



DevOps Fundamentals

Lesson Objectives

- Context Setting
- DevOps Introduction
- Overview of DevOps
- DevOps Tools
- DevOps practices
- Cloud : DevOps enabler
- DevOps steps



Context Setting

➤ **Disadvantages** of traditional project management

- Tightly controlled projects
- User Acceptance Testing is done and customer feedbacks are available only towards the end of the project
- Changes or Fixes are expensive
- Delayed delivery

Understanding Agile

➤ Advantages of Agile project management

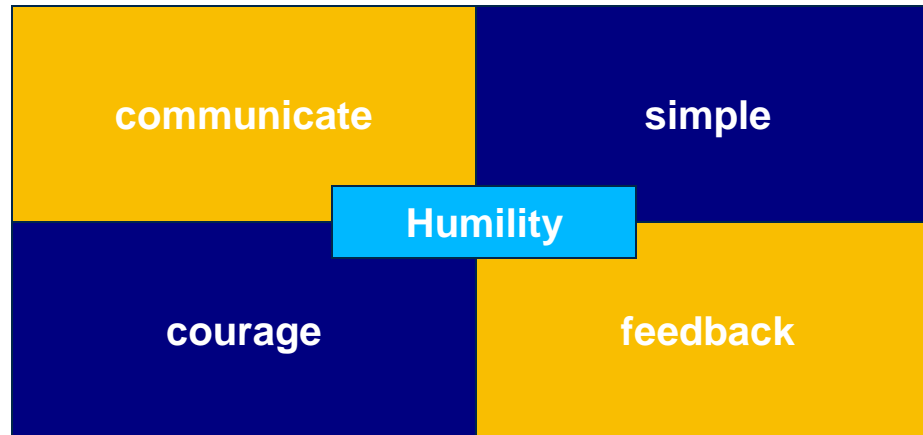
- Improved return on investment (RIO)
- Early detection and cancellation of failing products
- Higher quality software
- Improved control of a project
- Reduced dependence on individuals and increased flexibility

➤ Agile Concepts

- Customer Involvement : Person from customer's group joins the team of developers and helps select and prioritize the requirements to be implemented.
- Frequent and short releases: Frequently releasing pieces of software product provides the ability to deliver faster and expected results
- Facilitating Extraction: High Level and detailed level end-user requirements can be extracted using Agile
- Acceptance Test Criteria during requirement gathering: Acceptance tests are transformed into unit tests by developers before any other development activity

Understanding Agile : Values

- **Communication leads to valuable feedback which encourages simplicity which allows for courage to change**



Understanding Agile Development process

- **Iterative and evolutionary**
- **Timeboxing**
 - Set amount of time for iteration
 - Adapt future iteration based on the realities
- **Adaptive planning**
- **Incremental delivery**
- **Focused towards success than sticking with a plan**
- **Working software is valued and considered as a measure of progress**

Understanding Agile ways

➤ Scrum

- Management framework for incremental product development using one or more cross-functional, self-organizing teams

➤ Extreme Programming (XP)

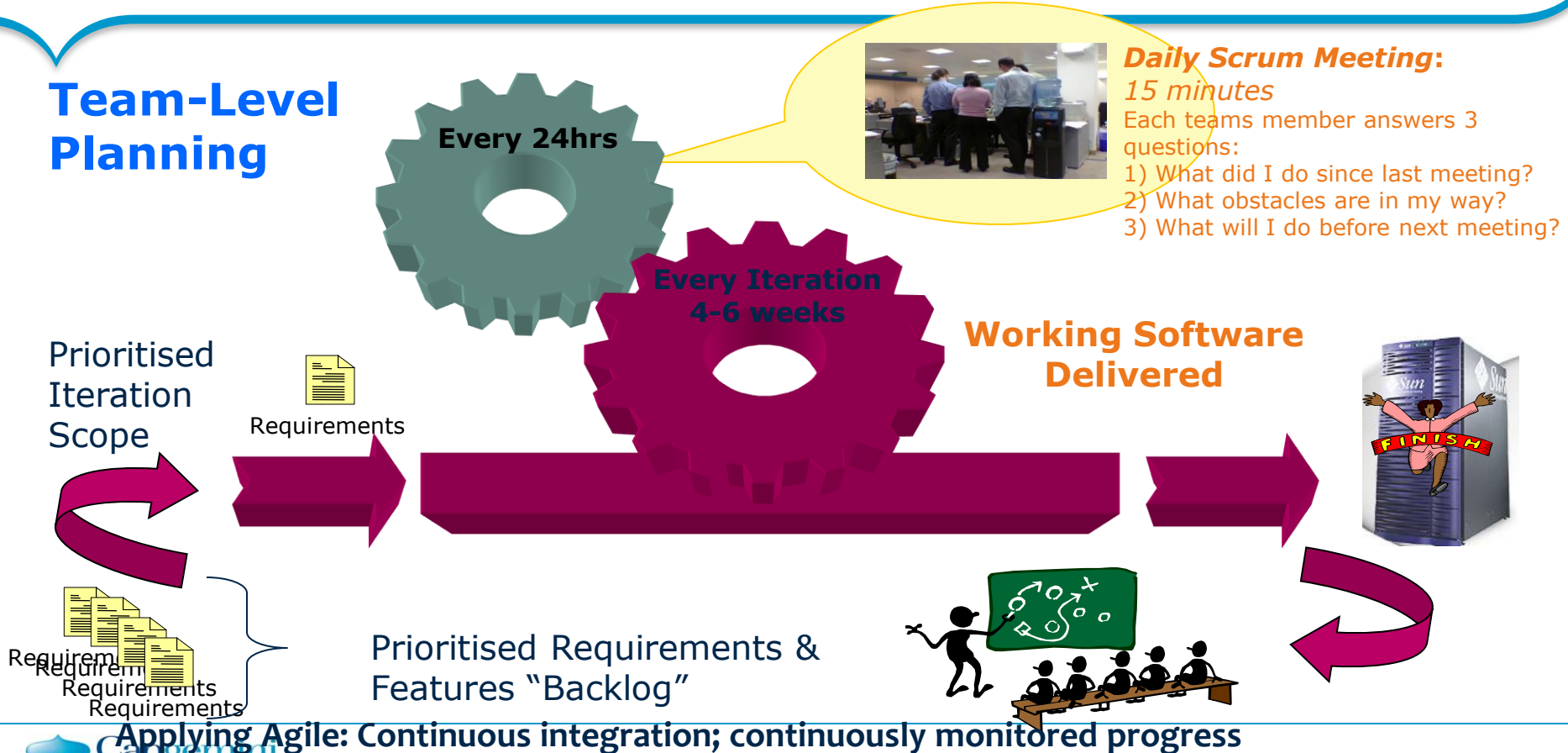
- Core software engineering practices essential to every agile project

➤ Lean, Kanban ...

➤ Your own

Understanding Agile project management - SCRUM

Team-Level Planning



DevOps Introduction : Why DevOps?



Why DevOps?

Innovation-Agility

Vs

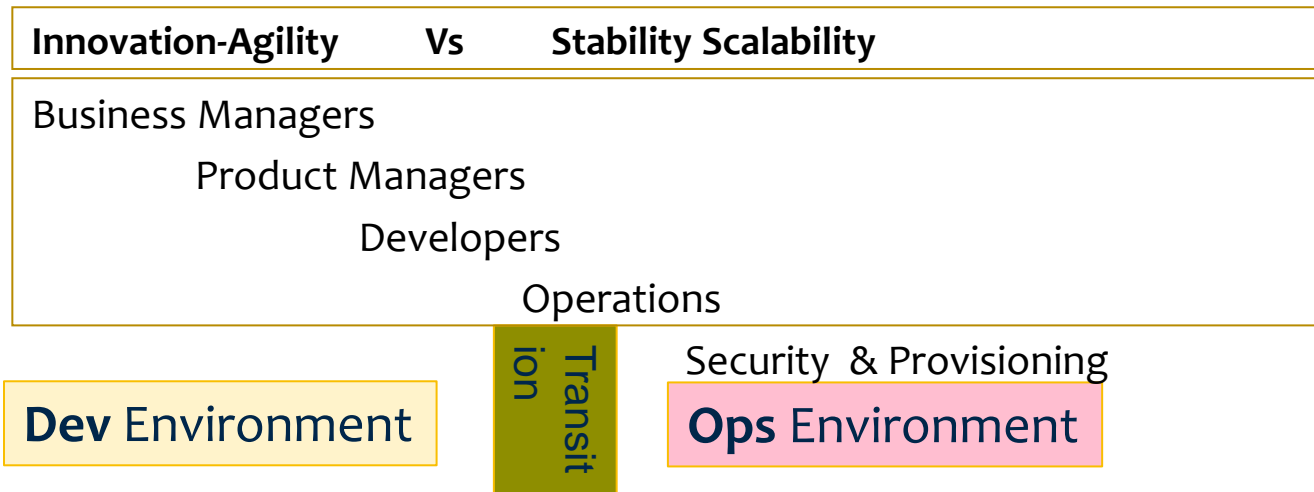
Stability Scalability

Dev Environment

Transit
ion

Ops Environment

Why DevOps?



What is DevOps?

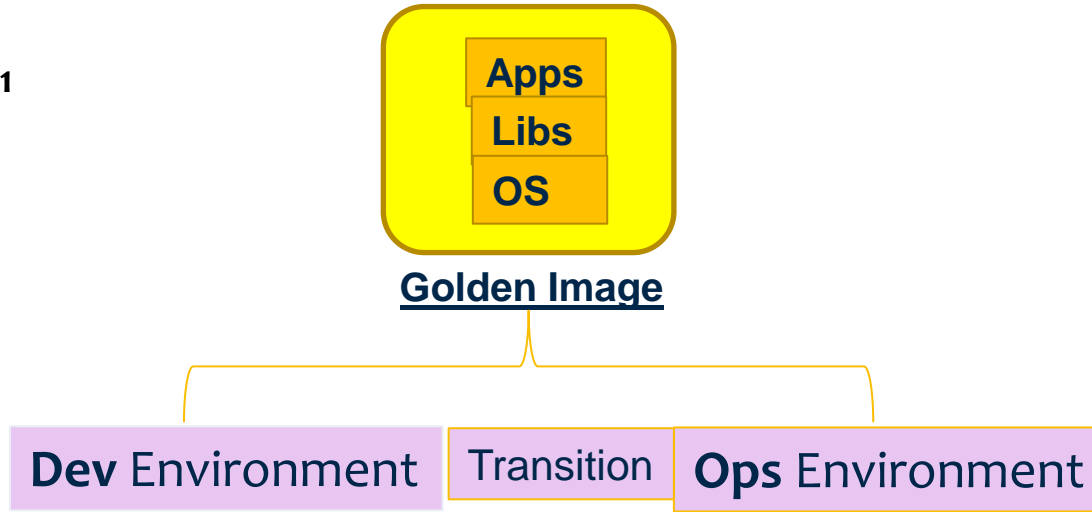
Dev Environment

Transition

Ops Environment

How DevOps?

➤ Option 1

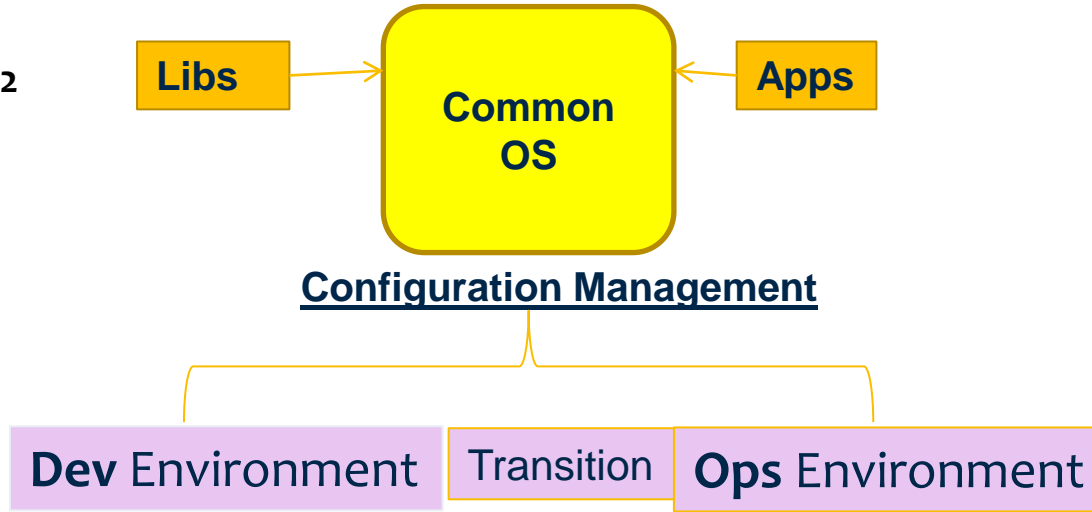


- **Desktops/workstations**
- **Virtual Machines**
- **Servers**

How DevOps?



Option 2

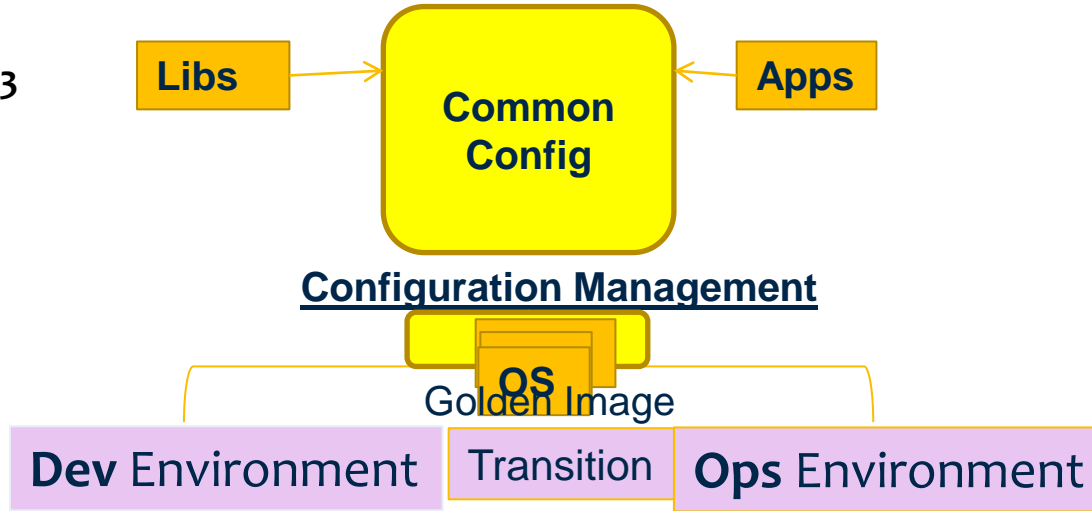


- Desktops/workstations
- Virtual Machines
- Servers

How DevOps?

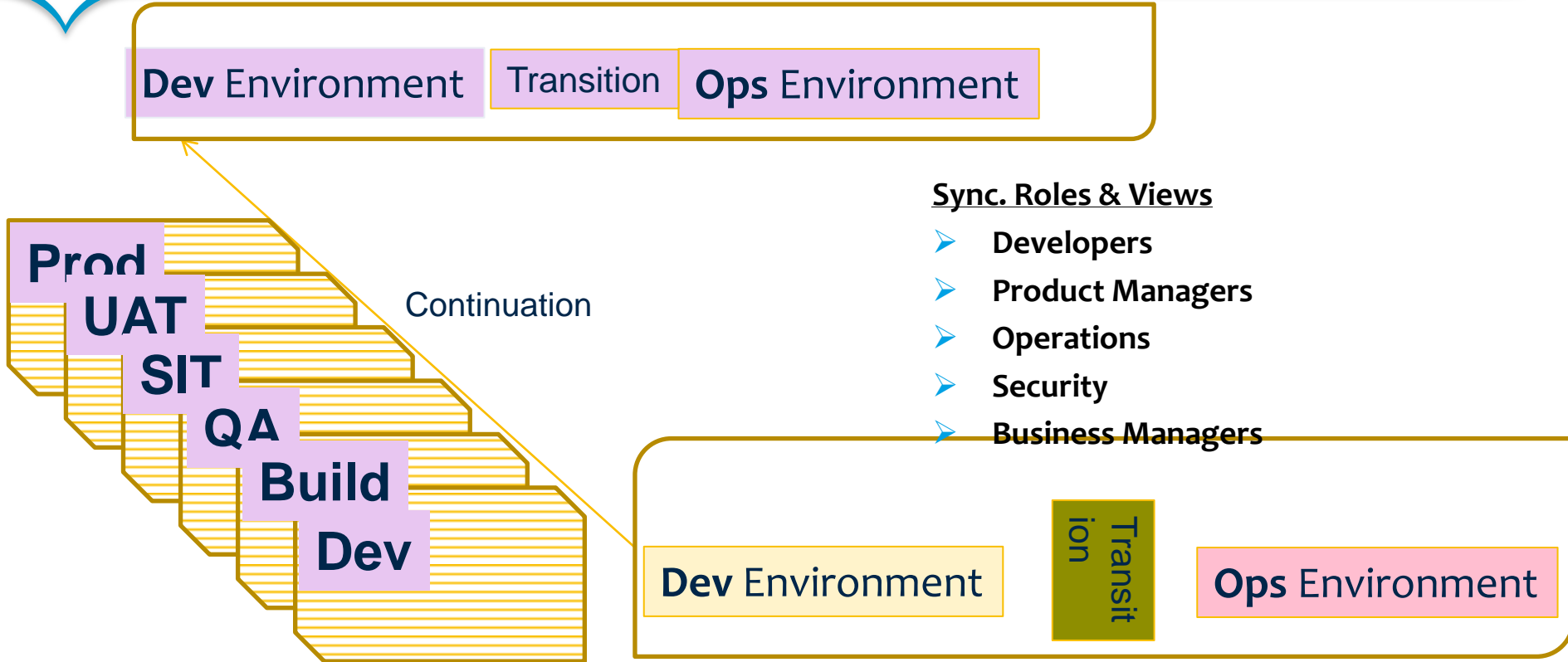


Option 3

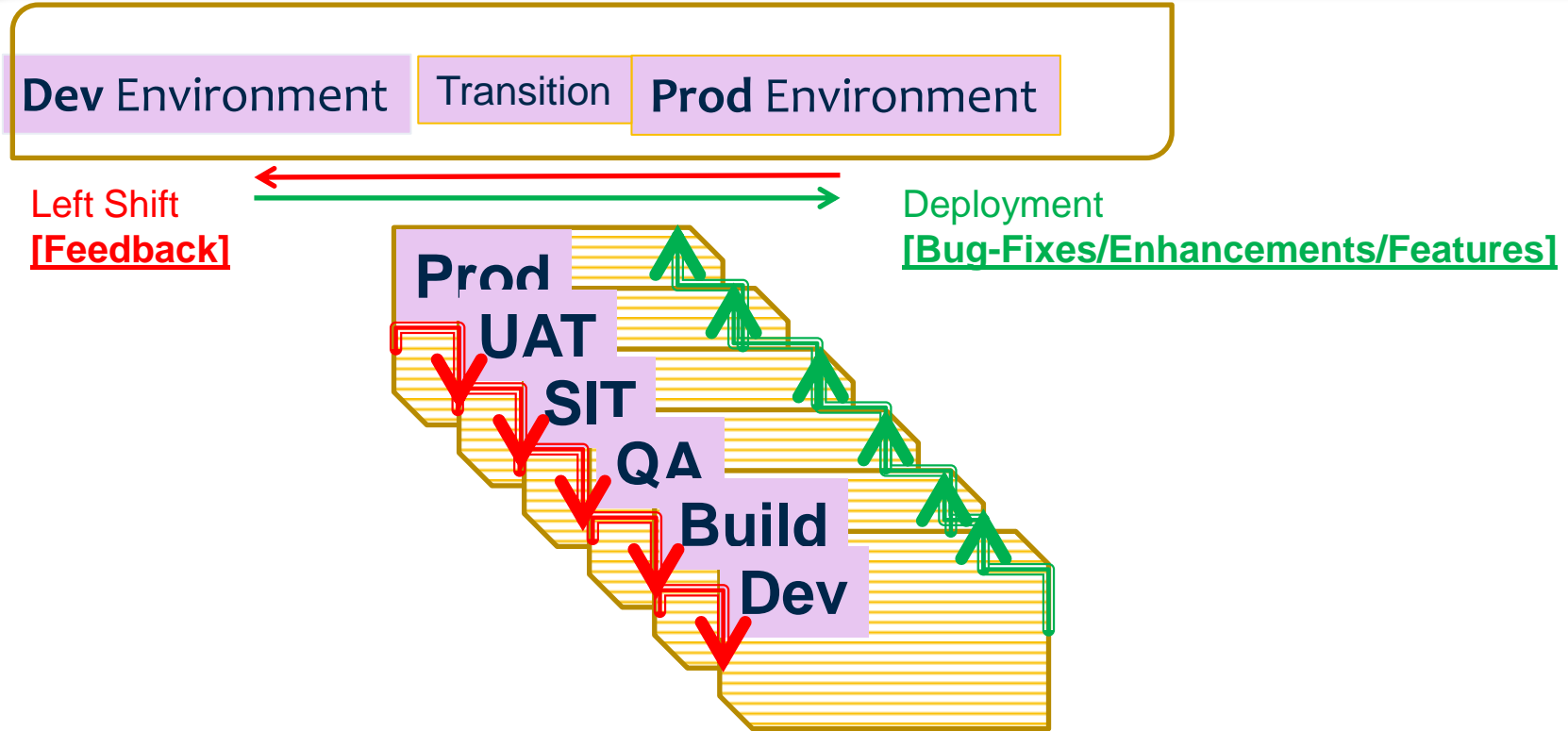


- Desktops/workstations
- Virtual Machines
- Servers

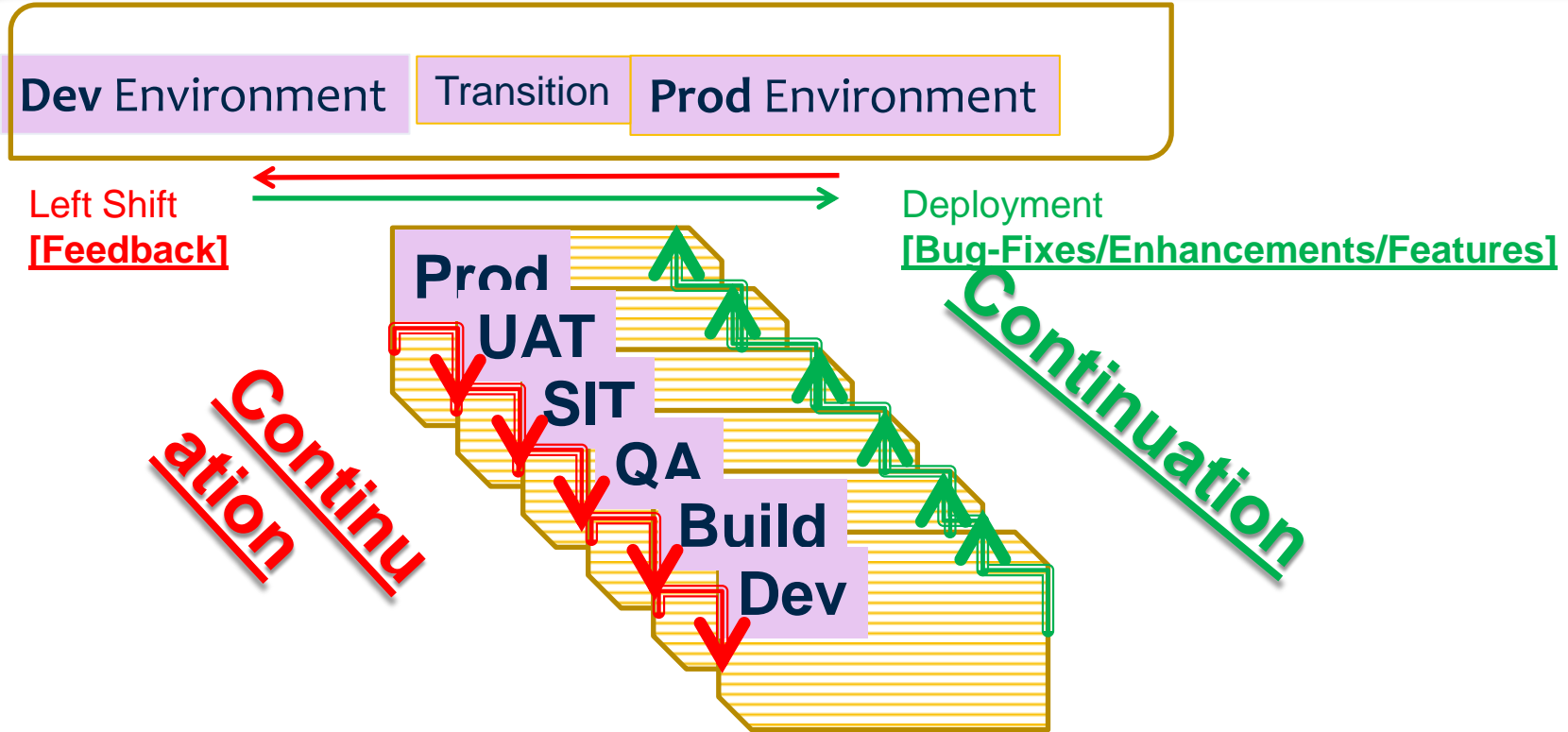
How DevOps?



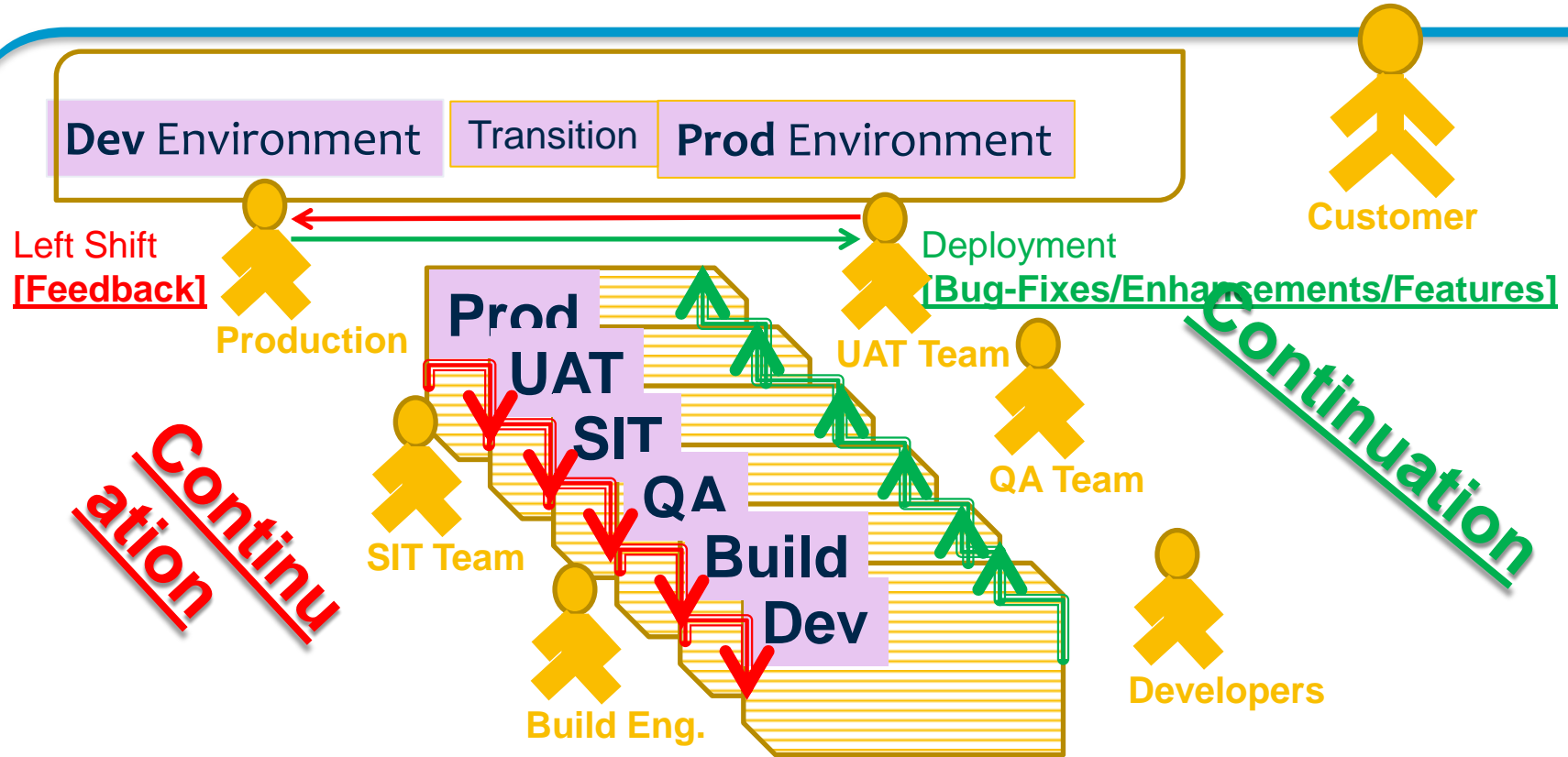
How DevOps?



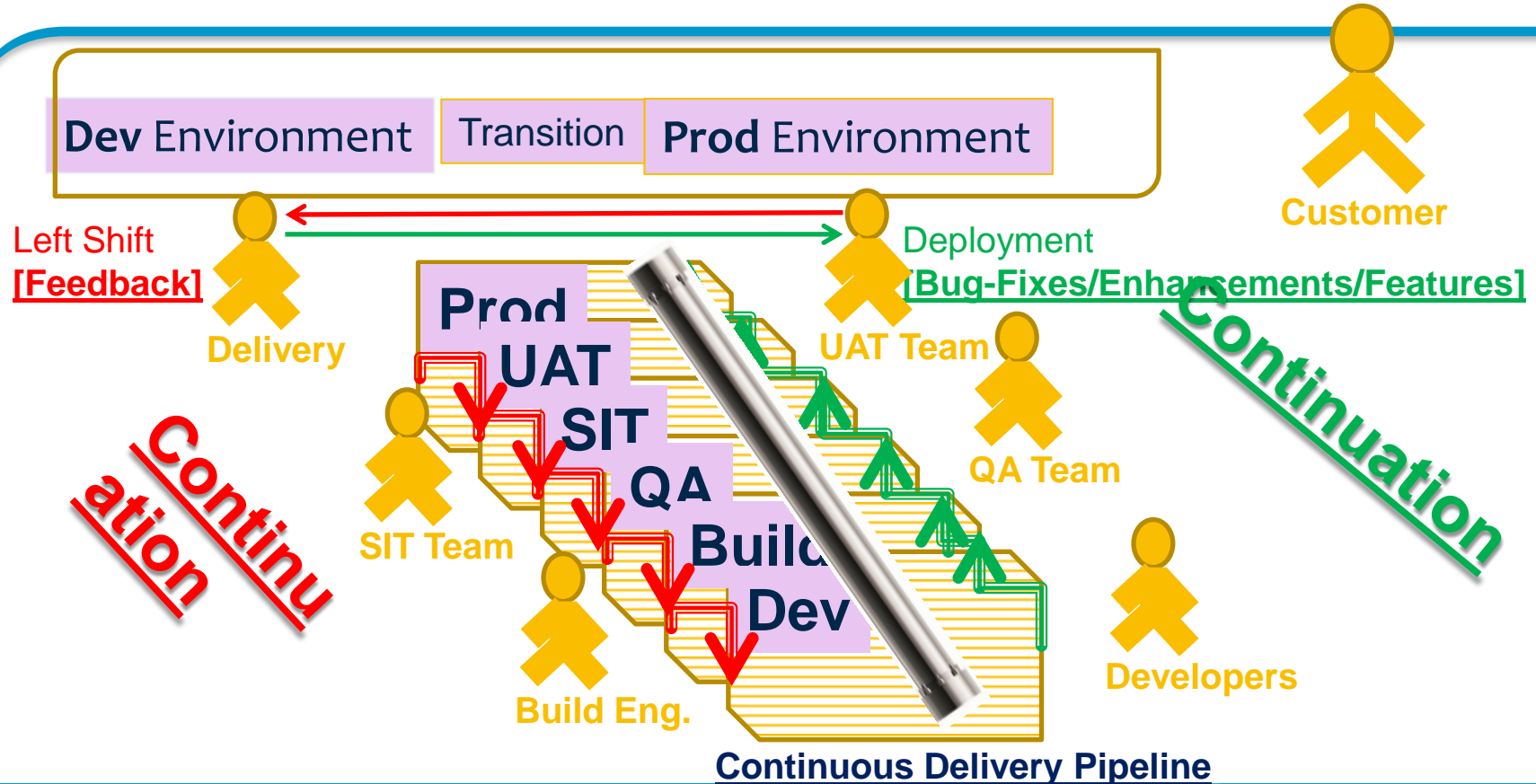
How DevOps?



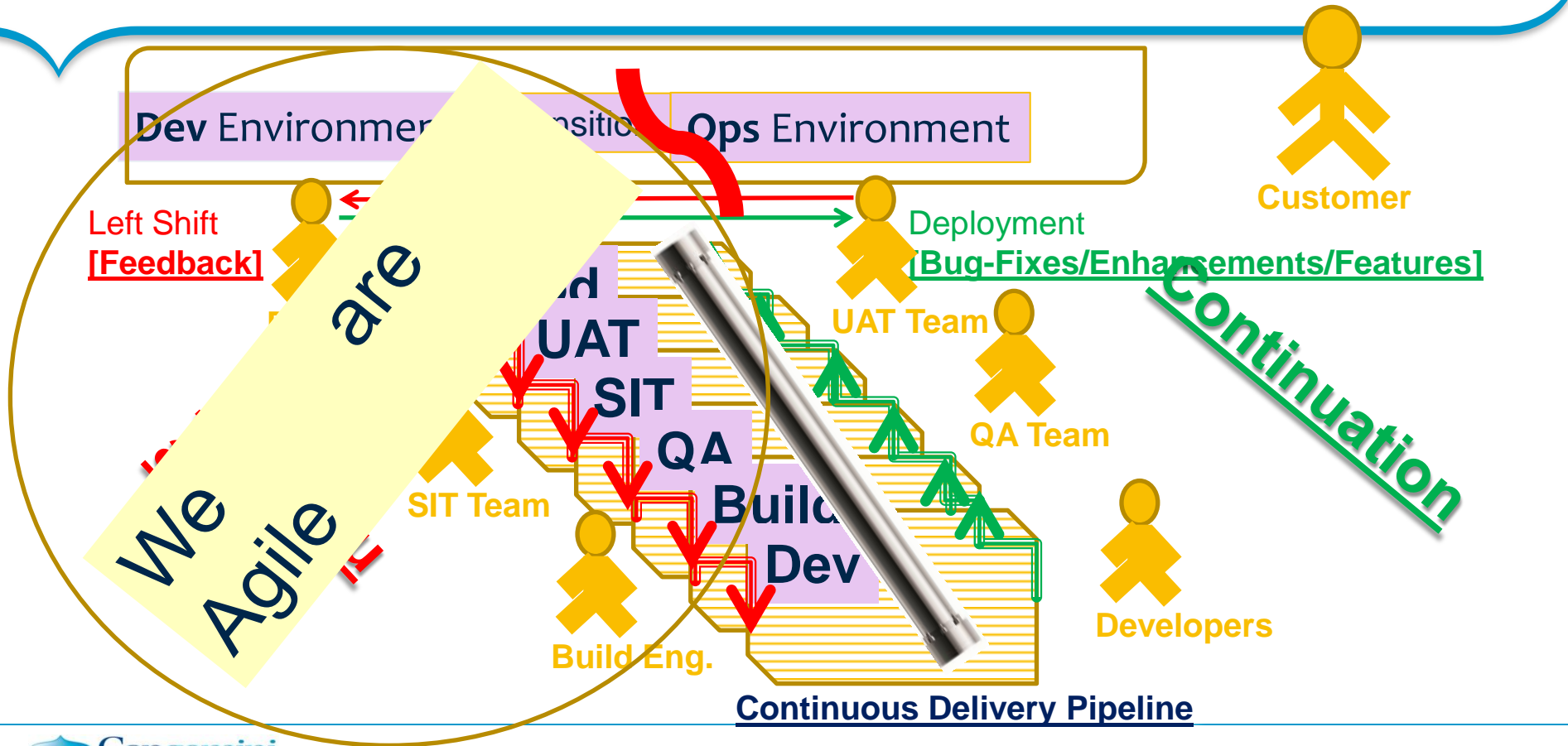
How DevOps?



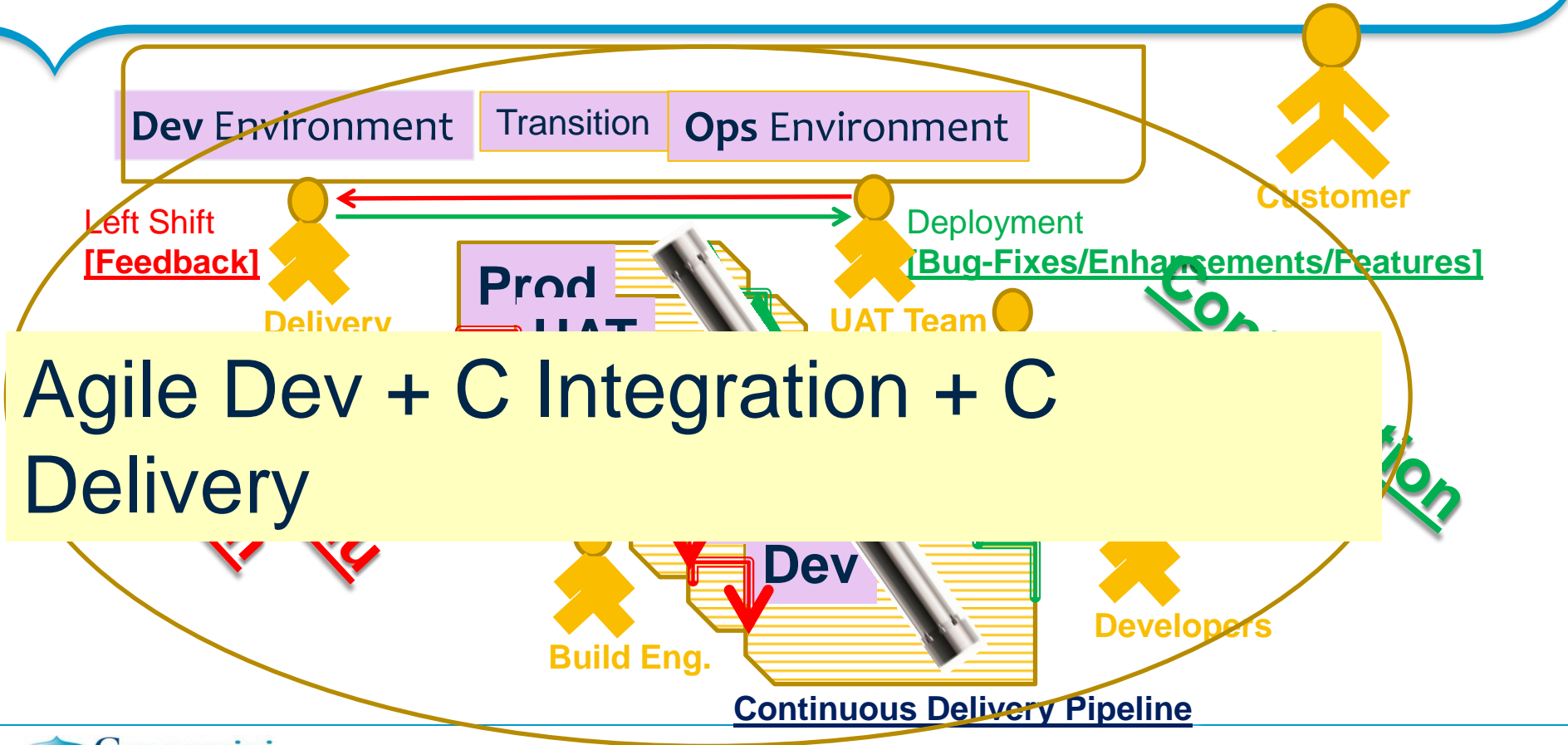
How DevOps?



Is DevOps same as Agile?

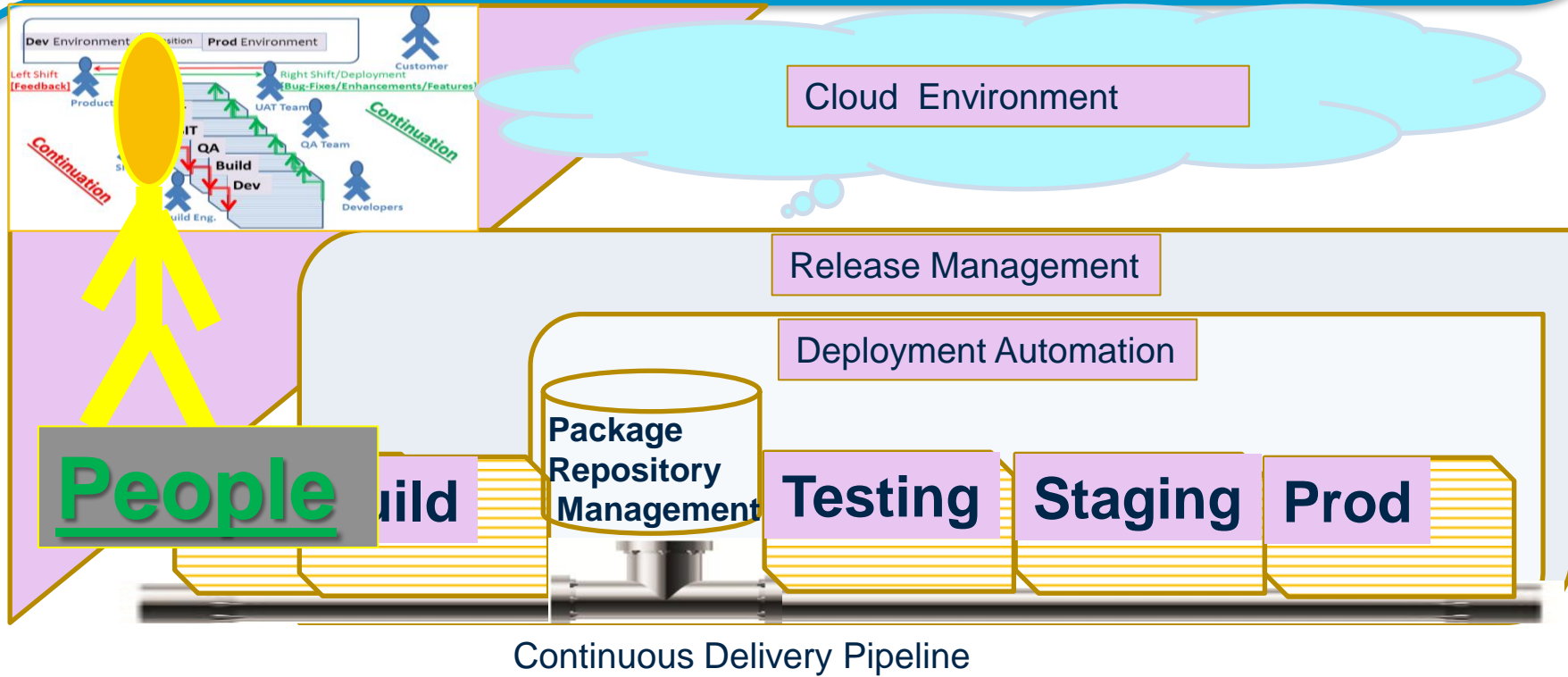


DevOps : Software Development Lifecycle

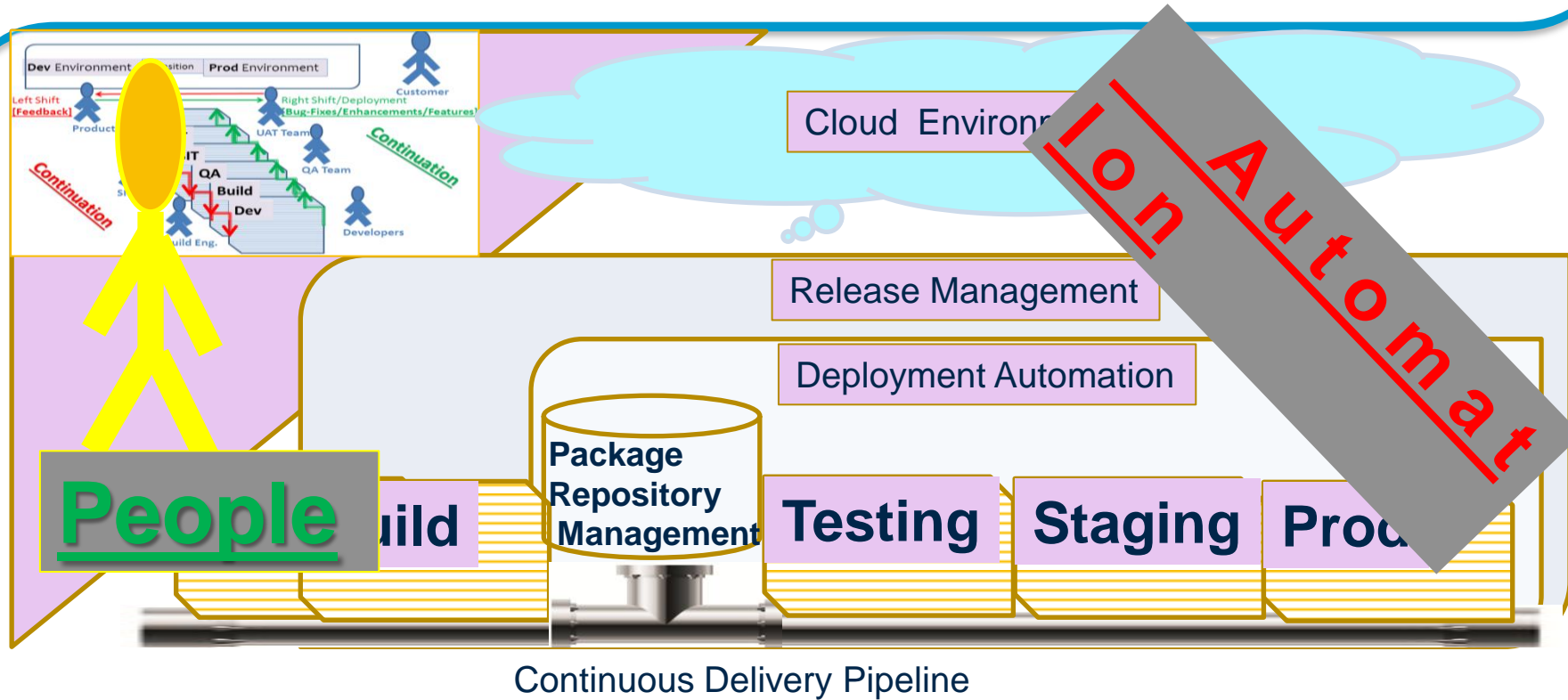


[illegible]

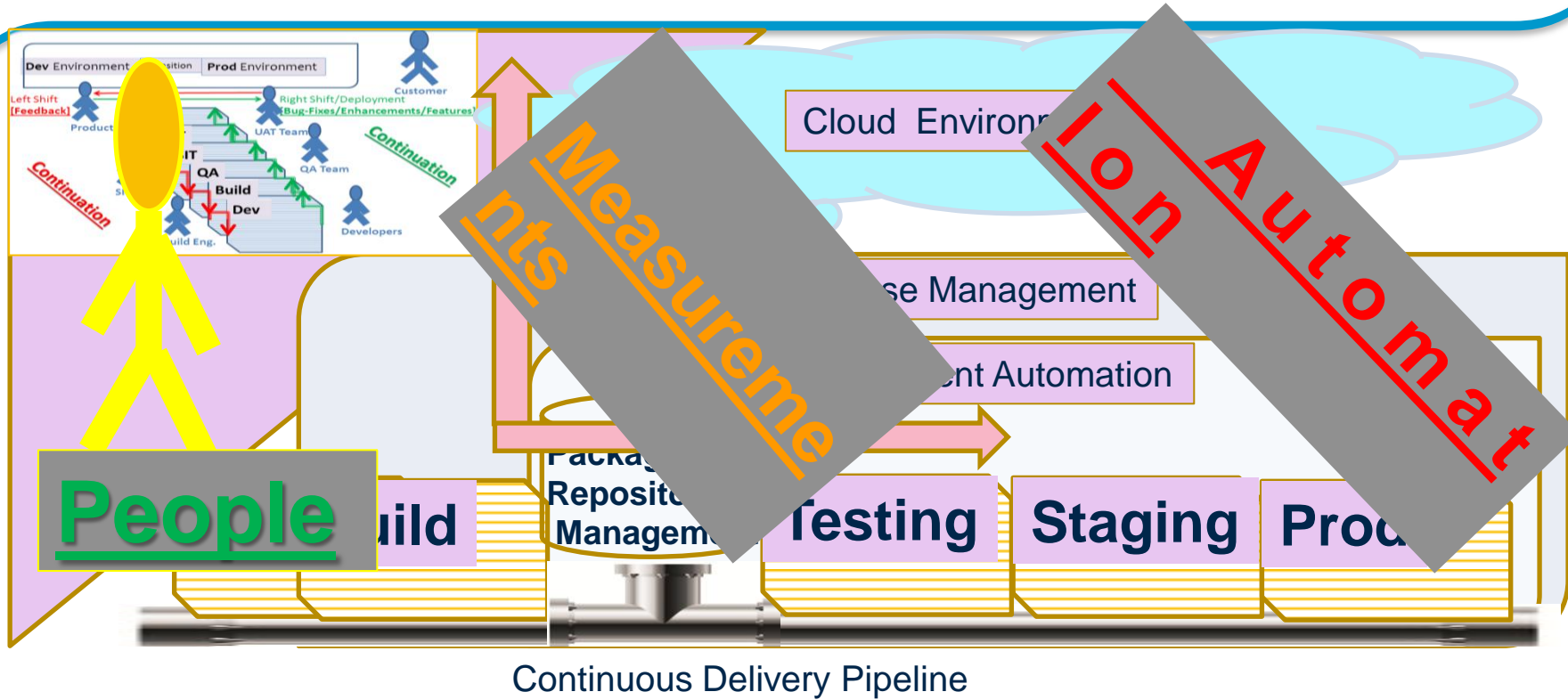
DevOps Continuous Delivery Principles



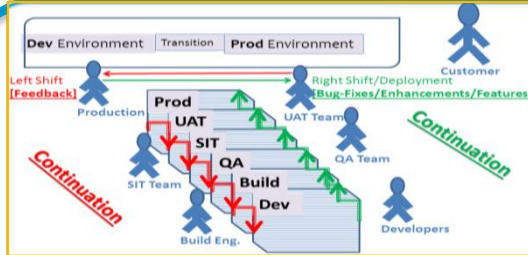
DevOps Continuous Delivery Principles



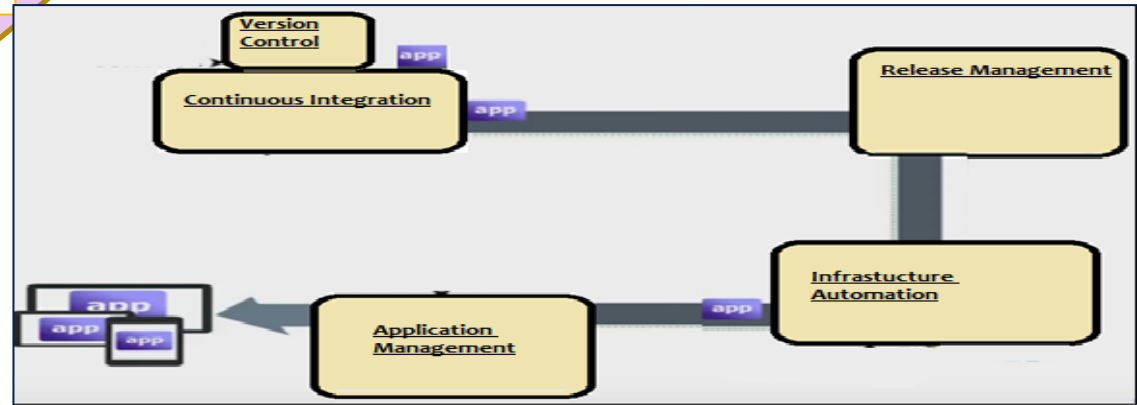
DevOps Continuous Delivery Principles



Overview of DevOps : Processes

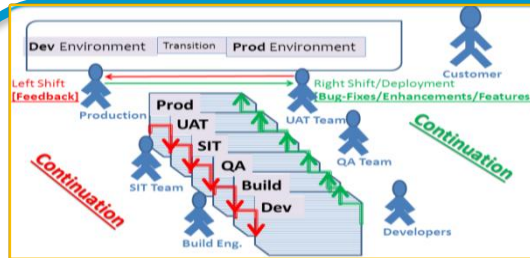


- Processes in DevOps
- Cross Platform/Technology tools



Continuous Delivery Pipeline

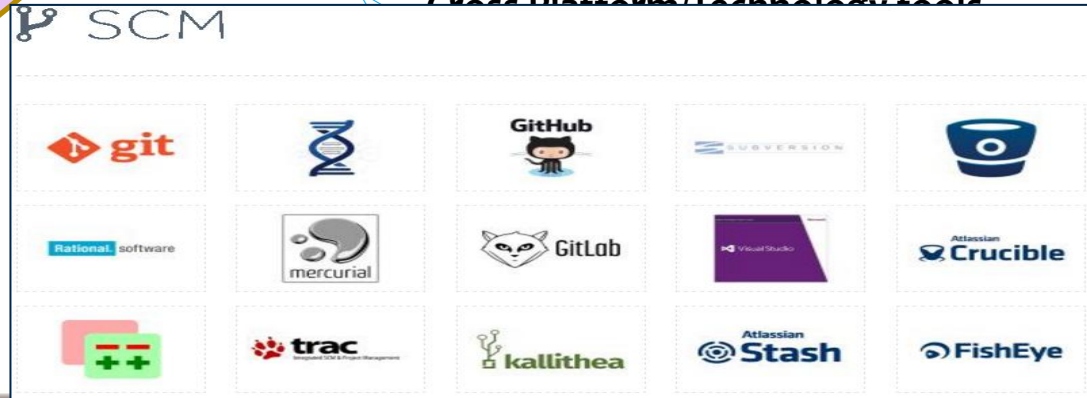
Overview of DevOps : Dev. Environment



Collaborative Development

- Project Planning
- Source Control Management
- Unit Testing

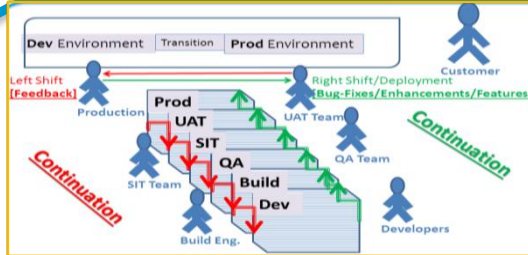
Cross Platform/Technology tools



Dev

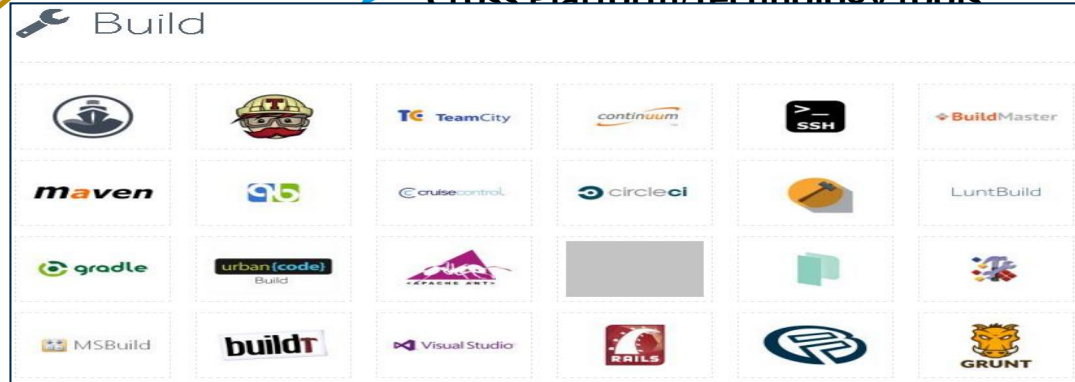
Continuous Delivery Pipeline

Overview of DevOps : Build Environment



- Perform the builds
 - Compile and unit test
 - Getting ready for deployment

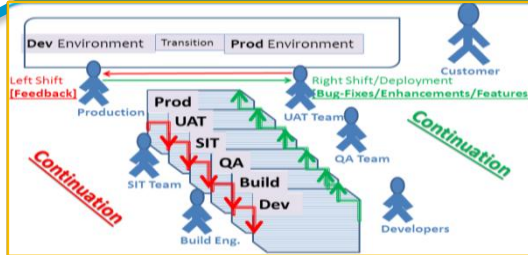
➤ Cross Platform/Technology tools



Build

Continuous Delivery Pipeline

Overview of DevOps : Repository Management



➤ Artifact Repository of ...

- Built binaries, Configuration files
- Deployment scripts
- Infrastructure as code

➤ Repository Management tools



Repository Management

artifactory

Nexus

archiva



nuget

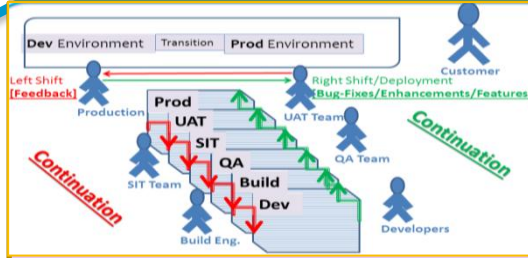


npm

Package
Repository
Management

Continuous Delivery Pipeline

Overview of DevOps : Testing Environment



Testing

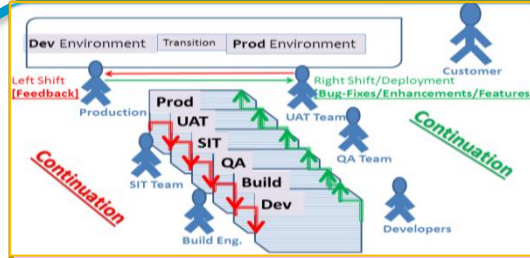
- UAT
- Integration
- Performance

Testing



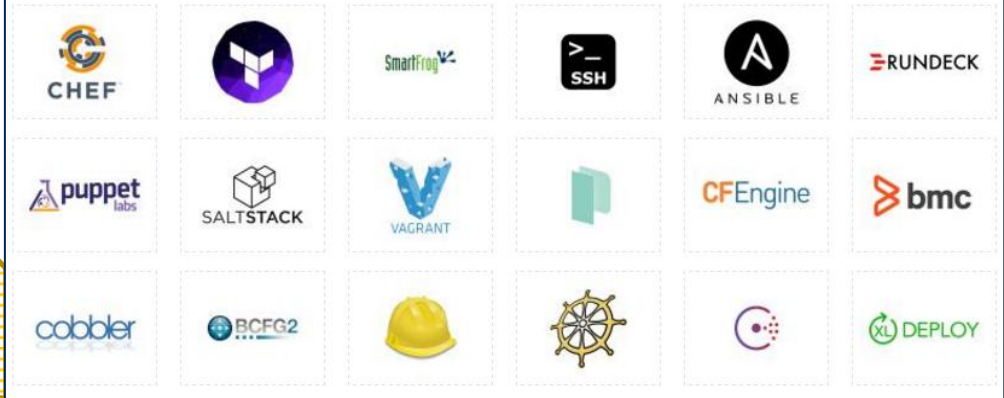
Continuous Delivery Pipeline

Overview of DevOps : Staging/Prod. Environment



Provisioning/ Configuration Management

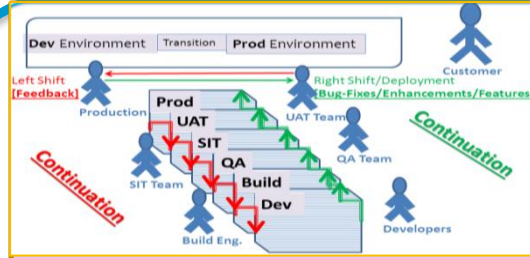
Provisioning / Configuration Management



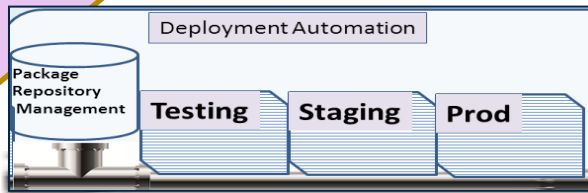
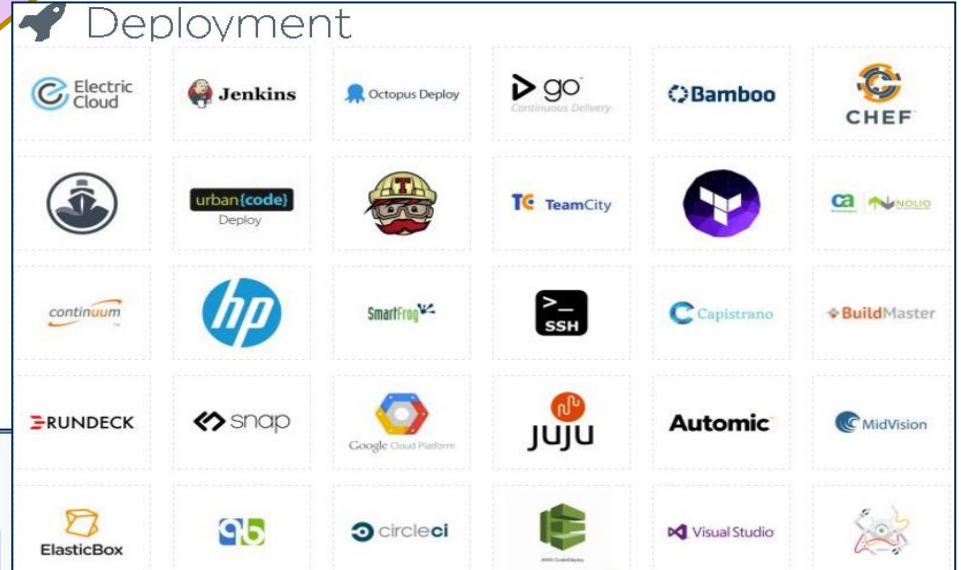
Stagin Prod

Continuous Delivery Pipeline

Overview of DevOps Landscape : Deployment Automation

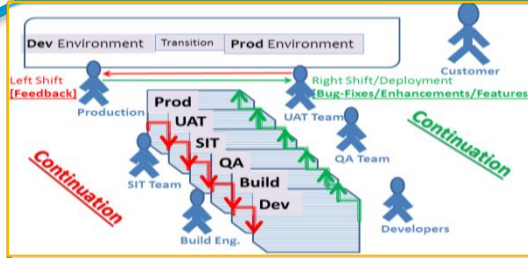


Deployment Automation tools

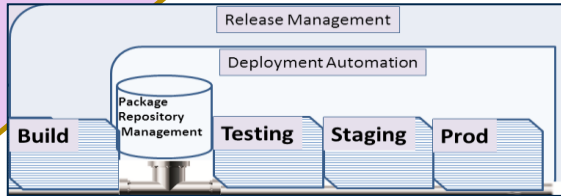
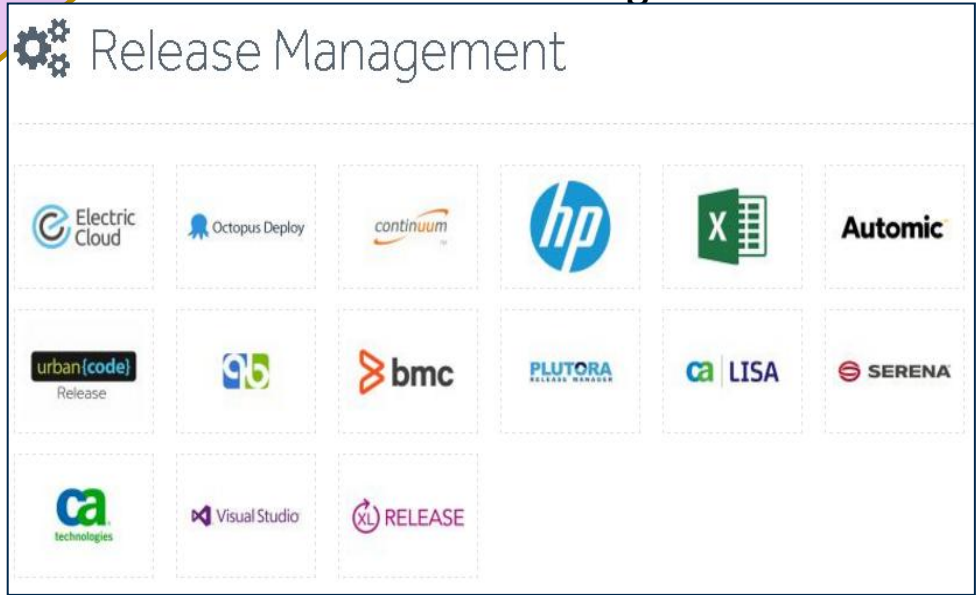


Continuous Delivery Pipeline

Overview of DevOps Landscape : Release Management

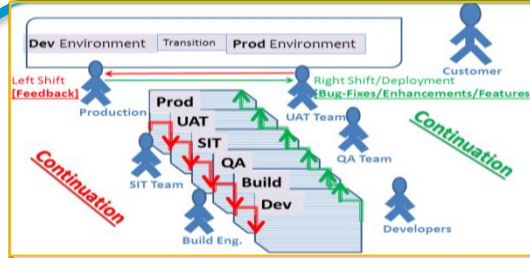


Automated Release Management tools



Continuous Delivery Pipeline

Overview of DevOps Landscape Cloud Provisioning



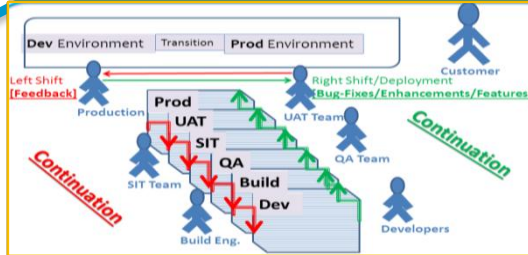
Cloud Provisioning tools



Cloud Environment

Continuous Delivery Pipeline

DevOps Practices : Infrastructure as Code

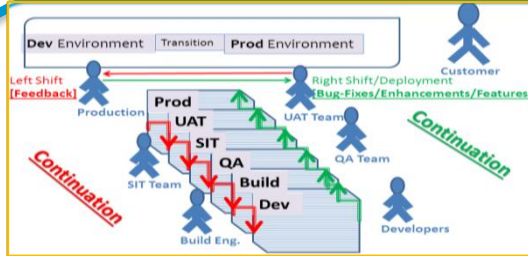


➤ Scale and the speed of provisioning infra structure configured to enable continuous delivery.

- Automated frequent builds on various configuration environments and instances on automated CI environments demands continued automation
- Providing scripts to automate the create the required environments at required speed just on demand
- Versioning the scripts in SCM
- Making changes to the scripts to introduce more varied environments

Continuous Delivery Pipeline

DevOps Practices : Infrastructure as Code



➤ Approaches to IaC script provisioning

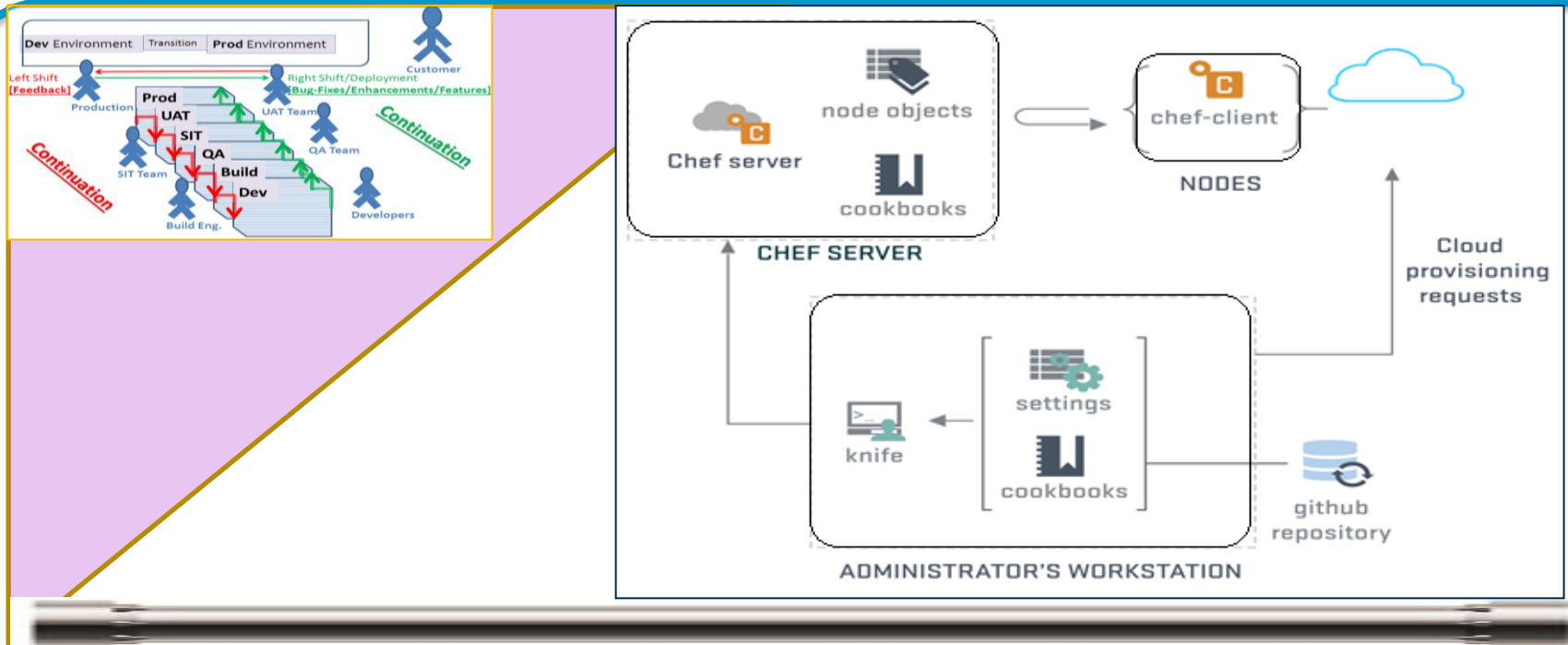
- Generic
- Application centric

➤ Generic IaC script provisioning

- Compliance mandates
 - Golden image creation/ updates
 - Launching of instances/modifications
-
- Example : Chef / Puppet

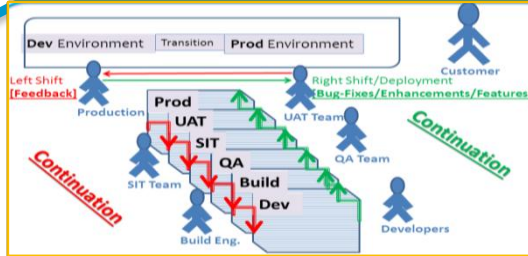
Continuous Delivery Pipeline

DevOps Practices : Infrastructure as Code : Chef



Continuous Delivery Pipeline

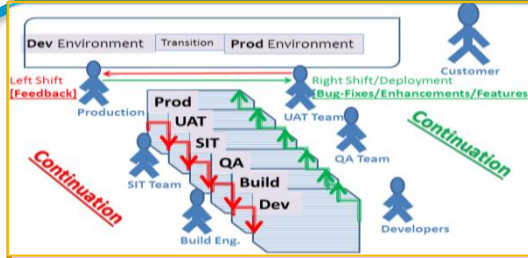
DevOps Practices : Infrastructure as Code : Chef script



```
bash "install_tomcat6" do
  tomcat_version_name = "apache-tomcat-#{node[:tomcat6][:version]}"
  tomcat_version_name_tgz = "#{tomcat_version_name}.tar.gz"
  user "root"
  cwd usr_share_dir
  not_if do ::File.exists?(::File.join(usr_share_dir,tomcat_version_name))
  end
  code <<-EOH
  wget http://archive.apache.org/dist/tomcat/tomcat-6/v#{node\[:tomcat6\]\[:version\]}/bin/#{tomcat\_version\_name\_tgz}
  tar -zxf #{tomcat_version_name_tgz}
  rm #{tomcat_version_name_tgz}
  chown -R #{node[:tomcat6][:user]}:#{node[:tomcat6][:user]}
  #{tomcat_version_name}
  EOH
end
```

Continuous Delivery Pipeline

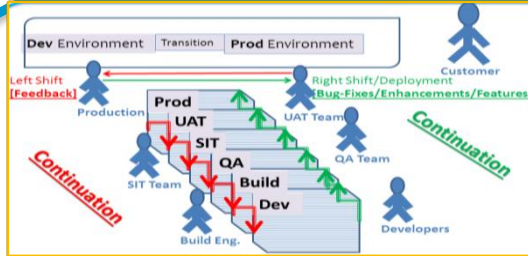
DevOps Practices : Continuous Integration



- **Faster “Source-Build-Test”**
- **Integrating builds into the common build, delivering it to a common cross team build server for system/application wide integration build’ on a regular basis.**
- **Address integration issues on a regular basis.**
- **Integration of results leads to early discovery and exposure of integration risks.**
- **Tool Example : Jenkins/Agile GO**

Continuous Delivery Pipeline

DevOps Practices : Automated Testing



➤ Includes testing for each environment in the pipeline

➤ **Dev. Environment**

- Unit, Sanity Tasting

➤ **CI Environment**

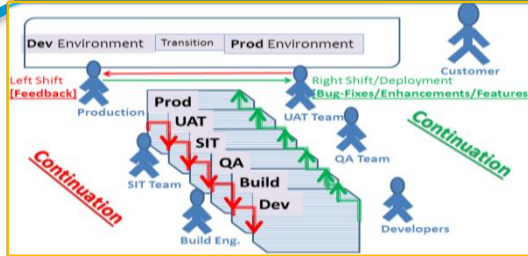
- Incremental Integration Testing

➤ **QA Environment**

- Functional , Usability Testing
- Compatibility Testing

Continuous Delivery Pipeline

DevOps Practices : Automated Testing



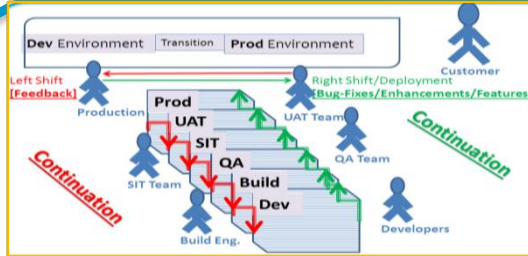
➤ Includes testing for each environment in the pipeline ...

- Staging Environment
 - Performance Testing
 - Stress Testing
 - Load Testing
 - End-To-End Testing
 - System Testing

➤ Bringing continuation in testing over different environments in the pipeline : Continuous Testing

Continuous Delivery Pipeline

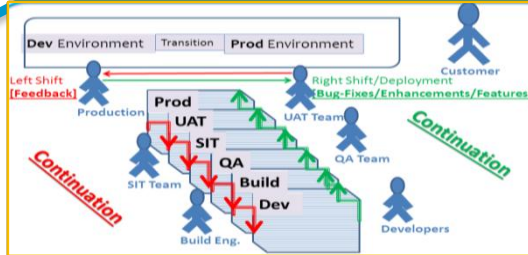
DevOps Practices : Continuous Testing



- Testing earlier and continuously across the life cycle giving continuous feedback
- AKA 'Left-Shift 'Testing
- Testing against production-like systems
- 'Ops' goal : Provide for production line ops-environments:
 - Test application behaves & performs well before it's ready for deployment.
 - The application delivery processes to be tested and validated upfront.

Continuous Delivery Pipeline

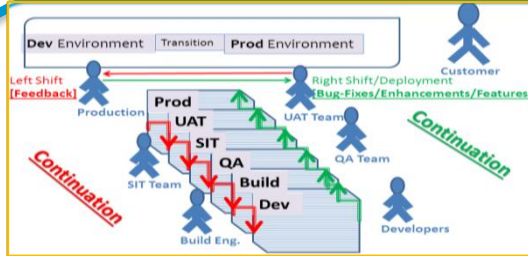
DevOps Practices : Using Automated Testing for Continuation



- Automating the ability to run the tests.
- Deploying test environments via automation
- Design how the tests will run unattended into a test framework.
- Make that test framework available to developers
- New Application architectures can have layer to enable deployment of test environments and automating tests
- Build test automation for security

Continuous Delivery Pipeline

DevOps Practices : Using Automated Testing for Continuation

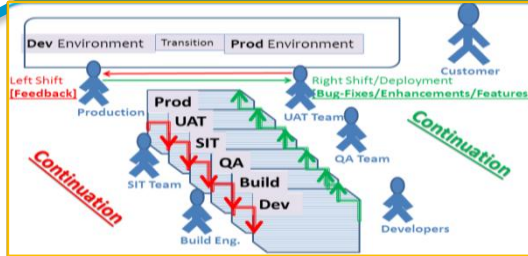


➤ High level steps

- Automating application deployment to test environments
- Collaborate around the application architecture to make it as easy as possible to automate application deployment. To test environment
- POC to establish the automation

Continuous Delivery Pipeline

DevOps Practices : Using Automated Testing for Continuation

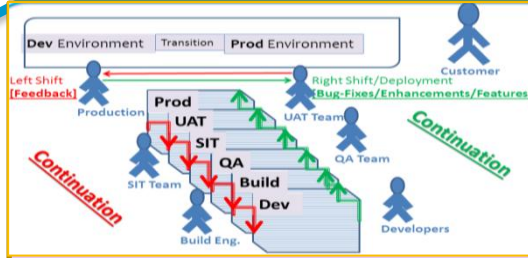


➤ High level steps

- Establish Framework that promotes the abstraction layer of your choice and provides reusable components for creating new scripts.
- Ensure every code change is committed with a test to the framework
- Every new test runs on very next build (or pipeline trigger)

Continuous Delivery Pipeline

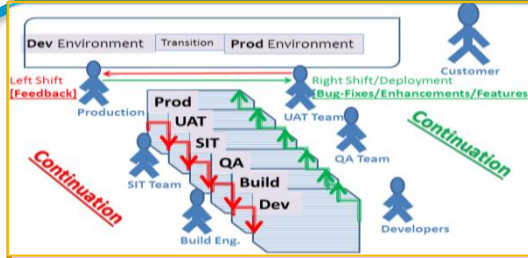
DevOps Practices : Continuous Deployment & Release Management



- Continuous deployment and release management raise the concept of continuous integration to the next level enabling creation of the delivery pipeline .
- This pipeline automates continuous deployment of software to QA environment, then to production in an efficient manner.
- Continuous release and deployment makes it possible to release new features to customers and users at the earliest possible..
- Correct selection of tooling and processes make up the core of DevOps to facilitate continuous integration, continuous release, and continuous deployment.

Continuous Delivery Pipeline

Cloud : DevOps Enabler

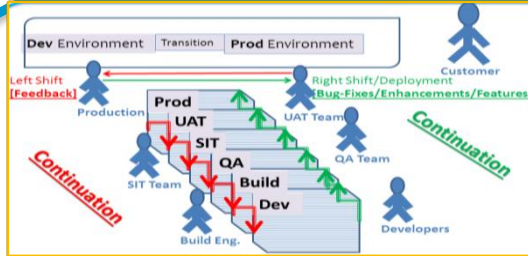


Advantages of Cloud based DevOps solutions

- Fast and dynamically environment provisioning or de-provisioning
- Out of the box Automation solutions
- Easy Service Virtualization
- Cloud DevOps Services Examples : IBM /AWS

Continuous Delivery Pipeline

DevOps steps ...



DevOps steps...

- Expanding the boundaries of Agile practices (Dev+Ops teams=scrum like)
- Adapting agile continuous testing (Reduce : test cycle, Increase : release quality)
- Build your delivery pipeline (Tools and other Ops options)
- Bring in the culture to regularly experiment for improvement and measure the results

Continuous Delivery Pipeline

Lesson Summary

- Context Setting
- DevOps Introduction
- Overview of DevOps
- DevOps Tools
- DevOps practices
- Cloud : DevOps enabler
- DevOps steps

