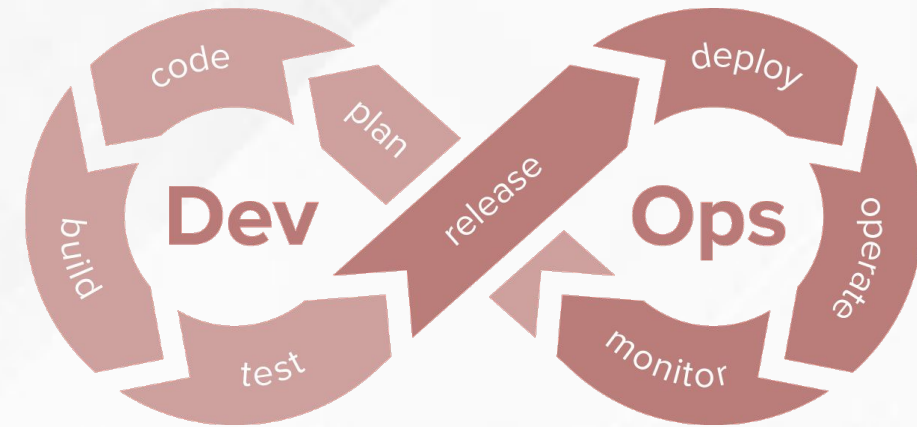




Unit-9

DevOps

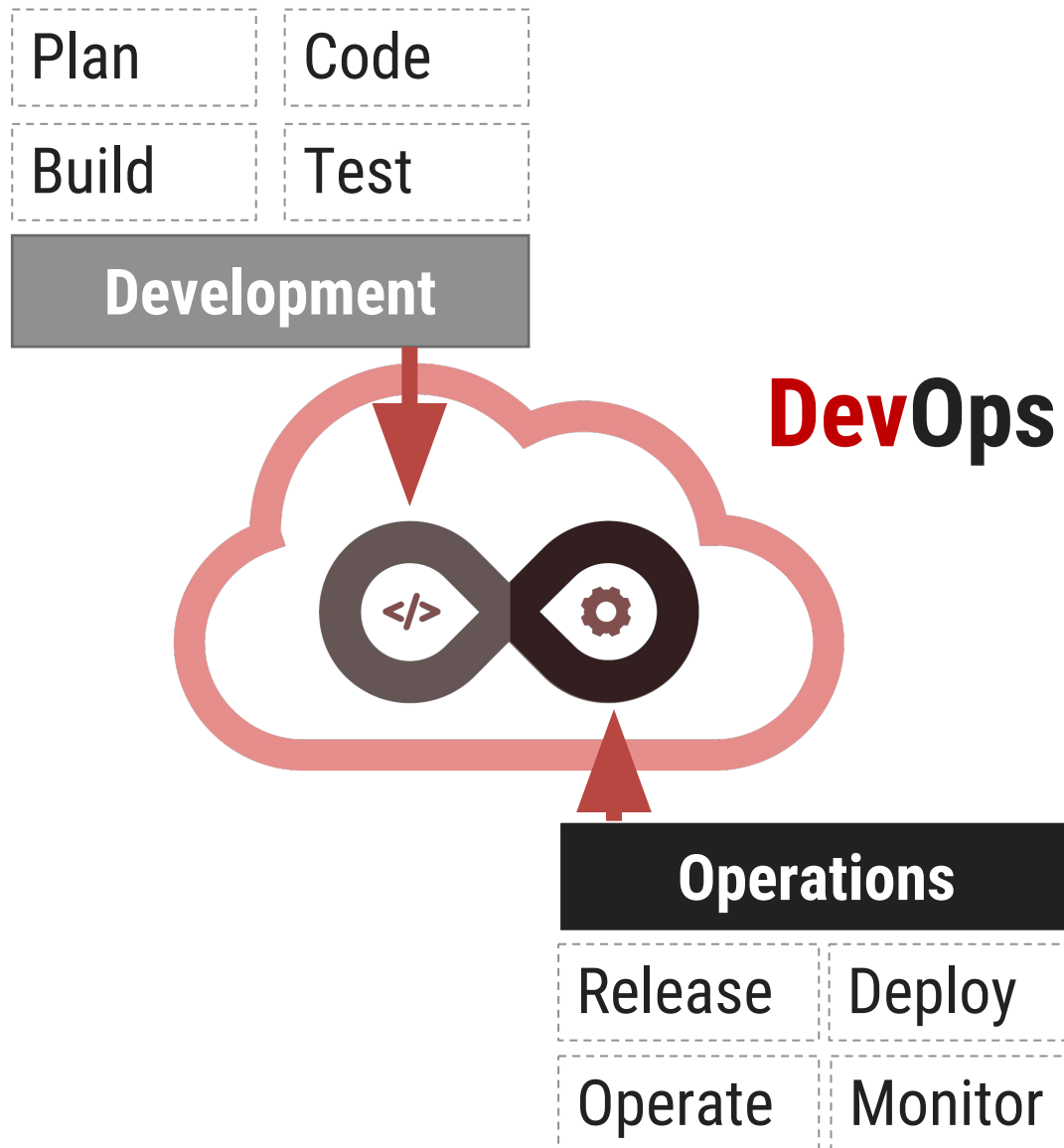




Outline

- What is DevOps?
- DevOps Importance and Benefits
- DevOps Principles and Practices
- 7 C's of DevOps Lifecycle for Business Agility
- DevOps and Continuous Testing
- How to Choose Right DevOps Tools
- Challenges with DevOps Implementation
- Must Do Things for DevOps
- Mapping My App to DevOps –Assessment
- Definition, Implementation, Measure and Feedback

What is DevOps?



- The word **DevOps** is the mixture of two words **Dev** (it represents **Software Development**) and **Ops** (It represents **IT Operations** e.g. Application and infrastructure planning, Automation implementation, incident response, and incident management, etc.)
- DevOps is a **group of practices & principals** that perfectly blends software development (Dev) and IT operations (Ops).
- The main **aim** of **DevOps** is to provide **continuous & high-quality delivery** of the software by making **shortening SDLC** (Software Development Life Cycle)
- DevOps **bridges the gap** between the **operation team & development team**, which generally used to work in isolation with another.

DevOps Importance and Benefits

1 Technical benefits

Continuous software delivery

Less complexity to manage

Faster resolution of problems

Maximizes Efficiency with Automation

2 Cultural benefits

Happier, more productive teams

Higher employee engagement

Greater professional development opportunities

3 Business benefits

Faster delivery of features

More stable operating environments

Improved communication and collaboration

More time to innovate (rather than fix/maintain)

Optimizes the Entire Business

Gets the focus on What Matters Most: People

DevOp Practices

- ❑ Process automation
- ❑ Continuous Integration (CI)
- ❑ Continuous Delivery (CD)
- ❑ Continuous Deployment
- ❑ Infrastructure as code
- ❑ Microservices
- ❑ Configuration management and many more.

DevOps Principles

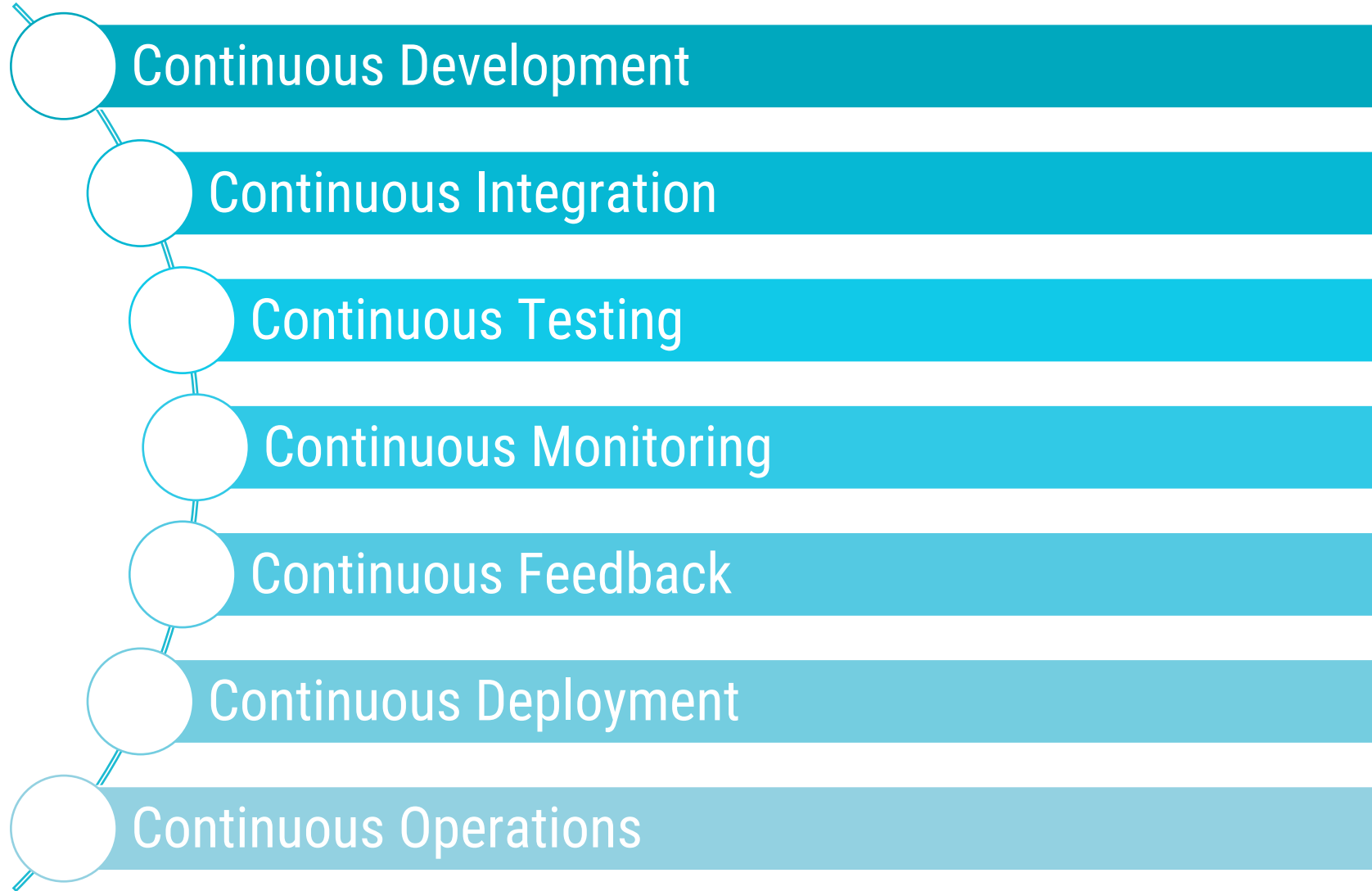
DevOps principles guide how to organize a DevOps environment.

- ❑ Incremental Releases
- ❑ Automation
- ❑ DevOps Pipeline
- ❑ Continuous Integration
- ❑ Continuous Delivery
- ❑ Continuous Monitoring
- ❑ Feedback Sharing
- ❑ Version Control
- ❑ Collaboration

7 C's of DevOps Lifecycle for Business Agility

- ❑ An agile relationship between development & operations are defined by DevOps.
- ❑ DevOps is a set of rules and procedures followed by the development team and operational engineers together from beginning to the final stage of the product.
- ❑ To understand the DevOps lifecycle phases is essentials to Learn DevOps.
- ❑ The DevOps Life cycle is described by 7'C
- ❑ The 7'C approach to DevOps requires an efficient integration of processes and tools to automate the seamless information exchange and execution.
- ❑ The 7'C approach to DevOps also requires efficient collaboration between all stakeholders empowering them to contribute more.
- ❑ The 7'C approach removes all deviation of information and execution.

7 C's of DevOps Lifecycle for Business Agility



7 C's of DevOps Lifecycle for Business Agility Cont.

1 Continuous Development

- **Planning** and **Coding** of the software are involved in this initial phase.
- During the planning phase, the **vision of the project** and **scope** is decided.
- After the planning, developers used to **begin coding** the application.
- For the planning no DevOps tools available, but many tools available to maintain the code.

2 Continuous Integration

- This is the **main stage** of the **DevOps lifecycle**.
- Developers should **commit** the **changes** to the **source code** more **frequently** as a software development practice. This should be on daily and weekly basis.
- Next step is to **build the every commit**, this is **used** for **early detection of problems** if they are present. Building the code involves not the compilation, but also includes **unit testing, integration testing, code review, and packaging**.
- The **new functionality** is **continuously integrated** with the **existing code**. So, there is continuous development of software. The newly written code needs to be integrated continuously and smoothly with the systems to reflect changes to the end-users.
- **Jenkins** is a famous **tool** used for **continuous integration**. Once the change in the Git repository is ready, then Jenkins fetches the committed code and prepares a build of that code. Then this **build** is **given to the test server** or the **production server**.

7 C's of DevOps Lifecycle for Business Agility Cont.

3 Continuous Testing

- In this phase, software continuously testing for bugs. The tools like TestNG, JUnit, Selenium etc are used for the constant testing.
- This phase, where the developed software is continuously testing for bugs. For constant testing, automation testing tools such as TestNG, JUnit, Selenium, etc are used.
- These testing tools allow QAs to test multiple code-bases in parallel to ensure that there is no flaw in the functionality. Docker Containers can be used for simulating the test environment.
- Replacing manually testing to Automation testing saves a lot of time and effort for executing the tests. The code is continuously integrated with the existing code, after the testing.

4 Continuous Monitoring

- Monitoring is the phase where important information about the use of software is recorded and processed to identify the problem areas. Monitoring is integrated with the operational capabilities of the software application. The system errors such as server not reachable, low memory, etc are resolved in this phase. It maintains the security and availability of the service.

7 C's of DevOps Lifecycle for Business Agility Cont.

5 Continuous Feedback

- By **analyzing** the **results** from the operations of the software, one can **improve the application development**.
- This can be achieved by **constant feedback** among the **development and the operations** of the next version of the current software applications.

6 Continuous Deployment

- This phase **deploys** the **code** to the **production servers**.
- This is essential to **ensure** that the **code** is **correctly working** on the **production** servers and **real environments**. The new code should be deployed continuously.
- The new code is deployed continuously, and **configuration management tools** play an essential role in executing tasks frequently and quickly. Here are some popular **tools** which are used in this phase, such as **Chef, Puppet, Ansible, and SaltStack**.

7 Continuous Operations

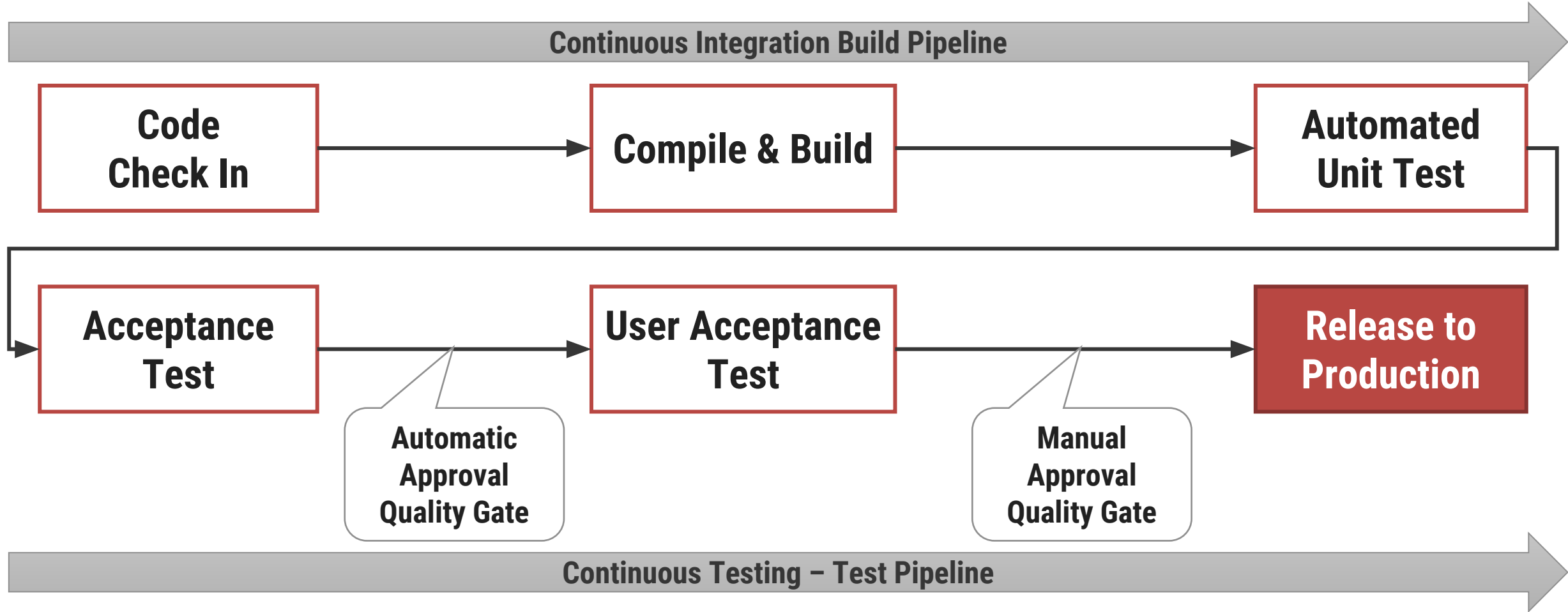
- All the operations in DevOps are based on the **continuity with complete automation** of the **release** process. It allow the organization to **accelerate the overall time to market**.
- It is clear that **continuity is the critical factor** in the **DevOps** in **removing steps that often distract the development**, take it longer to **detect issues** and produce a better version of the product after several months. With DevOps, we can make any software product more efficient and increase the overall count of interested customers in your product.

DevOps and Continuous Testing

- ❑ In DevOps **Continuous testing** is the most important process of the **continuous delivery pipeline** along with Continuous Integration in the same pipeline.
- ❑ The continuous testing in DevOps **contains** the **execution of automated tests repeatedly & continuously** against the codebase **in** the various **deployment environments**.
- ❑ Continuous testing contains **unit tests, static code analysis, security code analysis, integration tests & load and performance tests**. These sets of tests run in an automated continuous testing pipeline.
- ❑ To **test every line of code, every time at different stages** is almost **impossible** to do each time **manually** when a line of code is **updated** into version control. That's where continuous testing comes into the picture.

DevOps and Continuous Testing Cont.

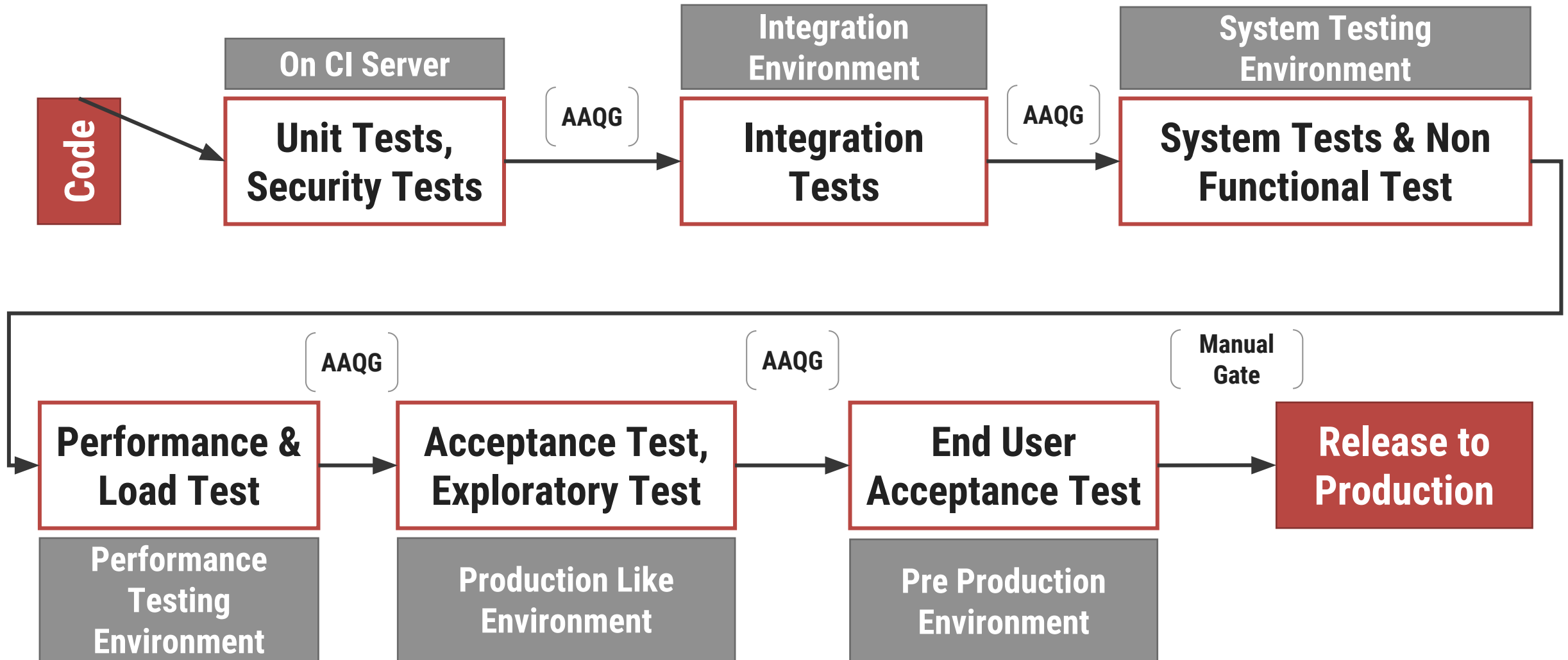
Continuous Delivery (CD) | Automated Deployment Pipeline



DevOps and Continuous Testing Cont.

AAQG = Automated Approval Quality Gate

Continuous Testing (CT)



DevOps and Continuous Testing (CT) Cont.

- As per the continuous testing (CT) **diagram**, **unit tests** are carried out on the **CI server**. It **tests** each unit of the **system** in an **isolation**.
- **Integration tests** are carried out on the **Integration environment** to **verify** the **components integrated**.
- **System tests** are carried out in the **system testing environment** where the system the integrated **components and interfaces** are **tested** through **system-level scenarios** in a system testing environment.
- **CT** progressively **becomes harder and longer** with the progression **towards** the **production environment** as environment complexity advances.
- The **test cases** are **required** to **update each time** at different phases. The **automated scripts** required to **update** because the code becomes more mature & progresses to a higher level of environment ill it gets into production.
- The **time to execute** the **tests increases** with the testing progress. E.g. the unit test may take very little time to execute, the integration tests, system tests or load tests may take a few hours to execute or may take a few days for execution.
- The **CT mainly running the automated test cases**. It also involves **certain manual tests** and **approval gates**, where few tests are executed manually, before pushing into the production.

Benefits of Continuous Testing in DevOps

- ❑ Early identification of critical bugs
- ❑ Smooth collaboration among developers, QA and Operations team
- ❑ At each stage of SDLC, it helps to assess the quality of software developed
- ❑ Leads to improvement in code quality by driving faster test results which
- ❑ Repeated testing ensures minimal failure rate for new releases
- ❑ Faster time to market with a viable product and continuous feedback mechanism

Choose Right DevOps Tools

- ❑ At the **time of adopting a DevOps approach** for building and operating software systems, one should **rely on modern tools** for each & every aspect of build, release, and operations activities.
- ❑ The **first step** to choose **Right DevOps Tools** is one need to **think more broadly** about what **company need**.
- ❑ Following are the guideline for selecting the right DevOps tools for any organization
- ❑ **Select tools which facilitate collaboration**
 - ❑ One should **give** proper **importance** to **collaboration** as a key selection aspect of tools.
 - ❑ Verify the **main purpose of the tool** to find **collaboration opportunities**.
 - ❑ **Check**, "How does the use of selected **tool helps in collaboration?**"
- ❑ **Select DevOps tools which provides APIs**
 - ❑ The company can get **mixture of new capabilities** from **multiple API-driven tools**.

Choose Right DevOps Tools Cont.

- ❑ Select the tools which can **store configuration in version control**
 - ❑ Select tools that provides **configuration to version control system** available in market.
- ❑ DevOps tools should **encourages learning**
 - ❑ Few tools for **DevOps** are so much involved and complicated for the **people** new to them; one should not expect everyone to **understand** or **adopt complex new tools immediately**. If management selects a tool that is **too tricky to use**, some people may **start opposing**, especially if management don't provide training.
 - ❑ Company should **select tools which is easy to use**.
- ❑ Bring people with company into the DevOps journey.
- ❑ **Avoid special production-only tools**
 - ❑ Select tool that enhance inter-team communications
 - ❑ Consider the entire organization as a system you are building.
 - ❑ **Select separate tools** for the **separate teams**.
- ❑ One should deploy shared tools for the collaborative teams.

Challenges with DevOps Implementation

❑ Lack of a **Standard Definition for DevOps**

- ❑ Many companies are still not have the **actual understanding of DevOps** e.g. the purpose behind it, or for that matter, the benefits of using it.
- ❑ Many people believe that DevOps overrules existing methodologies like Scrum which are practiced in most industries.
- ❑ Actually, DevOps helps to improve the execution of available methodologies.
- ❑ The implementation of DevOps becomes risky because of the lack of proper understanding.

❑ Lack of Vision

- ❑ It is difficult to have a vision without proper understanding and definition of the problem or solution. Because of it analysts may stuck to ambiguous circle.
- ❑ It is really more difficult to break this ambiguous circle & to become more receptive to the newer methods like DevOps for betterment – all at the same time!

❑ Shortage of Tool Knowledge

❑ Choice of Tools

❑ Lack of Tool Integration

❑ Cultural Challenges

❑ Isolated Teams

❑ Risk Analysis

The background features a light gray grid of squares, some of which are filled with a fine dot pattern. A large red semi-circle is at the top, and a red horizontal bar spans the middle. A central white hexagon with a dashed red border contains the text. In the top left, there is a small orange and yellow graphic resembling a comet or a cluster of dots. In the bottom right, there is a complex geometric pattern of overlapping triangles and lines in shades of gray.

***Thank
You***