



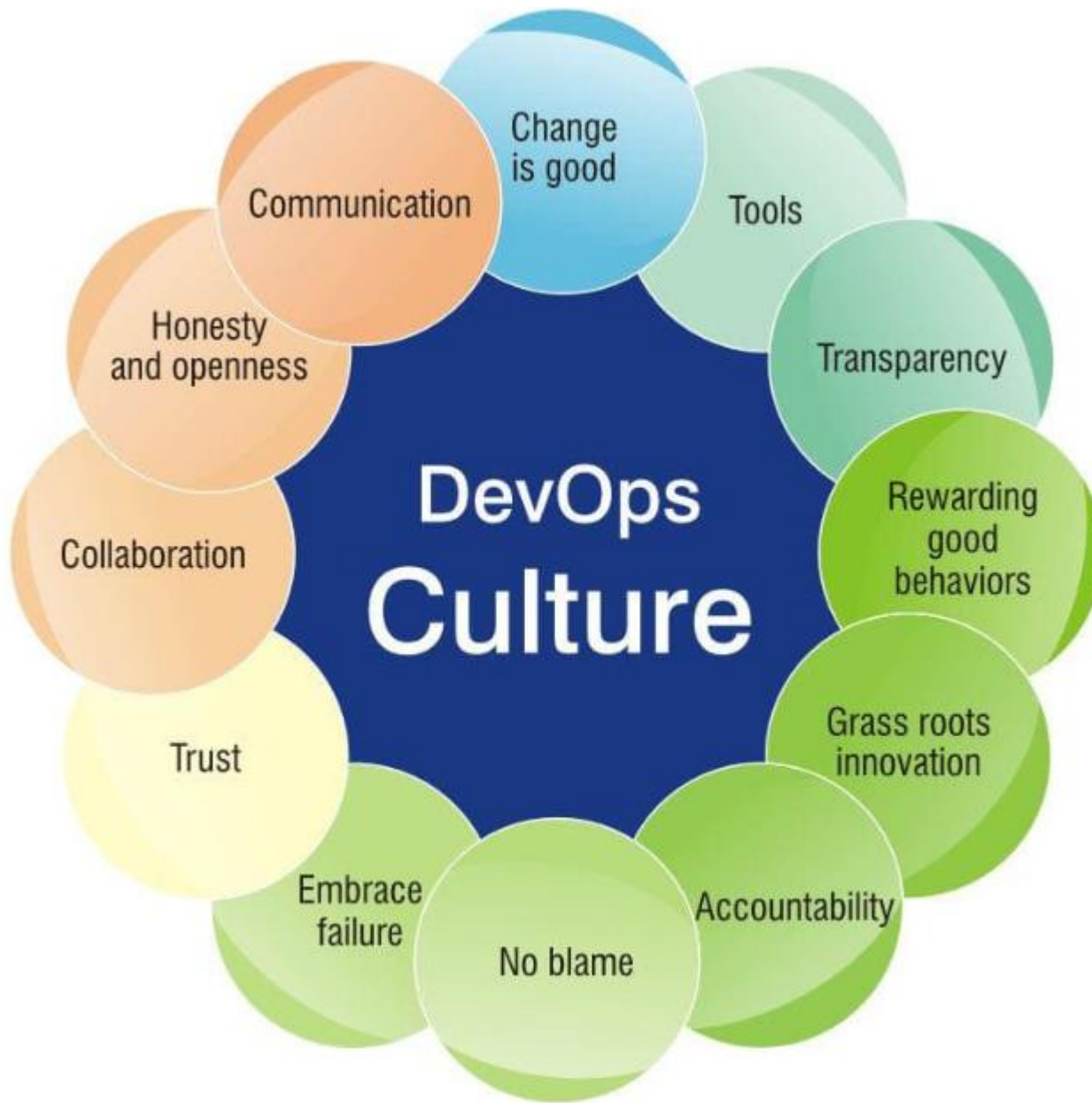
DevOps Principles

DevOps Training



What Is DevOps?

DevOps is a set of processes and practices to bridge the gap between Agile Software Development and Operations



Why DevOps?

- **Shorter Development Cycles**
- **Faster Innovation**
- **Reduce Implementation Failure**
- **Faster Recovery**
- **Better Communication and Cooperation**
- **Reduce Costs**

Top 3 Delivery Challenges

- Release management— Better understanding of risks, dependencies, compliance issues
- Release/Deployment coordination— Better tracking of discrete activities, faster escalation of issues, documented process control and granular reporting
- Release/Deployment Automation— Usually have existing automation but want to flexibly manage and drive this automation that can be invoked by non-operations resources in specific non-production environments

Why Devops?

- Agile Development— Addresses the gap between customer requirements and dev + testing teams— Cross-functional teams to design, develop, and test features/stories prioritized by the PO (Customer)— Focuses more on functional and non-functional readiness
- DevOps— Addresses the gap between dev + testing and Ops— Automated release management— Focuses on functional and non-functional plus operational and business readiness— Intensifies reusability and automation

DEV And OPS

- Developers work with Ops to understand the impact of code changes
- Developers now work more closely with production-equivalent systems
- Developers focuses on metrics required by Ops team like PSR
- Ops now have more clarity on infrastructure needs
- More automation on deployment
- Closely monitors the Dev – Test – Prod pipeline for each deployment with immediate feedback
- Better collaboration and communication

What are the roles of Devs and Ops?

- Dev
 - Create change
 - Add or modify features
- Ops
 - Create stability
 - Create or enhance services

Dev Ops



Tools



Automation



Access rights



Teams

Dev Ops



Tools



Automation



Access rights



Teams



Communication



Understanding



Integration

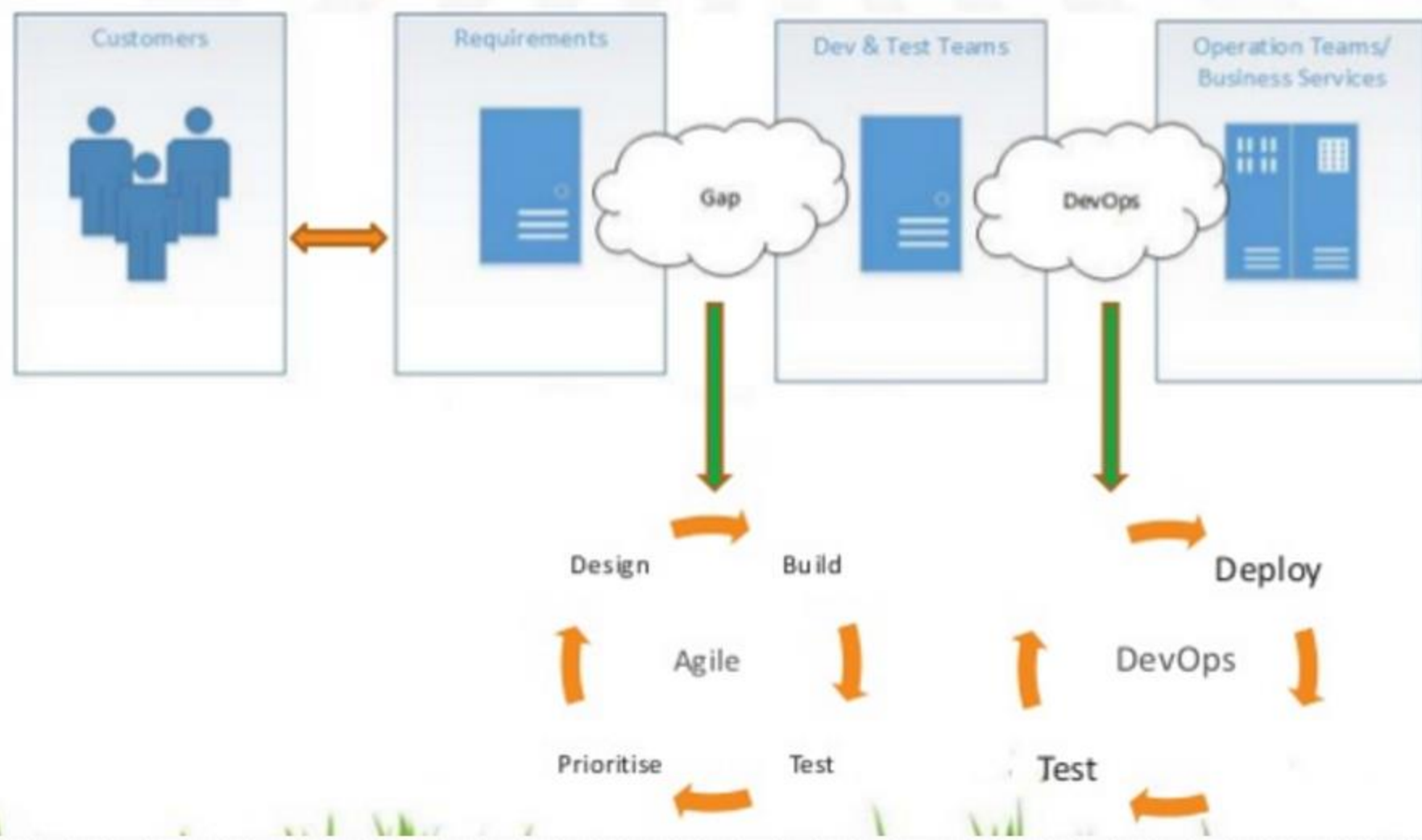


Relationships

The Problem

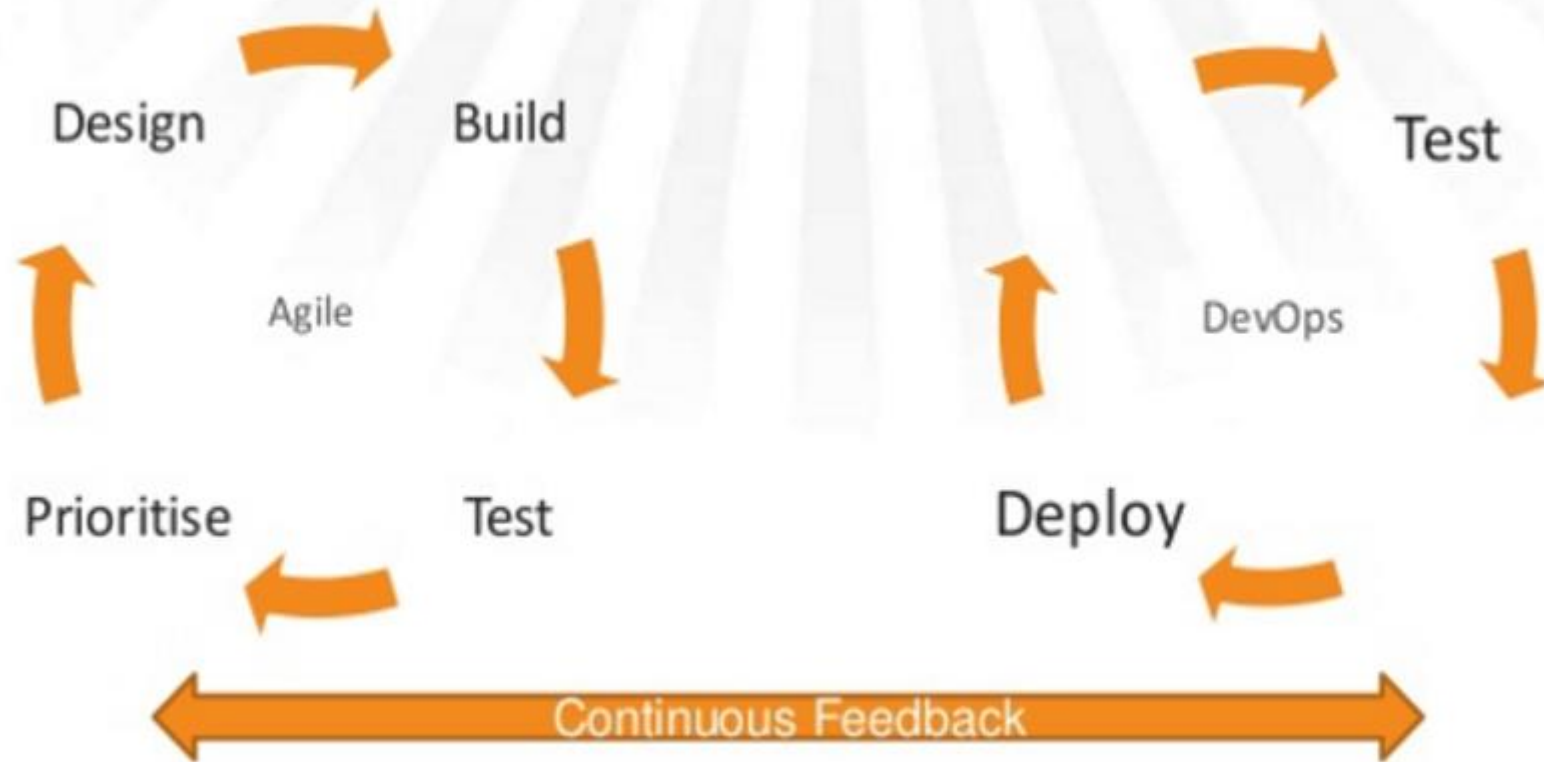
- Disconnect between the groups
- Results in conflict and inefficiencies
- Devs don't deploy consistent software
- Ops are motivated to resist change
- Development process is Agile
- Operations process is Static

Agile and DevOps



Agile + DevOps

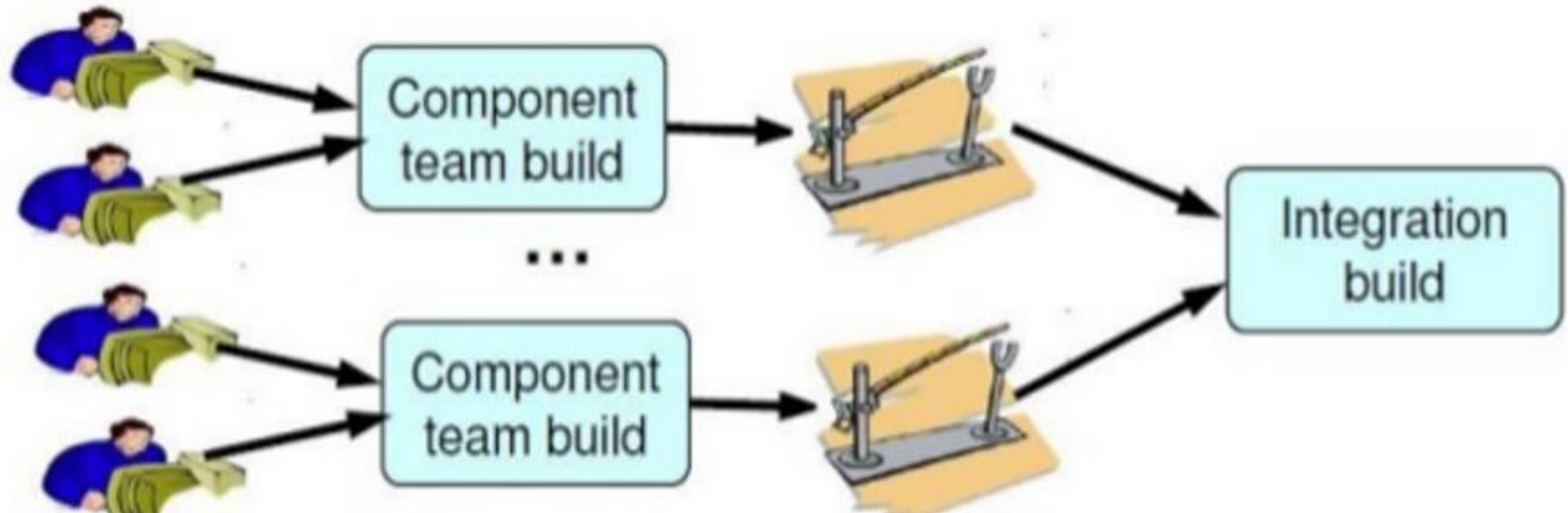
Continuous Integration extended as Continuous Delivery



Faster Delivery reduces risk

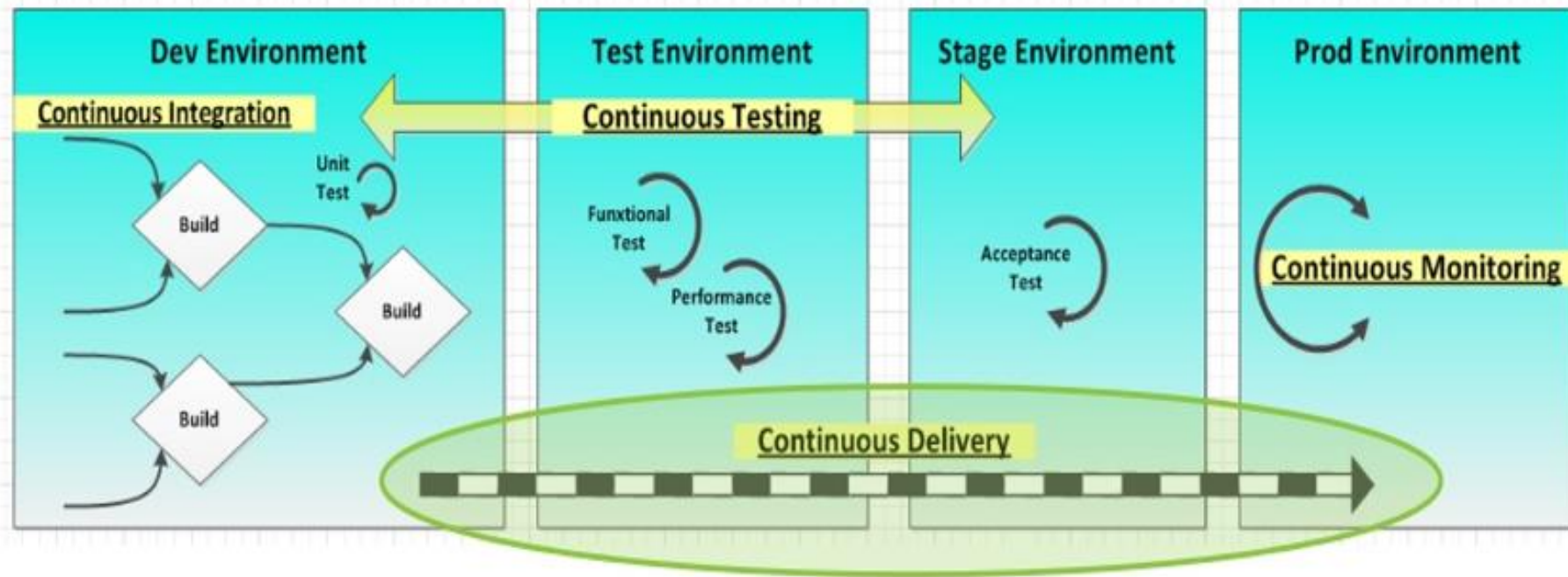
Continuous Integration

- Integrate the code changes by each developer so that the main branch remains up-to-date

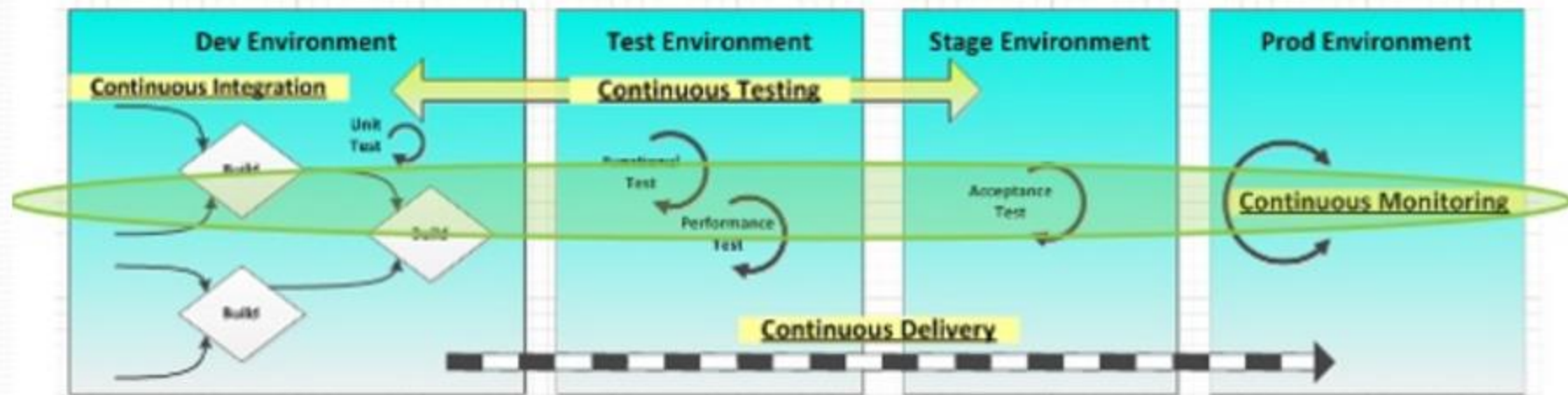
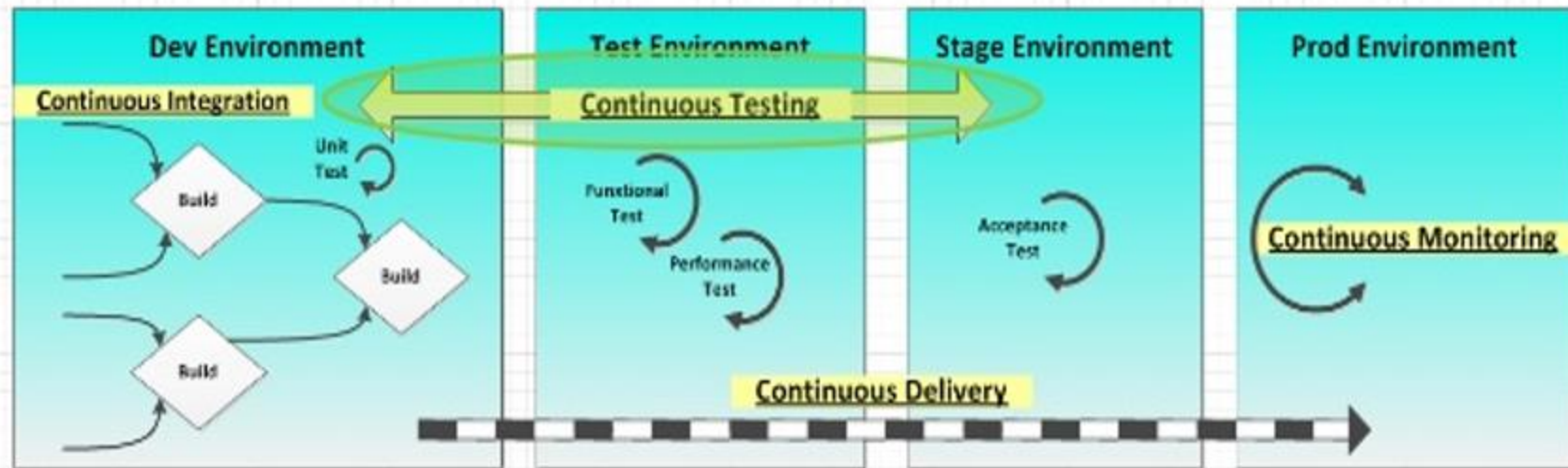


Continuous Delivery

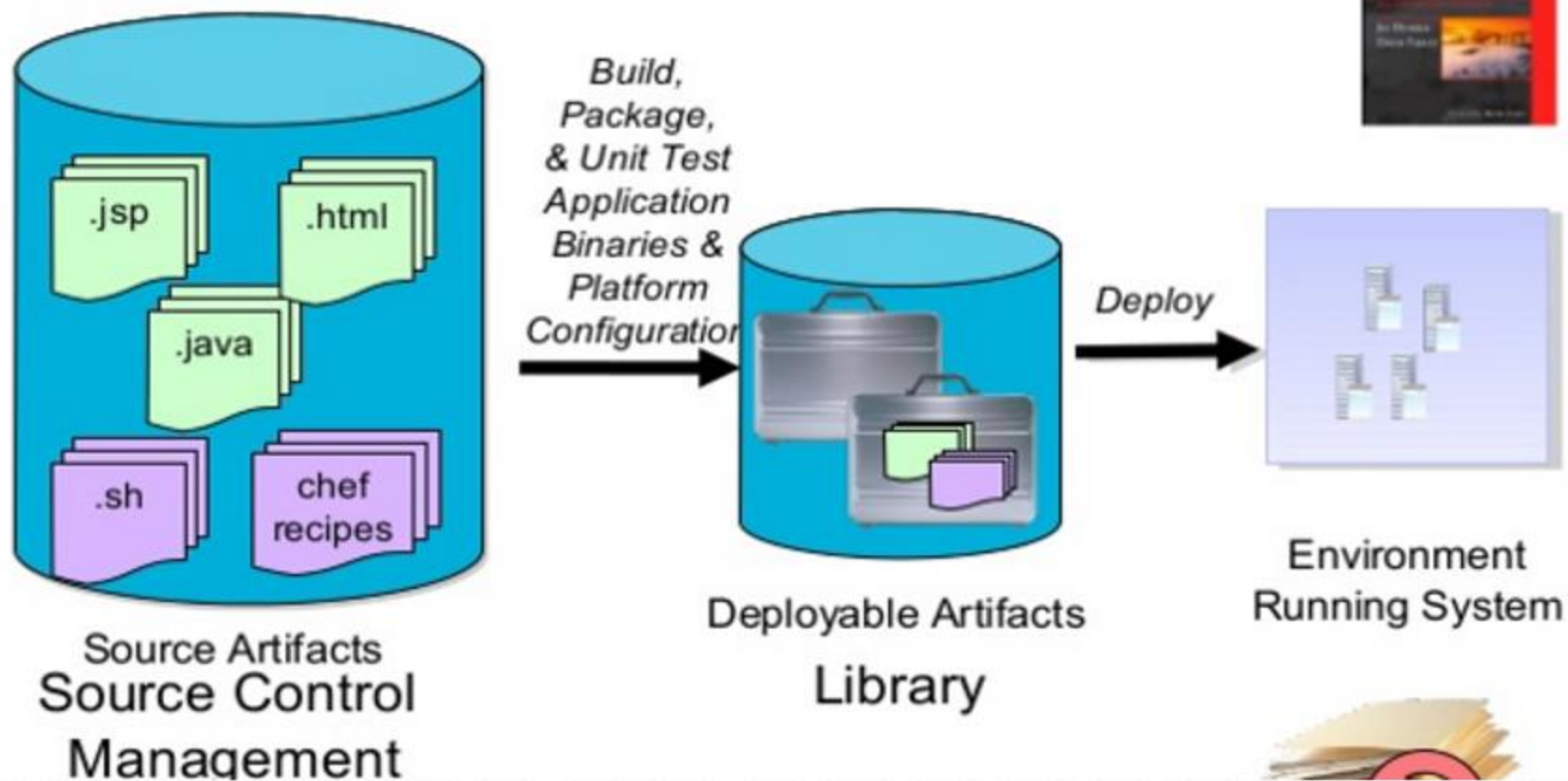
- Taking each CI build and run it through deployment procedures on production or production-equivalent environments.



Continuous Testing and continuous Monitoring



Delivery Pipeline



Common Elements of the Software Supply Chain

 **sonarqube**

 **Nexus**

maven

 **Jenkins**

 **JIRA**

 **git**

 **puppet**
labs

 **RUNDECK**


CHEF

 **docker**


SUBVERSION

 **Apache**
Tomcat

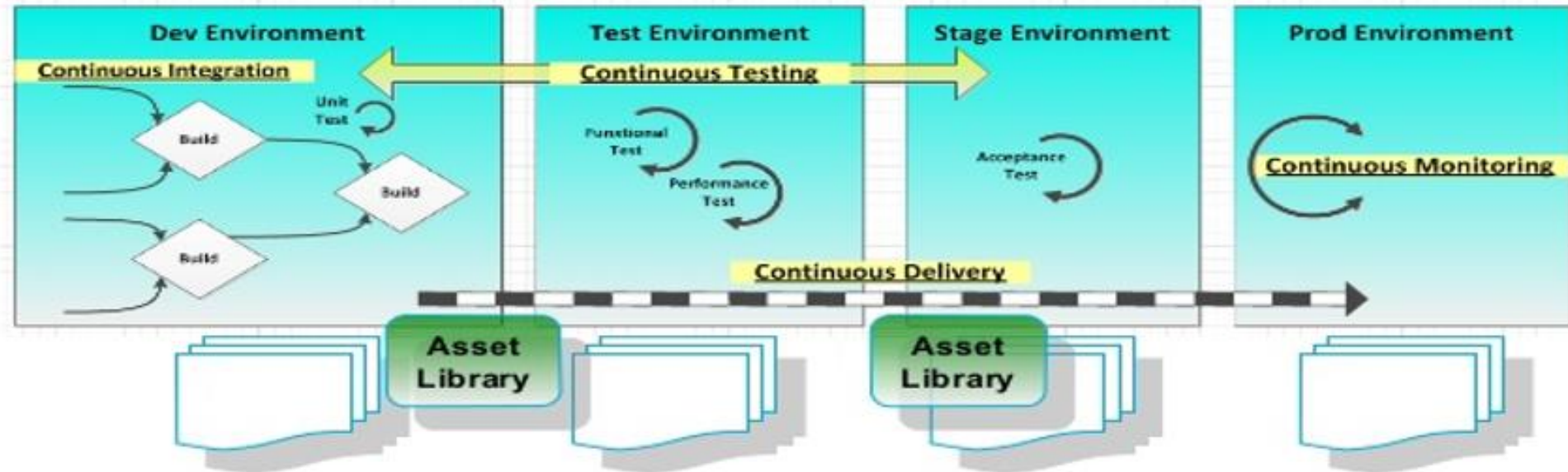
DevOps Practices

- Version Control For All
- Automated Testing
- Proactive Monitoring and Metrics
- Kanban/Scrum
- Visible Ops/Change Management
- Configuration Management
- Incident Command System
- Continuous Integration/Deployment/Delivery
- “Put Developers On Call”
- Virtualization/Cloud/Containers
- Toolchain Approach
- Transparent Uptime/Incident Retrospectives

Key Technical Capabilities of DevOps

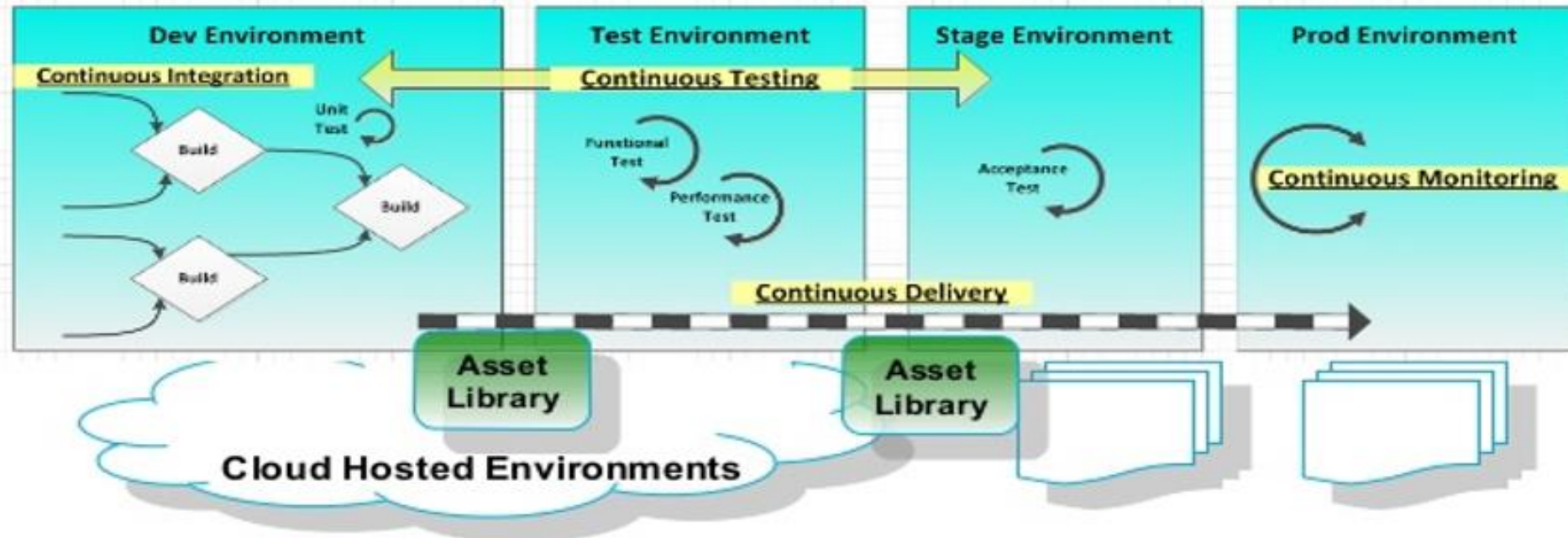
- Version Control Systems – Git, GitHub, Artifactory, Nexus
- Automation / Scripting – Python, Ruby, Bash, Linux Administrations etc
- Build Tools – Maven, Ant
- CI Tools – Jenkins, Team city, Bamboo
- App/Web Servers – Deployment Automation
- Infrastructure Provisioning and Management – Puppet, Chef, Ansible, Docker
- Cloud - AWS
- Daily Ops

The Variants of Continuous Delivery



2. Deploy to Dev, QA and Prod hosted on Physical Servers (no Cloud)

The Variants of Continuous Delivery



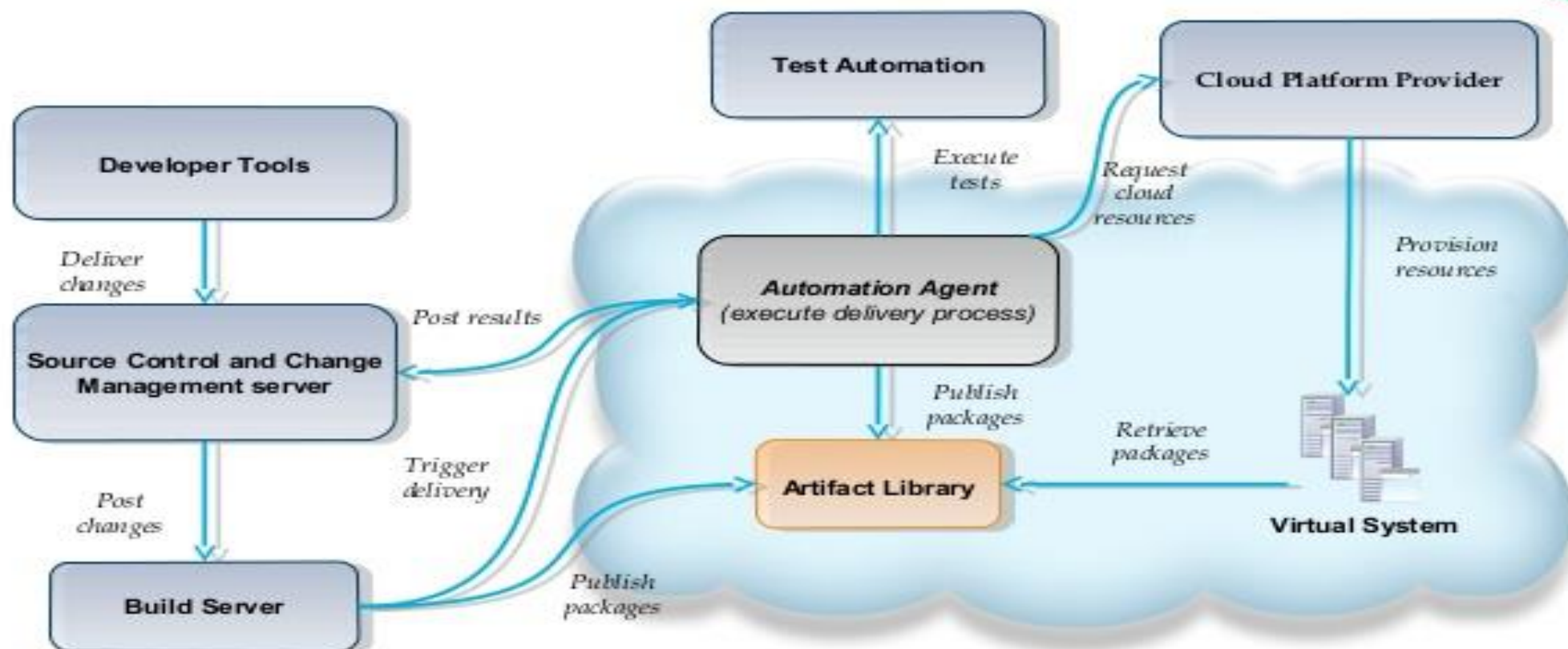
3. Deploy to Dev and QA hosted on Private or Public Cloud. Prod is on-prem physical servers
(very common)

Continuous Delivery Adoption Maturity

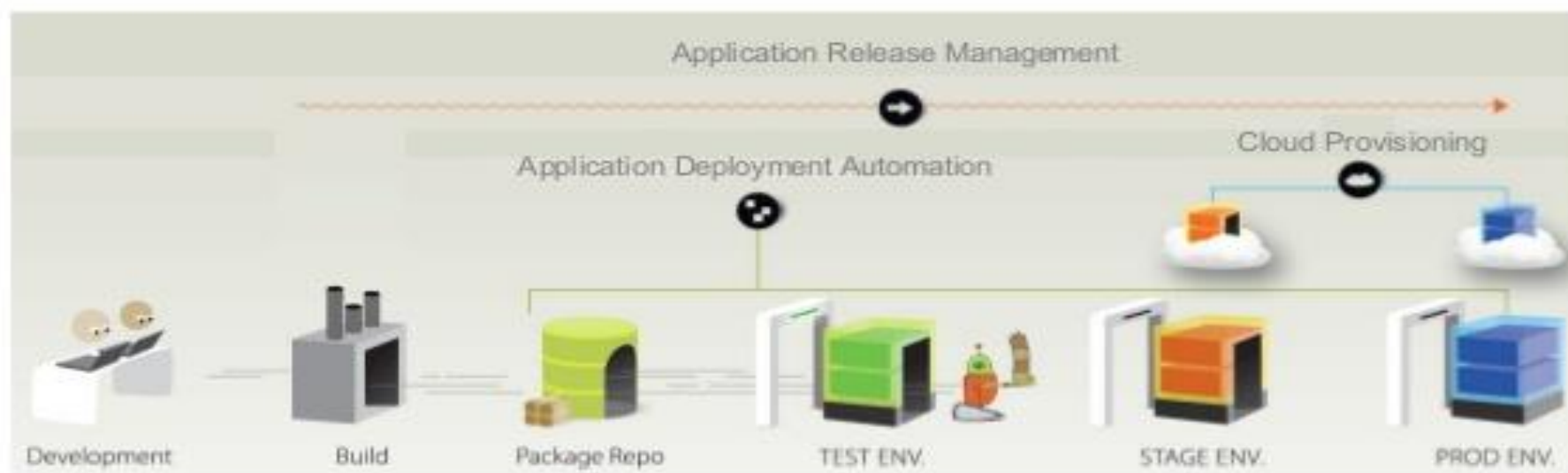


- **Common Source Control**
 - **Automated Builds (Build Definitions)**
 - **Continuous Integration (CI)**
 - **Automated Delivery**
 - **Continuous Delivery to Test (CD)**
 - **Continuous Delivery to Production-like Systems (Infrastructure as Code)**
 - **Continuous Delivery thru Prod (Cloud based)**

Continuous Delivery flow



DevOps Tool Chain



Accelerate Flow To Production

- Reduce batch size
- Automated environments mean identical dev/test/prod environments
- Create safety through automation
 - Continuous Integration/Testing
 - Automated Regression Testing
 - Continuous Delivery
 - Continuous Deployment
 - Feature Flags (A/B testing)
 - Security Testing

Metrics

- Capture, learn, improve.
- Assists in:
 - Capacity Planning
 - Trend Analysis
 - Fault Finding
- Simple as saving Tomcat access info
- Plotted on a graph over time

Before

- Talk about functional requirements
- Talk about **non-functional** requirements
 - Security
 - Backups
 - Availability
 - Upgradeability
 - Configuration Mgmt
 - Monitoring
 - Logging
 - Metrics

During

- Communication
- Source Control
- Automate Builds
- Automate Tests
- Automate Deployments (Dev, Test and Prod)
- Collate App and System Metrics

After

- Release
 - Retrospective Meetings
 - Continue to Run Tests
 - Monitor Applications and Systems
- Issues (Yes, they do happen)
 - Post Mortem Meetings

Dev Ops

A culture and mindset for collaborating
between developers and operations

it's
Q & A
TIME!



THANK YOU!

training@laksans.com

@copyright of www.laksans.com