HITACHI
Inspire the Next

DevOps Continuous Integration

From Hitachi Vantara

Andrés Pérez
Enterprise Architect, Enterprise Architecture Group
July 2018

Agenda

HITACHI
Inspire the Next

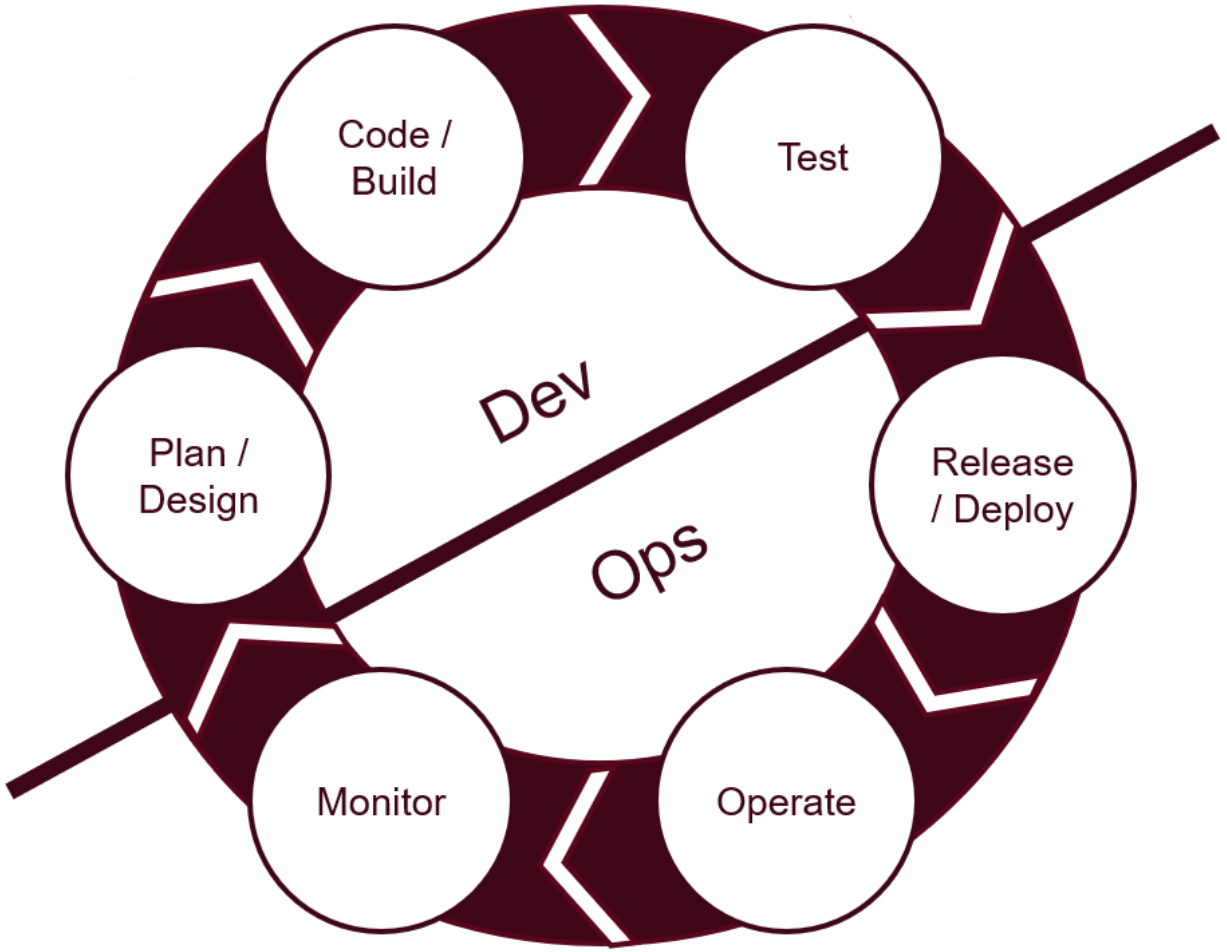
- What is DevOps?
  - Lifecycle
  - Maturity & Phases
  - Challenges & Goals
- Focus on Continuous Integration
  - Lifecycle Diagram
  - Solution Repository
  - Testing
  - Handling CI

What is DevOps?

HITACHI
Inspire the Next

Key concepts:

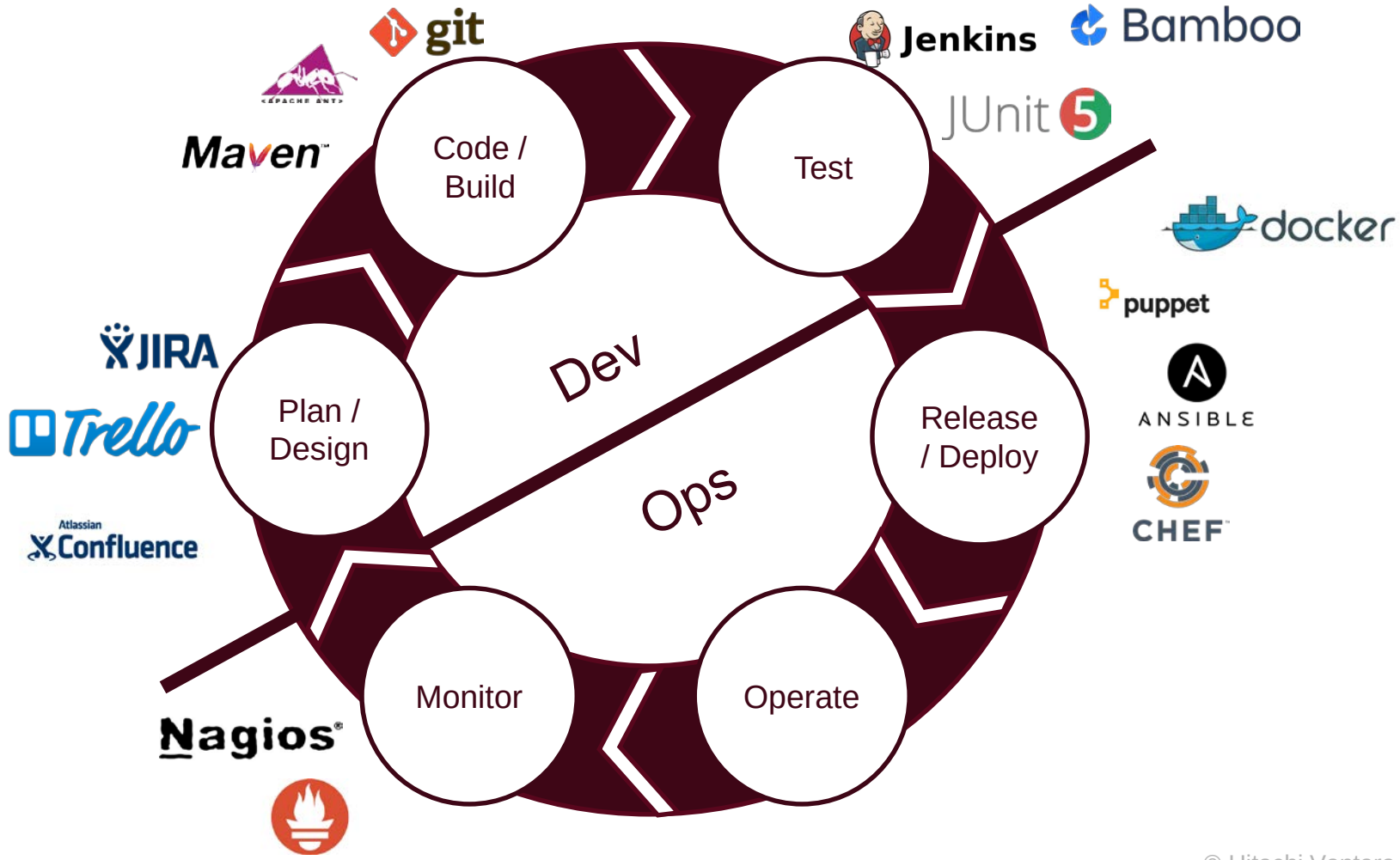
- DevOps is a methodology to create software solutions
- DevOps is based on the integration between software developers and system administrators
- DevOps helps to manufacture software faster, with higher quality, lower cost and a very high frequency of releases



DevOps Lifecycle diagram

HITACHI
Inspire the Next

- Identify tools used on each stage



HITACHI
Inspire the Next

Phases of the DevOps Maturity

DevOps Transition Path

Phases of DevOps Maturity

DevOps Transition path

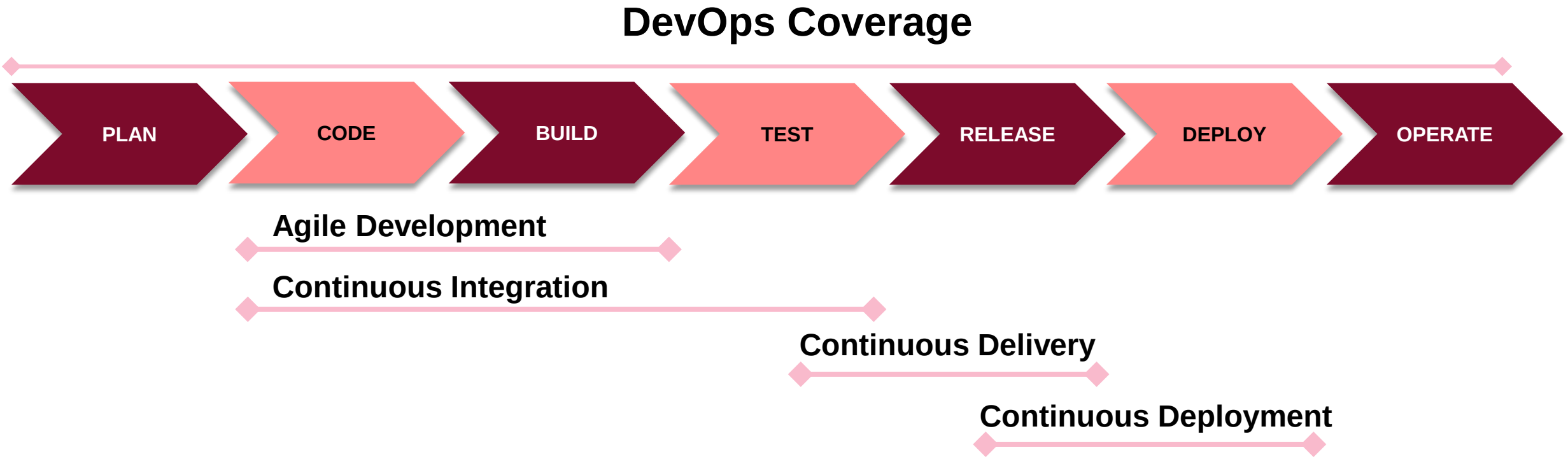
HITACHI
Inspire the Next

The process for adopting DevOps within an organization has several phases that evolve from each other:

- **Continuous Integration (CI)**.
- **Continuous Delivery**.
- **Continuous Deployment (CD)**.

Phases of DevOps Maturity

HITACHI
Inspire the Next



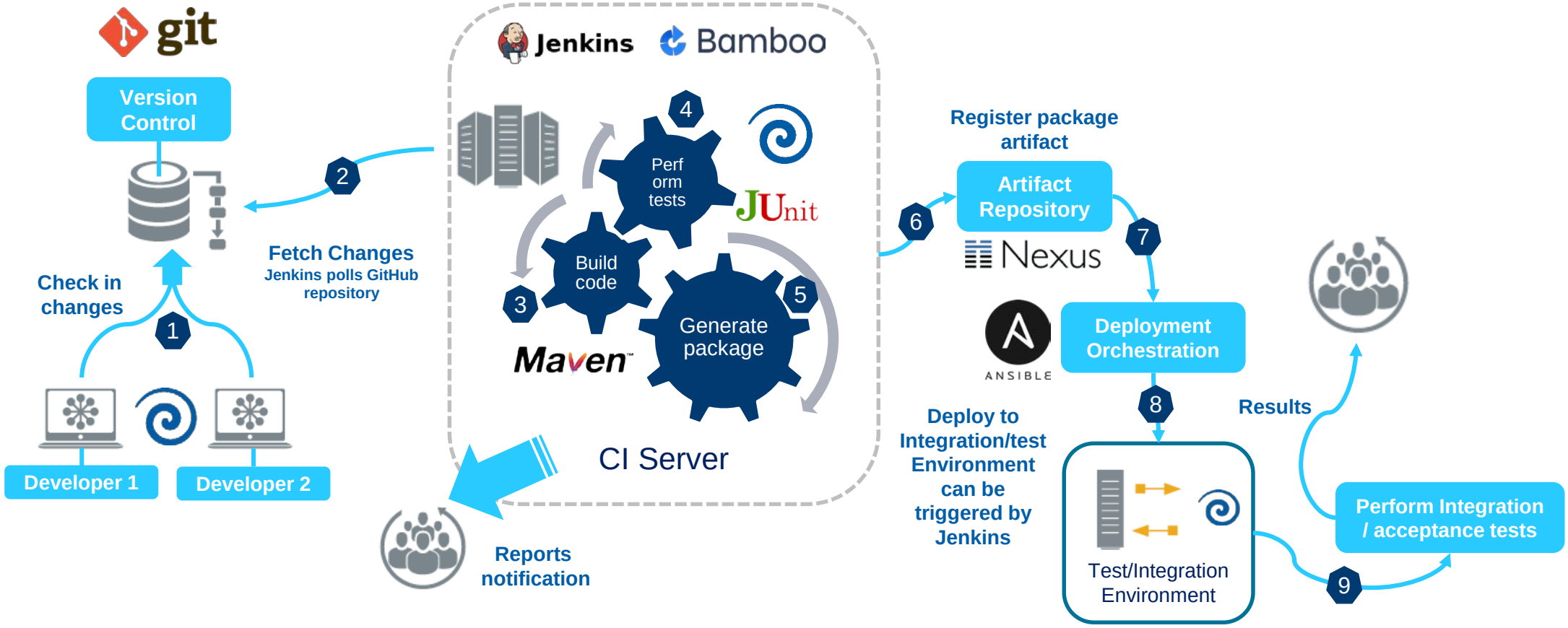
Note: Each phase shown here is the next step in an evolving DevOps practice

DevOps End-to-End Model

Continuous Integration

HITACHI
Inspire the Next

PLAN | CODE | BUILD | TEST | RELEASE | DEPLOY | OPERATE



HITACHI
Inspire the Next

Guidelines for Planning and Design

Recommendations

DevOps Guidelines for Pentaho Projects

Planning & Design

HITACHI
Inspire the Next

Planning and Design is the startup point of your workflow and the DevOps engine.

Here are some key points:

- **Collaborative Culture**
- **Educate**
- **Iterate**
- **Be Agile**

HITACHI
Inspire the Next

Challenges & Goals of DevOps

Solve Challenges with DevOps

HITACHI
Inspire the Next

Before the DevOps method, software development what done by separate teams that were not integrated, making the whole process slower and inefficient.

These are the main challenges which DevOps addresses:

- Dev is often unaware of QA and Ops roadblocks that prevent the program from working as anticipated
- QA and Ops are typically working across many features and have little context of the business purpose and value of the software
- Each group has opposing goals that can lead to inefficiency and finger pointing when something goes wrong

Achieve Goals with DevOps Implementation

HITACHI
Inspire the Next

- Improve deployment procedures and timing
- Decrease the amount of failures with software deliveries
- Rapidly adopt new business requirements
- Improve resolution procedures and mean time to recovery

HITACHI
Inspire the Next

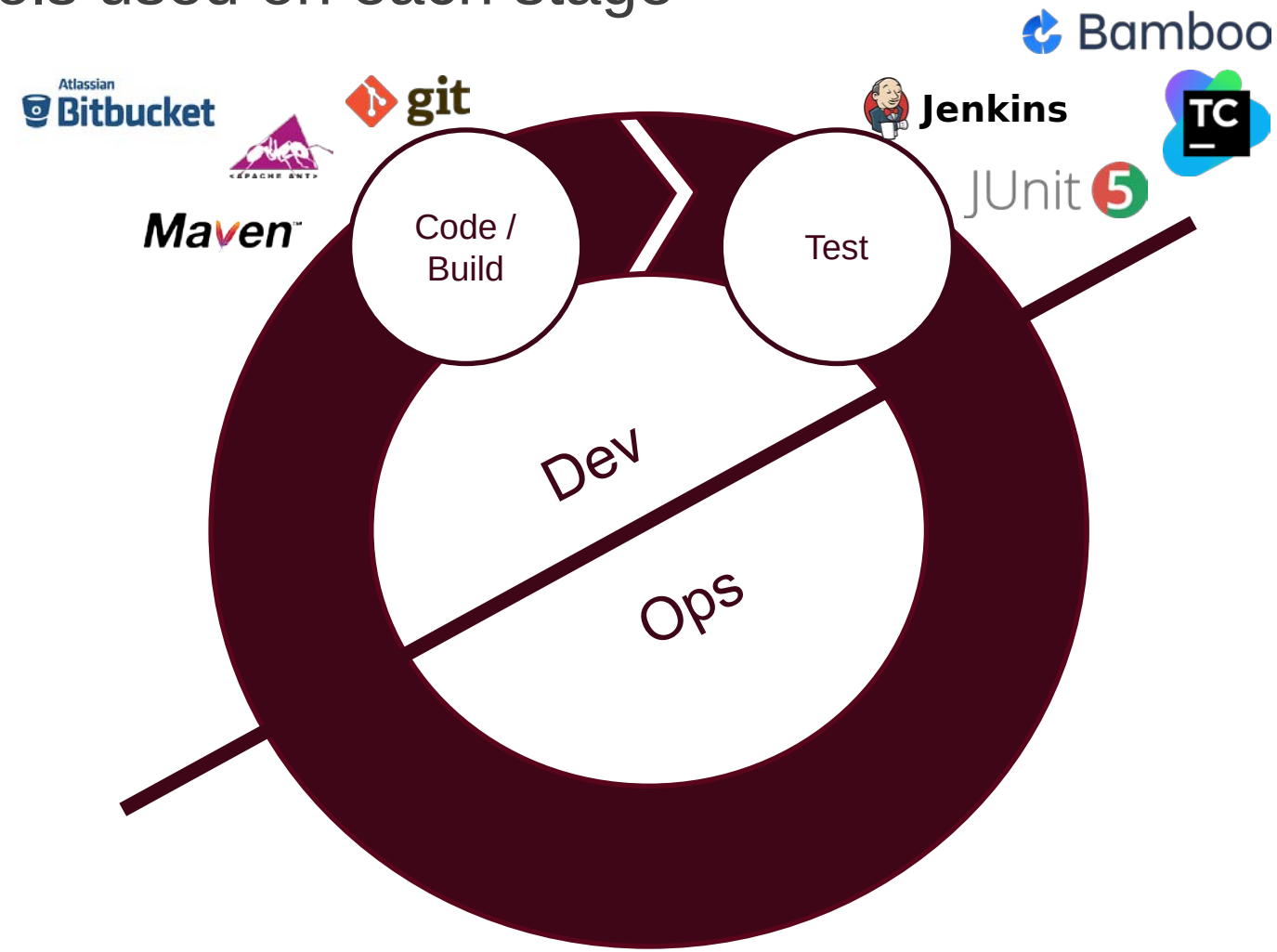
Focus on Continuous Integration (CI)

CI Adoption Illustrated

Continuous Integration Lifecycle

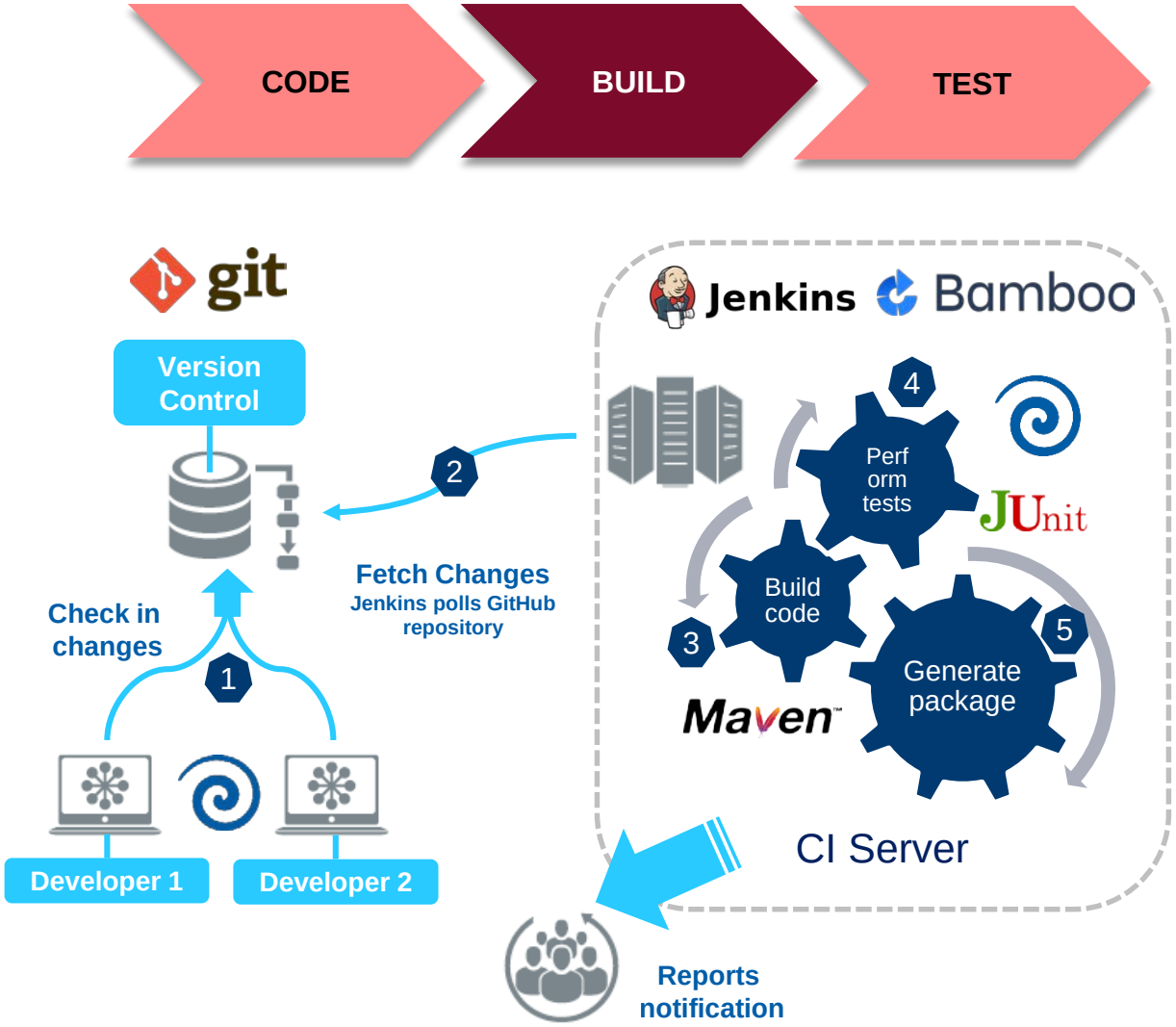
HITACHI
Inspire the Next

- Identify tools used on each stage



Continuous Integration End-to-End

HITACHI
Inspire the Next



HITACHI
Inspire the Next

Solution Repository

Store and Maintain Code

Solution Repository

HITACHI
Inspire the Next

The PDI Client (also known as Spoon) offers several different types of file storage:

- Supported
  - Pentaho EE Repository = UAT/Production environments
  - File-based solution = Development/Testing environments
- Unsupported
  - Database Repository
  - File Repository

Pentaho EE Repository

HITACHI
Inspire the Next

The Pentaho Repository stores transformations, jobs, and schedules in a central environment through the Pentaho Server; this is the recommended enterprise deployment approach. The main features of this approach are:

- Basic version control system to keep history, track changes and platform for collaboration
- Storage, handling, and maintenance of the solution
- Integration with Enterprise Security

Note: Since version control is handled with GIT, you can disable it for the Pentaho Repository through configuring the Pentaho Server

For more information on working with the Pentaho EE Repository, please visit https://help.pentaho.com/Documentation/8.1/Products/Data_Integration/Repositories

File-based Solution

HITACHI
Inspire the Next

This consists of storing the PDI solution code directly in the file system. An enterprise tool is required on top of the file system solution to handle centralization of the code, history tracking, and the collaborative approach.

This approach is very popular and useful for integration with VCS enterprise tools such as Git, SVN, BitBucket, etc.

A file-based solution is the preferred and recommended option to integrate Pentaho developments into the organization standards.

Version control system

HITACHI
Inspire the Next

A version control system in your Development and Test environments is recommended to:

- Provide a central location for policies and experience regarding Development systems.
- Keep track of changes if you want to use some advanced features, such as fork, tag, etc.
- Simplify work progress between team members.
- Integrate with the standards defined in your organization by IT.
- **Integrate the code in the CI automatic lifecycle.**

How to work with VCS

Recommendations

HITACHI
Inspire the Next

Here are the main recommendations to use VCS with Pentaho:

- Standardize the solution structure.
- Each developer has their own copy in the local environment as a clone/checkout of the main project.
- Use branching wisely, based on your active projects and your organization model.
- The release strategy is tied to the VCS workflow to maintain tracking.

HITACHI
Inspire the Next

Standardize the Solution

Standardize the Solution Structure

HITACHI
Inspire the Next

All projects require some kind of standard structure, due to the fact there are several moving parts inherent to it. Typically, the structure needs:

- A repository to store the solution code or resources
- An organized way to manage environment configuration, including settings that are environment-aware, such as usernames, passwords, database connections, etc.
- A simple method to deploy and start working on
- A simple method for testing and packaging the solution

The idea of our CI approach is to simplify the above while providing certain level of customization ideas referring to each use case.

Solution Folder Structure

Example

HITACHI
Inspire the Next

```
etl/bin                  ← General execution scripts, to place any execution/shell script needed in the solution
     /base               ← Main execution scripts of the framework
   /environments         ← Base folder for the environment configuration and setup
     /env1
       /runtime1
         /.kettle
         /.pentaho
       /runtime2
    /env2
       /runtime1
   /model                ← SQL scripts, Mondrian models, etc. should be placed here
   /repository           ← The ETL code (.ktr & .kjb)
   /src                  ← Additional custom Java code and location of the JUnit testing main class
extensions/              ← Java extensions not related to PDI (such as Hadoop input formats)
plugins/                 ← Additional PDI plugins or existing plugins with fixes
pom.xml                  ← Main pom file that guides packaging procedure
```

HITACHI
Inspire the Next

Unit Testing

Unit Testing

Introduction

HITACHI
Inspire the Next

Unit tests are a key stage in the continuous integration automation process, because their results indicate whether each unit works correctly and efficiently on its own.

The concept is to develop each test use case per each non-trivial function or part of the solution.

On ETL solutions with Pentaho, the fundamental principle is to **use PDI to test PDI**.

Advantages:

- Encourage changes
- Simplifies integration
- Works as code documentation
- Makes it easier to locate issues in the solution

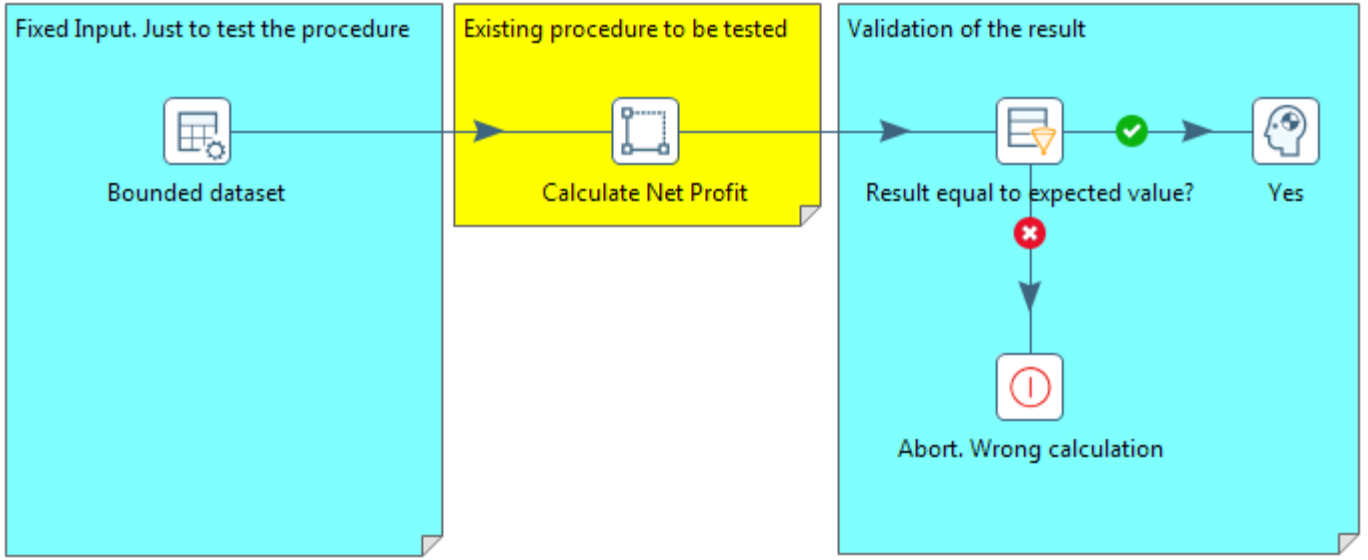
Use PDI to Test PDI

What does it mean?

HITACHI
Inspire the Next

Let’s assume that we have a centralized and common procedure to calculate “net profit” across the solution, which receives as parameters the different amounts required by the defined formula.

The idea is create a PDI transformation with a bounded data set as input and execute the procedure (in this particular case using a mapping transformation). At the end, the process is able to compare the procedure result with the expected value and decide if it works properly.



Use PDI to Test PDI

Recommendations and Tips

HITACHI
Inspire the Next

- Unit tests are an important part in the success of the project. Therefor it needs to be included as part of the efforts when developers are creating solution functionalities
- Testing procedures can be a transformation or a job that groups together other jobs and transformations depending on the complexity and objective of the test. In case of Jobs, the entire job needs to executed successfully for the unit test to be successful
- Evaluate results of each single unit test to decide if each piece of code is suitable for use

HITACHI
Inspire the Next

Code Promotion

Going towards automatic deployments

Code Promotion

When your project moves through its lifecyle, the deployment process will make sure all necessary project artefacts become available on the required Pentaho Server machines.

Code Promotion

HITACHI
Inspire the Next

The PDI toolkit offers multiple functionalities that can support and automate your deployment process.

- Usage of **import_export** script provided as part of the Pentaho Server to export/import from/to Pentaho EE Repository
- Using the Pentaho Server's File Management REST APIs. A set of jobs/transformation can be created to upload the solution code to the Pentaho Server repository

Code Promotion

Recommendations and tips

HITACHI
Inspire the Next

- Do not perform manual intervention to the exported packages
- Configuration such as database connections, environment variables, etc. should be done on each environment without changing the code on the fly to adapt to it
- **Database connections management**
  - Once you deploy the project jobs and transformations to the Pentaho Repository, the shared.xml file will no longer be used to hold the database connections. Once deployed, the connection information will be stored in the Pentaho Repository
  - The deployment process uses the connection information that is stored within the transformation and job's xml definition. As a consequence, you would need to update the connection manually or use the REST API to do so (automated approach)

HITACHI
Inspire the Next

Handling Continuous Integration

Continuous Integration Server

HITACHI
Inspire the Next

This is key for the automation of tasks in the DevOps path.

The CI Server handles the process of getting the source code of the solution, performs unit tests, and finally creates the package release based on the results.

Advantages:

- Immediate detection and solution of issues, avoiding last-minute chaos
- Immediate execution of the unit tests getting reports every time
- Indicators of project/solution quality

The main recommendation is to make use of a dedicated tool to handle such tasks, like Jenkins, TeamCity, BuildBot, Bamboo, etc.

References

HITACHI
Inspire the Next

- Pentaho 8.1 Product Page
  https://www.pentaho.com/product/version-8-1
- Pentaho 8.1 Documentation
  https://help.pentaho.com/Documentation/8.1
- Support Portal
  https://support.pentaho.com
- Best Practices
  https://support.pentaho.com/hc/en-us/categories/200888603-Best-Practices
- Enterprise Architecture Group
  http://www.pentaho.com/service/enterprise-support
- Professional Services
  http://www.pentaho.com/service/consulting-services

Questions?