DevOps: The 5 c+ 1 way

Hoo Chuan Wu
DevOps Specialist

Agenda

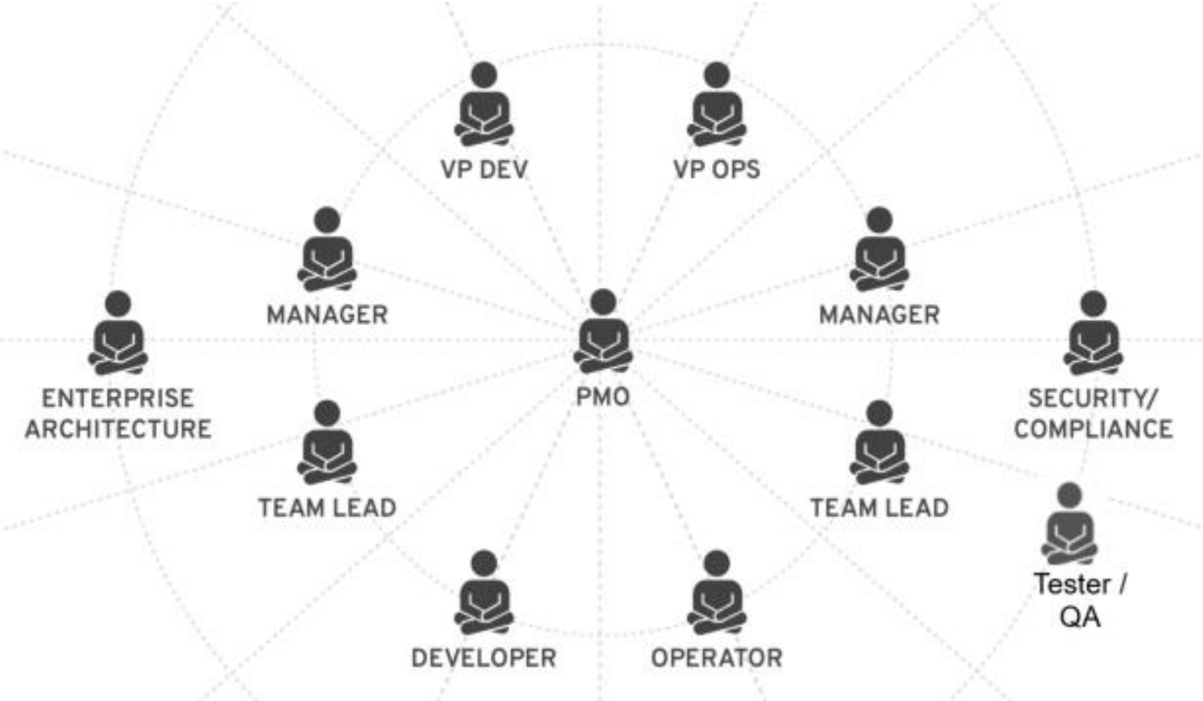
- Regulated DevOps
- Red Hat's Value Proposition
  - Productivity, Agility, Release Velocity
- 5 + 1 View Model
- Example of Bi-modal DevOps
- DevOps with OCP

DevOps

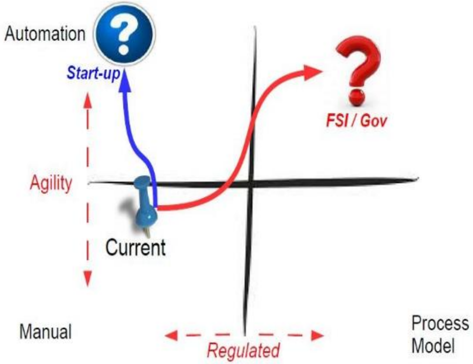
Practice…
Communication
Collaboration

Teaming Culture

Automation is key



Regulated DevOps



Governance and Compliance Needs

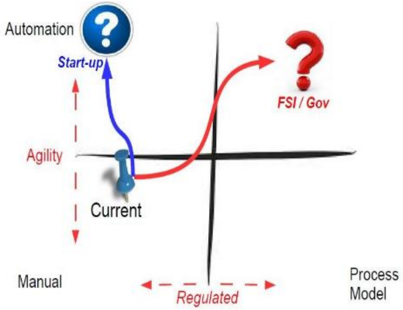
1. How do you ensure the ***maker*** is not the ***checker*** ?

2. How do you ensure the development team is using the right source ***code base*** that represent the current production run-time ?

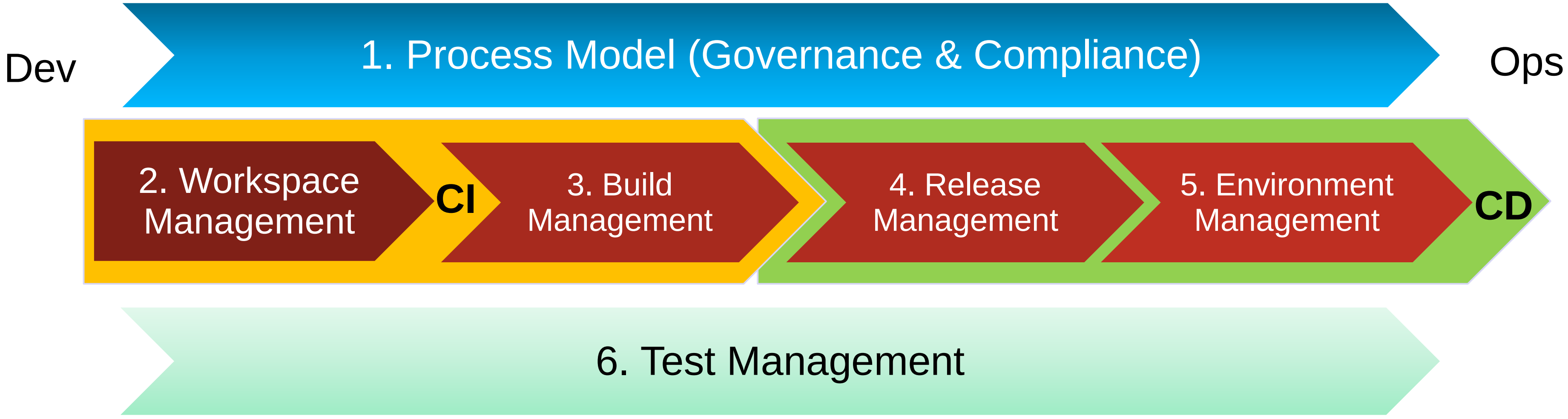
3. How do you ensure the build ***pipeline integrity*** is intact when the production run-time is updated ?

https://www.linkedin.com/pulse/different-faces-devops-chuan-wu-hoo/

Regulated vs Regular DevOps



Level-1 DevOps



Dev

1. Process Model (Governance & Compliance)

Ops

2. Workspace Management

CI

3. Build Management

4. Release Management

5. Environment Management

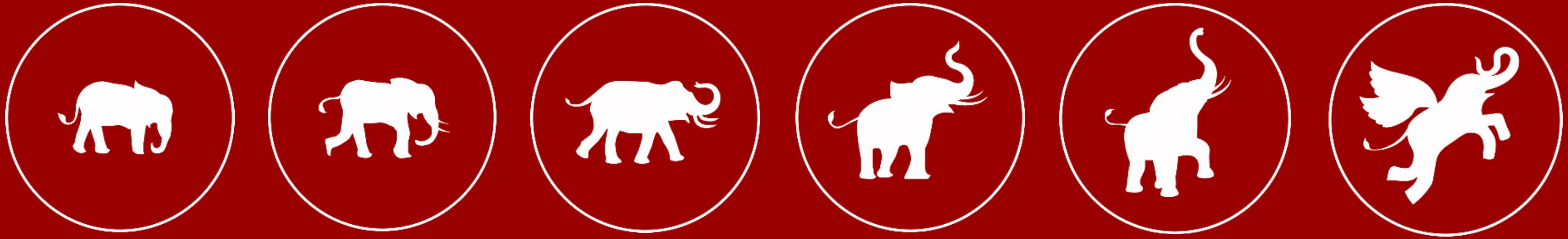
CD

6. Test Management

Test & Security

Digital Darwinism

Bi-modal DevOps Journey



Re-Org to DevOps

Automation: Ansible

CI & CD Deployment Pipeline

Self-Service, On-Demand, Elastic Infrastructure

Advanced Deployment Techniques

Microservices

(and flying elephants!)

Address Business Driver

Increase Business Agility

- Reduce the time to service
- Reduce the time to validate a hypothesis

Increase Quality of Service

- Reduce the number of service disruptions
- Minimise or avoid service downtime
- Improve service performance (elasticity/scalability)
- Improve quality of release

Increase Operational Efficiency

- Reduce resource utilisation
- Increase personnel efficiency
- Reduce over-time

Red Hat's Value Proposition

Increase Business Agility

- Improve Team Productivity
- Increase Release Velocity

Increase Quality of Service

- Service Performance (Elasticity & Scalability)
- OCP – Portability, Resiliency (self-healing)
- OCP/ VM – Pending Availability
  - Blue/Green Deployment
  - Parallel Testing (OCP / VM – Pend Availability)
  - Test env (release + persistent env)

Increase Operational Efficiency

- Workflow Approval integrated with Automation
  - **“Single Button Click”** approach
- Improve Team Productivity

The 5+1 View Model

Dev

Re-Org to DevOps

1.Process Model

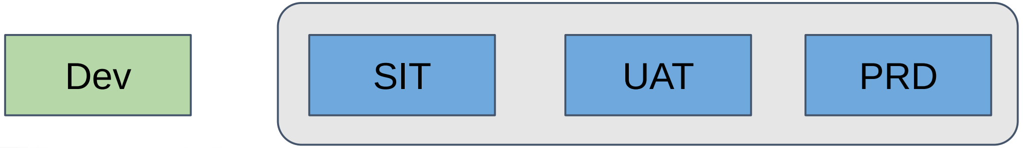
(communication + collaboration → automation)

Ops

Activities

Artifacts

- **Workflow** (consolidated CM plans)
  - Tracks the Production Baseline
  - Manages the Release Tag
  - Preempts invalid promotion


Dev

SIT

UAT

PRD

Test & Security

https://www.linkedin.com/pulse/begin-your-devops-journey-red-hat-chuan-wu-hoo/

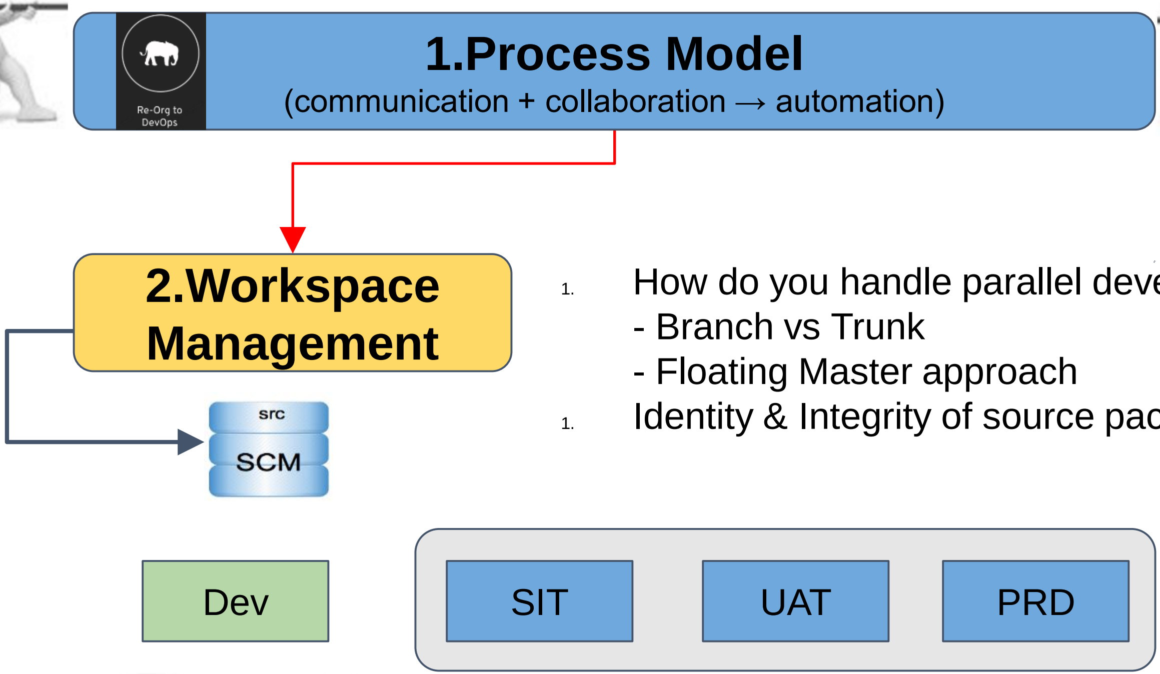
The 5+1 View Model

Dev

Ops

Activities

Artifacts



1.Process Model

(communication + collaboration → automation)

2.Workspace Management

Dev

SIT

UAT

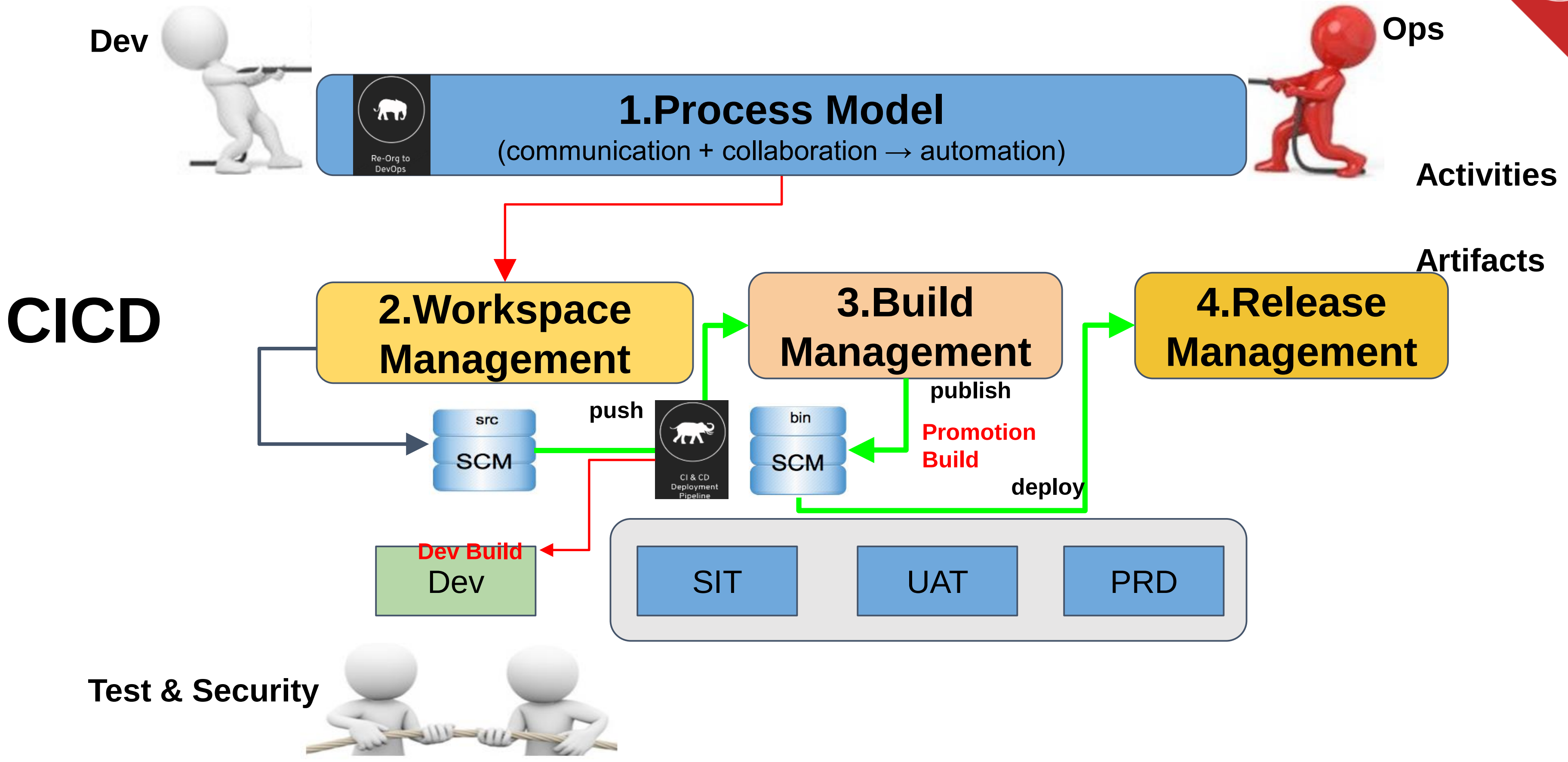
PRD

SCM Usage Model

1. How do you handle parallel development ?
   - Branch vs Trunk
   - Floating Master approach
1. Identity & Integrity of source package

Test & Security

The 5+1 View Model



Dev

Ops

1.Process Model

(communication + collaboration → automation)

Re-Org to DevOps

Activities

Artifacts

CICD

2.Workspace Management

3.Build Management

4.Release Management

push

publish

src

SCM

CI & CD Deployment Pipeline

bin

SCM

Promotion Build

deploy

Dev Build

Dev

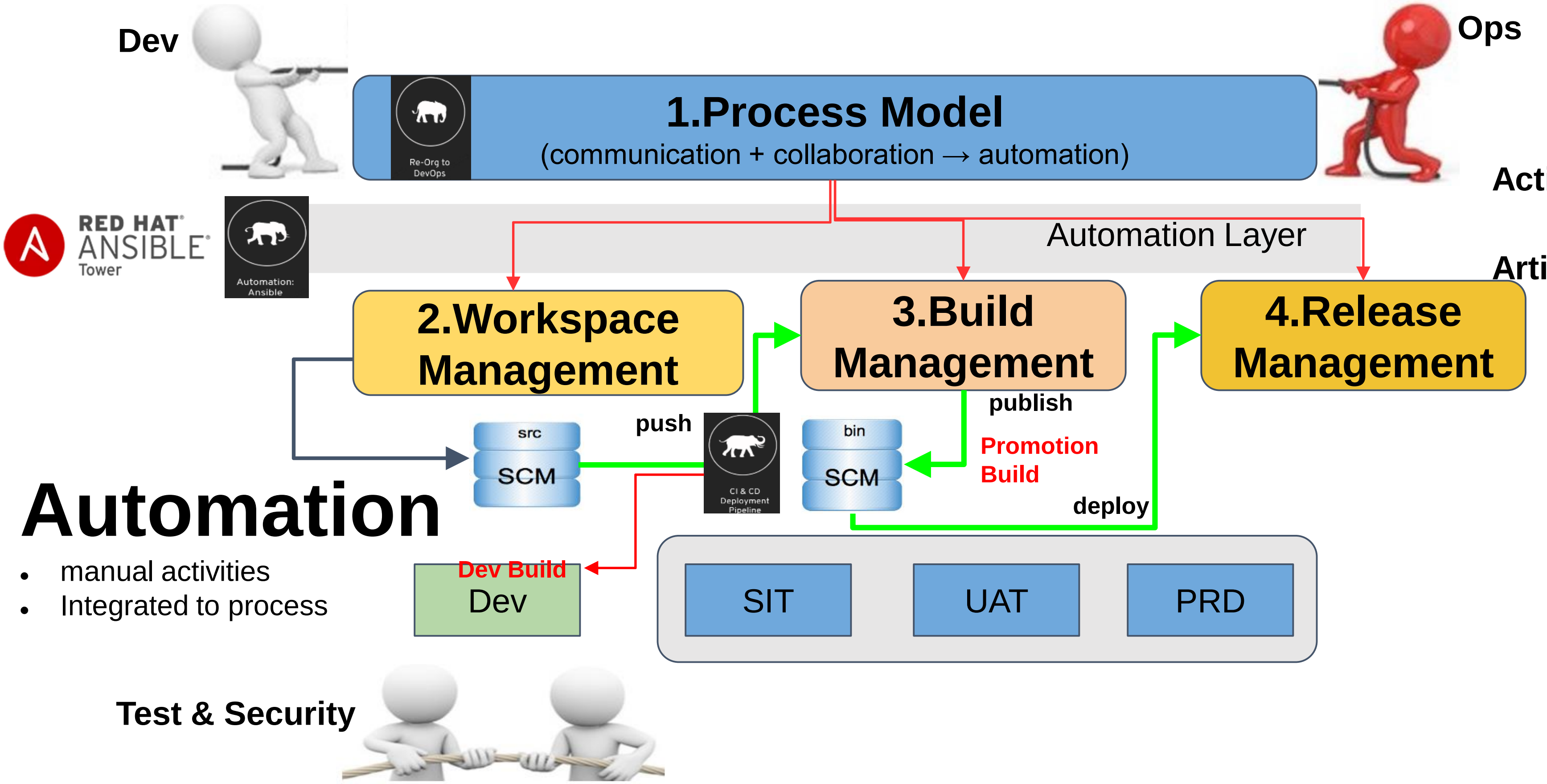
SIT

UAT

PRD

Test & Security

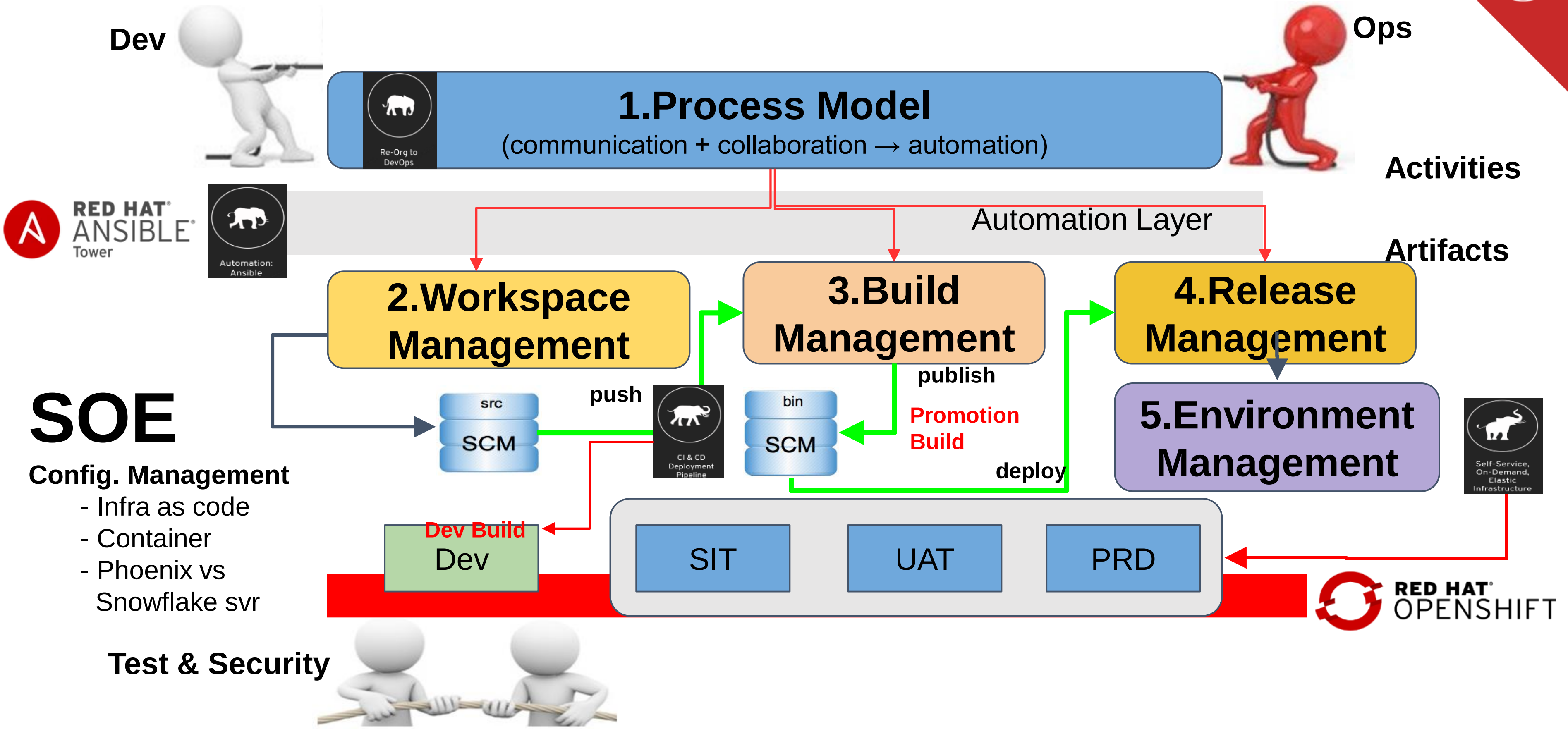
The 5+1 View Model



Automation

- manual activities
- Integrated to process

The 5+1 View Model



Dev

Ops

Activities

Artifacts

1.Process Model

(communication + collaboration → automation)

Re-Org to DevOps

RED HAT ANSIBLE Tower

Automation: Ansible

Automation Layer

2.Workspace Management

3.Build Management

4.Release Management

5.Environment Management

src

SCM

push

CI & CD Deployment Pipeline

bin

SCM

publish

Promotion Build

deploy

Self-Service, On-Demand, Elastic Infrastructure

Dev Build

Dev

SIT

UAT

PRD

RED HAT OPENSHIFT

SOE

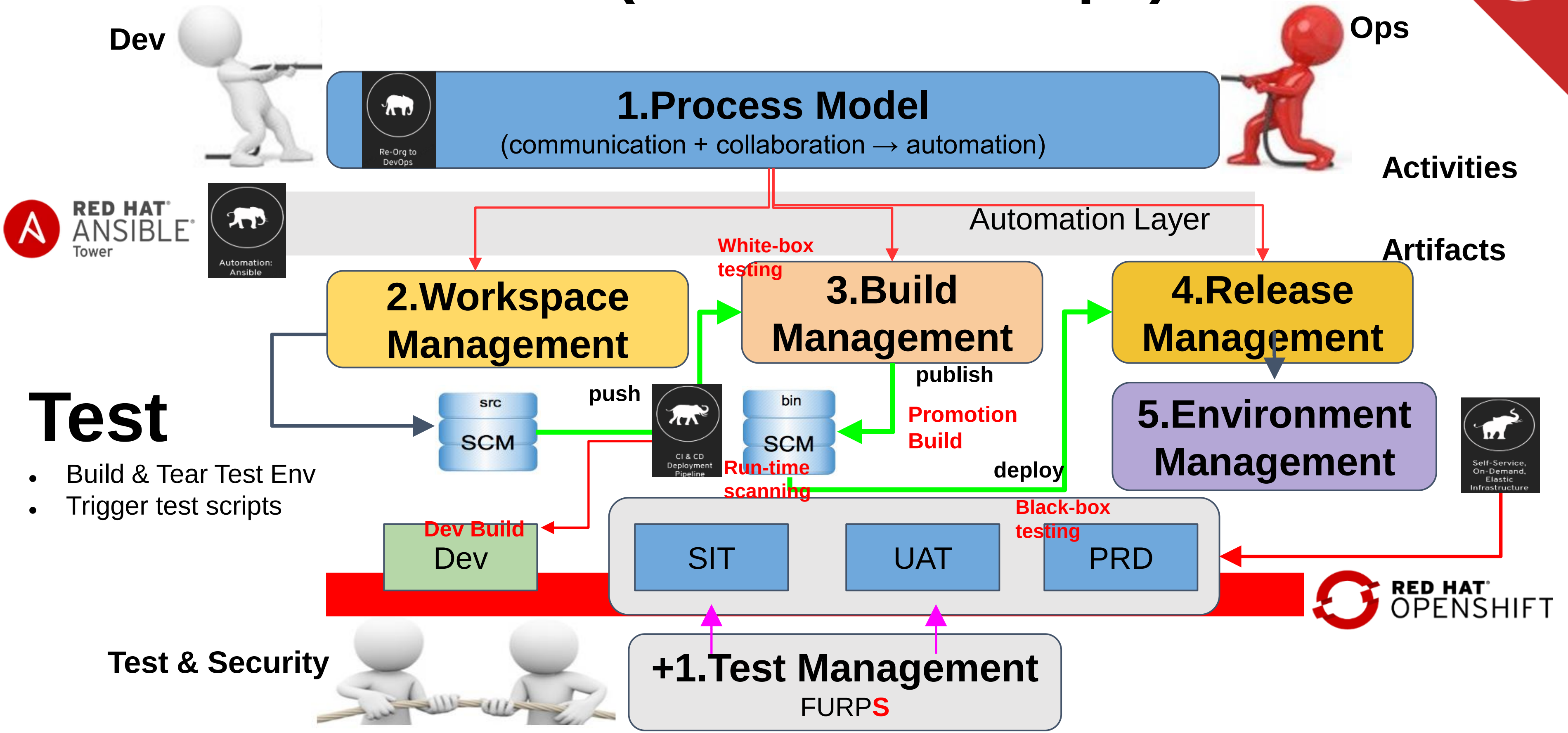
Config. Management

- Infra as code
- Container
- Phoenix vs Snowflake svr

Test & Security

https://www.linkedin.com/pulse/red-hat-devops-deployment-using-openshift-container-platform-hoo/

The 5+1 View Model (Bi-modal DevOps)



Dev

Ops

1.Process Model

(communication + collaboration → automation)

Re-Org to DevOps

Activities

RED HAT ANSIBLE Tower

Automation: Ansible

Automation Layer

Artifacts

White-box testing

2.Workspace Management

3.Build Management

4.Release Management

5.Environment Management

Self-Service, On-Demand, Elastic Infrastructure

Test

- Build & Tear Test Env
- Trigger test scripts

src SCM

push

CI & CD Deployment Pipeline

bin SCM

publish

Promotion Build

Run-time scanning

deploy

Dev Build

Dev

Black-box testing

SIT

UAT

PRD

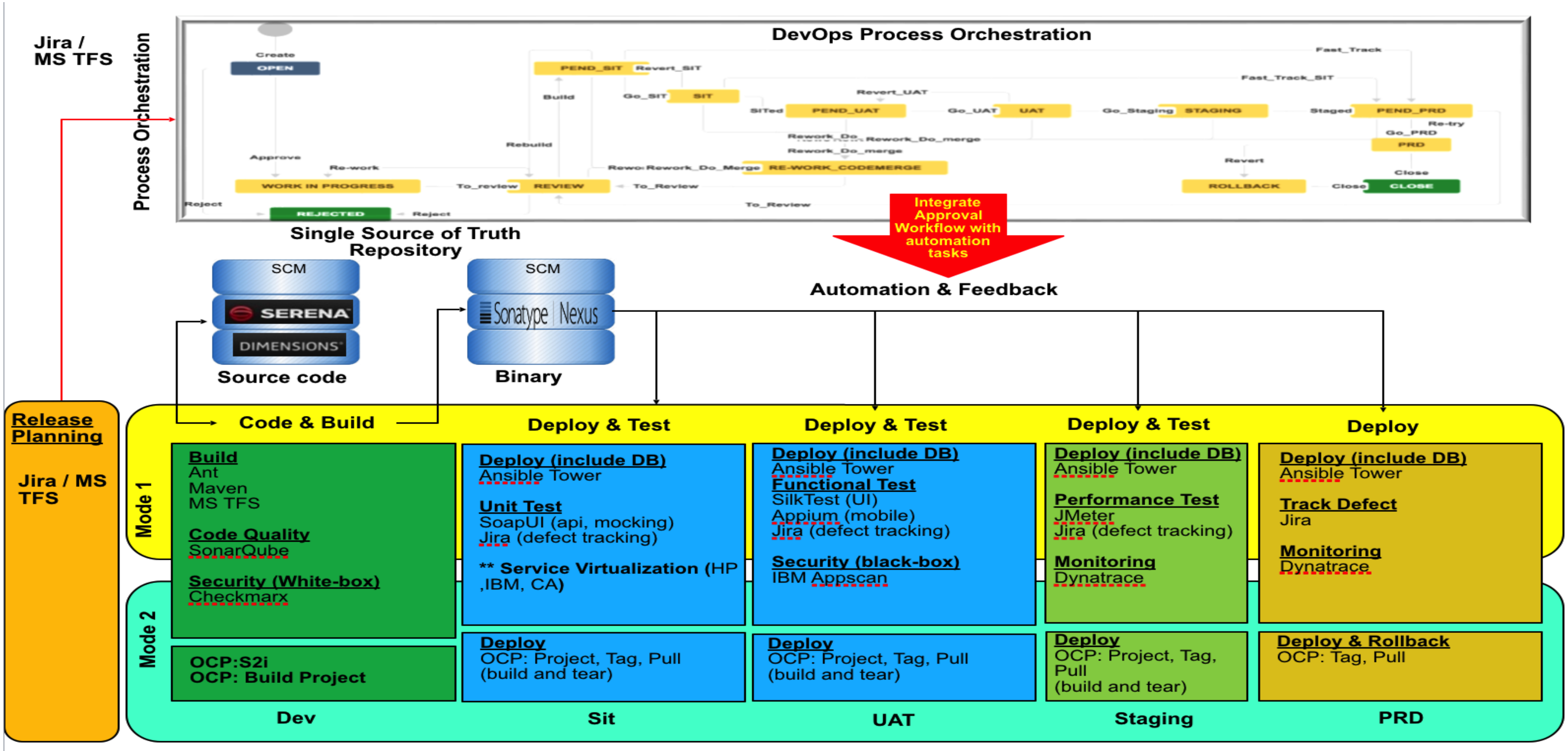
RED HAT OPENSHIFT

Test & Security

+1.Test Management

FURPS

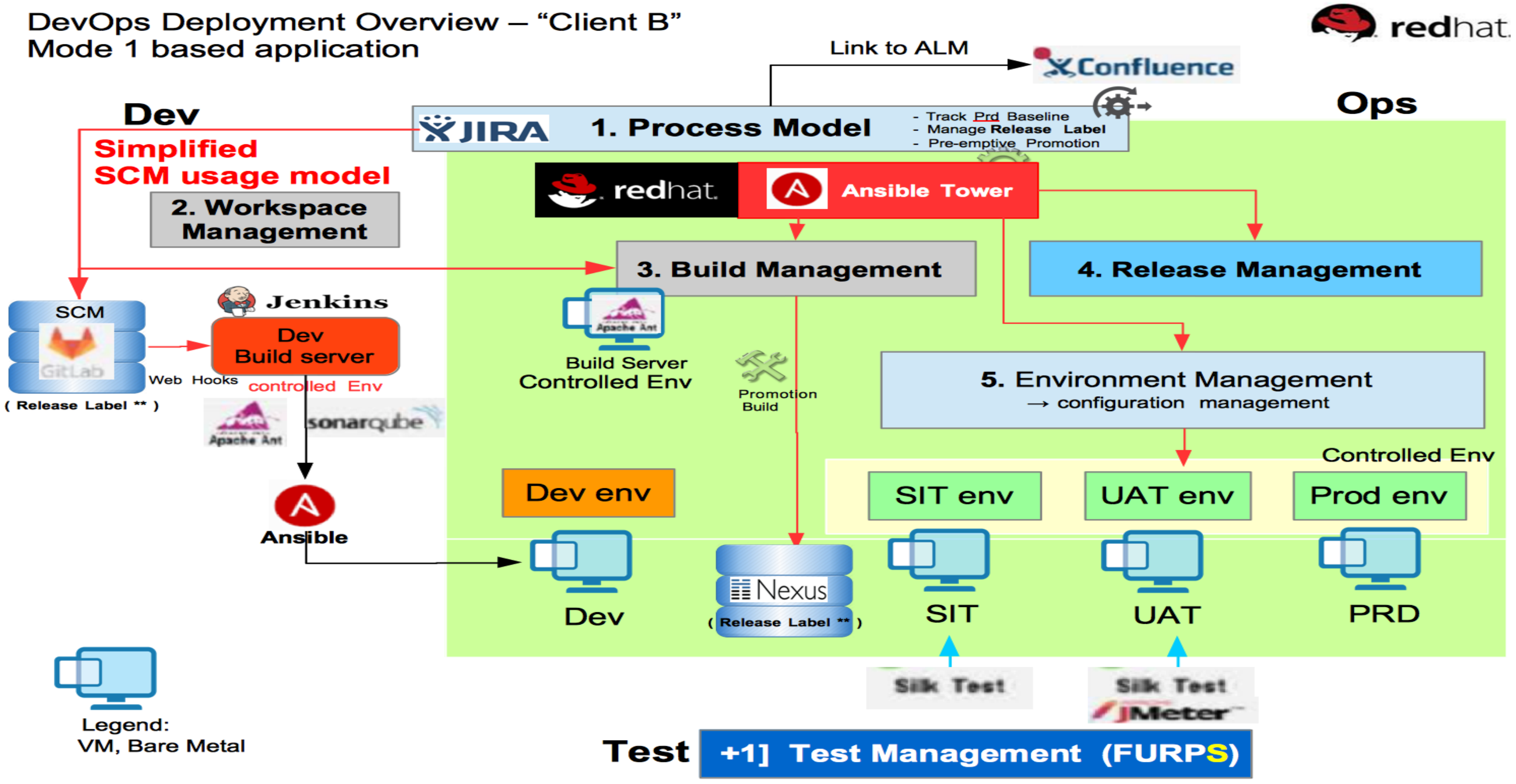
Level-1 DevOps – eg. Client's tools supporting practice



	Dev	Sit	UAT	Staging	PRD
	Code & Build	Deploy & Test	Deploy & Test	Deploy & Test	Deploy
Mode 1	**Build** Ant Maven MS TFS **Code Quality** SonarQube **Security (White-box)** Checkmarx	**Deploy (include DB)** Ansible Tower **Unit Test** SoapUI (api, mocking) Jira (defect tracking) ** **Service Virtualization** (HP ,IBM, CA)	**Deploy (include DB)** Ansible Tower **Functional Test** SilkTest (UI) Appium (mobile) Jira (defect tracking) **Security (black-box)** IBM Appscan	**Deploy (include DB)** Ansible Tower **Performance Test** JMeter Jira (defect tracking) **Monitoring** Dynatrace	**Deploy (include DB)** Ansible Tower **Track Defect** Jira **Monitoring** Dynatrace
Mode 2	**OCP:S2i** **OCP: Build Project**	**Deploy** OCP: Project, Tag, Pull (build and tear)	**Deploy** OCP: Project, Tag, Pull (build and tear)	**Deploy** OCP: Project, Tag, Pull (build and tear)	**Deploy & Rollback** OCP: Tag, Pull

OCP: Red Hat Openshift Container Platform

A Live Mode 1 DevOps Deployment



DevOps Deployment Overview – "Client B"
Mode 1 based application

redhat

Link to ALM

Confluence

Dev

Ops

Simplified
SCM usage model

JIRA

1. Process Model

- Track Prd Baseline
- Manage **Release** Label
- Pre-emptive Promotion

2. Workspace Management

redhat

Ansible Tower

3. Build Management

4. Release Management

SCM

GitLab

(Release Label **)

Jenkins

Dev
Build server

Web Hooks

controlled Env

Apache Ant

sonarqube

Apache Ant

Build Server
Controlled Env

Promotion
Build

5. Environment Management
→ configuration management

Controlled Env

Dev env

SIT env

UAT env

Prod env

Ansible

Dev

Nexus

(Release Label **)

SIT

UAT

PRD

Silk Test

Silk Test

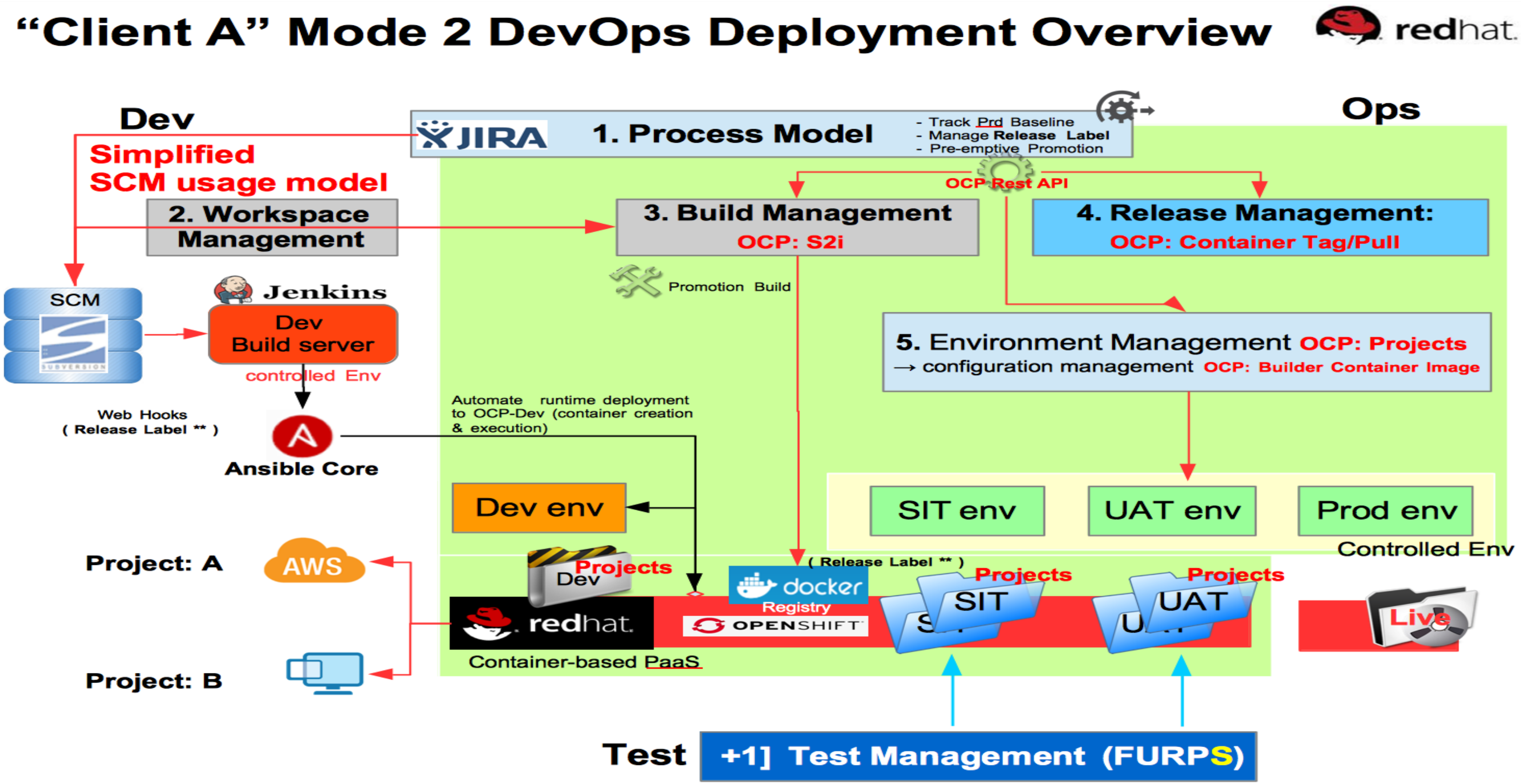
JMeter

Legend:
VM, Bare Metal

Test

+1] Test Management (FURPS)

A Live Mode 2 DevOps Deployment



"Client A" Mode 2 DevOps Deployment Overview

redhat.

Dev

Ops

Simplified
SCM usage model

JIRA

1. Process Model

- Track Prd Baseline
- Manage Release Label
- Pre-emptive Promotion

OCP Rest API

2. Workspace Management

3. Build Management
OCP: S2i

4. Release Management:
OCP: Container Tag/Pull

Promotion Build

5. Environment Management OCP: Projects
→ configuration management OCP: Builder Container Image

SCM

Jenkins

Dev
Build server

controlled Env

Web Hooks
(Release Label **)

Ansible Core

Automate runtime deployment to OCP-Dev (container creation & execution)

Dev env

SIT env

UAT env

Prod env

Controlled Env

(Release Label **)

Project: A

AWS

Project: B

Dev Projects

docker Registry

redhat.

OPENSHIFT

Container-based PaaS

SIT Projects

UAT Projects

Live

Test

+1] Test Management (FURPS)

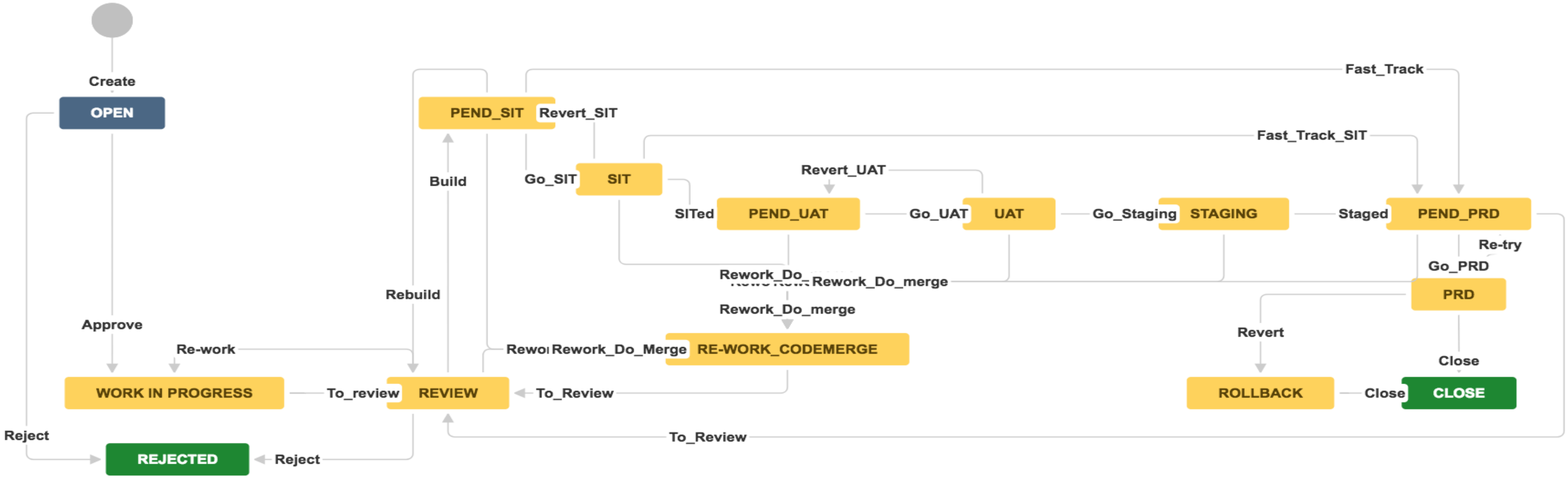
Red Hat OpenShift eases DevOps Practice

5+1 View Model – DevOps Approach

Defines the DevOps practices in the areas of 1) Process Model , 2) Workspace Management, 3) Build Management, 4) Release Management, 5) Environment Management (inclusive of configuration management) , and +1) Test Management.

	DevOps Focus Area	OpenShift - OCP
1	**Process Model** Orchestrate DevOps activities	- Integrate OCP to Process using OC client - Maintain Release Tag across SVN (src) and OCP image (bin)
2	**Workspace Management** SCM usage model+release tag management	
3	**Build Management**	- OCP Source-to-Image (S2i) + OCP controlled SOE (Build Env) - Tag application container image published to Registry
4	**Release Management**	- Leverage on Tag & Registry to identify release source → SVN - Using OC client to pull "Tag" image into Projects - Registry – source of Bin repository - Projects – Env (SIT, UAT, PRD)
5	**Environment Management** **- Configuration Management**	- Environment using Projects (Dev, SIT, UAT, PRD) - Project creation using Template - Provides signing of Builder container image → CM approval - Ensure SOE with signed Builder image
6	**Test Management**	- Enable parallel testing with "Build & Tear" test environment

DevOps Workflow - Activities



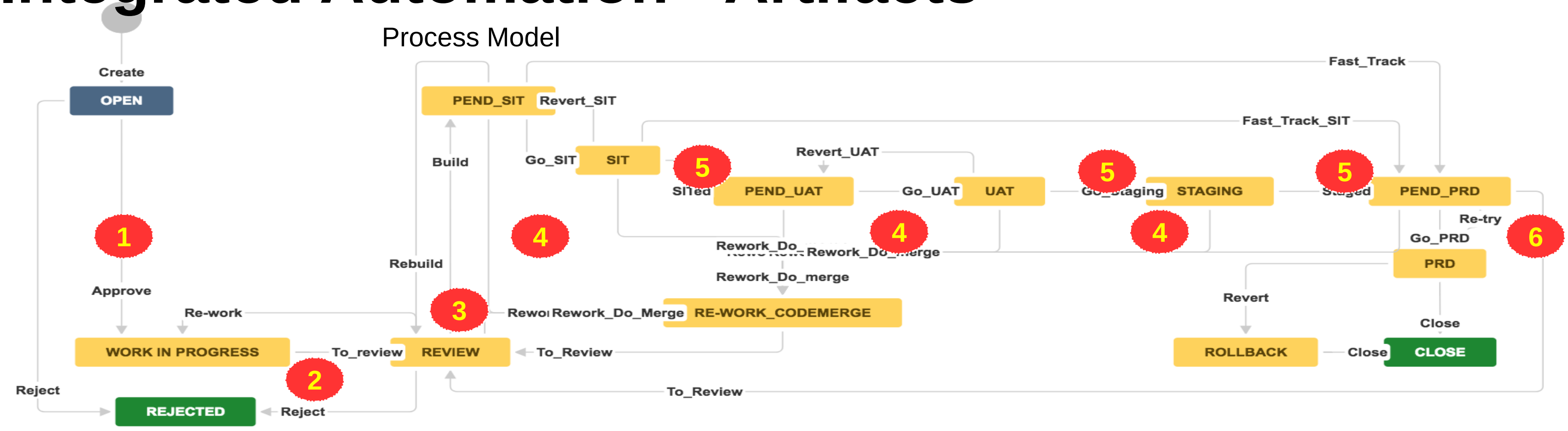
1. Simplify high ceremonial process

- across various key stakeholders

2. Integrate approval with automation

Integrated Automation - Artifacts

Process Model



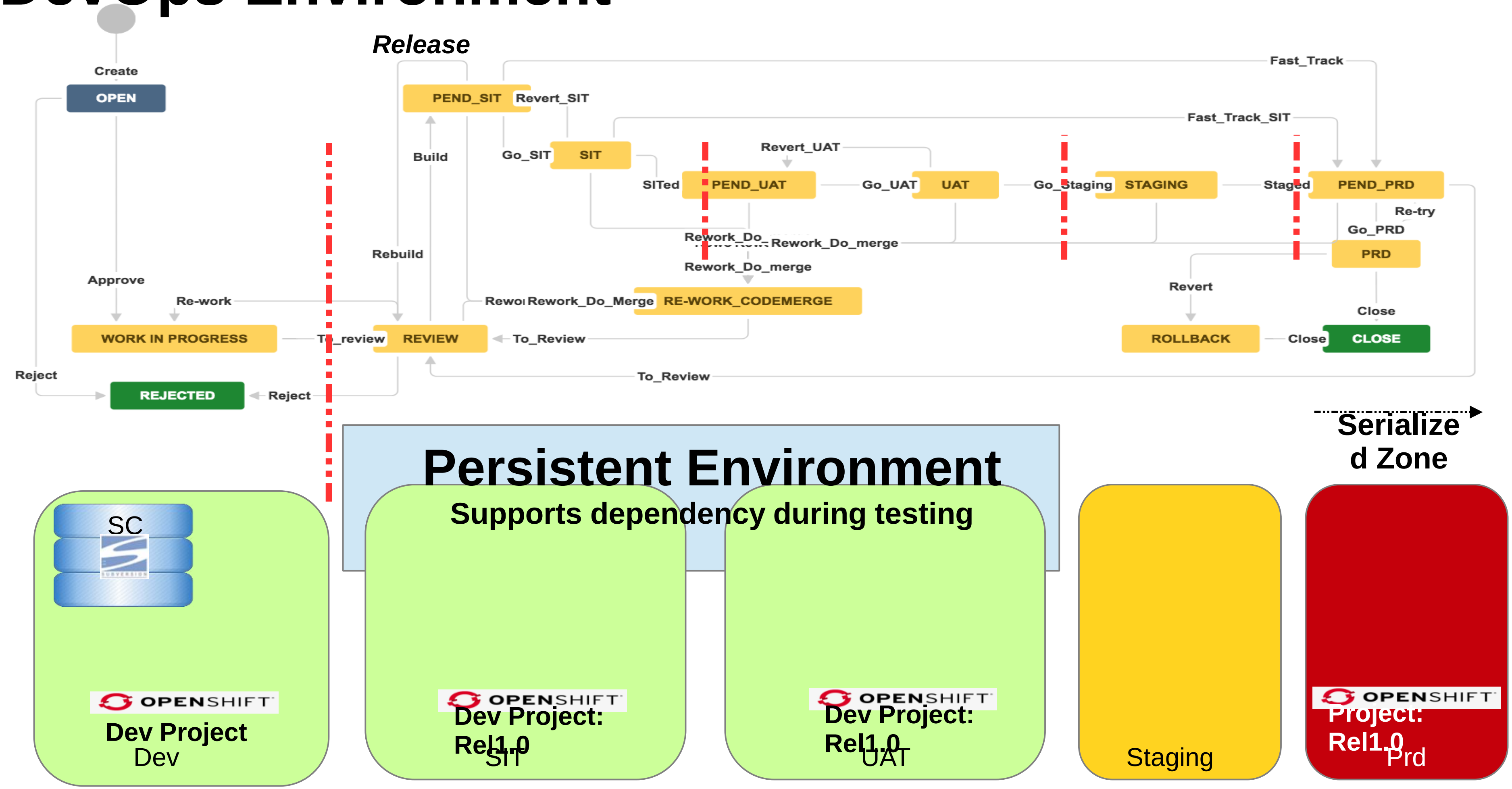
SCM
Usage
Model

OCP
Integration

OPENSHIFT

1. Auto create dev **Branch** using Prd_baseline
2. Auto create **Tag + lock** Tag upon dev completion
3. Auto checkout source for **Build + Publish** to Registry
4. Auto create Env, **deploy**, & start Application
5. Auto **create and teardown** of test env
6. Auto **deploy** & start Application

DevOps Environment



Release

Create

OPEN

Approve

Re-work

WORK IN PROGRESS

To_review

REVIEW

Reject

REJECTED

Reject

Rebuild

Build

PEND_SIT

Revert_SIT

Go_SIT

SIT

SITed

PEND_UAT

Revert_UAT

Go_UAT

UAT

Go_Staging

STAGING

Staged

PEND_PRD

Fast_Track

Fast_Track_SIT

Re-try

Go_PRD

PRD

Rework_Do

Rework_Do_merge

Rework_Do_merge

Rewo Rework_Do_Merge

RE-WORK_CODEMERGE

To_Review

To_Review

Revert

ROLLBACK

Close

Close

CLOSE

Serialize d Zone

Persistent Environment

Supports dependency during testing

SC

OPENSHIFT

Dev Project

Dev

OPENSHIFT

Dev Project: Rel1.0

SIT

OPENSHIFT

Dev Project: Rel1.0

UAT

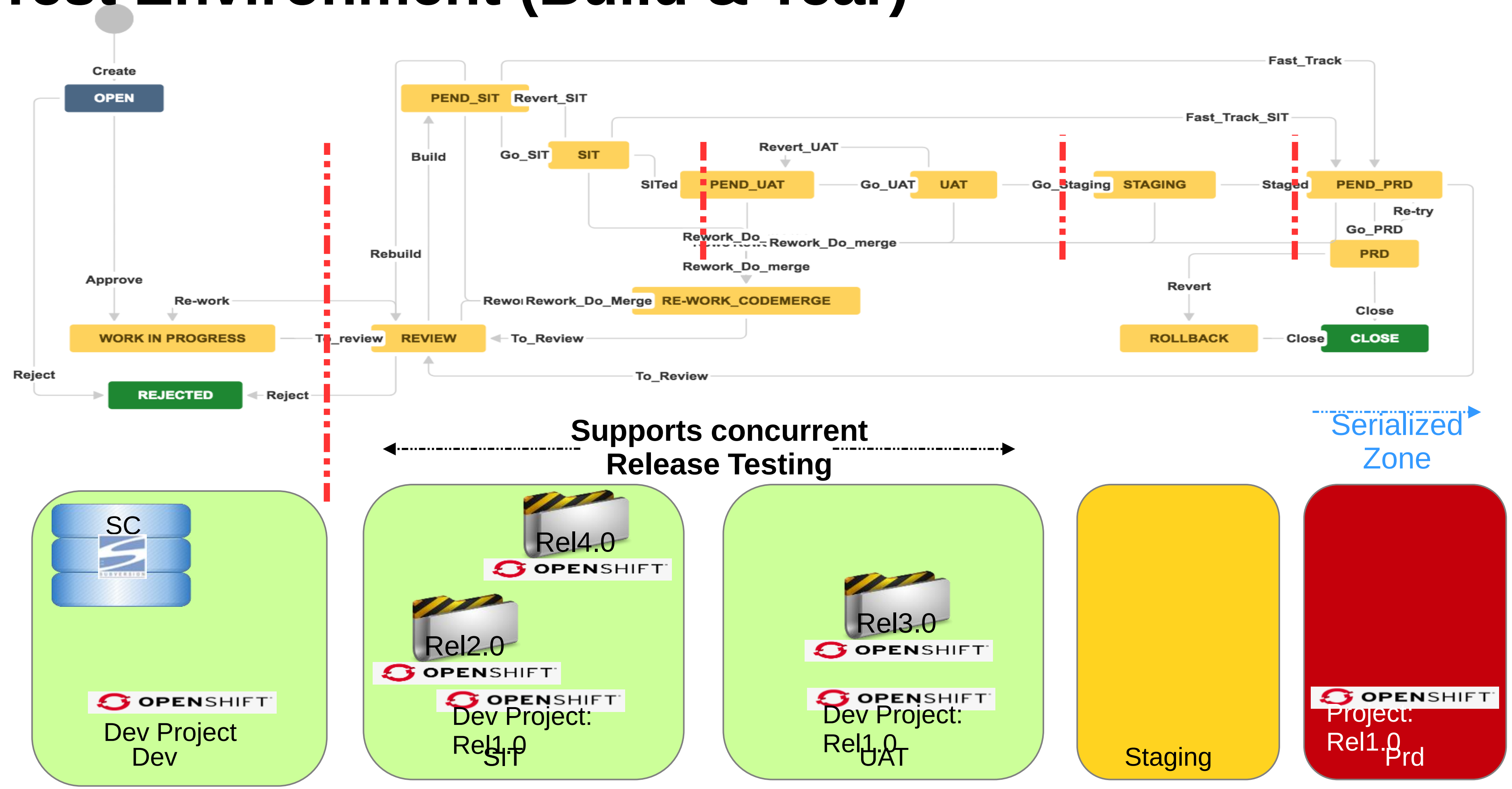
Staging

OPENSHIFT

Project: Rel1.0

Prd

Test Environment (Build & Tear)



Create
OPEN
PEND_SIT
Revert_SIT
Fast_Track
Fast_Track_SIT
Build
Go_SIT
SIT
Revert_UAT
SITed
PEND_UAT
Go_UAT
UAT
Go_Staging
STAGING
Staged
PEND_PRD
Re-try
Go_PRD
Rework_Do
Rework_Do_merge
PRD
Rebuild
Rework_Do_merge
Approve
Re-work
Revert
Rewoı Rework_Do_Merge
RE-WORK_CODEMERGE
Close
WORK IN PROGRESS
To_review
REVIEW
To_Review
ROLLBACK
Close
CLOSE
Reject
To_Review
REJECTED
Reject

Supports concurrent
Release Testing

Serialized
Zone

SC
OPENSHIFT
Dev Project
Dev

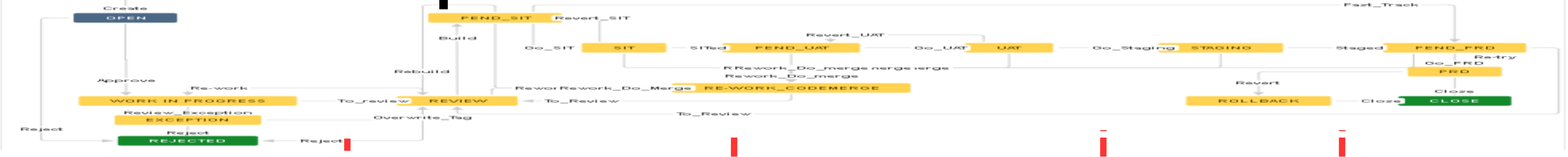
Rel4.0
OPENSHIFT
Rel2.0
OPENSHIFT
OPENSHIFT
Dev Project:
Rel1.0
SIT

Rel3.0
OPENSHIFT
OPENSHIFT
Dev Project:
Rel1.0
UAT

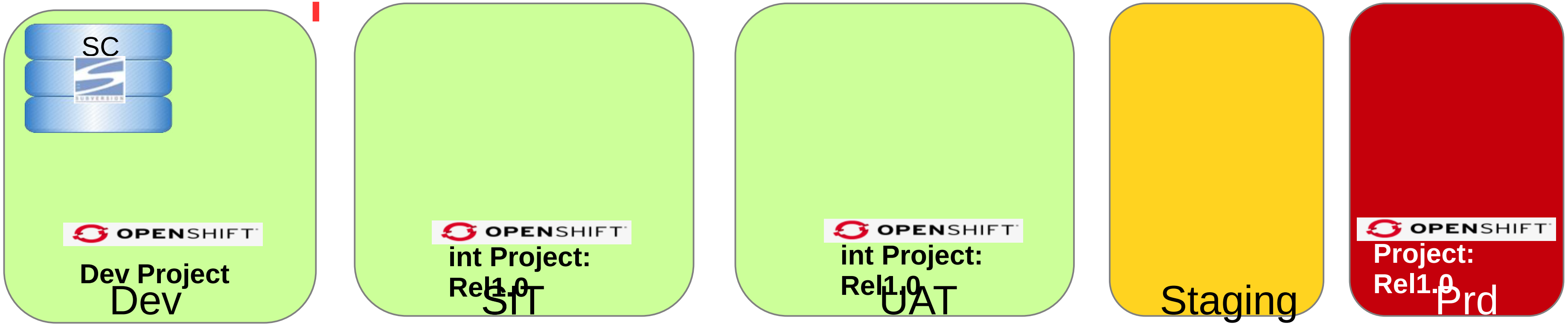
Staging

OPENSHIFT
Project:
Rel1.0
Prd

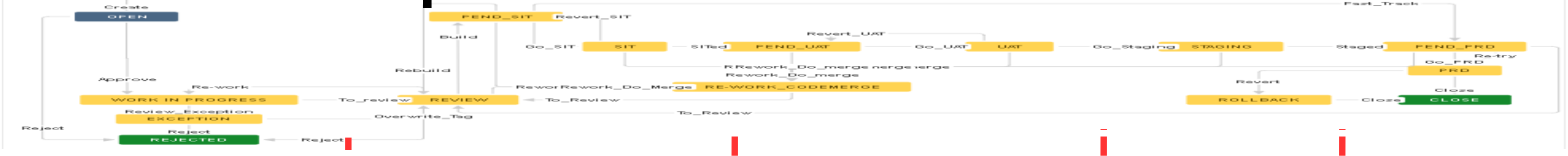
Interface : Step 1



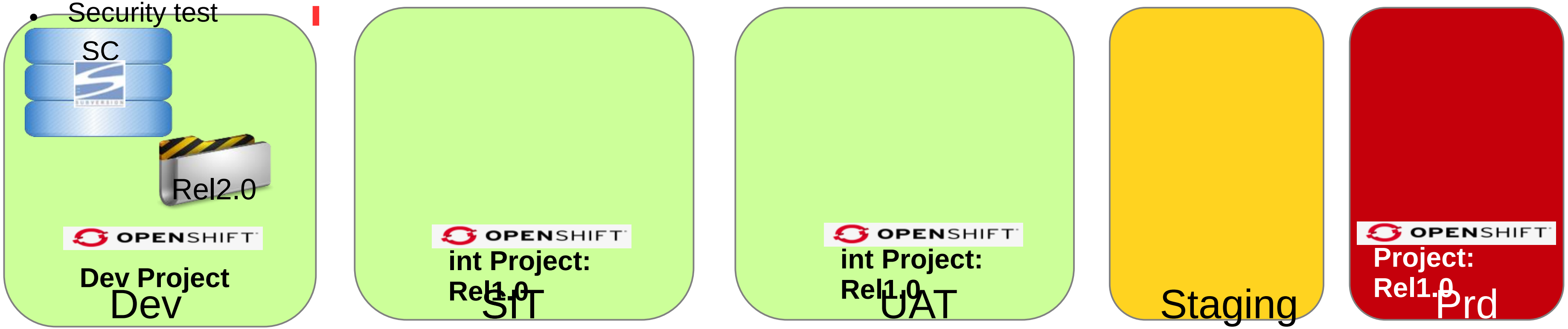
- Team raise Change Request (CR)



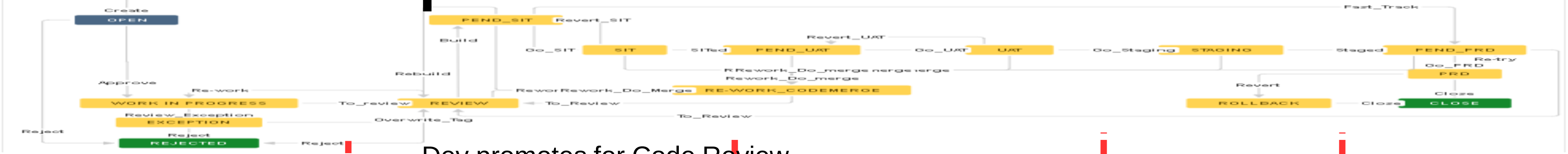
Interface : Step 2



- PM approves CR
  - auto create dev branch (using production code base)
- Develop …
  - CI with Jenkins
  - Unit testing
  - Code quality
  - Security test



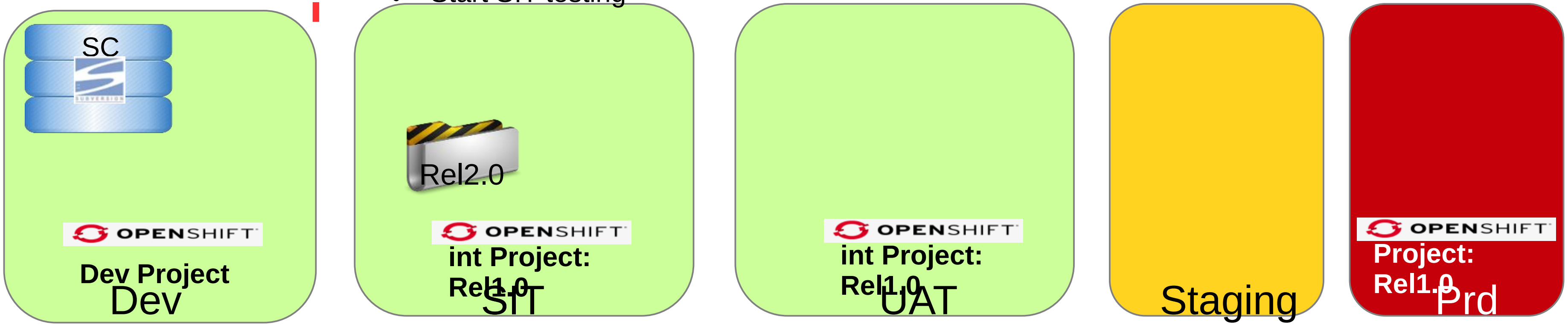
Interface : Step 3



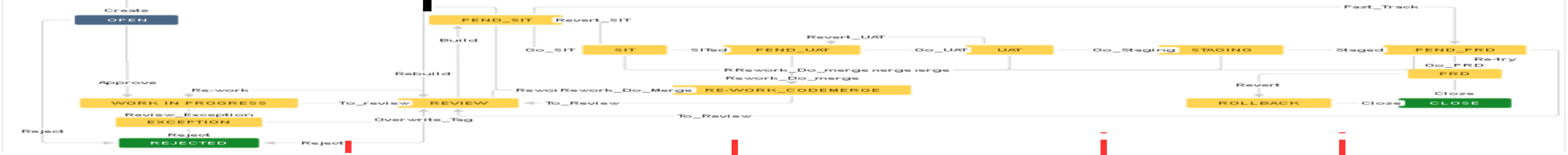
- Dev promotes for Code Review
  - Auto freeze source code → branch to tag

- PM approves
  - Auto build App container: move code from tag to OCP for source-to-image (s2i) → App container published to Registry
  - Auto create SIT env (OCP project)
  - Auto deploy app container in SIT project and start Pod

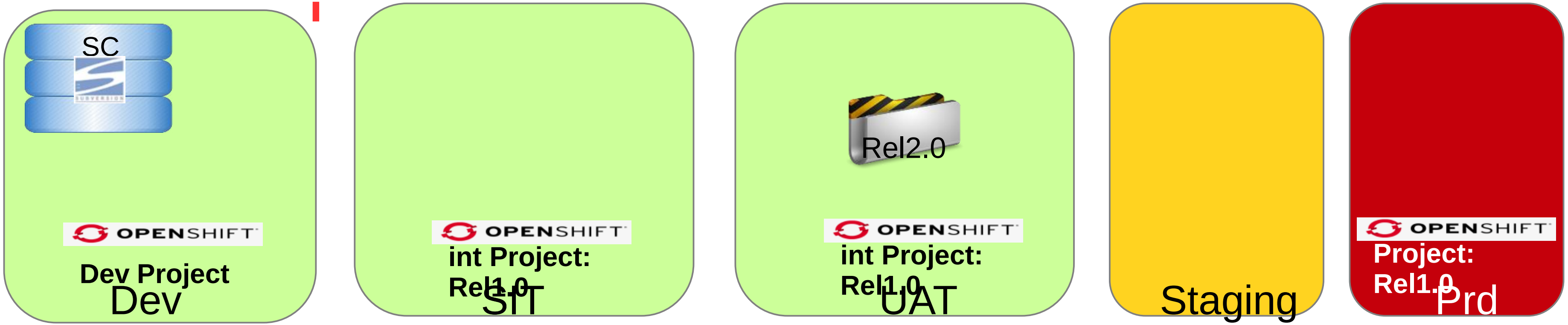
  - Start SIT testing



Interface : Step 4



- Promote to UAT
  - Auto remove SIT project
  - Auto create UAT env (OCP project)
  - Auto deploy app container in project and start Pod
  - Start UAT Testing



Dev Project

Dev

int Project: Rel1.0

SIT

int Project: Rel1.0

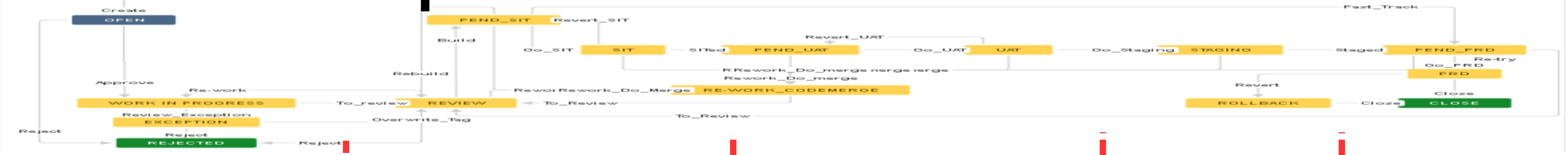
UAT

Staging

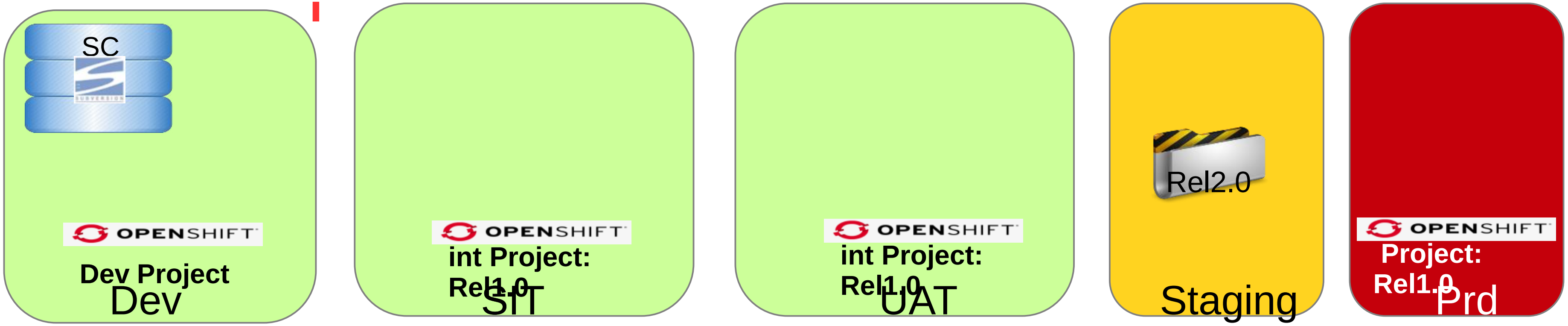
Project: Rel1.0

Prd

Interface : Step 5



- Promote to Staging
  - Auto remove UAT project
  - Auto create Stage env (OCP project)
  - Auto deploy app container in project and start Pod
  - Start Performance Testing



Dev Project

Dev

int Project: Rel1.0

SIT

int Project: Rel1.0

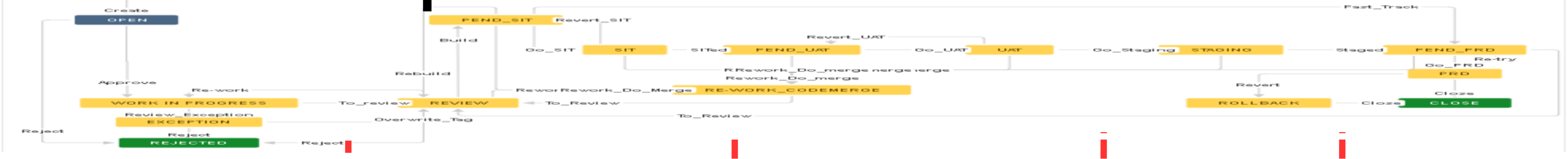
UAT

Staging

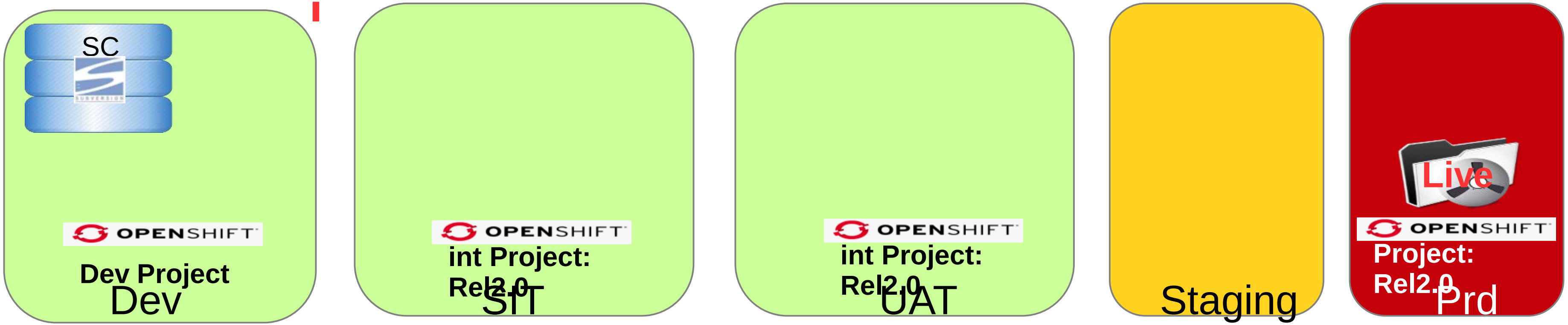
Project: Rel1.0

Prd

Interface : Step 6



- Promote to Production
- Auto remove Stage project
- Auto deploy app container in Production project and start Pod
- Close CR



SC

OPENSHIFT

Dev Project

Dev

OPENSHIFT

int Project: Rel2.0

SIT

OPENSHIFT

int Project: Rel2.0

UAT

Staging

Live

OPENSHIFT

Project: Rel2.0

Prd

Thank You