MARCH 30, 2022

brillioone.ai HPE

High Performance Engineering

ANALYST VIEWS

01
Gartner

Through 2022, 75% of DevOps initiatives will fail to meet expectations due to issues around organizational learning and change.

02
McKinsey

60% of the CIOs felt their organization's tech debt had risen perceptibly over the past three years, tech debt consumes as high as 20% of technology budget

03
Gartner

By 2024, 60% of organizations will have switched from multiple point solutions to value stream delivery platforms to streamline application delivery, up from 20% in 2021

WE SEE CHALLENGES ON THE FLOOR

Setting up **DevOps pipeline** is nightmare and we do not have the right skills

How do I ensure my developers spend **more time adding features** than fixing build and deployment failures

How do i drive a **better developer experience** in my team

It's tedious work to **collate tasks across engineering value stream** in one place

How can I ensure **time to market** and **deliver value** to my customers

How do we bring in **uniformity and consistency of practices** and **speed up delivery,** while staying relevant to the ever-changing application architecture?

How to reduce the **handoffs** between teams which lead to **siloed visibility**?

DEVELOPER EXPERIENCE HAS SEEN A PARADIGM SHIFT

- Developers start using **multiple tools** along with writing code

- Developers spend significant time on other aspects like **integrations, system dependencies, tool orientations** apart from focusing on coding

- Courtesy continuous poor interactions with tool sets everyday, the developer experience turns worse, leading to **sub standard product quality**

- Developers now own the process of **coding, deploying and running** their applications without much difficulty
- A **seamless flow of the code** through build, commit and deploy stages is enabled

- Realizing that an improved Developer Experience can lead to an **enhanced Customer Experience** too

- Popular belief **'Developers know it all'**
- Minimal Focus on Developer Experience

- With increasing tools and processes, **multiple challenges crop up** in the quest of moving the code to production stage

- As a result, the overall **time to market** for the solutions increases drastically

- Realizing that developers should **solely focus on writing code**
- Toolkit comprising of **frameworks and reusable libraries** should be used to analyze, refine, optimize, and migrate existing pipeline

- This is done through the right level of **collaboration** and **unification** between the dev, ops, security, and other discreetly operating teams

HENCE THE NEED FOR A HIGH-PERFORMANCE ENGINEERING ENABLER

Improved Developer Experience + **Drive A cultural Shift** = **Faster to Market** + **Drive organizational change**

Optimizing the current workflows

A mind-set change

Quicker release cycles

Product Ways of Working

Adopting DevOps practices at scale

A united and a shared identity between development ,testing and operations

A blend of plethora of tools and plugins that reduces Time to Market
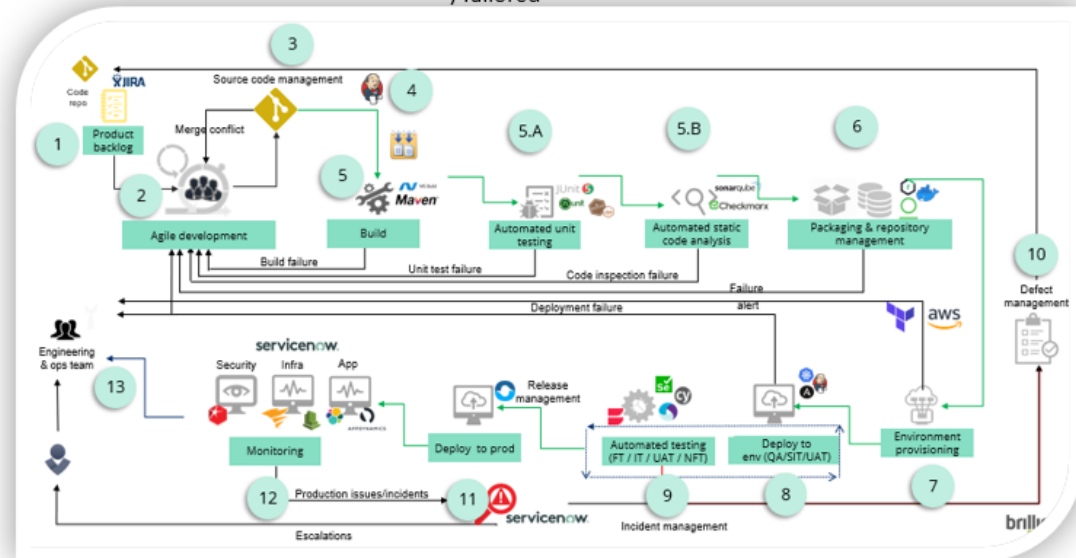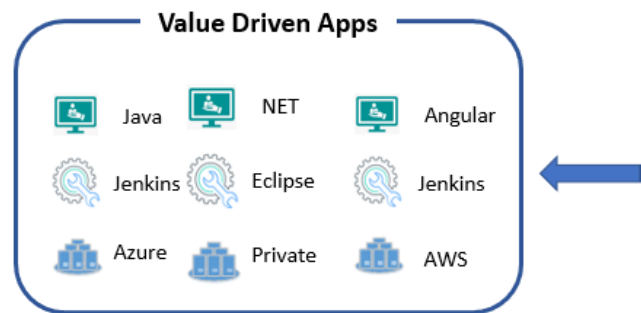
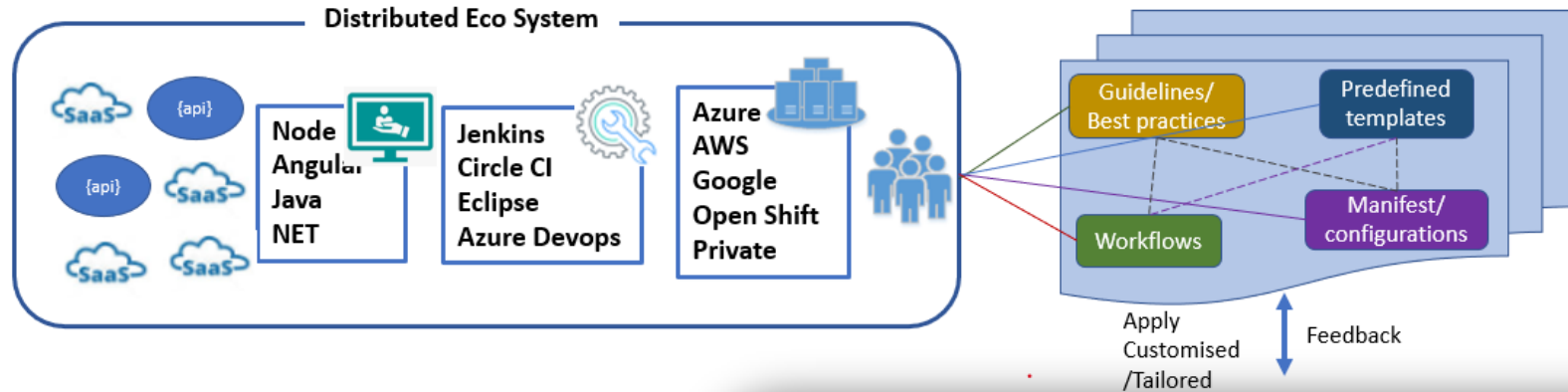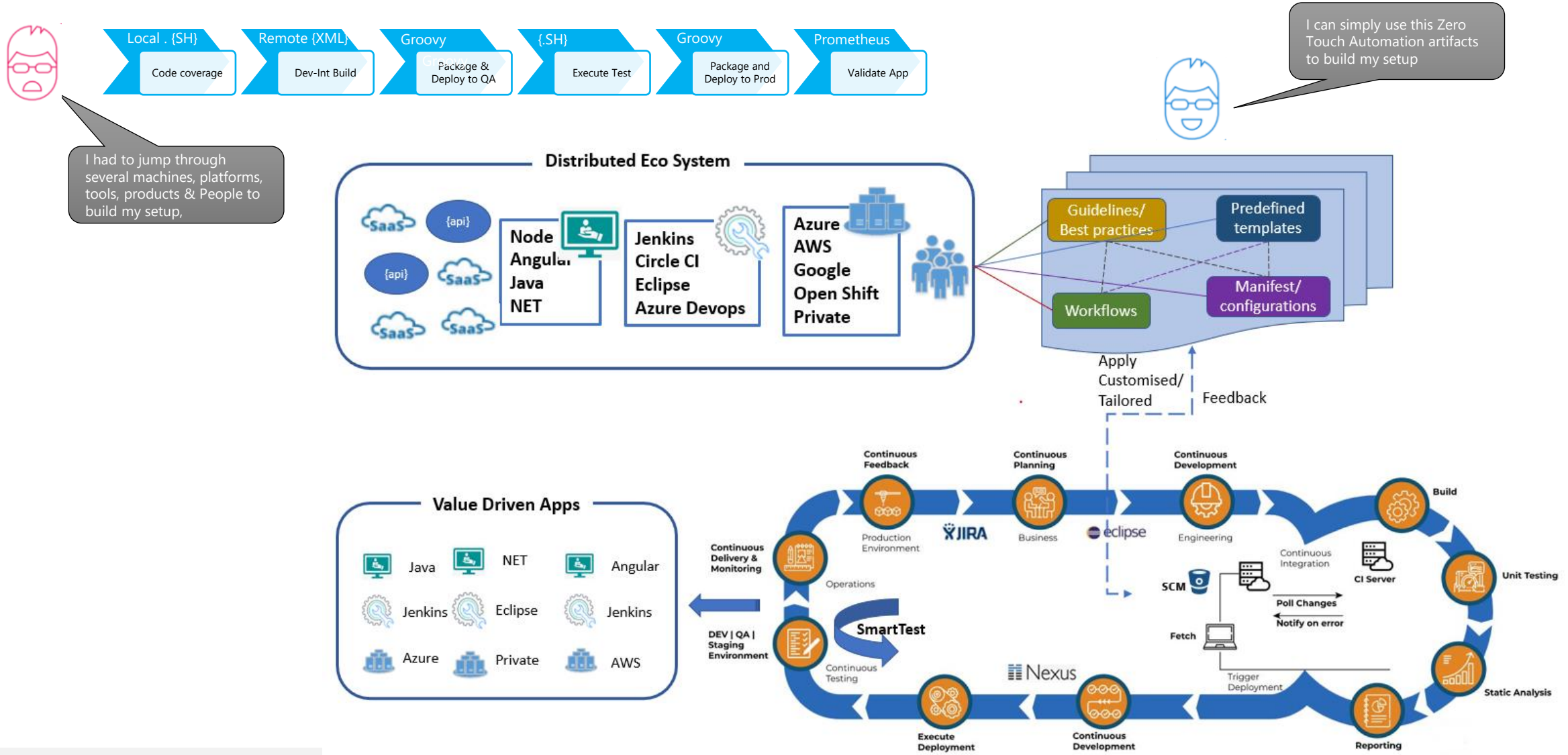Delivering sustainable business value

TO ENHANCE DEVELOPER'S EXPERIENCE

I can simply use this Zero Touch Automation artifacts to build my setup

Local . {SH} — Code coverage
Remote {XML} — Dev-Int Build
Groovy — Package & Deploy to QA
{.SH} — Execute Test
Groovy — Package and Deploy to Prod
Prometheus — Validate App

I had to jump through several machines, platforms, tools, products & People to build my setup,

Distributed Eco System

SaaS {api} {api} SaaS SaaS SaaS

Node
Angular
Java
NET

Jenkins
Circle CI
Eclipse
Azure Devops

Azure
AWS
Google
Open Shift
Private

Guidelines/ Best practices
Predefined templates
Workflows
Manifest/ configurations

Apply Customised /Tailored

Feedback

Value Driven Apps

Java | NET | Angular
Jenkins | Eclipse | Jenkins
Azure | Private | AWS

DEVELOPER'S JOURNEY SIMPLIFIED...

BY AUTOMATING THE VALUE STREAM

VALUE STREAM DELIVERY PLATFORM
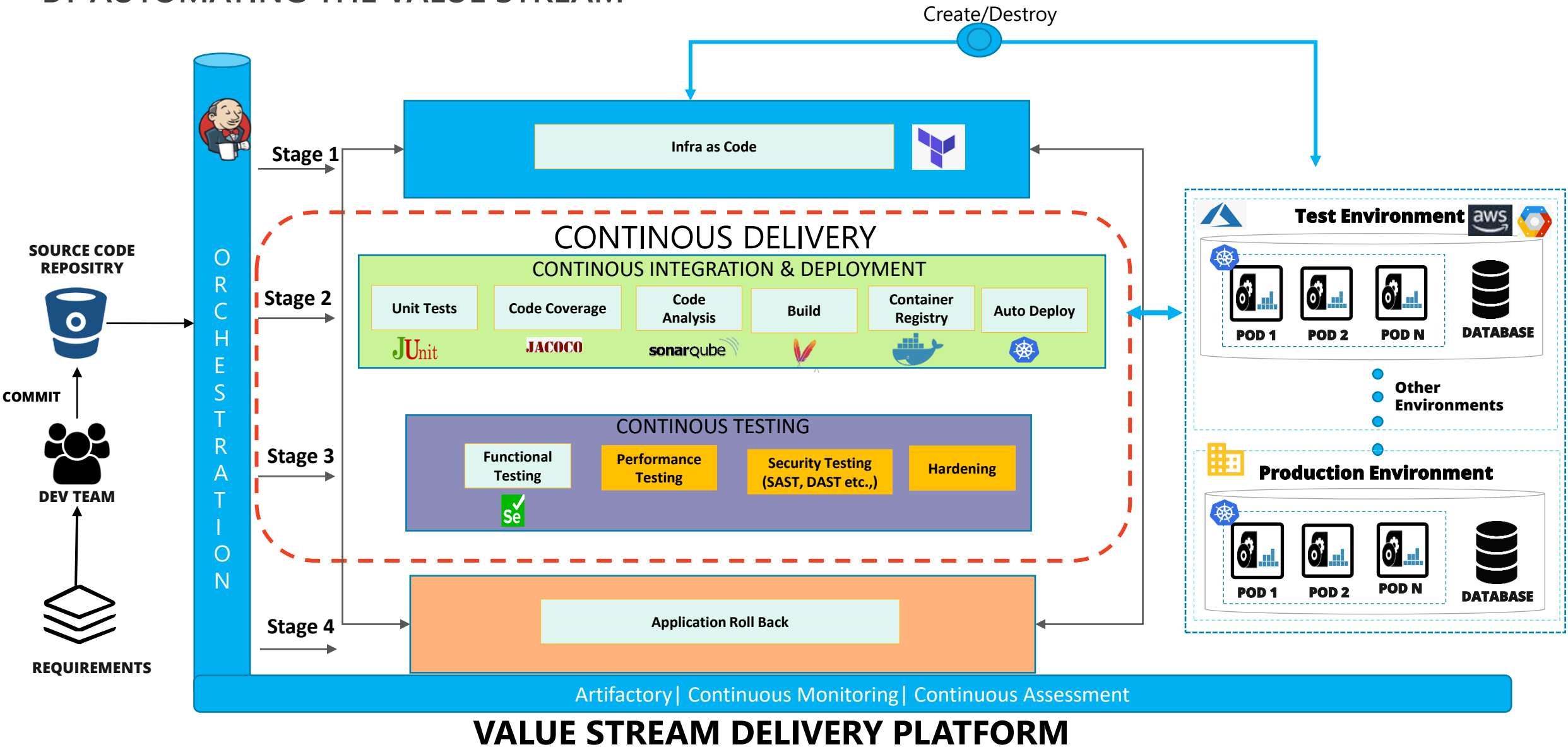
INTEGRATING TOOLS ACROSS THE APPLICATION LIFECYCLE

CONTINUOUS INTEGRATION

CODE REVIEW & ANALYSIS / SAST

TEST AUTOMATION & COVERAGE / DAST

VERSION CONTROL

ENVIRONMENT PROVISIONING/ CONTINUOUS DEPLOYMENT

REQUIREMENTS MANAGEMENT

Seamless orchestration

MONITORING

Code Quality Assurance

360-degree view of Functionality Coverage

OUR EXPERTISE IN DEVOPS TOOLS

Enhanced Code and Environment Management

Effective asset Management based on Demand

Improved Decision making through Holistic View of all requirements

Proactive approach to improve business continuity & efficiency

You coded it, You built it
You Built, You test it
You tested it , You deploy it
You deployed it , You own it

You Built it,
You own it

No fear release

No fear of speed
No fear of breakdown
No fear of being out of control
No fear of being non-complaint

BUSINESS USE CASES

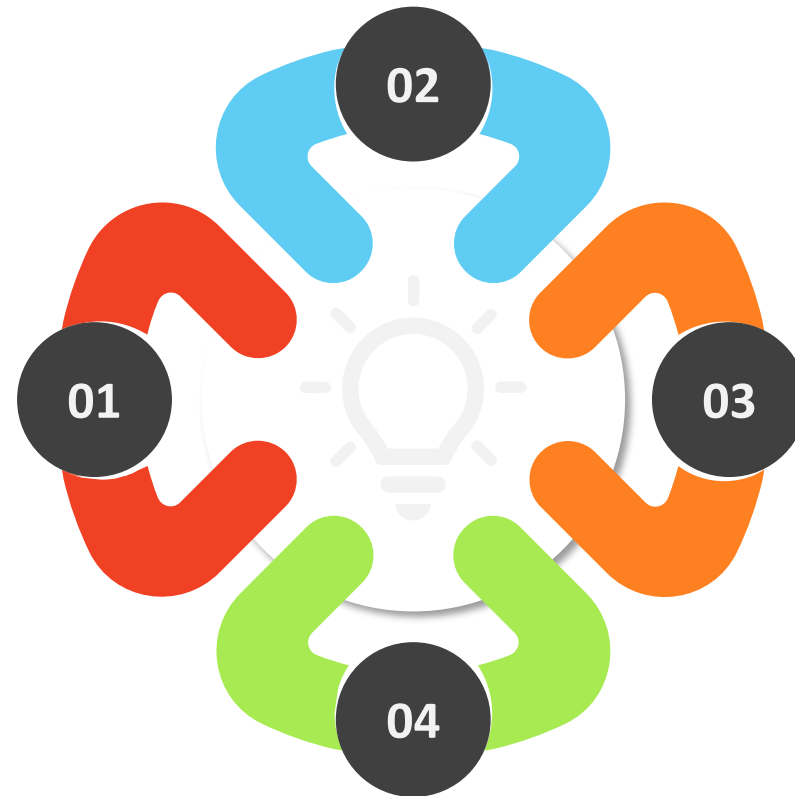**Traditional Application Development Lifecycle Management**

Tool Stack Used: Jenkins, Azure, TF

Microservices Application Development Lifecycle Management

Tool Stack Used: Jenkins, Azure, TF , K8s

Other Extensions /Integrations

Tool Stack Used: ServiceNow, JIRA ,Netsuite

MS Azure native/hybrid services

Tool Stack Used: JIRA, AZ DevOps , Service Now

Customer Benefits

Enhancing DevOps charter with quick wins

Reusable assets for cross portfolio

Thought leadership to transformation by leveraging existing investment

End-to-End DevSecOps framework with industry standard best practices on both cloud & non cloud environments

Pre-configured & scalable to customised workflow/requirements

BUSINESS IMPACT DELIVERED

Faster time to market

Improve team productivity

Continuous Service Delivery

Reduction in build failures

Increased collaboration through single point of truth

Improvement in cycle time
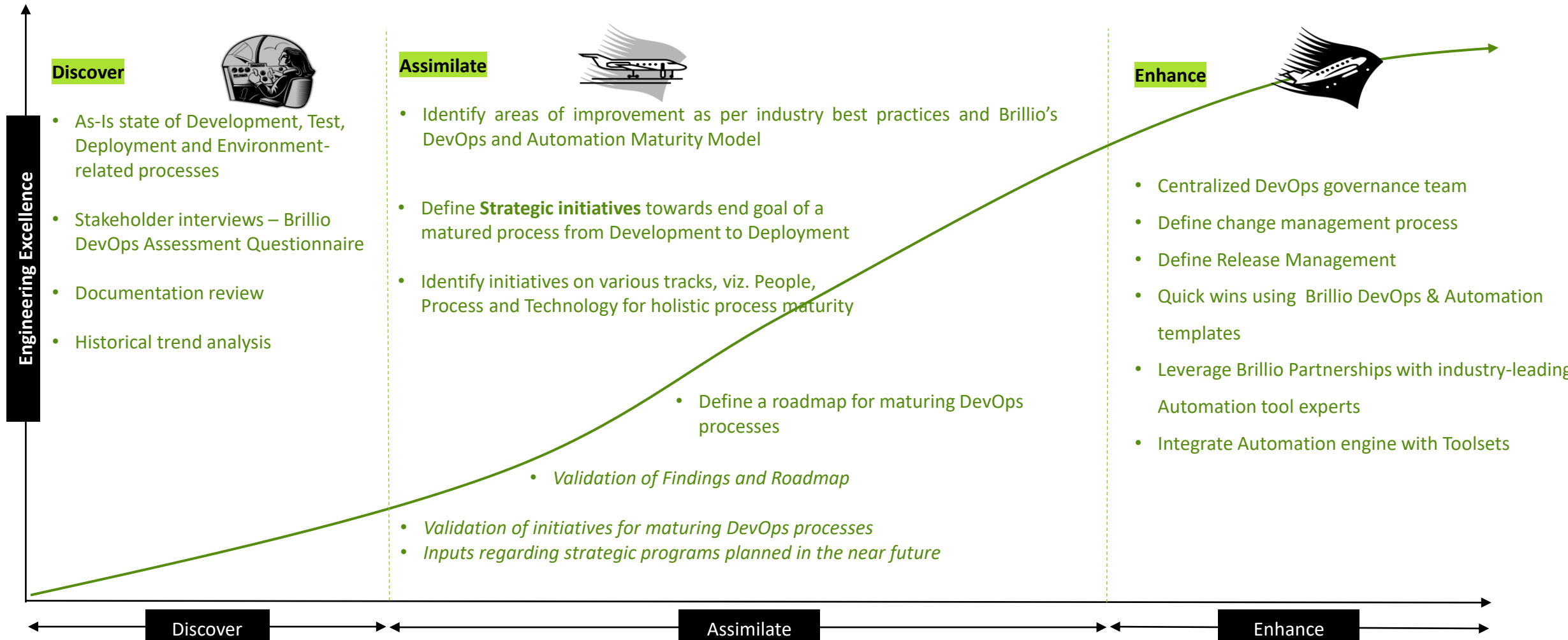
Faster realization of investment

Improved transparency across your value stream

Reduced Time to Recovery

IMPLEMENTATION APPROACH

Discover

- As-Is state of Development, Test, Deployment and Environment-related processes

- Stakeholder interviews – Brillio DevOps Assessment Questionnaire

- Documentation review

- Historical trend analysis

Assimilate

- Identify areas of improvement as per industry best practices and Brillio's DevOps and Automation Maturity Model

- Define **Strategic initiatives** towards end goal of a matured process from Development to Deployment

- Identify initiatives on various tracks, viz. People, Process and Technology for holistic process maturity

- Define a roadmap for maturing DevOps processes

- *Validation of Findings and Roadmap*

- *Validation of initiatives for maturing DevOps processes*
  Inputs regarding strategic programs planned in the near future

Enhance

- Centralized DevOps governance team

- Define change management process

- Define Release Management

- Quick wins using Brillio DevOps & Automation templates

- Leverage Brillio Partnerships with industry-leading Automation tool experts

- Integrate Automation engine with Toolsets

Engineering Excellence

Discover | Assimilate | Enhance

brillio one.ai

Thank You!

LET US CRAFT SOMETHING

AMAZING. **TOGETHER!**