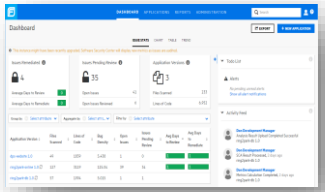
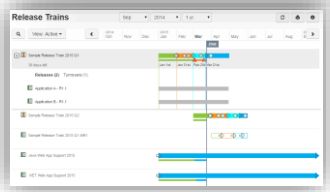
Release Management for the multi-modal Enterprise

Kevin A. Lee – kevin.lee@microfocus.com
Senior Solutions Architect

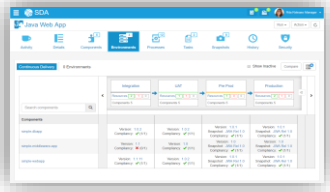
DevOps and Release Management Webinars (UK)



- **Automating DevSecOps**: How to embed security into your continuous delivery pipelines



- **Enterprise DevOps**: Release Management for the multi-modal Enterprise



- **Continuous Delivery Pipelines**: Automating the value stream through continuous release

https://www.microfocus.com/campaign/serena-release-management/

Agenda

01. Application Release Management Challenges

02. Release Management “Good” Practices

03. Micro Focus Release Control

04. Demonstration

05. Q&A

Application Release Management Challenges

Modern Software Lifecycle Disruptions

METHODOLOGY
DevOps Adoption


Waterfall


Iterative


Agile


DevOps

APPLICATIONS
Software Complexity


Monolithic

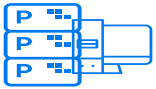

Client-Server


N-Tier


Microservices + APIs

INFRASTRUCTURE
Cloud Transformation

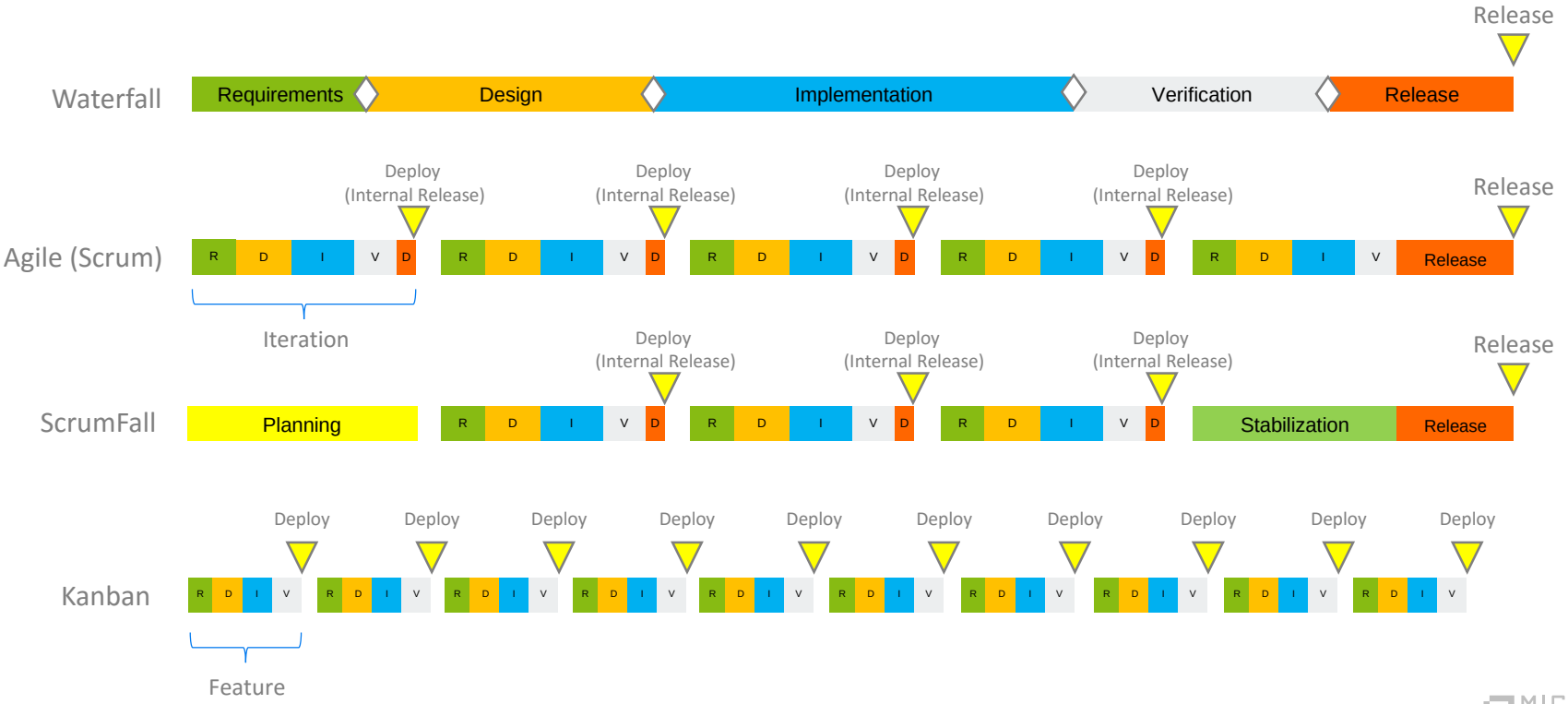

Mainframe


Open Systems


Virtualization/Cloud

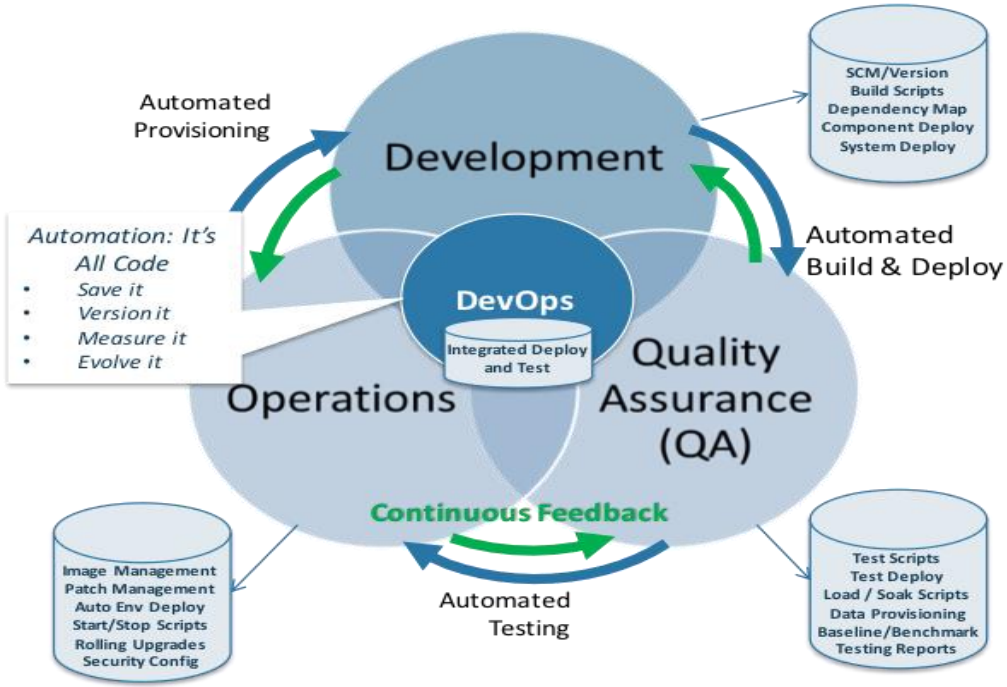

Cloud Native

Shift to Continuous Release



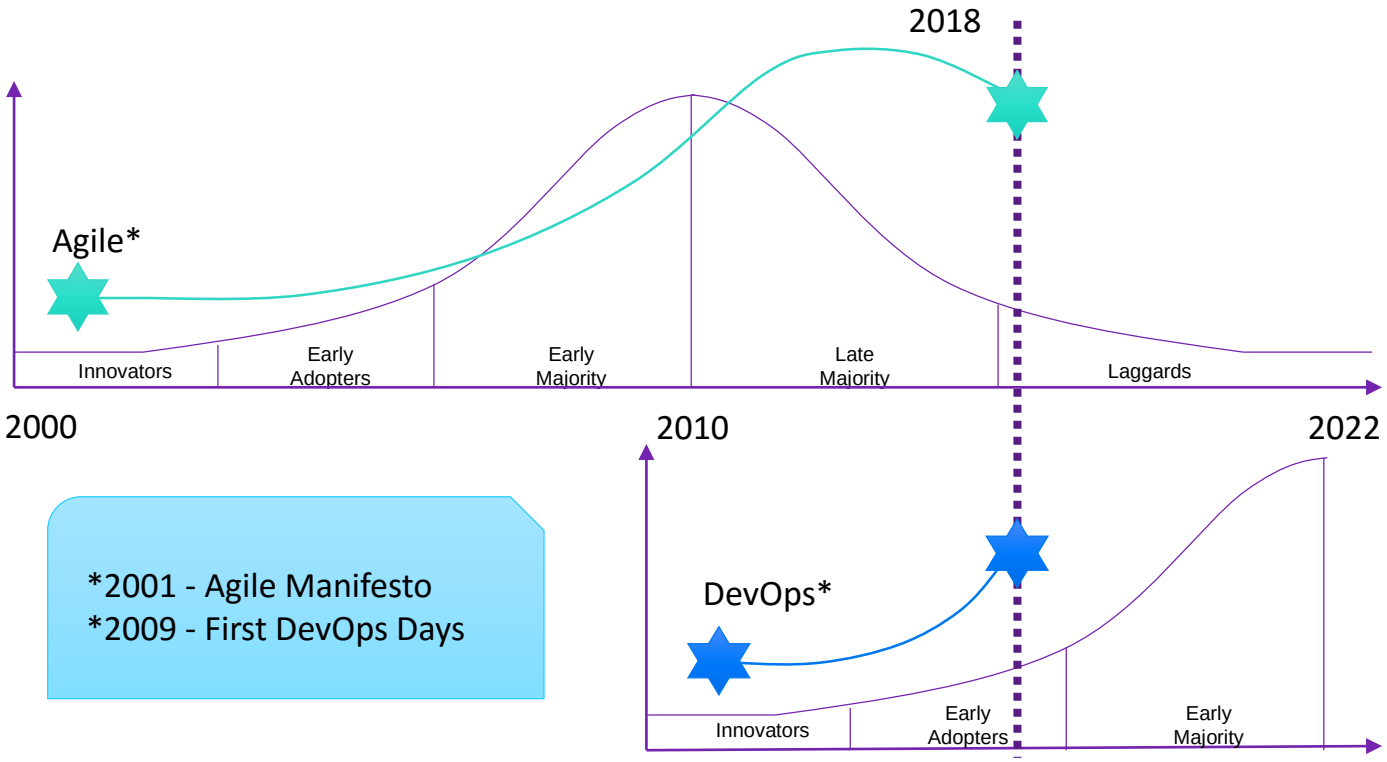
Adoption of DevOps

DevOps definition: **DevOps** (development and operations) is an enterprise software development phrase used to mean a type of agile relationship between development and IT operations. The goal of **DevOps** is to change and improve the relationship by advocating better communication and collaboration between these two business units. (webopedia.com)



Release Management?

DevOps is still maturing



*2001 - Agile Manifesto
*2009 - First DevOps Days

DevOps definition varies widely

People - Culture

Systems Thinking

Value Stream Management

Continuous Feedback

Multidisciplinary Teams

Autonomous Teams

Experimentation & Learning

Process - Techniques

Agile

Continuous Delivery

Application Performance Management

Microservices

Shift Left

Release Management

Containerization

Infrastructure as Code

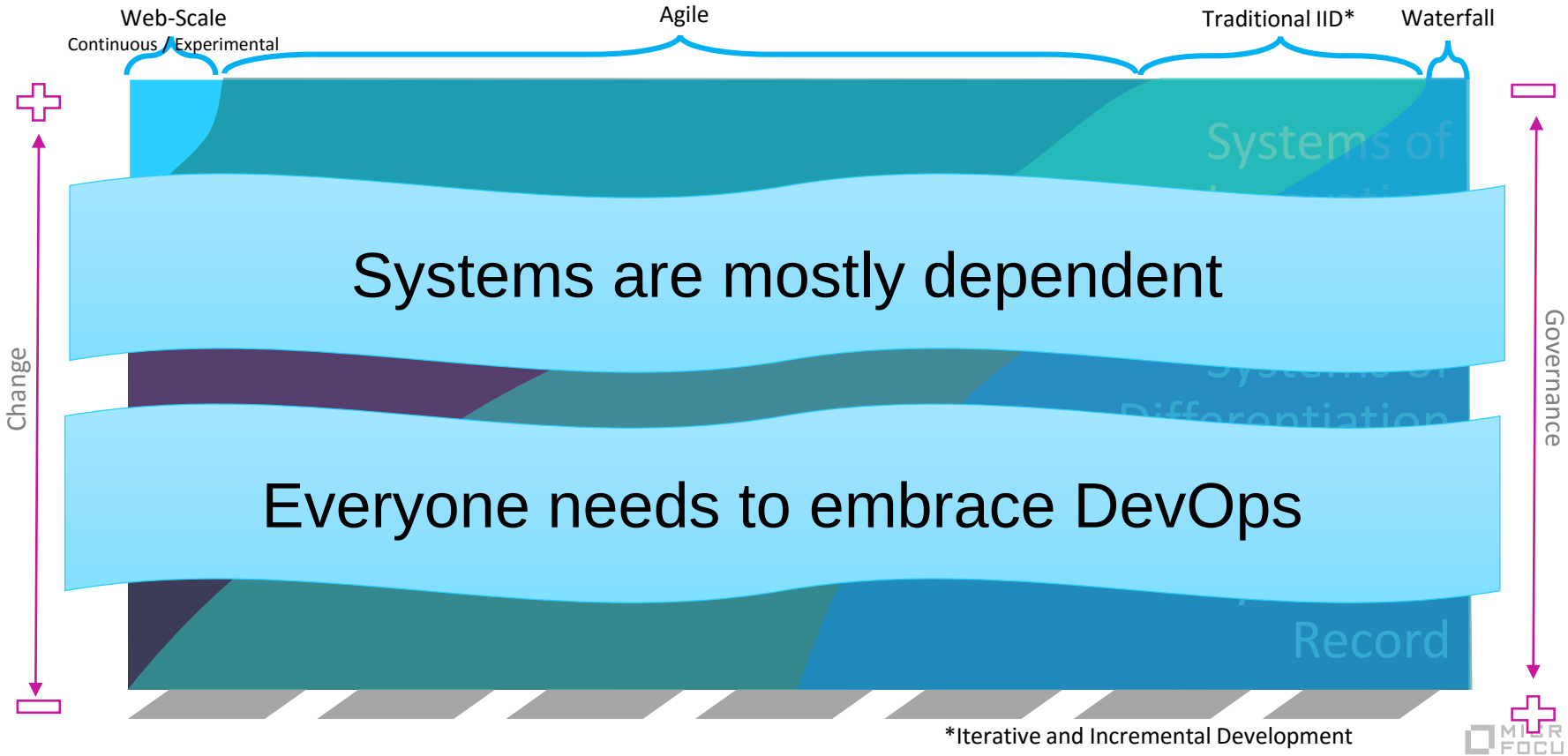
Tools - Automation

“Systems Thinking” is a central tenet

Systems Thinking *is the DevOps goal of being aware of how your actions not only affect your team, but all the other teams involved in the* ***release*** *process. It is a cultural journey achieved by* ***continual experimentation*** *and amplification of feed back loops.*

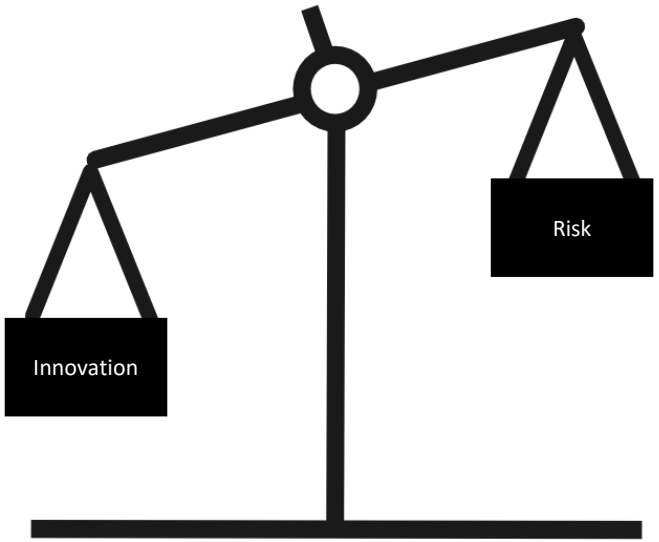
Thinking in Silos	• Lack of dependency planning • Misaligned priorities • Finger pointing
Systems Thinking	• Visibility and Transparency • Shared Goals • Ownership and Collaboration

Most organisations are/will be “Multi-Modal”



Enterprises need to balance Agility and Control

"Move fast without breaking things"



Governance, Risk and Compliance	Sarbanes-Oxley, PCI-DSS, HIPAA …	Segregation of Duties	Information Security, Audit
Multi-platform Development	Distributed	Mainframe	Mobile
Dependent Applications	Integration to Core Systems, Back Office Systems	Integration to third-party providers	
Shared Environments	Multiple Apps running on Infrastructure	Schedule Access to Test	
Customer Representative Environments	Performance Testing		

Release Management – Micro Focus

End-to-end view of releases to (dependent) systems – one way to achieve System Thinking whilst balancing Agility (Innovation) and Control (Risk)

Release Planning

Dependency Management

Environment Management

Release Execution

Deployment Planning

Deployment Execution

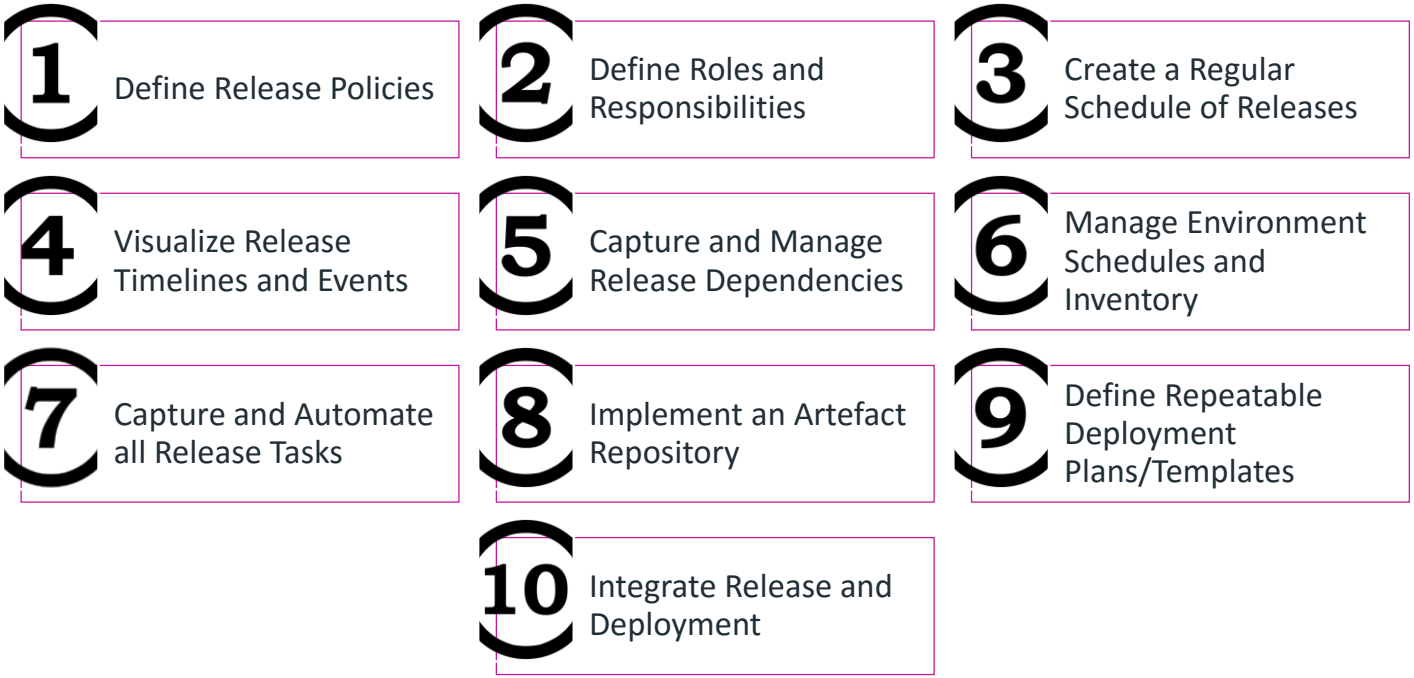
Integration to Change Management

Release Management needs to adapt to DevOps

- Engage all stakeholders (Dev/Test/Release/Ops).
- Provide governance and audit but remain accessible and easy to use.
- Agnostic to “development” and “deployment” tools.
- Agnostic to “development” approach – multi-modal enablement.
- Align to Service / Change Management as it exists in the Enterprise.
- Capture KPIs/Metrics to demonstrate value of DevOps/RM

Release Management Good Practices

Release Management “Good” Practices



Define Release Policies

*Determines under which circumstances different **Release Types** should be used as well as the standard set of milestones that selecting a particular **Release Type** implies in the **Release Calendar**.*

- Major Release:
  - Use for planned quarterly full content releases, e.g. v1.0, v2.0
  - Dev → Integration → UAT → Pre Prod→ Production
- Minor Release:
  - Use for planned monthly incremental content releases, e.g. v1.1, v2.1
  - Dev → UAT → Pre Prod → Production
- Patch Release:
  - Use for planned defect fixes, e.g. v1.1.1, v2.1.1
  - Dev → Pre Prod → Production
- Emergency Release:
  - Use for unplanned production outages
  - Dev → Production
  - *(typically not displayed on Release Calendar)*

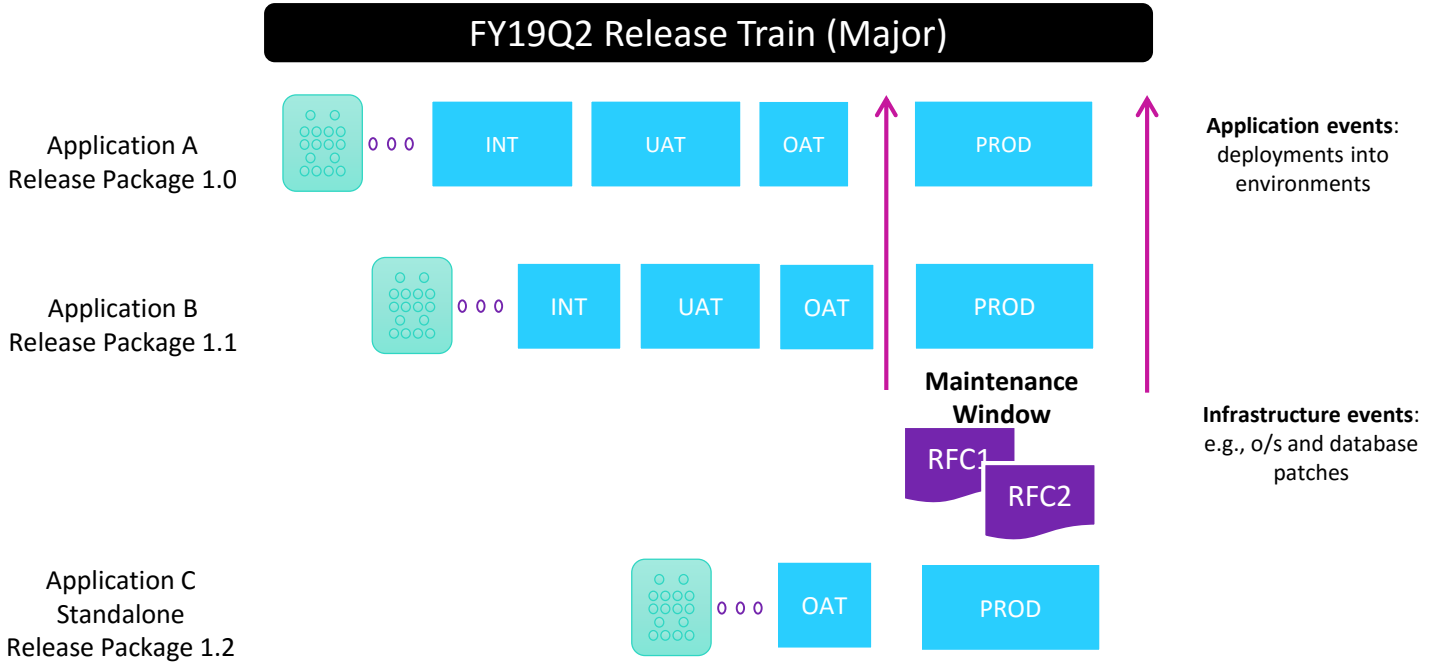
Define Roles and Reponsiblities (RACI)

	Product Owner	Release Manager	Developer	Tester	Deployment Engineer	Executive
Planning	C	CAR	C	C	C	I
Build	A	I	R			
Packaging	A	C	R			
Deployment	I	A	RI (for INT)	RI (for UAT)	RI (for PROD)	I
Approval	R (for INT)	AR		C	C	R
Rollback	I	A		I	R	I
Review	C	RA	C	C	C	I

Responsible - those who do the work to achieve a task
Accountable - those who are ultimately accountable for the correct and thorough completion of the task
Consulted – those who provide input to the task
Informed – those who need to be kept informed about the task

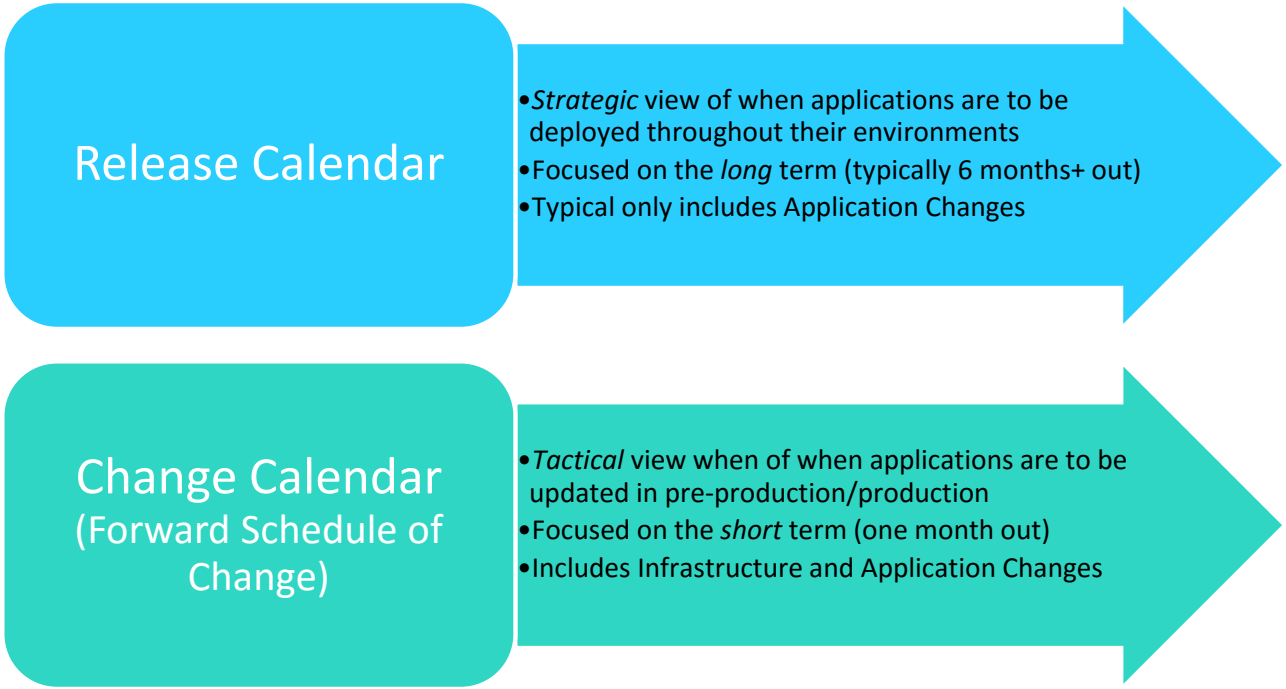
Create a Regular Schedule of Releases

*Releases Trains can be used to coordinate the deployment of multiple applications. Release Trains are scheduled on a recurring basis, i.e. quarterly, monthly and **Release Content** aligned to the Release Train depending on business needs.*



Visualise Release Timelines and Events

*The **Release Calendar** is a set of published milestones that details when Releases are planned to transition through the different development, test and production stages.*

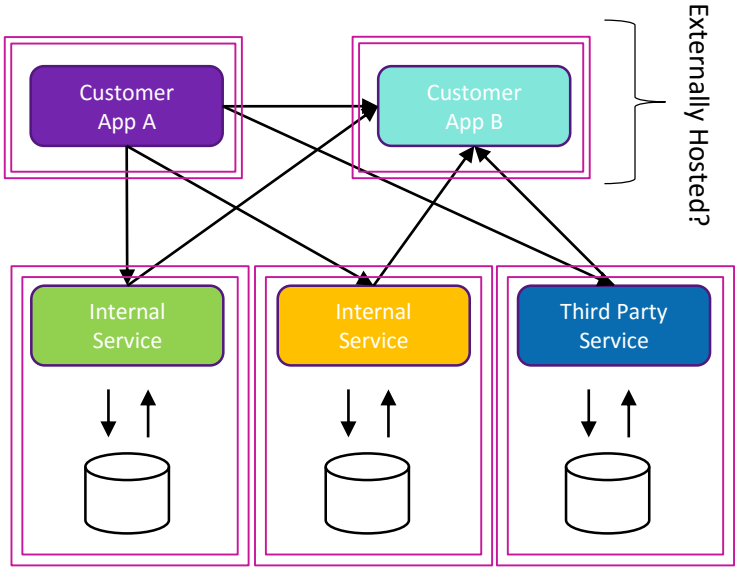


- Release Management adds an entry to the Change Calendar via Release Planning

Capture and Manage Release Dependencies

*Use **Application Dependency Mapping** to capture the dependencies between applications and ensure the impact and delivery of new releases is managed and controlled.*

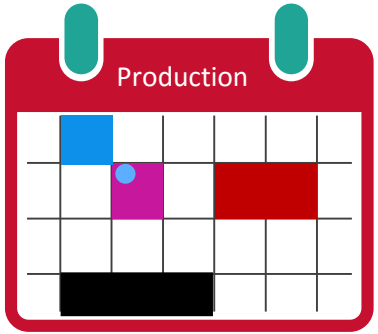
- Capture dependencies between systems, applications and components:
  - Code level
  - Test level
  - Data / Runtime …
- Use Releases (**Release Trains**) to manage the impact and delivery of dependent applications
- Schedule and manage access to Environments to ensure there are no conflicts
- Use Application Dependency Mapping tool and/or CMS/CMDB even Excel spreadsheet to start



Independent services with cross communication through apis

Manage Environments Schedules and Inventory

*The purpose of **Environment Management** is to ensure that all activities related to Test and Production Environments are scheduled, approved and managed and that an **Environment Inventory** of installed software versions also be recorded.*



Application	Version
App A	1.0
App B	1.1
…	…

- Environments are logical and define schedules for:
  - Agreed Maintenance Windows
  - Blackouts (e.g. Change Freezes), Whiteouts
  - Scheduled/Approved/Rejected Deployments
- **Environment Groups** can be used to group a number of similar environments together and that can be deployed to in parallel, e.g. “Test Group” contains: Integration, UAT, Performance
- Restricted environments should be controlled/permissioned with relevant **Approvals**.
- Environments should visualise the **Inventory** of Deployed Application Versions, from a **CMS** or through linkage to an external Deployment tool.

Capture and Automate all Release Tasks

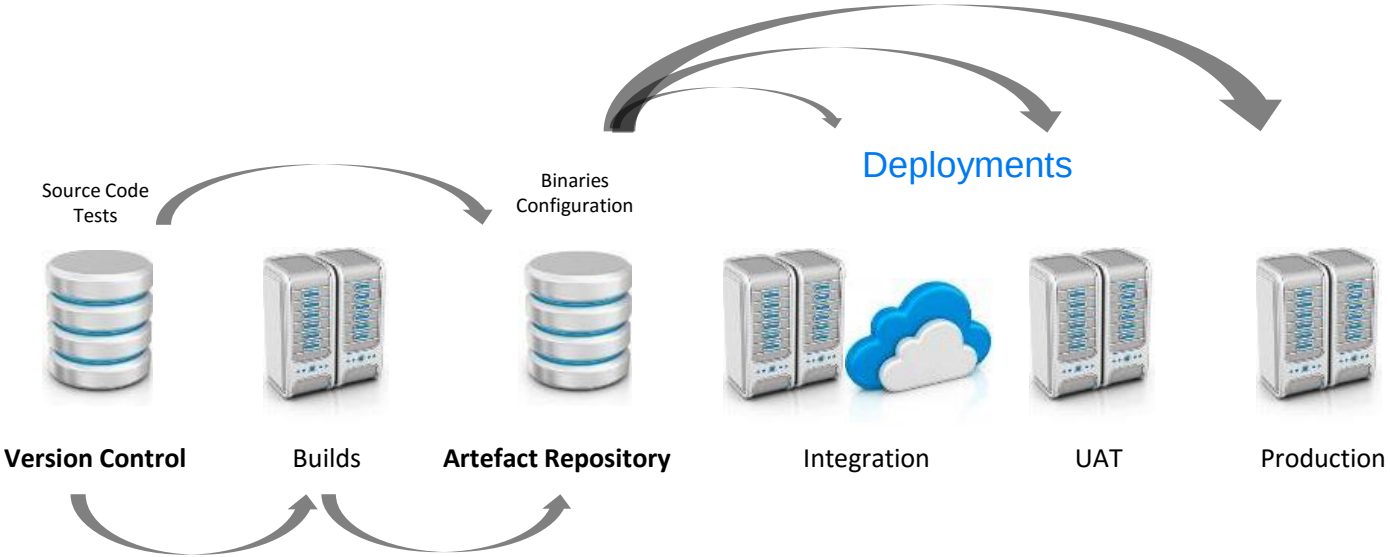
*The purpose of Release Planning is to assign authorized **Changes** to **Release Packages** and to define the scope and content of Releases. The Release Planning process develops a schedule for building, testing and deploying the Release.*

- Planning Release Trains, Application Releases, Turnovers
- Assigning Content to Release Trains, Application Releases, Turnovers
- Reviewing Implementation/Test/Rollback Plans
- Assigning Resources
- Creating Backup Requests
- Creating Outage Notifications
- Integration to Change Management (FSC)
- Post Implementation Review

A Release Manager is essentially the “Project Manager” for a Release

Implement an Artefact Repository

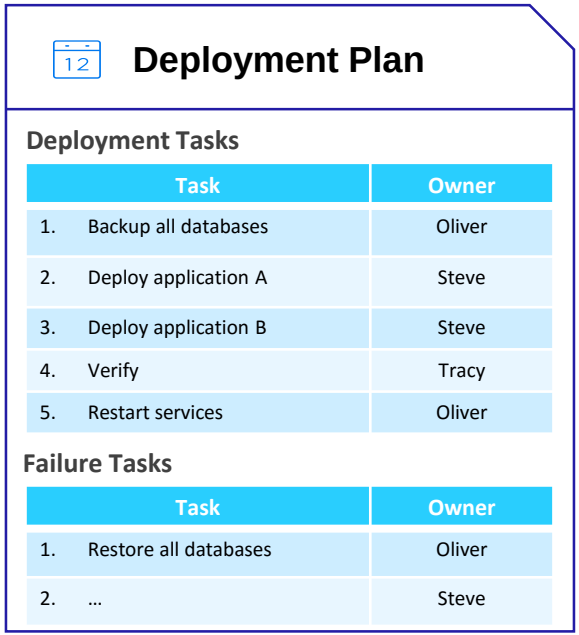
An Artefact Repository is used to securely store releasable artefacts (i.e. built binaries, database scripts etc.). Formal environments should only be deployed to be extracting versions artefacts from the Artefact Repository.



An Artefact Repository should to be used to capture not just artefacts but the metadata around them, i.e. what has been deployed to where, by whom.

Define Repeatable Deployment Plans (Templates)

*A **Deployment Plan** defines the sequence of operations or steps that must be carried out to deliver changes into a target system environment. The individual operations within a deployment plan can be executed manually or automatically.*



Deployment Plan

Deployment Tasks

	Task	Owner
1.	Backup all databases	Oliver
2.	Deploy application A	Steve
3.	Deploy application B	Steve
4.	Verify	Tracy
5.	Restart services	Oliver

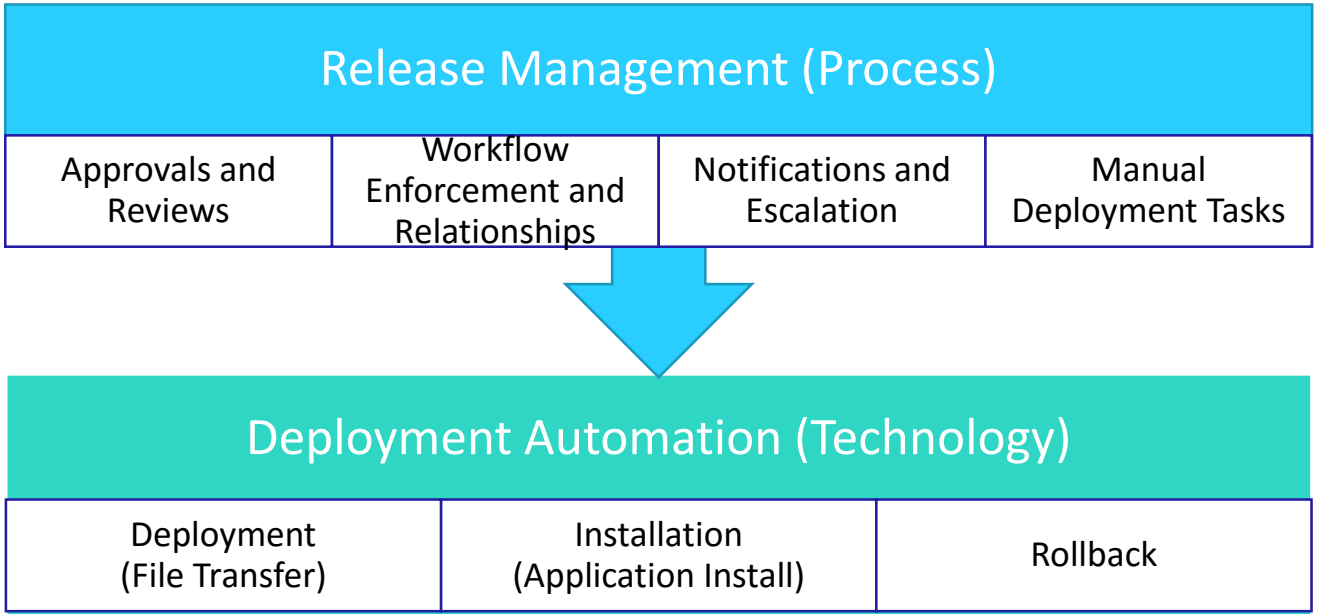
Failure Tasks

	Task	Owner
1.	Restore all databases	Oliver
2.	…	Steve

- Deployment Plans should be:
  - **Repeatable**
  - **Rehearsable**
  - **Timebound**
  - **Scheduled**
  - **Approved**
- Deployment plans should be constructed based on what has been deployed to the previous environments
- Deployment plans should be able to be saved as **templates** for reuse
- Deployment plans should include both **Deployment** (new) and **Failure** (rollback) tasks.

Integrate Release and Deployment

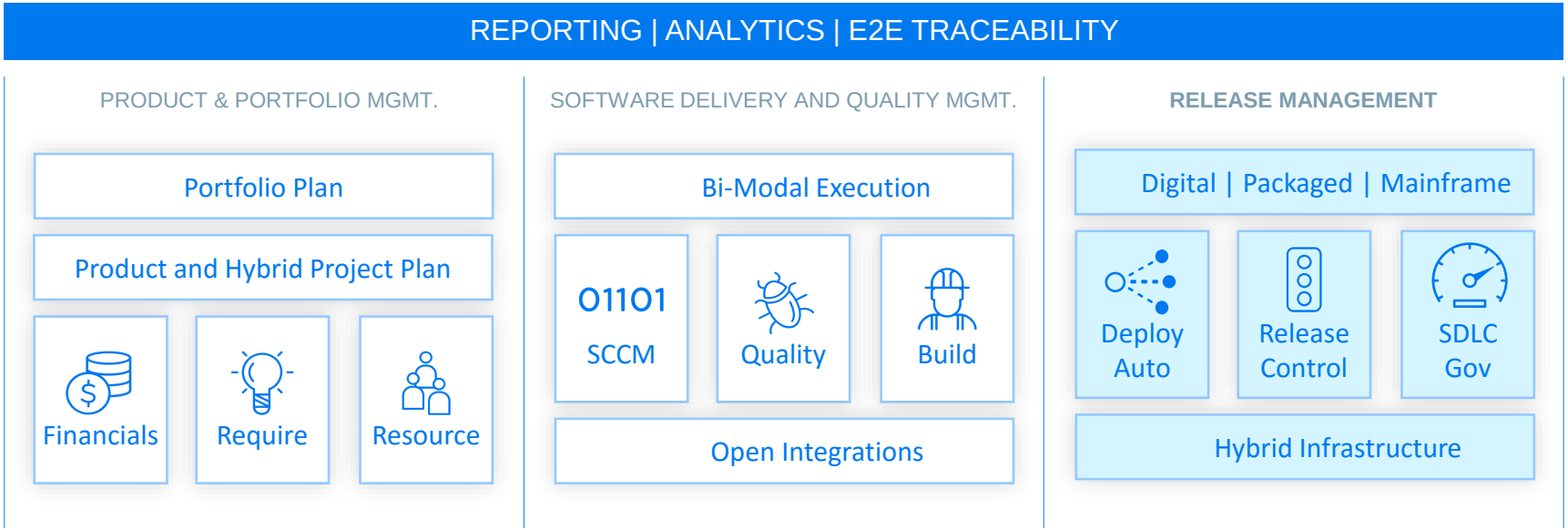
Release Management *is the process responsible for Planning, Scheduling and Controlling the movement of Releases to Test and Live.* ***Deployment*** *is the activity responsible for movement of new or changed hardware, software, documentation etc.*



Micro Focus Solutions

Micro Focus Portfolio and Lifecycle Management

Accelerating an organization's ability to securely deliver high-quality applications at enterprise scale and manage, track, and unlock the value of investments you've already made.



Micro Focus Release Management Solutions

Deployment Automation

orchestrate ● automate ● deploy ● audit

ORCHESTRATE COMPLEX DEPLOYMENTS ACROSS ALL ENVIRONMENTS

Release Control

plan ● track ● deliver ● integrate ● audit

PLAN, TRACK, MONITOR AND RELEASE COMPLEX APPLICATIONS ACROSS ANY TECHNOLOGY

Micro Focus Release Control

Release Control enables organizations to plan, control, and orchestrate their release processes for both mainframe and distributed systems, from definition to deployment with a visual release calendar and automated approval processes.

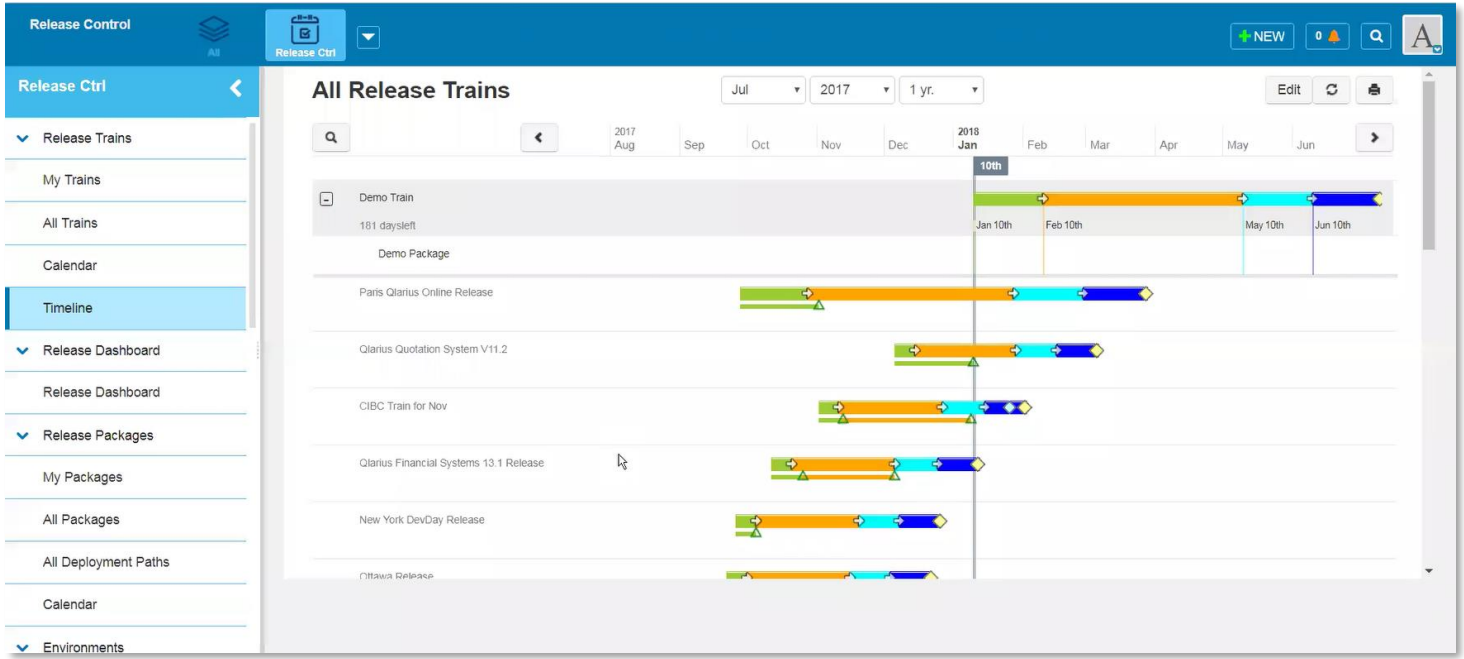
Plan, track, and control all release content

VISUALIZE & CUSTOMIZE ALL RELEASE ACTIVITIES SIMPLY AND EASILY

RELEASE PLANNING AND EXECUTION CO-ORDINATION

REDUCES RELEASE RISK BY PROVIDING VISIBILITY OF RELEASE PROCESSES TO ALL AREAS OF YOUR ORGANIZATION

ENSURES AUDIT AND COMPLIANCE BY TRACKING ALL RELEASE ACTIVITIES



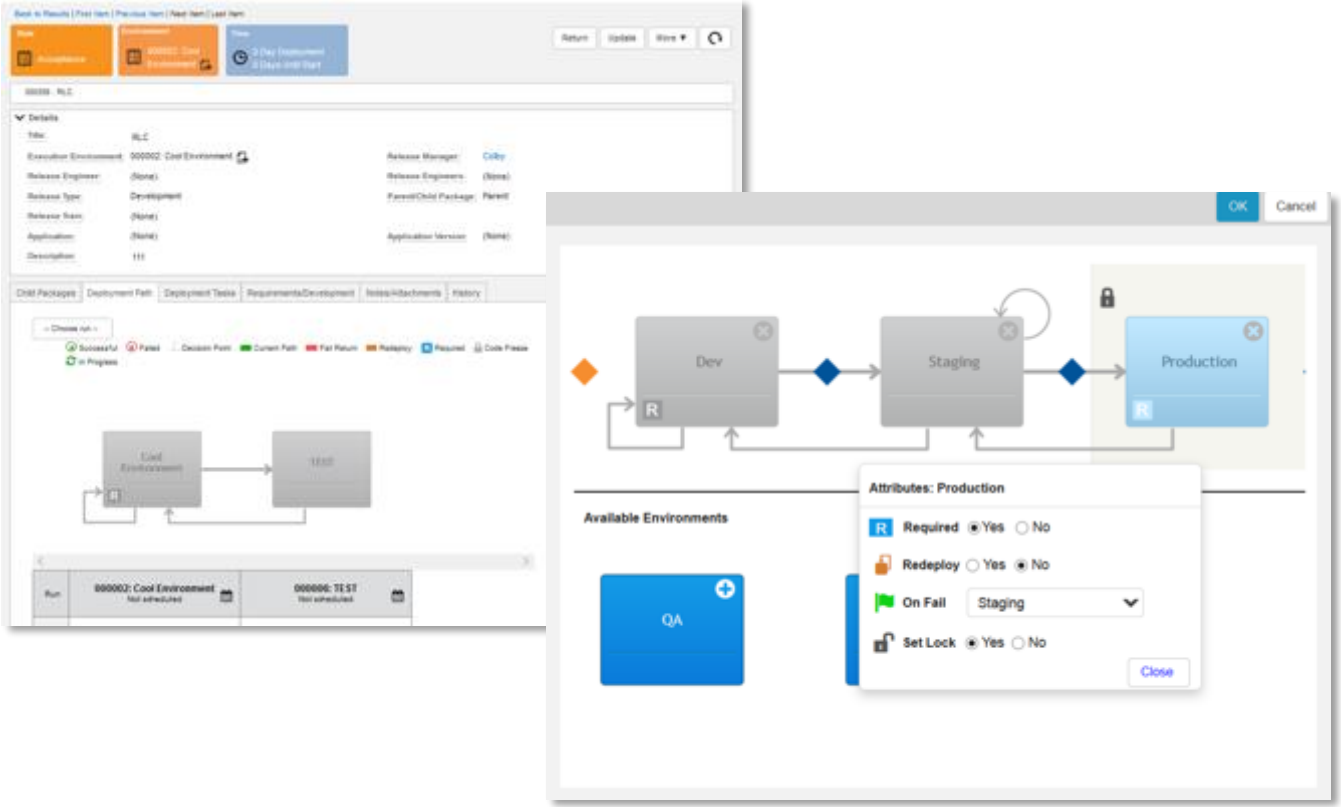
MICRO FOCUS

Automate deployments, tasks and actions

AUTOMATICALLY PROMOTE MULTIPLE APPLICATIONS ACROSS ENVIRONMENTS IF PROCESS RULES ARE MET

ENFORCE ENVIRONMENT SEQUENCING, RE-DEPLOY AND RETROFIT PATHS FOR APPLICATIONS OR TRAINS

RELEASE MORE OFTEN, RELEASE MORE EACH TIME, AND REDUCE THE RELEASE WINDOW

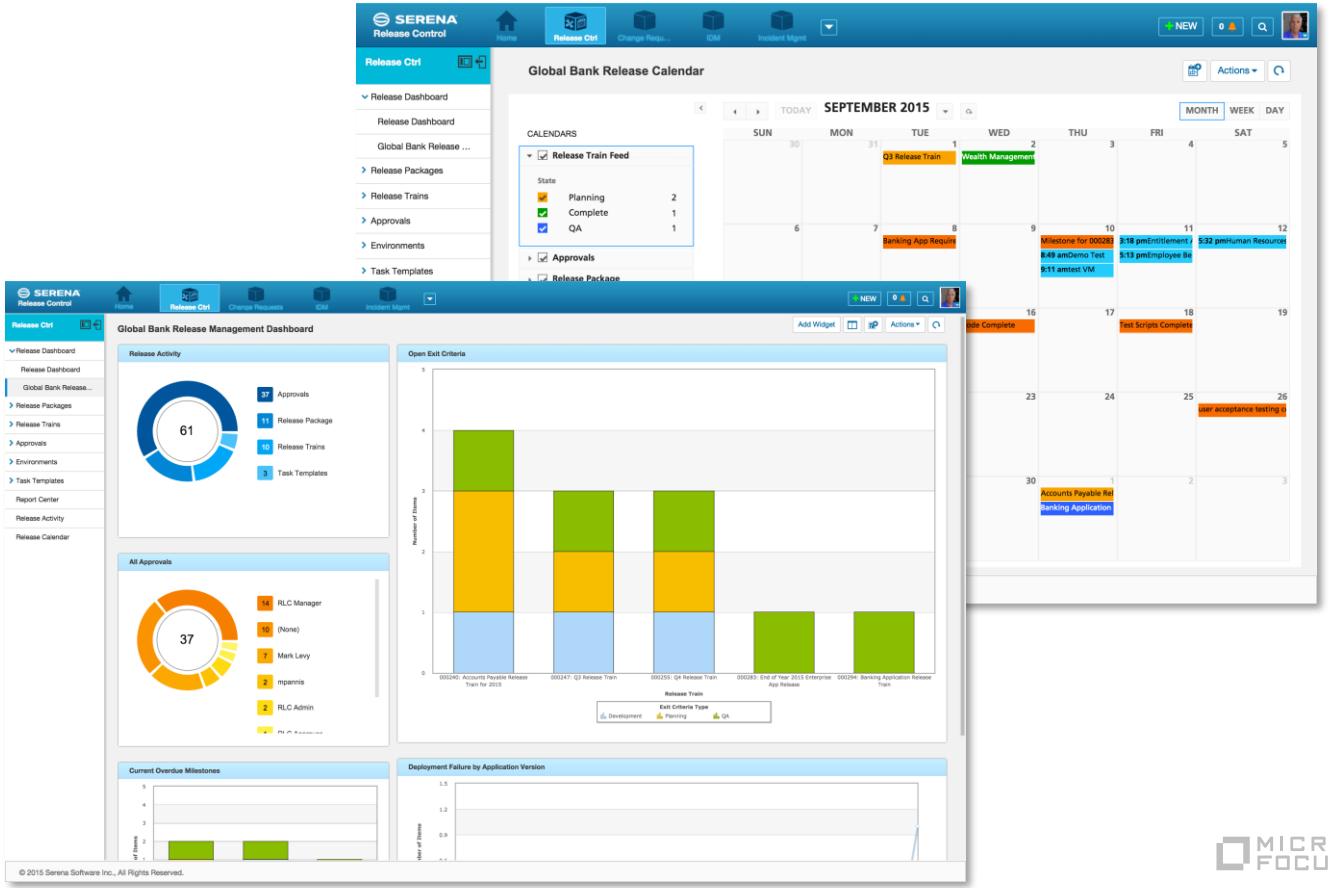


Actionable insight from dashboards and reports

VIEW UPDATES ON ALL RELEASE ACTIVITIES, FROM EXECUTIVE-LEVEL STATUS TO DEPLOYMENT TASK DETAILS IN REAL-TIME

SHARED CALENDARS AND A CONFIGURABLE TIMELINES PROVIDE VISIBILITY ACROSS YOUR ENTIRE ORGANIZATION

MAINTAIN A SINGLE SOURCE OF TRUTH TO KEEP THE ENTIRE ORGANIZATION ALIGNED



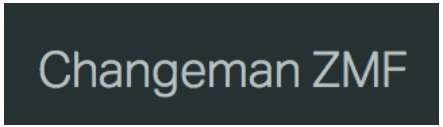
Cross Product Integration & Collaboration Capabilities

EASE OF INTEGRATION TO EXISTING CUSTOMER INFRASTRUCTURE

OUT-OF-THE-BOX SUPPORT FOR COMMON APPLICATION DEPLOYMENT TOOLS AND SYSTEMS

AVAILABLE PLUGIN CATEGORIES

- REQUEST PROVIDERS
- DEPLOYMENT UNIT PROVIDERS
- EXECTUION PROVIDERS



Release Control benefits:

- **Visibility and Tracking**: Centralized release calendar, process metrics, and dashboards
- **Compliance and Control**: Process enforcement, routing, and approvals with built-in audit trails and reports
- **Multi-Platform Support**: The only release management solution that supports both mainframe and distributed
- **Process Automation**: Automation and workflow that replace manual process steps to enforce your release policies and processes
- **Integration**: Flexible and open architecture that allows easy integration to current enterprise tools via an extensible, enterprise ready plugin architecture

Release Control: Customer Proof

"With Serena (now part of Micro Focus) Release Control we have a reliable and repeatable release process, greater visibility into the release activities for all of our stakeholders and significantly improved confidence in our compliance with international laws and internal audit requirements."

Director Change and Release, Global Financial Services Leader

- Reduced release window; production release reduced from 3 hours to 35 minutes
- Anticipated benefits of up to $3M annually through broad use of Release Control
- Gained an internal rate of return of 185 percent over five years

World's largest credit solutions provider

- Eliminated audit and compliance risks
- Provided stakeholders clear insight into the impact of changes to its respective applications and assets
- Automated promotion of changes into production removing major source of errors
- Delivered greater environment stability, predictability and reliability

Global Financial Services Leader

Demonstration

Q&A

Kevin A. Lee
kevin.lee@microfocus.com
+44 (0)7799 072507

Thank You!!