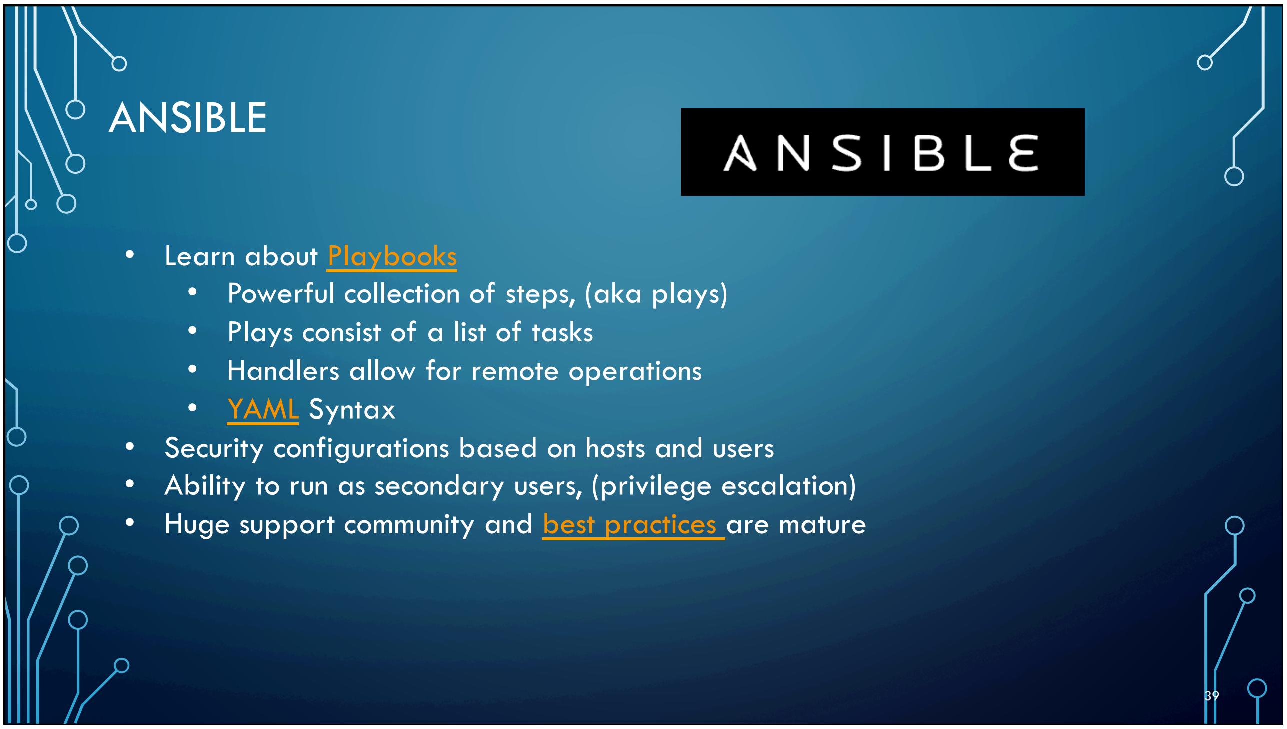
| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 91 En<br>Xlr<br>XL Release | 92 En<br>Ur<br>UrbanCode Release | 93 En<br>Bm<br>BMC Release Process | 94 En<br>Ca<br>CA Release Automation | 95 En<br>Au<br>Automic | 96 En<br>Pl<br>Plutora Release | 97 En<br>Sr<br>Serena Release | 98 Pd<br>Tfs<br>Team Foundation | 99 Fm<br>Tl<br>Trello | 100 Pd<br>Jr<br>Jira | 101 Fm<br>Rf<br>HipChat | 102 Fm<br>Sl<br>Slack | 103 Fm<br>Fd<br>Flowdock | 104 Pd<br>Pv<br>Pivotal Tracker | 105 En<br>Sn<br>ServiceNow |
| 106 Os<br>Ki<br>Kibana | 107 Fm<br>Nr<br>New Relic | 108 En<br>Dt<br>Dynatrace | 109 Os<br>Ni<br>Nagios | 110 Os<br>Zb<br>Zabbix | 111 En<br>Dd<br>Datadog | 112 Os<br>El<br>Elasticsearch | 113 Fm<br>Ad<br>AppDynamics | 114 En<br>Sp<br>Splunk | 115 Fm<br>Le<br>Logentries | 116 Fm<br>Sl<br>Sumo Logic | 117 Os<br>Ls<br>Logstash | 118 Os<br>Sn<br>Snort | 119 Os<br>Tw<br>Tripwire | 120 En<br>Ff<br>Fortify |

XebiaLabs

Follow @xebialabs

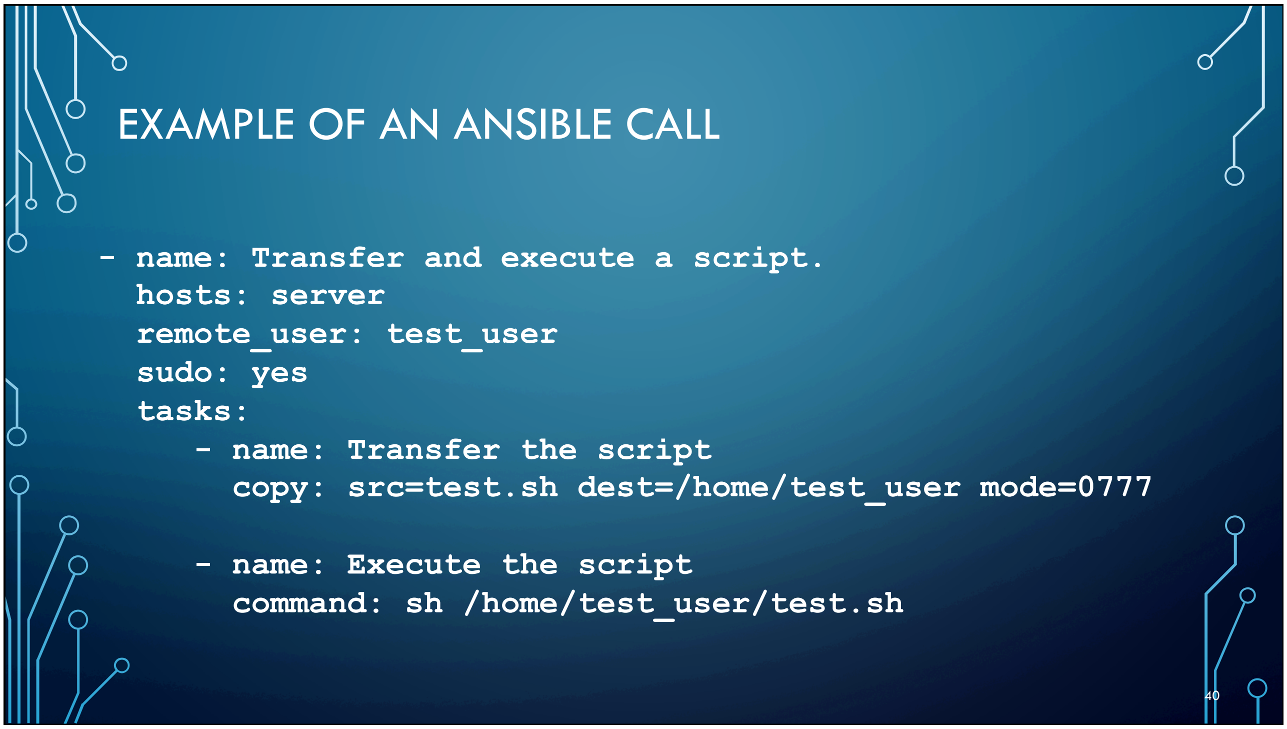
https://xebialabs.com/periodic-table-of-devops-tools/

38



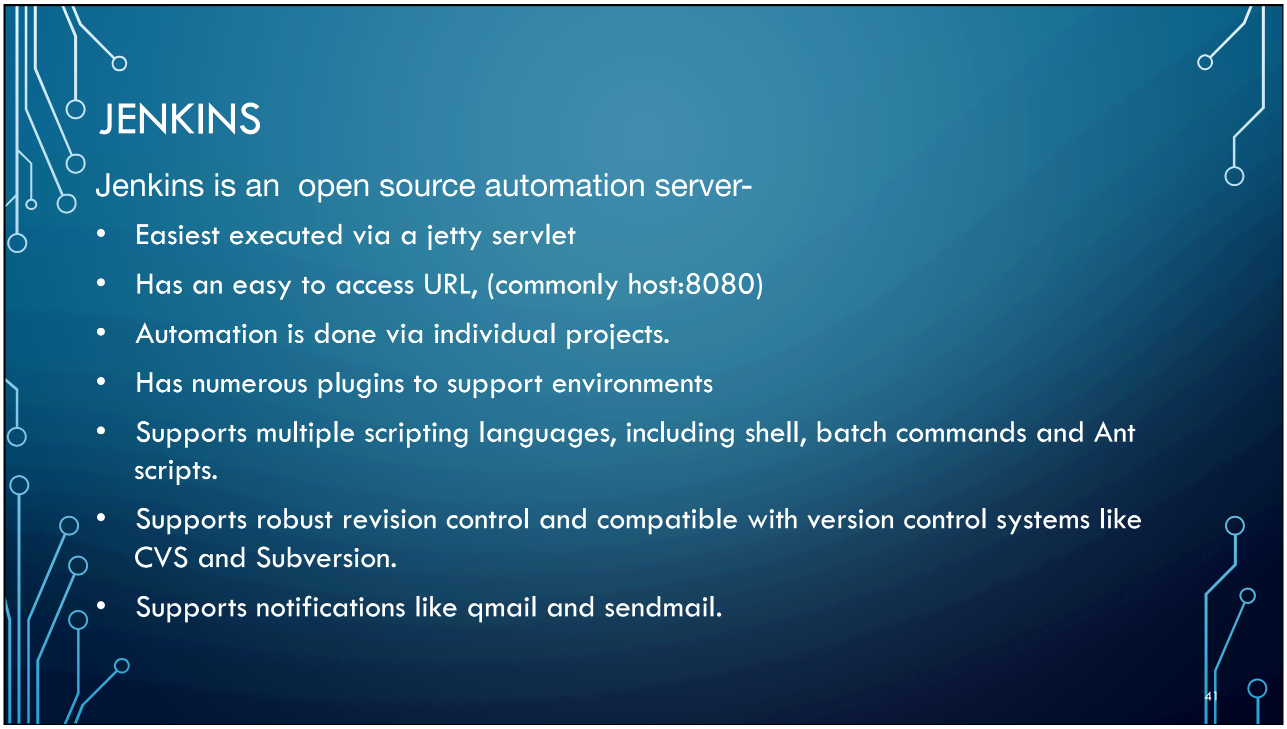
## ANSIBLE

- Learn about Playbooks
  - Powerful collection of steps, (aka plays)
  - Plays consist of a list of tasks
  - Handlers allow for remote operations
  - YAML Syntax
- Security configurations based on hosts and users
- Ability to run as secondary users, (privilege escalation)
- Huge support community and best practices are mature



## EXAMPLE OF AN ANSIBLE CALL

```
- name: Transfer and execute a script.
  hosts: server
  remote_user: test_user
  sudo: yes
  tasks:
     - name: Transfer the script
       copy: src=test.sh dest=/home/test_user mode=0777

     - name: Execute the script
       command: sh /home/test_user/test.sh
```

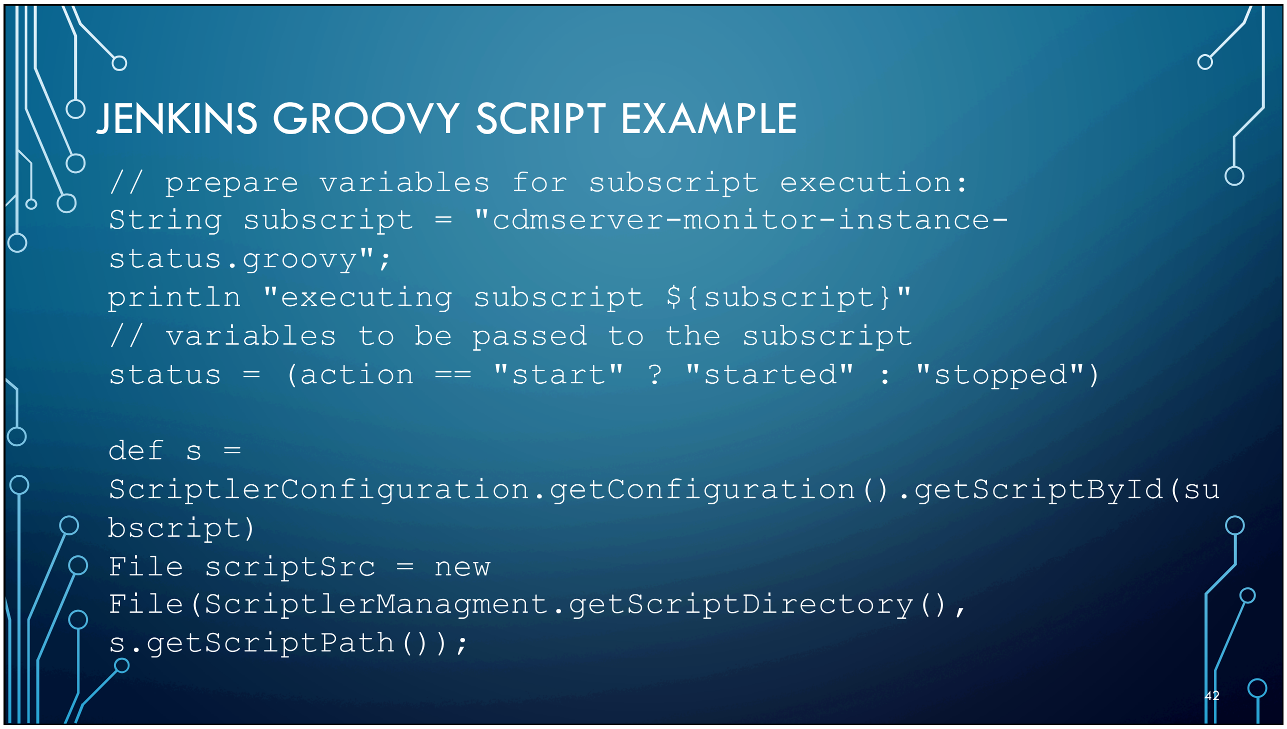


# JENKINS

Jenkins is an  open source automation server-

- Easiest executed via a jetty servlet
- Has an easy to access URL, (commonly host:8080)
- Automation is done via individual projects.
- Has numerous plugins to support environments
- Supports multiple scripting languages, including shell, batch commands and Ant scripts.
- Supports robust revision control and compatible with version control systems like CVS and Subversion.
- Supports notifications like qmail and sendmail.

41

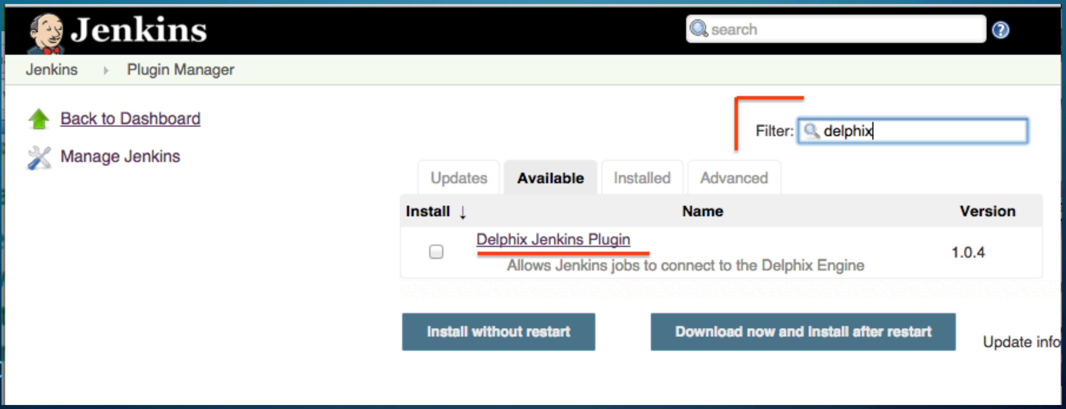


# JENKINS GROOVY SCRIPT EXAMPLE

```
// prepare variables for subscript execution:
String subscript = "cdmserver-monitor-instance-
status.groovy";
println "executing subscript ${subscript}"
// variables to be passed to the subscript
status = (action == "start" ? "started" : "stopped")

def s =
ScriptlerConfiguration.getConfiguration().getScriptById(su
bscript)
File scriptSrc = new
File(ScriptlerManagment.getScriptDirectory(),
s.getScriptPath());
```

42

# JENKINS PLUGIN FOR DELPHIX



Filter: delphix

| Install ↓ | Name | Version |
|---|---|---|
| □ | Delphix Jenkins Plugin<br>Allows Jenkins jobs to connect to the Delphix Engine | 1.0.4 |

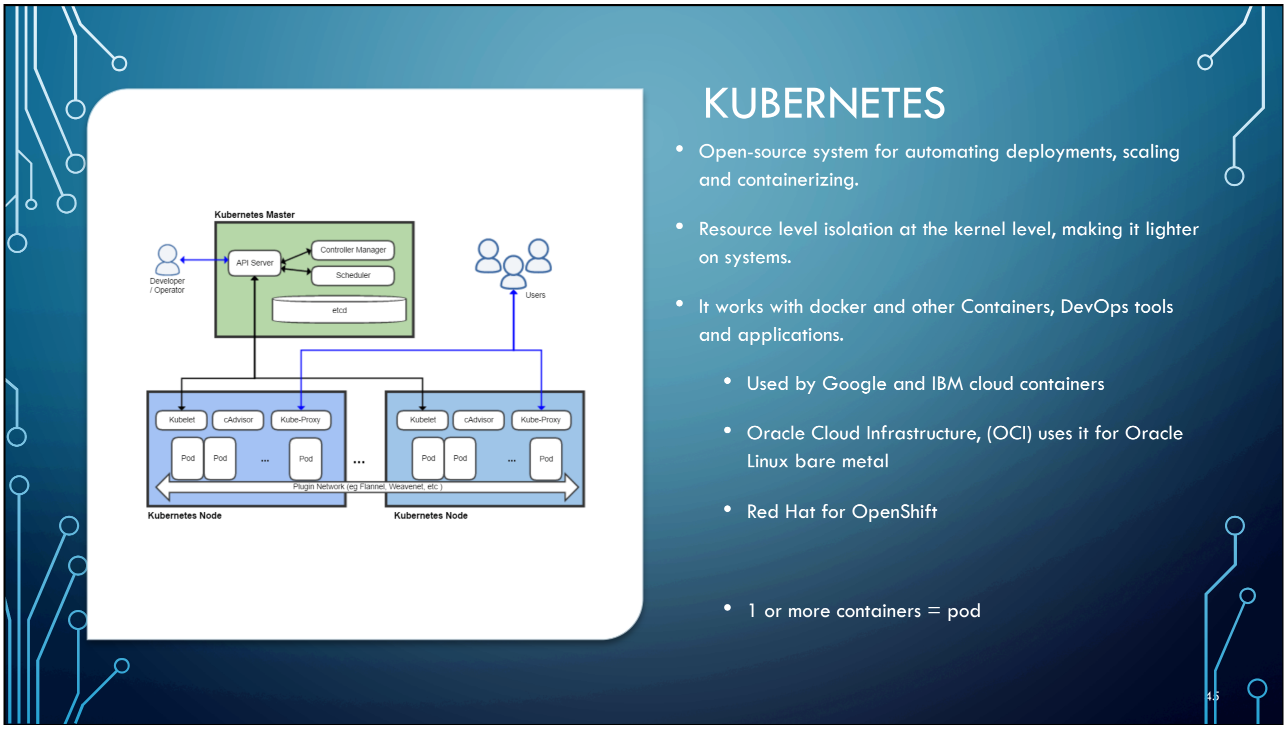
https://github.com/delphix/delphix-jenkins-plugin

# PUPPET

- Review, automate, deploy and invest in all your software.
- Control and enforce consistency across environments and platforms.
- Share, test and enforce changes on-premise and in the cloud.

**Similar to Puppet**

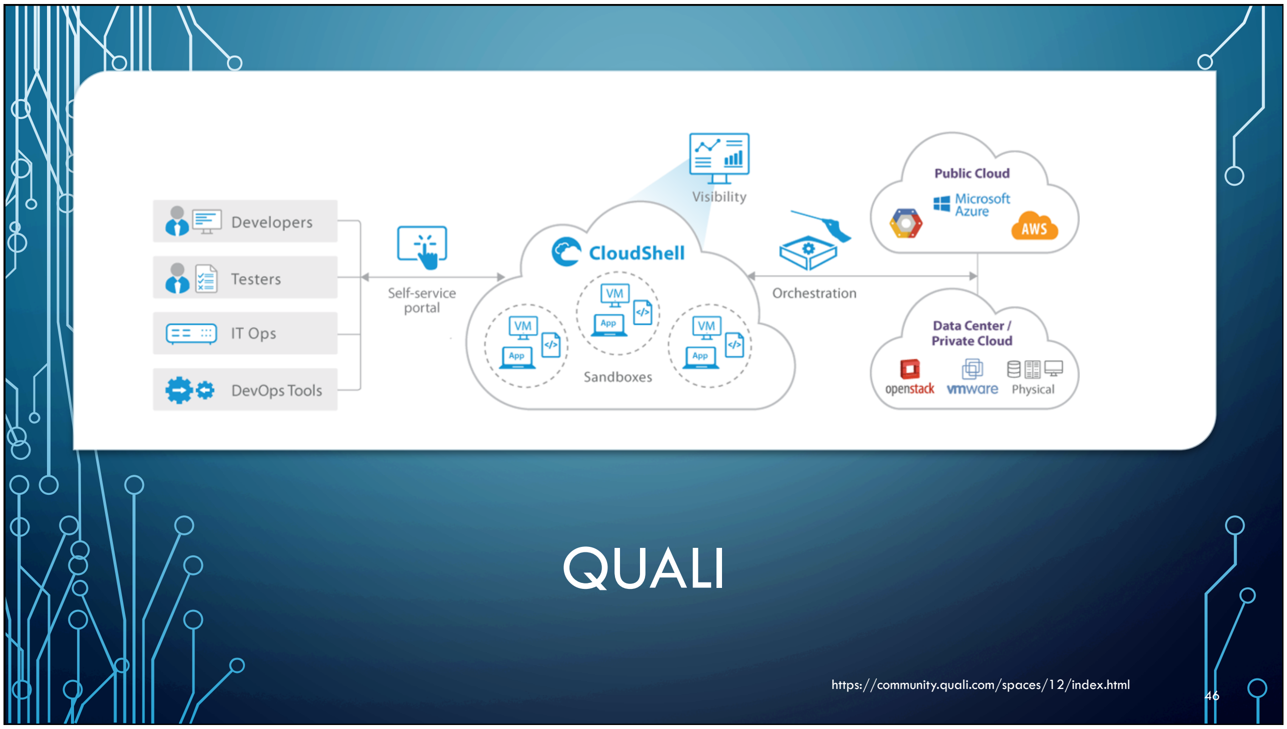
- Chef
- Jenkins



## KUBERNETES

- Open-source system for automating deployments, scaling and containerizing.
- Resource level isolation at the kernel level, making it lighter on systems.
- It works with docker and other Containers, DevOps tools and applications.
  - Used by Google and IBM cloud containers
  - Oracle Cloud Infrastructure, (OCI) uses it for Oracle Linux bare metal
  - Red Hat for OpenShift
  - 1 or more containers = pod

45

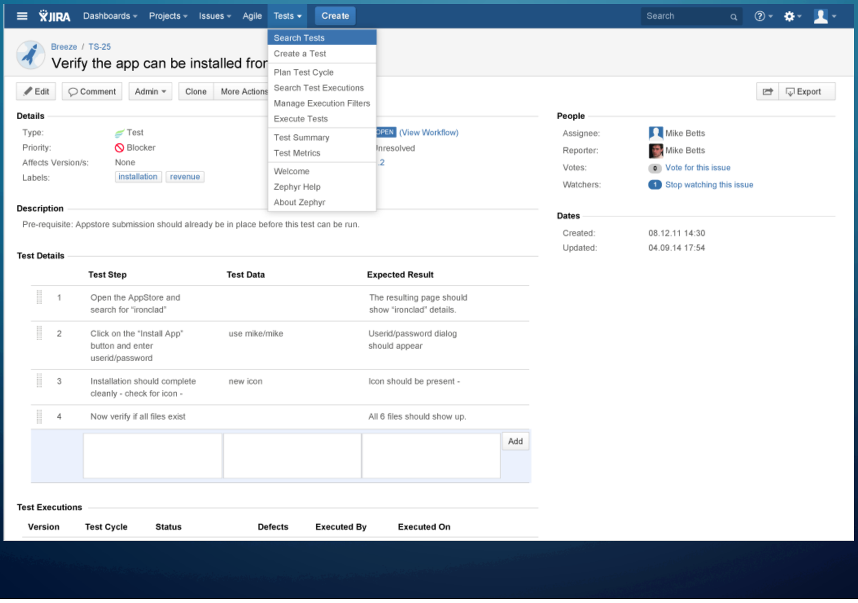


## QUALI

https://community.quali.com/spaces/12/index.html

46

## JIRA FOR TESTING



Breeze / TS-25

Verify the app can be installed fro

**Details**

Type: Test

Priority: Blocker

Affects Version/s: None

Labels: installation revenue

OPEN (View Workflow)

Unresolved

2

**People**

Assignee: Mike Betts

Reporter: Mike Betts

Votes: 0 Vote for this issue

Watchers: 1 Stop watching this issue

**Dates**

Created: 08.12.11 14:30

Updated: 04.09.14 17:54

**Description**

Pre-requisite: Appstore submission should already be in place before this test can be run.

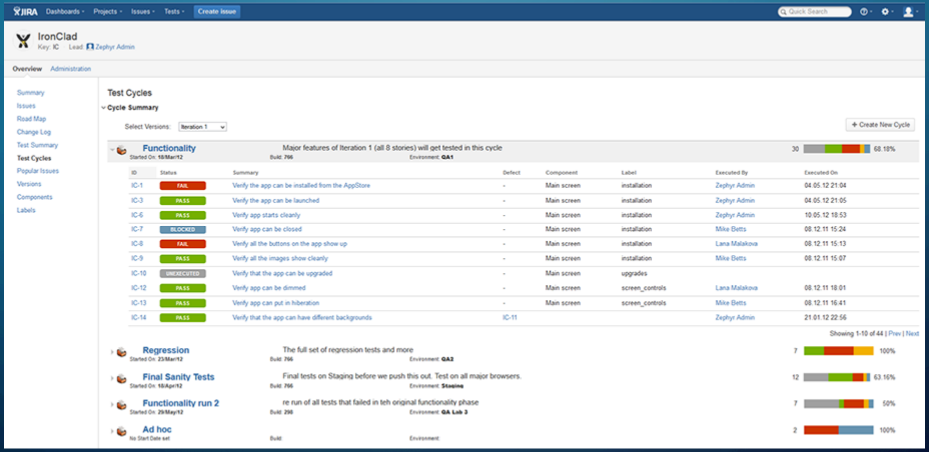
**Test Details**

| | Test Step | Test Data | Expected Result |
|---|---|---|---|
| 1 | Open the AppStore and search for "ironclad" | | The resulting page should show "ironclad" details. |
| 2 | Click on the "Install App" button and enter userid/password | use mike/mike | Userid/password dialog should appear |
| 3 | Installation should complete cleanly - check for icon - | new icon | Icon should be present - |
| 4 | Now verify if all files exist | | All 6 files should show up. |

**Test Executions**

| Version | Test Cycle | Status | Defects | Executed By | Executed On |
|---|---|---|---|---|---|

## JIRA FOR TESTING



IronClad

Key: IC Lead: Zephyr Admin

Test Cycles

Cycle Summary

Select Versions: Iteration 1

**Functionality** Started On: 18/Mar/12 — Major features of Iteration 1 (all 8 stories) will get tested in this cycle — Build: 766 — Environment: **QA1** — 30 — 68.18%

| ID | Status | Summary | Defect | Component | Label | Executed By | Executed On |
|---|---|---|---|---|---|---|---|
| IC-1 | FAIL | Verify the app can be installed from the AppStore | - | Main screen | installation | Zephyr Admin | 04.05.12 21:04 |
| IC-3 | PASS | Verify the app can be launched | - | Main screen | installation | Zephyr Admin | 04.05.12 21:05 |
| IC-6 | PASS | Verify app starts cleanly | - | Main screen | installation | Zephyr Admin | 10.05.12 18:53 |
| IC-7 | BLOCKED | Verify app can be closed | - | Main screen | installation | Mike Betts | 08.12.11 15:24 |
| IC-8 | FAIL | Verify all the buttons on the app show up | - | Main screen | installation | Lana Malakova | 08.12.11 15:13 |
| IC-9 | PASS | Verify all the images show cleanly | - | Main screen | installation | Mike Betts | 08.12.11 15:07 |
| IC-10 | UNEXECUTED | Verify that the app can be upgraded | - | Main screen | upgrades | | |
| IC-12 | PASS | Verify app can be dimmed | - | Main screen | screen_controls | Lana Malakova | 08.12.11 18:01 |
| IC-13 | PASS | Verify app can put in hiberation | - | Main screen | screen_controls | Mike Betts | 08.12.11 16:41 |
| IC-14 | PASS | Verify that the app can have different backgrounds | IC-11 | | | Zephyr Admin | 21.01.12 22:56 |

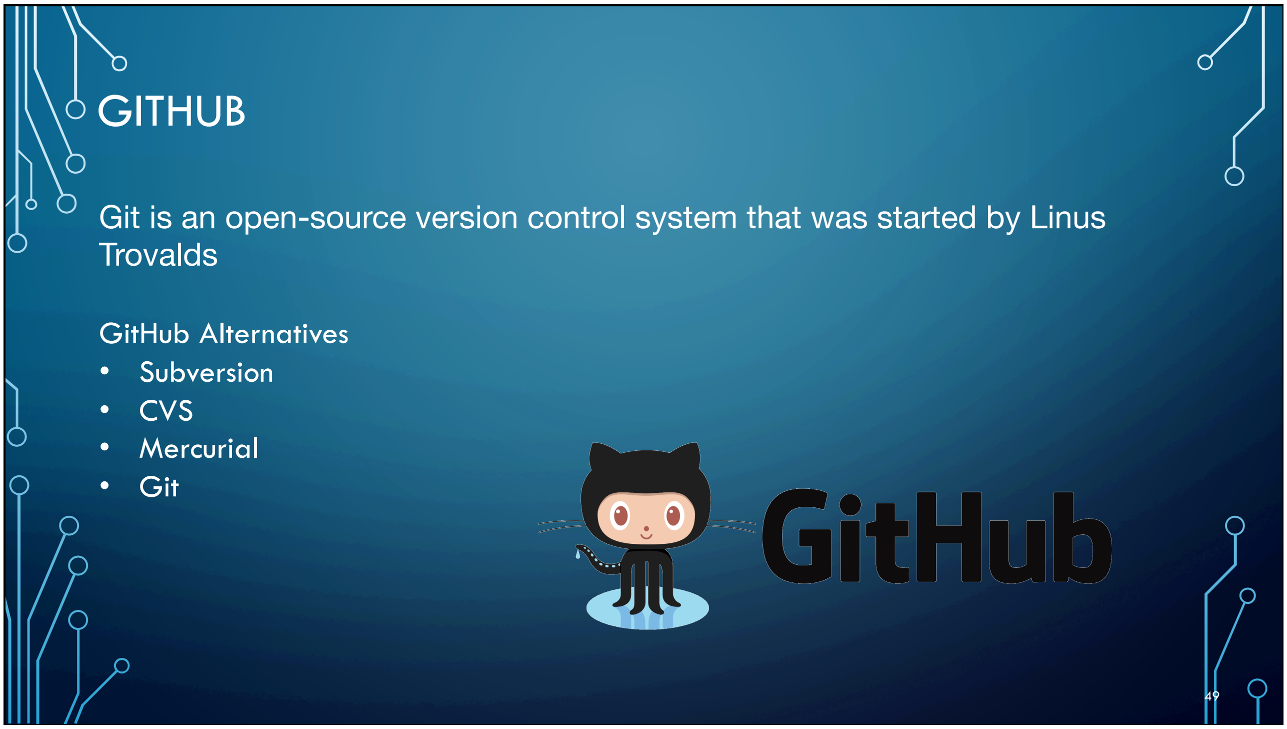
Showing 1-10 of 44 | Prev | Next

**Regression** Started On: 23/Mar/12 — The full set of regression tests and more — Build: 766 — Environment: **QA2** — 7 — 100%

**Final Sanity Tests** Started On: 18/Apr/12 — Final tests on Staging before we push this out. Test on all major browsers. — Build: 766 — Environment: **Staging** — 12 — 63.16%

**Functionality run 2** Started On: 29/May/12 — re run of all tests that failed in teh original functionality phase — Build: 298 — Environment: **QA Lab 3** — 7 — 50%

**Ad hoc** No Start Date set — Build: — Environment: — 2 — 100%



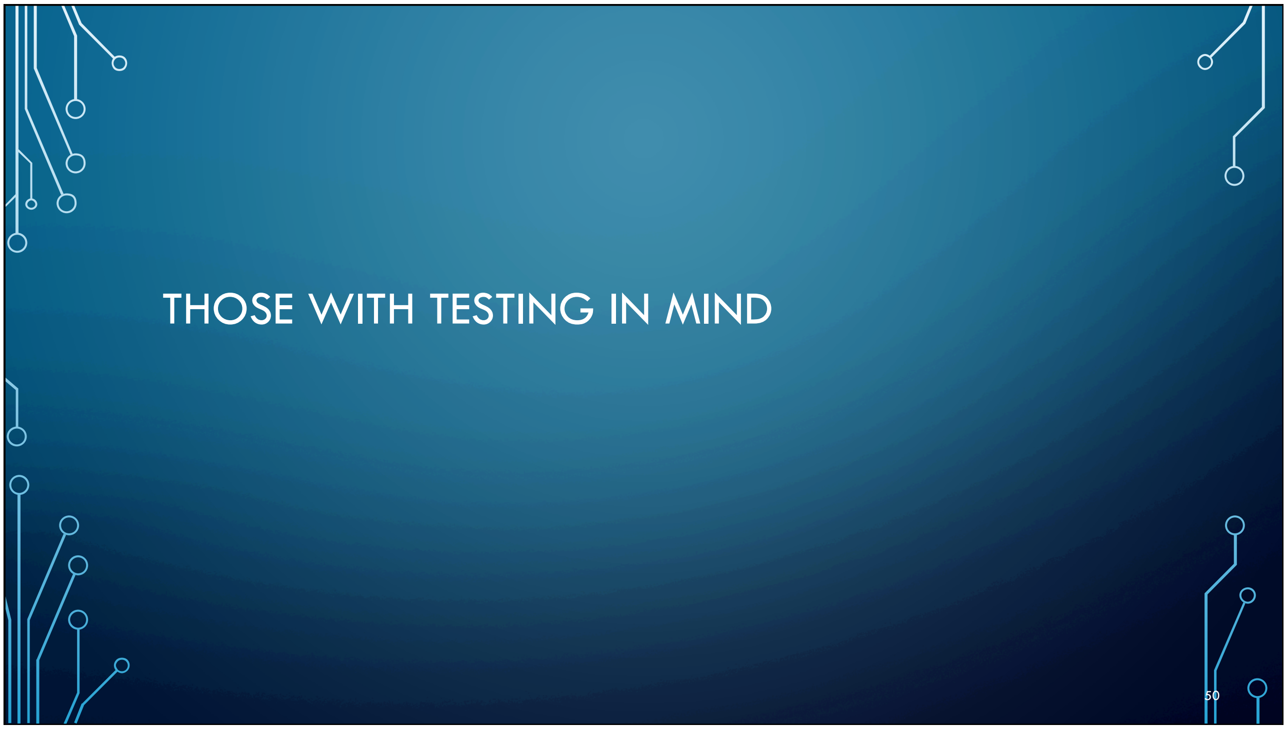
## GITHUB

Git is an open-source version control system that was started by Linus Trovalds

GitHub Alternatives

- Subversion
- CVS
- Mercurial
- Git

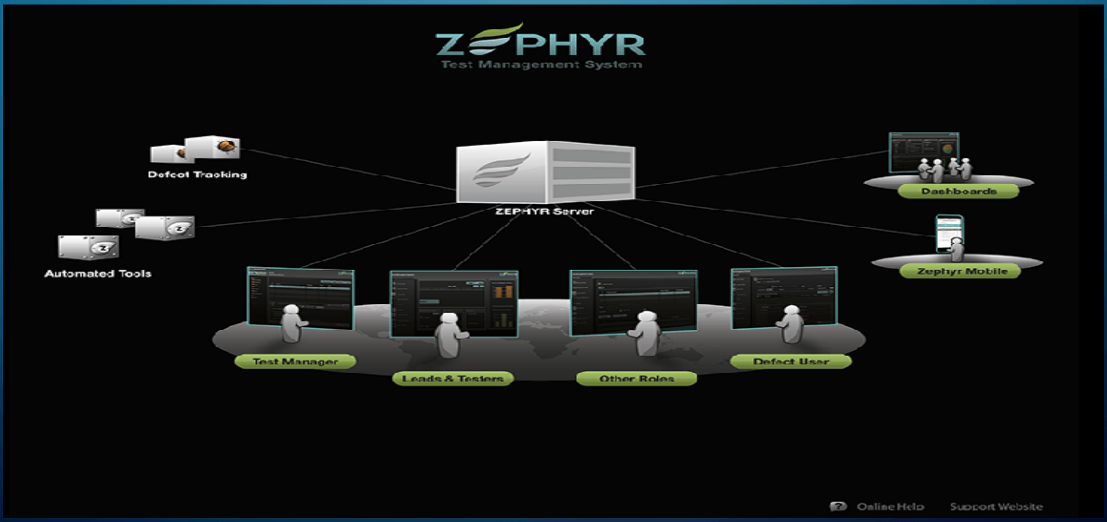
49



## THOSE WITH TESTING IN MIND

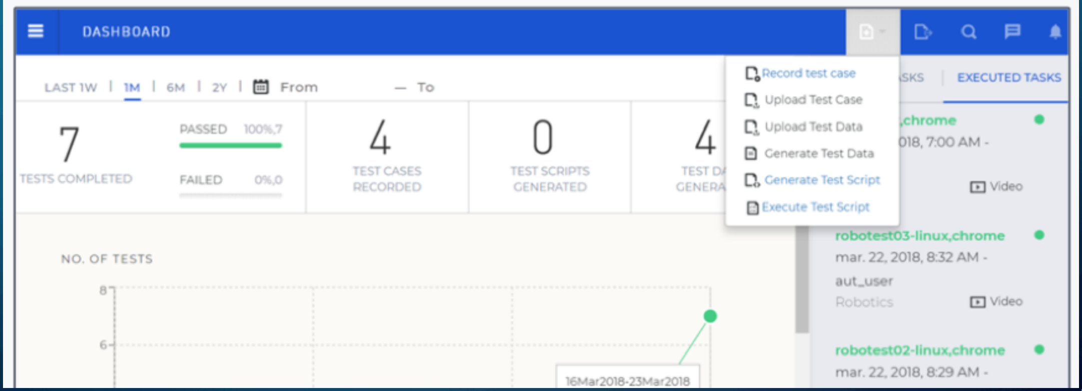
50

# ZEPHYR



51

# AUTONOMIQ



7 TESTS COMPLETED

PASSED 100%,7

FAILED 0%,0

4 TEST CASES RECORDED

0 TEST SCRIPTS GENERATED

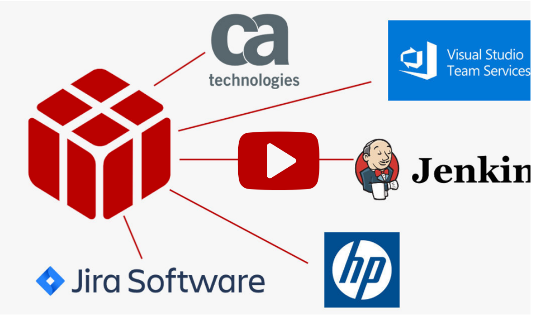
4 TEST D GENERA

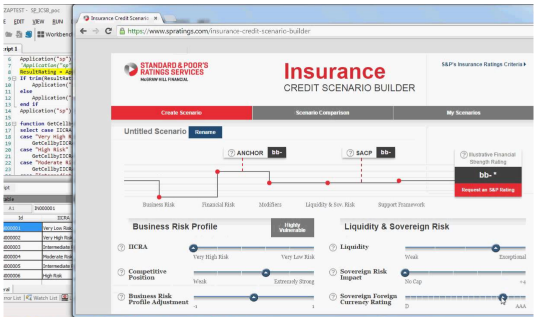
,chrome
018, 7:00 AM -

robotest03-linux,chrome
mar. 22, 2018, 8:32 AM -
aut_user
Robotics

robotest02-linux,chrome
mar. 22, 2018, 8:29 AM -

52





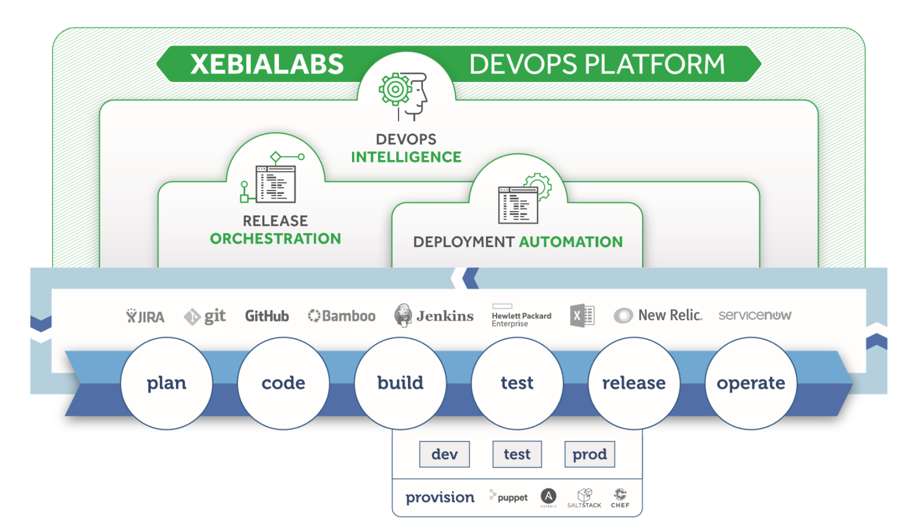
```
Application("sp")
'Application("sp"
ResultRating = Ap
If trim(ResultRat
    Application("
else
    Application("
end if
Application("sp")

function GetCellb
select case IICRA
case "Very High R
    GetCellbyIICR
case "High Risk"
    GetCellbyIICR
case "Moderate Ri
    GetCellbyIICR
```

| A1 | IN000001 |
|---|---|
| Id | IICRA |
| 000001 | Very Low Risk |
| 000002 | Very High Risk |
| 000003 | Intermediate |
| 000004 | Moderate Risk |
| 000005 | Intermediate |
| 000006 | High Risk |

# ZAPTEST

53

# XEBIALABS DEVOPS PLATFORM



XEBIALABS DEVOPS PLATFORM

DEVOPS INTELLIGENCE

RELEASE ORCHESTRATION

DEPLOYMENT AUTOMATION

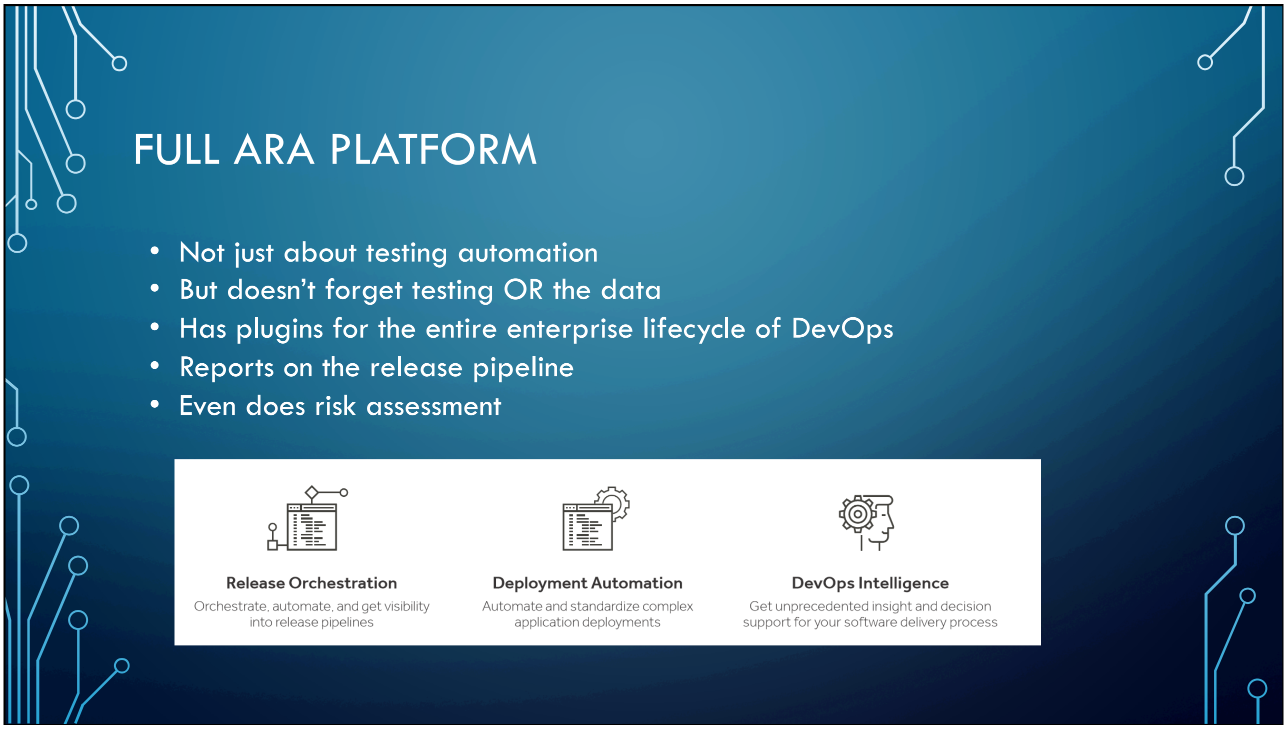
JIRA git GitHub Bamboo Jenkins Hewlett Packard Enterprise New Relic. servicenow

plan code build test release operate

dev test prod

provision puppet SALTSTACK CHEF

54



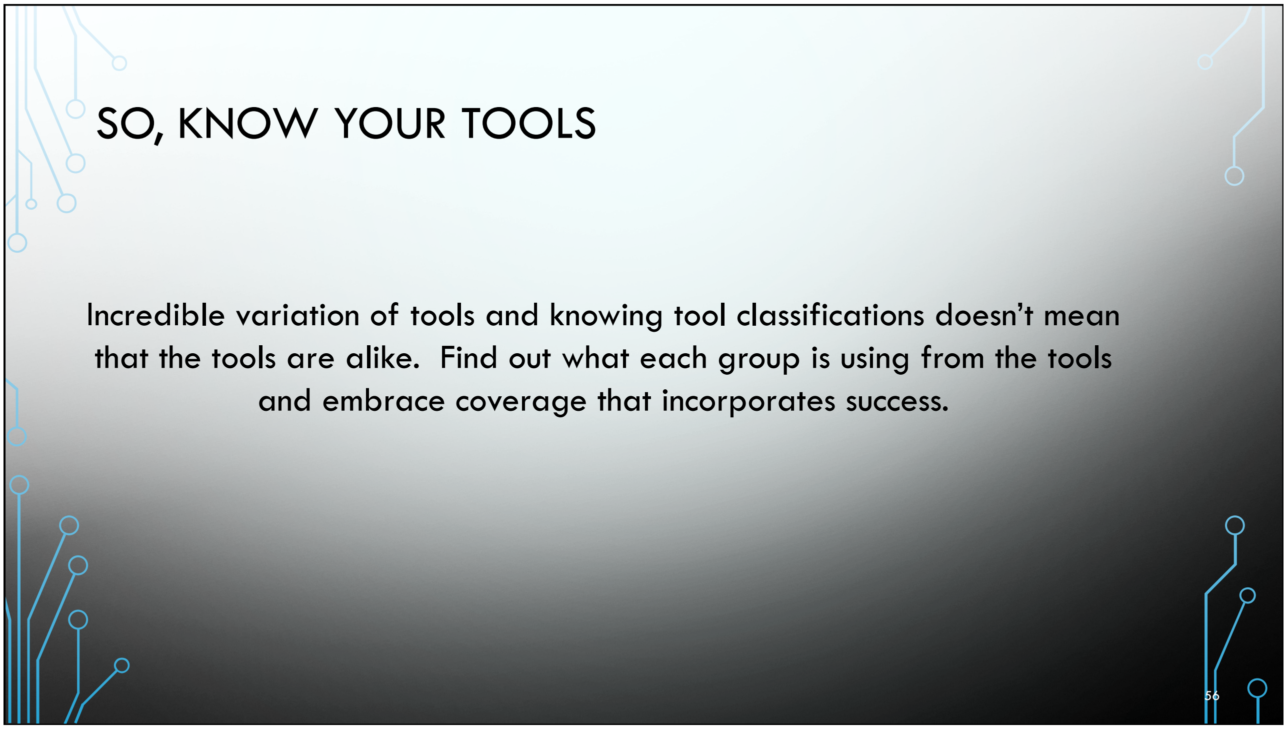
# FULL ARA PLATFORM

- Not just about testing automation
- But doesn't forget testing OR the data
- Has plugins for the entire enterprise lifecycle of DevOps
- Reports on the release pipeline
- Even does risk assessment

**Release Orchestration**
Orchestrate, automate, and get visibility into release pipelines

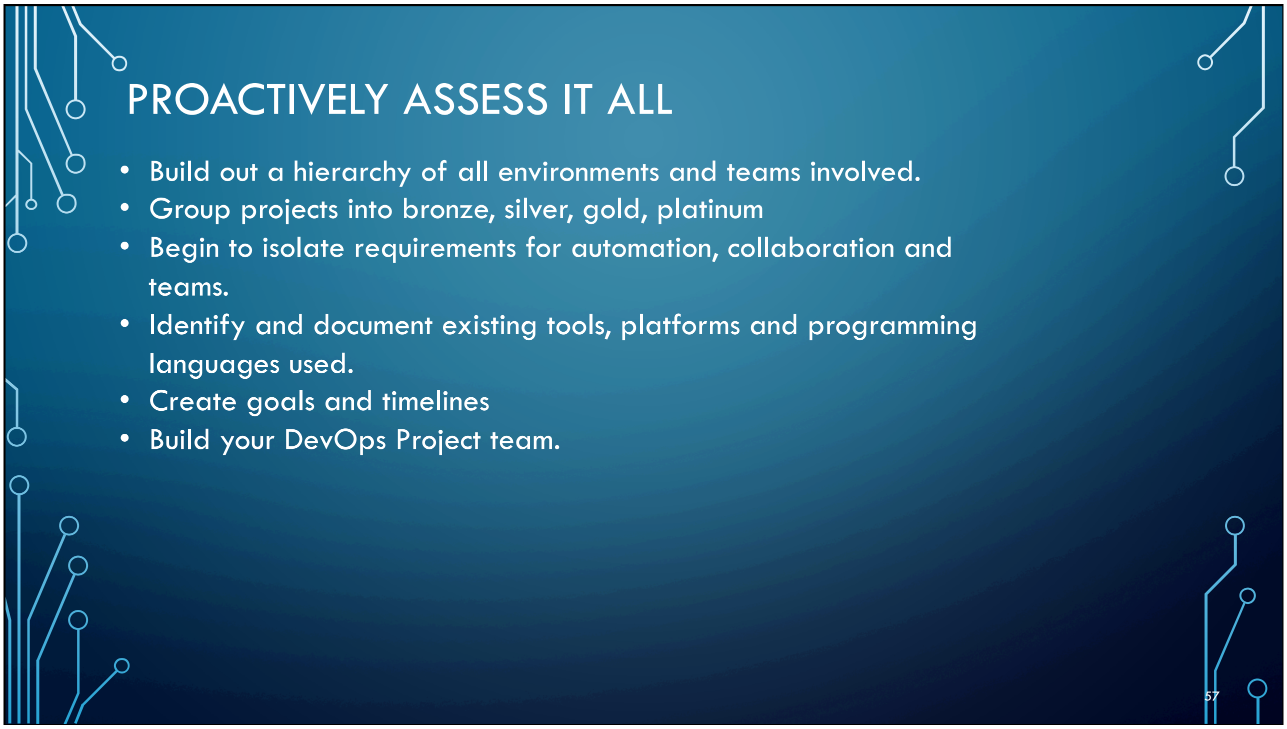
**Deployment Automation**
Automate and standardize complex application deployments

**DevOps Intelligence**
Get unprecedented insight and decision support for your software delivery process



# SO, KNOW YOUR TOOLS

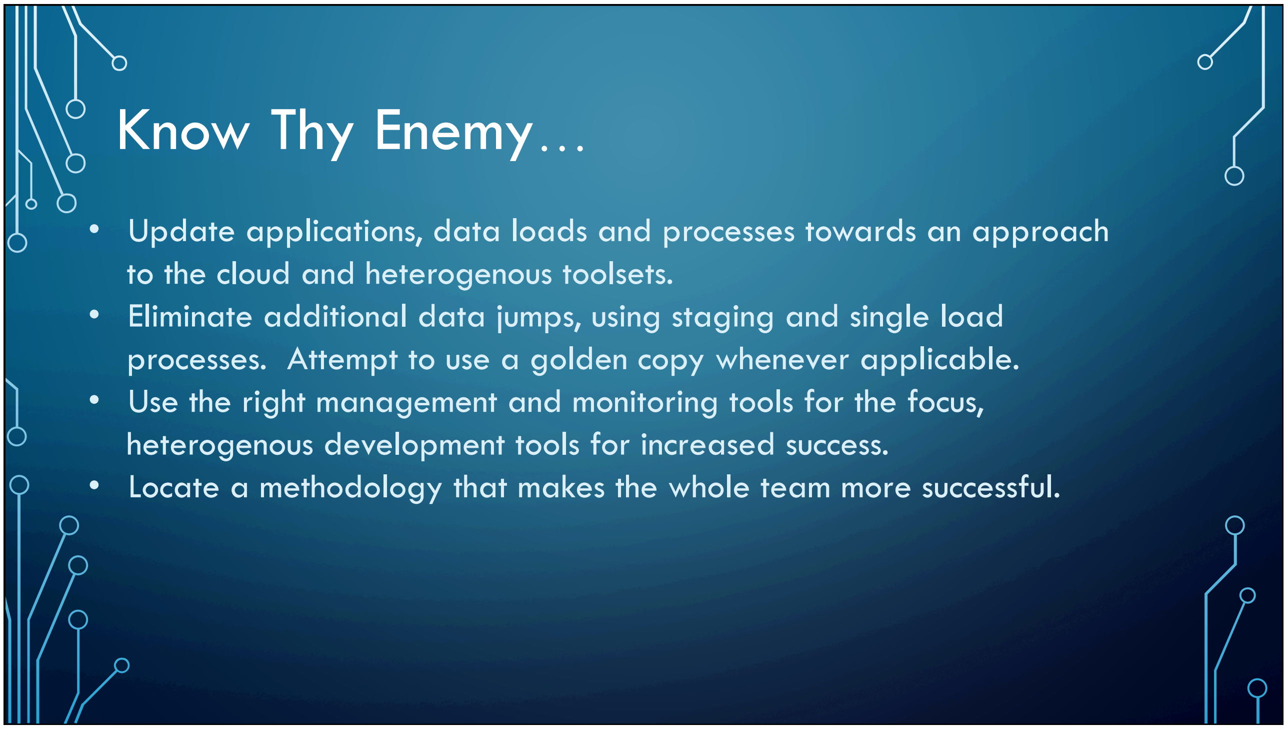
Incredible variation of tools and knowing tool classifications doesn't mean that the tools are alike. Find out what each group is using from the tools and embrace coverage that incorporates success.



## PROACTIVELY ASSESS IT ALL

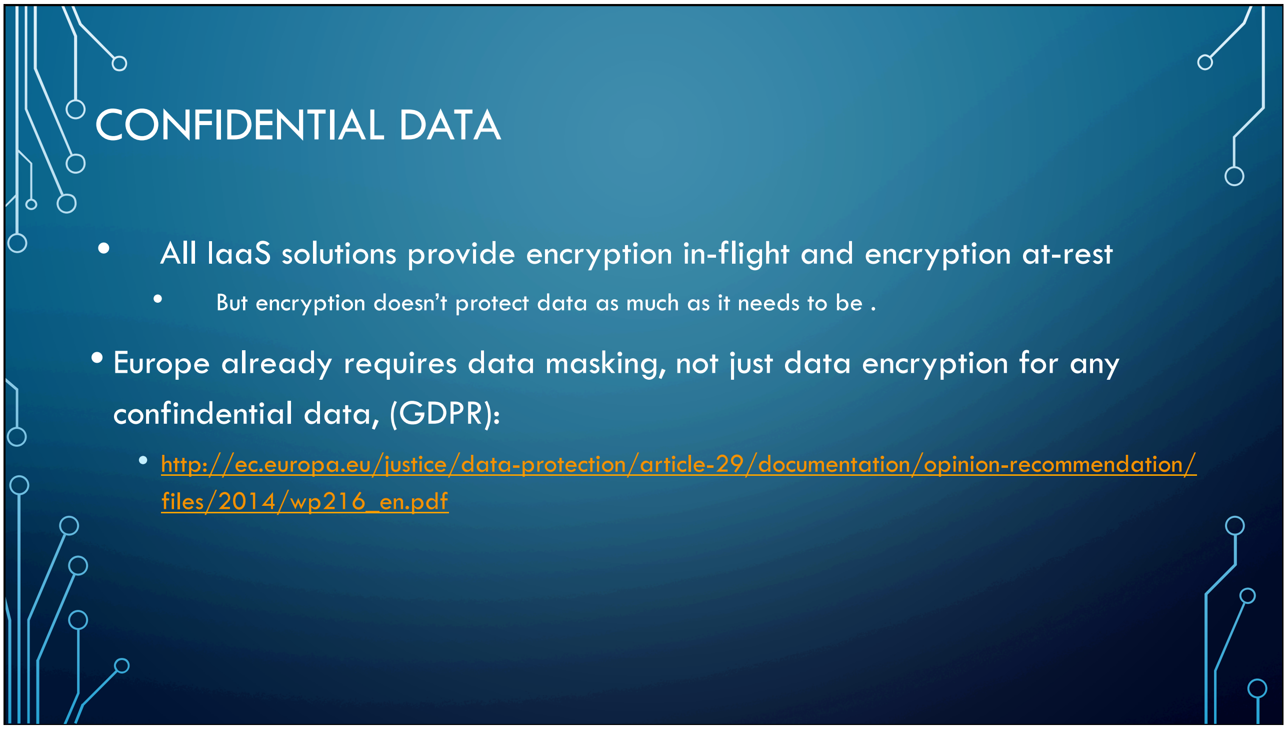
- Build out a hierarchy of all environments and teams involved.
- Group projects into bronze, silver, gold, platinum
- Begin to isolate requirements for automation, collaboration and teams.
- Identify and document existing tools, platforms and programming languages used.
- Create goals and timelines
- Build your DevOps Project team.

57



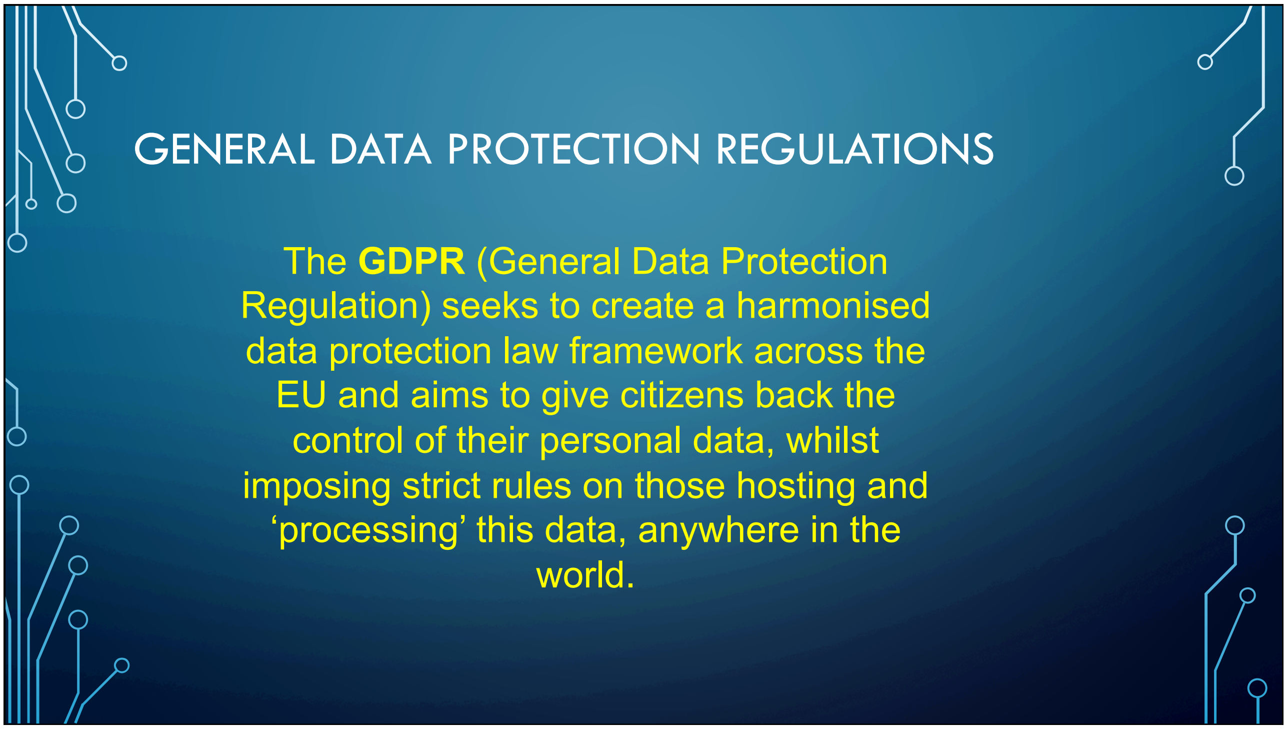
## Know Thy Enemy…

- Update applications, data loads and processes towards an approach to the cloud and heterogenous toolsets.
- Eliminate additional data jumps, using staging and single load processes. Attempt to use a golden copy whenever applicable.
- Use the right management and monitoring tools for the focus, heterogenous development tools for increased success.
- Locate a methodology that makes the whole team more successful.



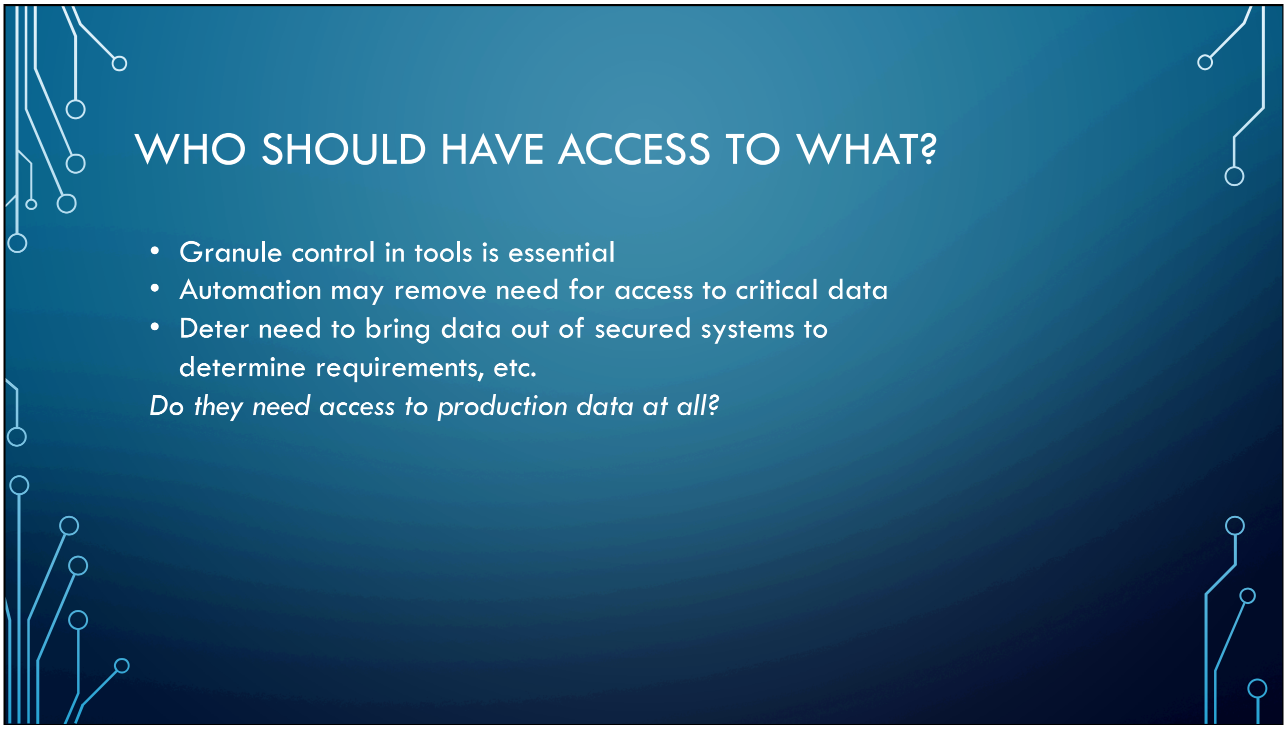
## CONFIDENTIAL DATA

- All IaaS solutions provide encryption in-flight and encryption at-rest
  - But encryption doesn’t protect data as much as it needs to be .
- Europe already requires data masking, not just data encryption for any confindential data, (GDPR):
  - http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf



## GENERAL DATA PROTECTION REGULATIONS

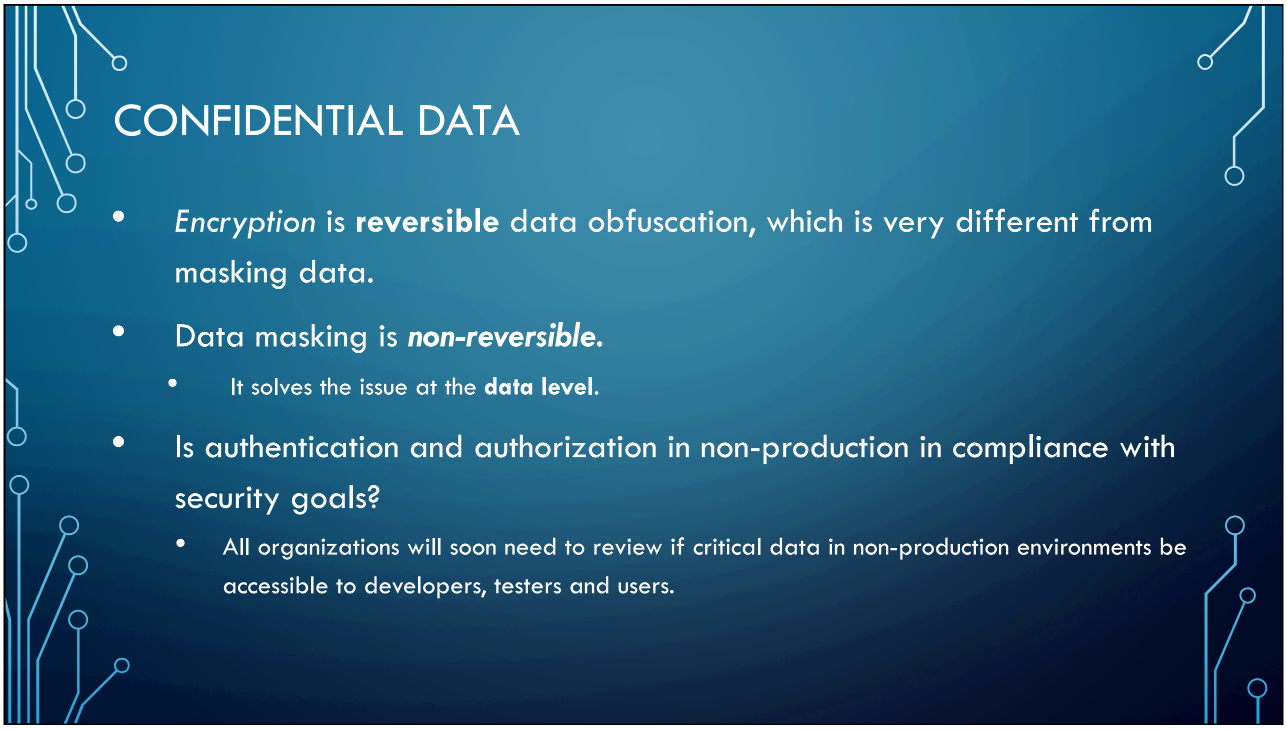
The **GDPR** (General Data Protection Regulation) seeks to create a harmonised data protection law framework across the EU and aims to give citizens back the control of their personal data, whilst imposing strict rules on those hosting and ‘processing’ this data, anywhere in the world.



## WHO SHOULD HAVE ACCESS TO WHAT?

- Granule control in tools is essential
- Automation may remove need for access to critical data
- Deter need to bring data out of secured systems to determine requirements, etc.

*Do they need access to production data at all?*



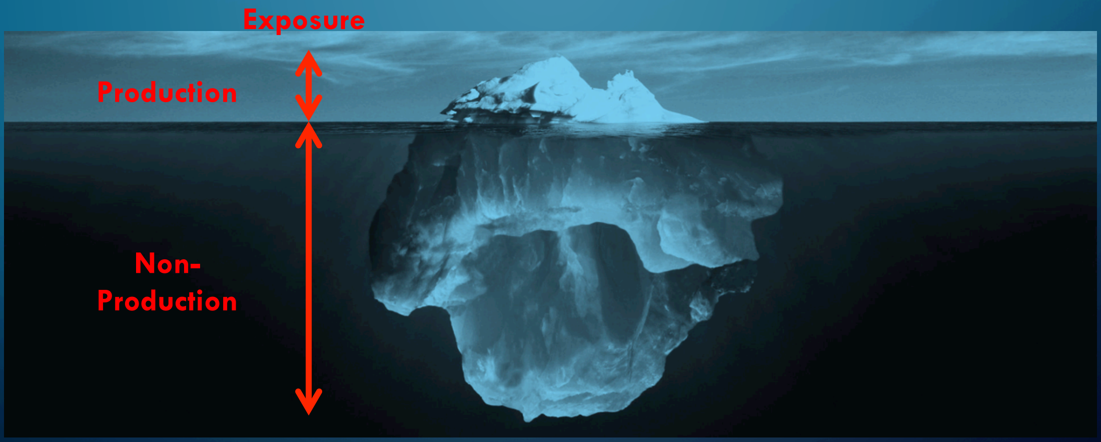
## CONFIDENTIAL DATA

- *Encryption* is **reversible** data obfuscation, which is very different from masking data.
- Data masking is ***non-reversible.***
  - It solves the issue at the **data level.**
- Is authentication and authorization in non-production in compliance with security goals?
  - All organizations will soon need to review if critical data in non-production environments be accessible to developers, testers and users.

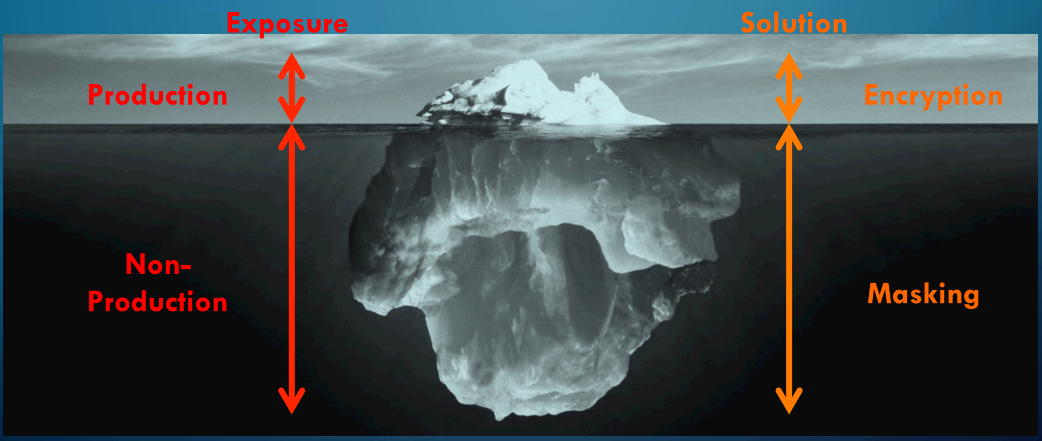
## WHY MASKING IS PART OF THE ANSWER

- Masking personally-identifiable, (PII, HIPAA, PCI, etc.) information renders it useless from a security standpoint
- Resolves both the *technical* and personal responsibility issue.
- The data can be masked before it moves to non-production, removing unnecessary risk.

## CONFIDENTIAL DATA



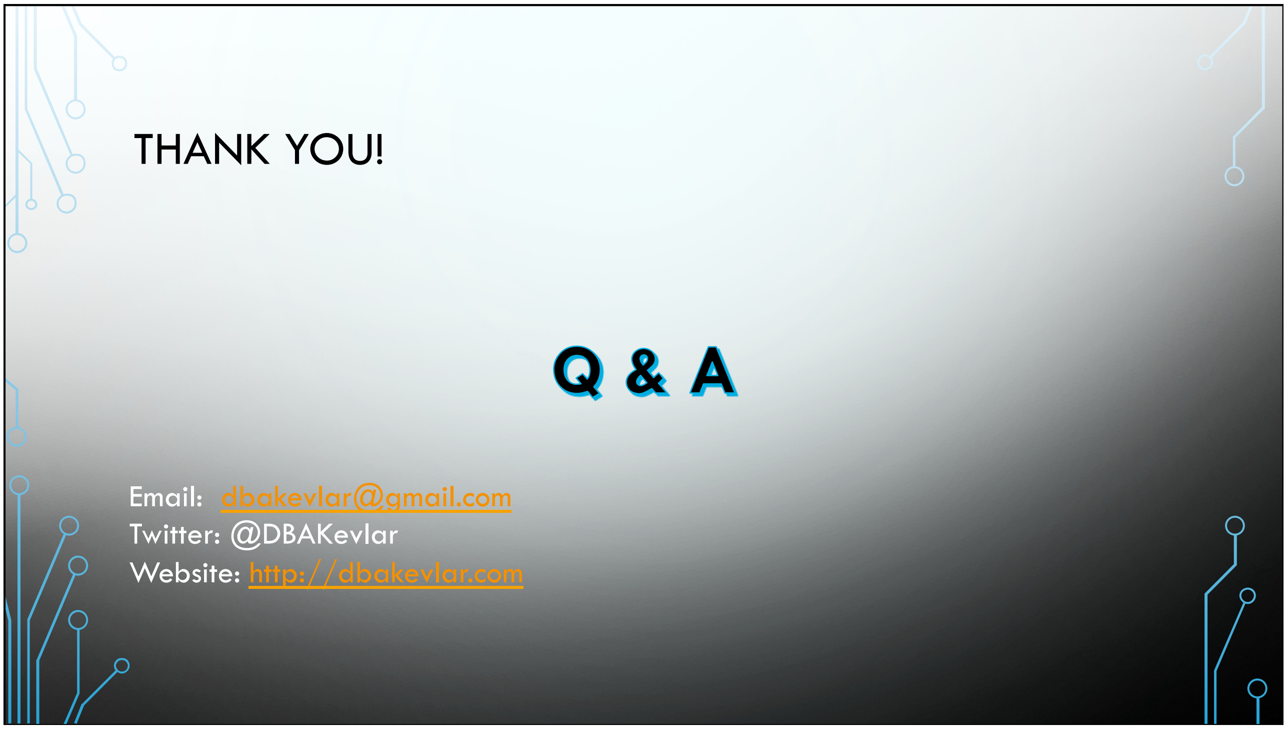
## CONFIDENTIAL DATA



## HOW TO SUCCEED WITH DEVOPS TOOLS

Culture is one of the biggest challenges in DevOps:

- Automating repeatable and complex processes, (scripting and automation tools)
- Get data out of the way of DevOps, (Containers, Data Pods and Virtualization)
- Locate heterogeneous tools/scripting languages that can support multiple tiers of technology.
- Understanding that automation does not lessen the value of the contributors, but increases it.
- Read- The Phoenix Project and the DevOps Handbook!



## THANK YOU!

**Q & A**

Email: dbakevlar@gmail.com
Twitter: @DBAKevlar
Website: http://dbakevlar.com