

# DevOps



**Caltech**

Center for Technology &  
Management Education

## Post Graduate Program in DevOps



# Introduction to DevOps

# Learning Objectives

By the end of this lesson, you will be able to:

- State the principles of DevOps
- Explain the challenges in the traditional approach
- Describe how DevOps helps in overcoming challenges faced in traditional approach
- Describe the DevOps tools
- Explain DevSecOps and its importance



# DevOps Overview and Principles



# DevOps

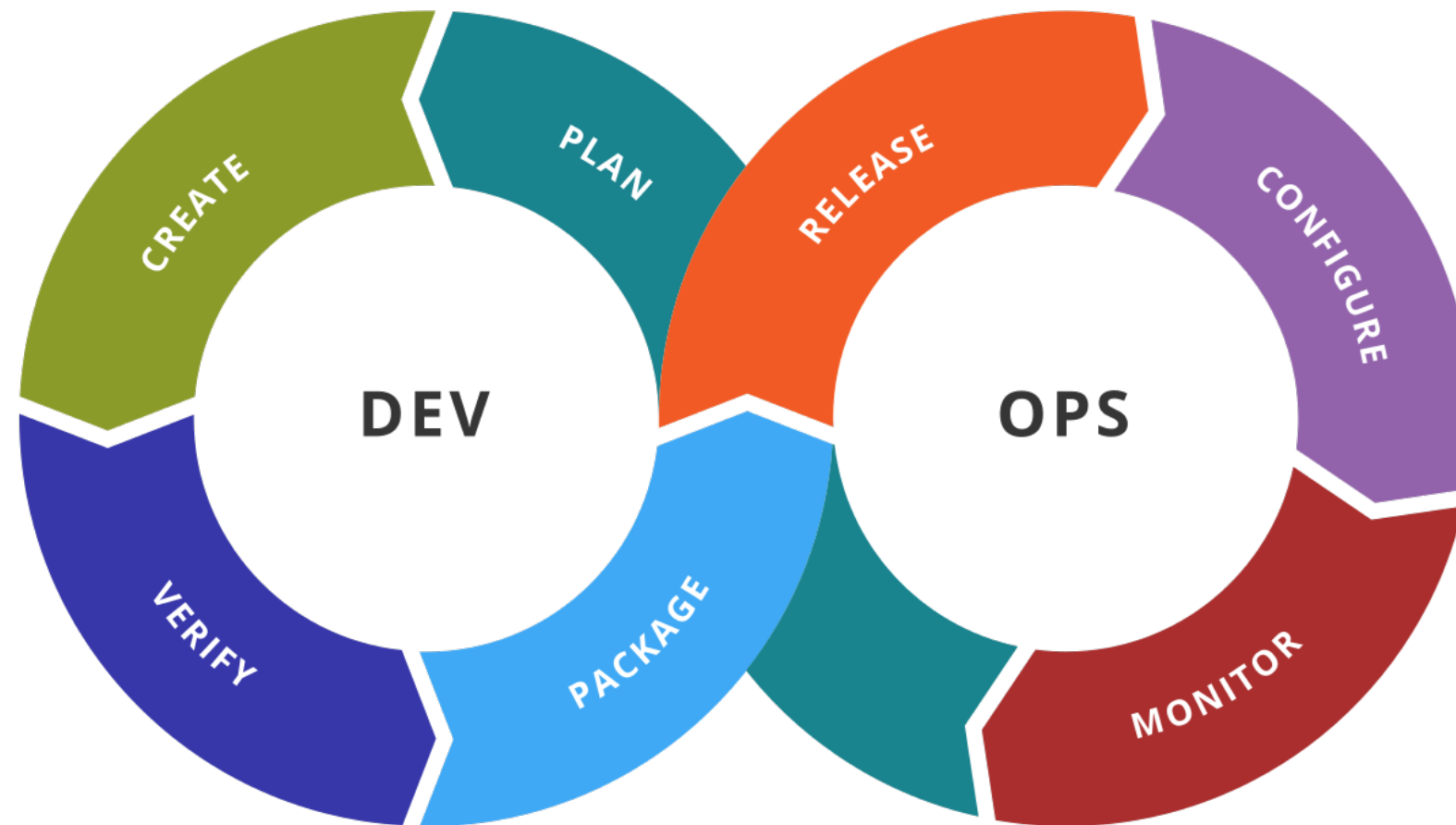
DevOps is short for **Development** and **Operations**. It concentrates on collaboration between developers and other parties involved in building, deploying, operating, and maintaining software systems.

# History of DevOps

- Patrick Debois, a Belgian consultant, project manager, and agile practitioner is one among the initiators of DevOps.
- A presentation on “10+ Deploys per Day: Dev and Ops Cooperation at Flickr” helped in bring out the ideas for DevOps and resolve the conflict of “ It’s not my code, it’s your machines! ”
- DevOps blends lean thinking with agile philosophy.

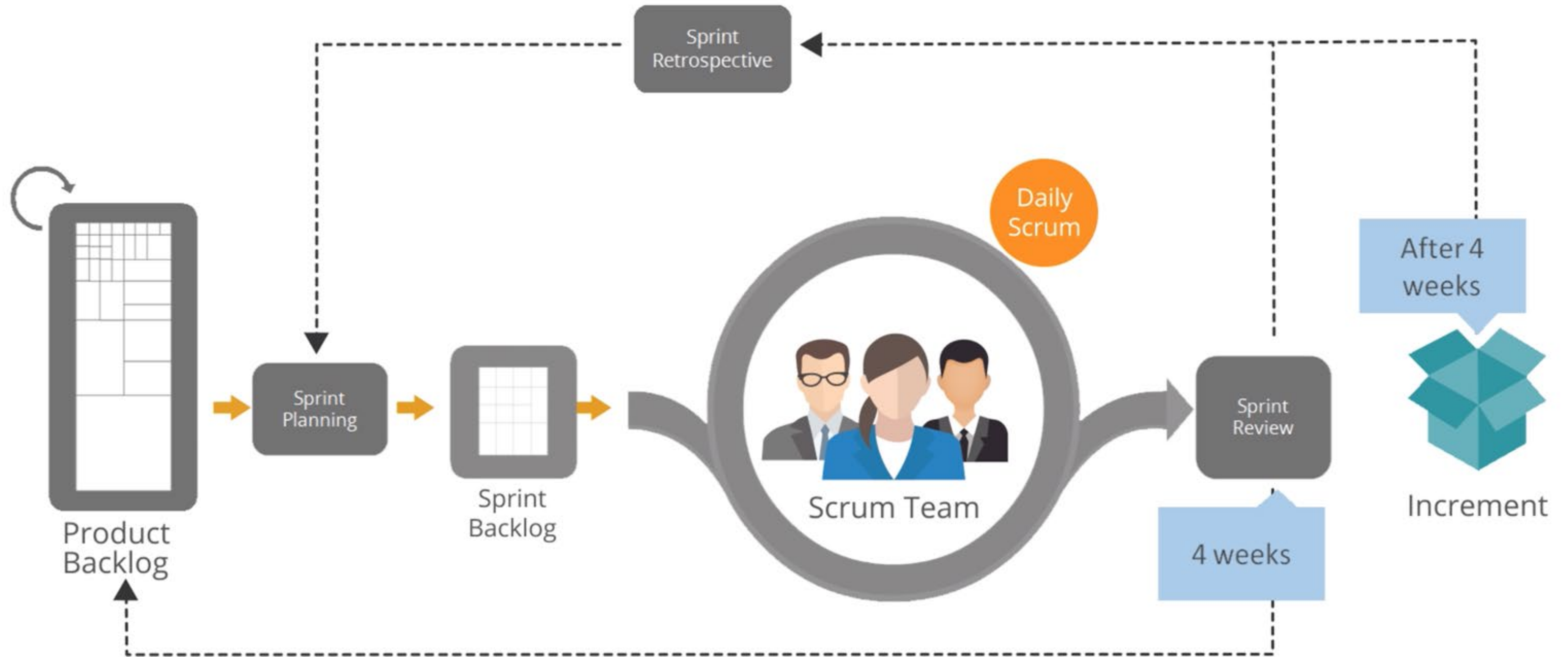


# Overview of DevOps



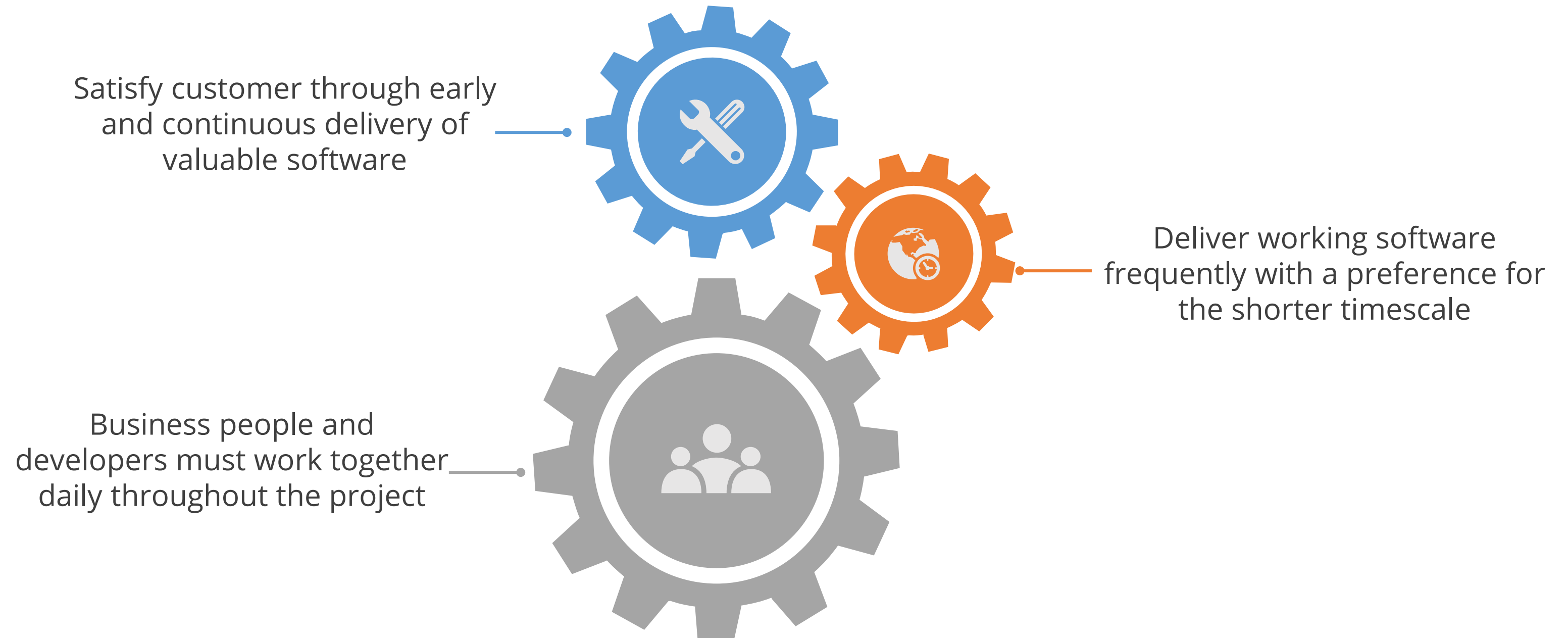
- DevOps is an agile relationship between development and IT operations
- DevOps is the abbreviation for **D**evelopment and **O**perations
- The Development includes Plan, Create, Verify, and Package
- The Operations include Release, Configure, and Monitor

# Agile Example: Scrum

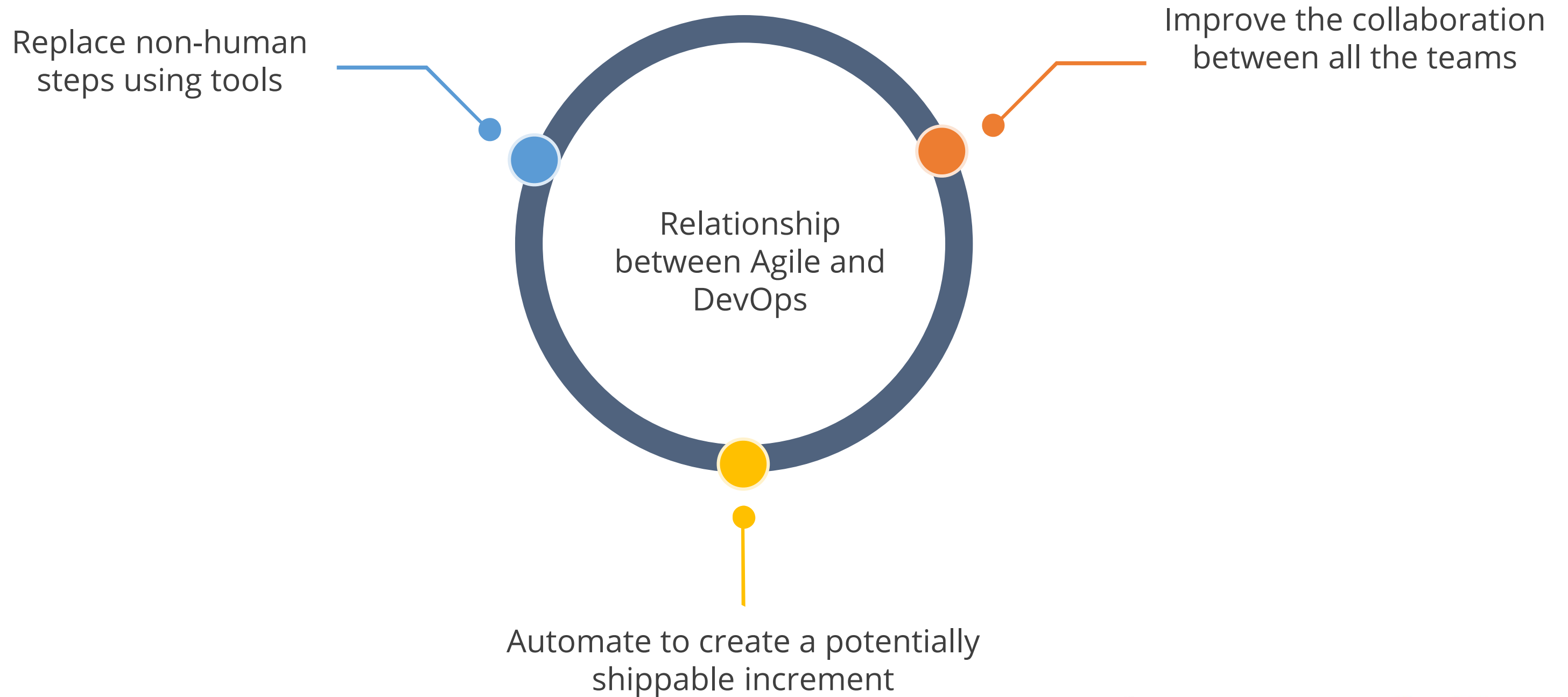




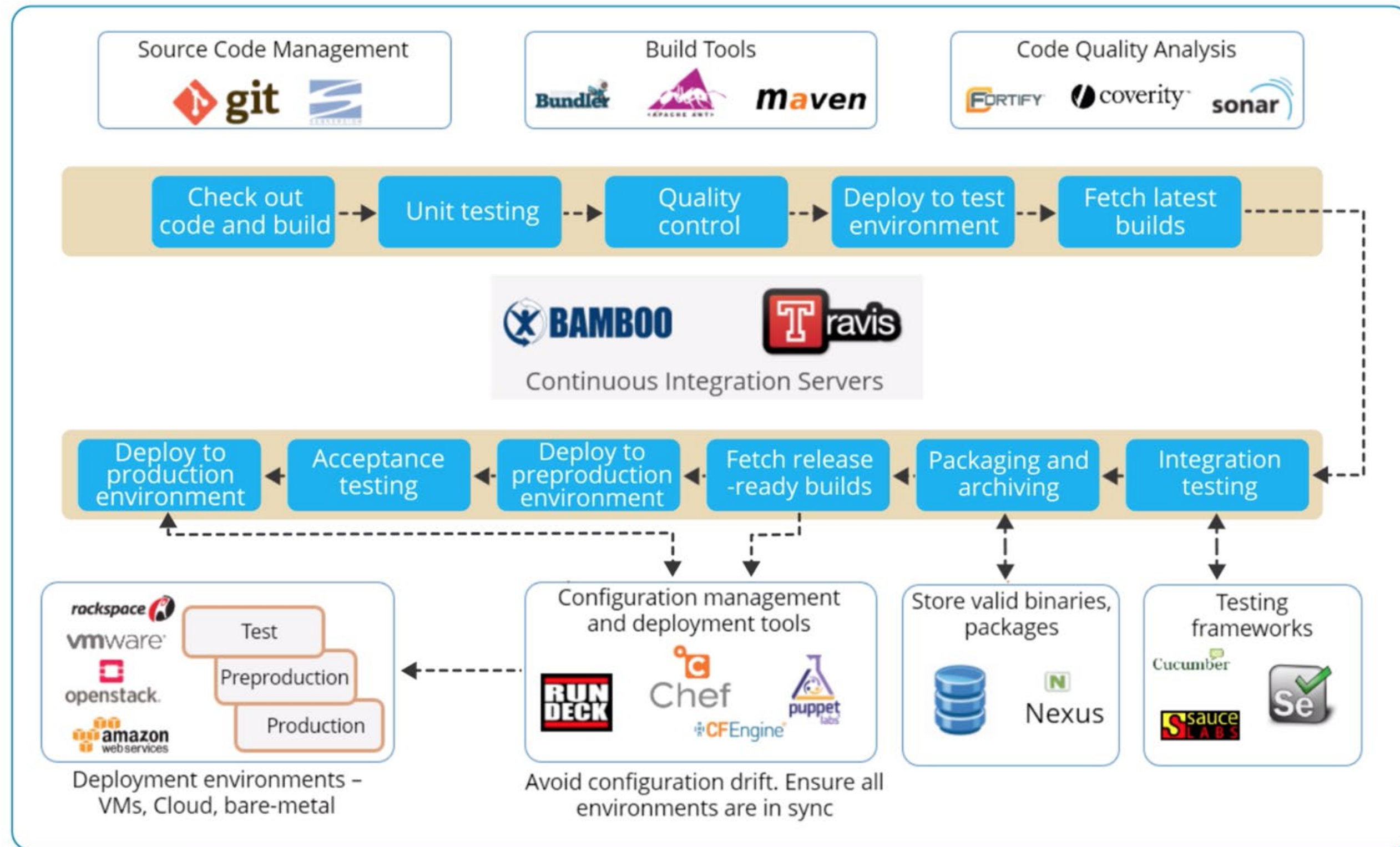
# Relationship Between Agile and DevOps



# Relationship Between Agile and DevOps



# Agile and DevOps Example



# DevOps Toolchains



Monitoring  
Performance



Releasing into  
Production



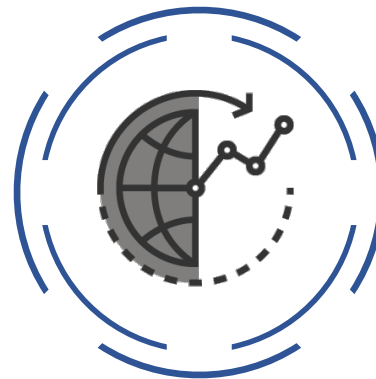
Building  
Applications



Code Development  
and Unit Testing



Configuration  
Management



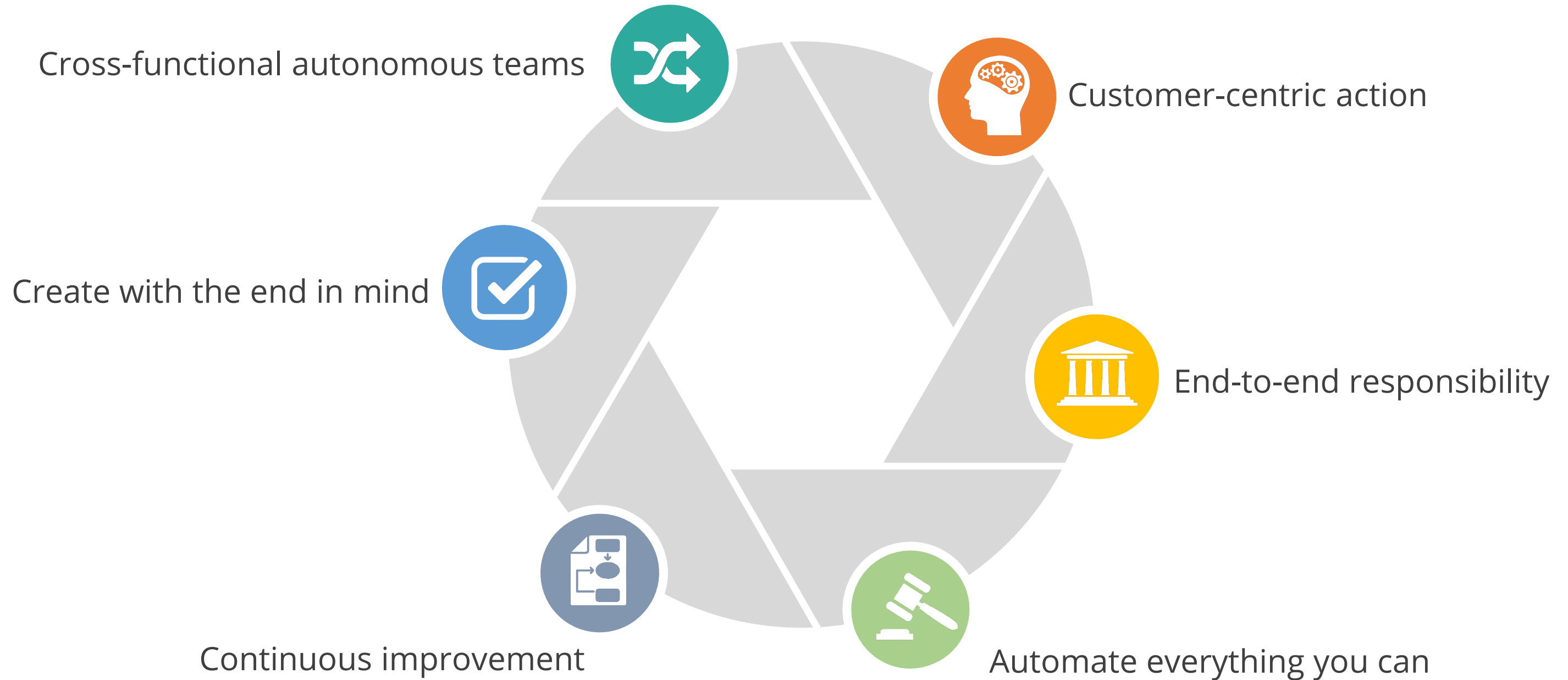
Integration and  
Performance Testing



Packing the  
Application



# DASA DevOps Principles



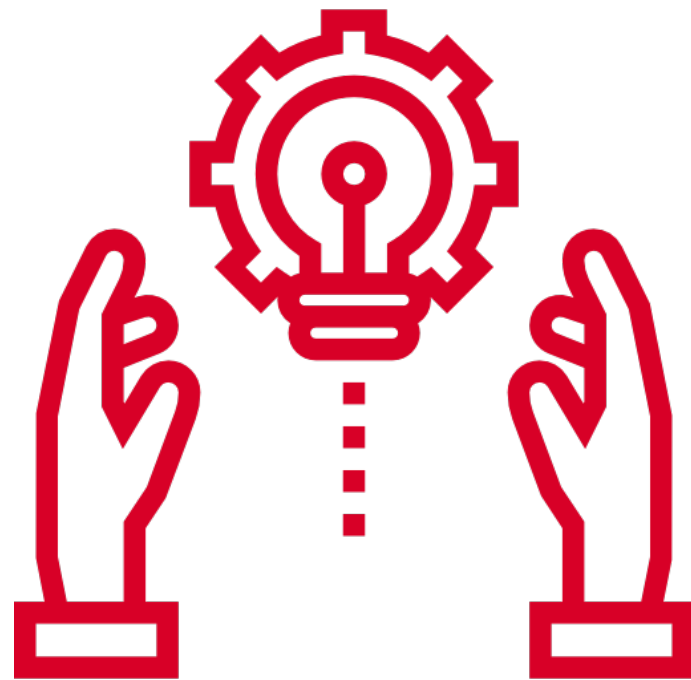
## Case Study



Amazon switched to DevOps at the development phase of their web services, popularly known as AWS.

# Challenges in the Traditional Approach

# Waterfall Method



Most of the development teams use waterfall method, which is time-consuming because of the larger size of the developer team, testers, and the code involved.



# Productivity



Codes that are large and bundled into release will result in jammed production and lower the productivity.

# Difficult to Achieve Goal



Less investment on resources and constant work make it difficult for the developers to achieve goal or an outcome.

# Investment in Schedule Planning Systems



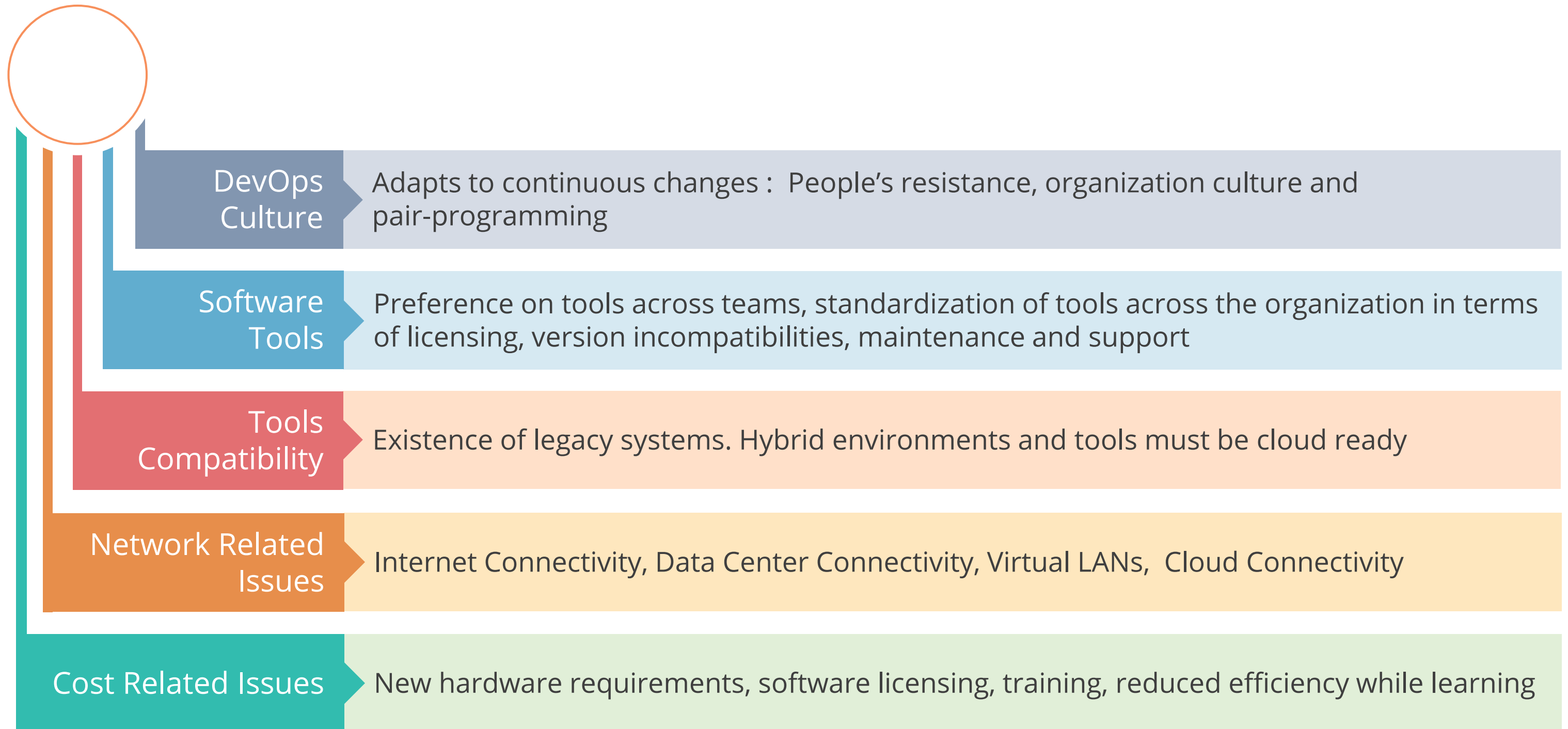
More money is invested in schedule planning systems which are sensitive and inaccurate. As a result, it consumes more time to manage the systems.

# Constantly Changing Challenges



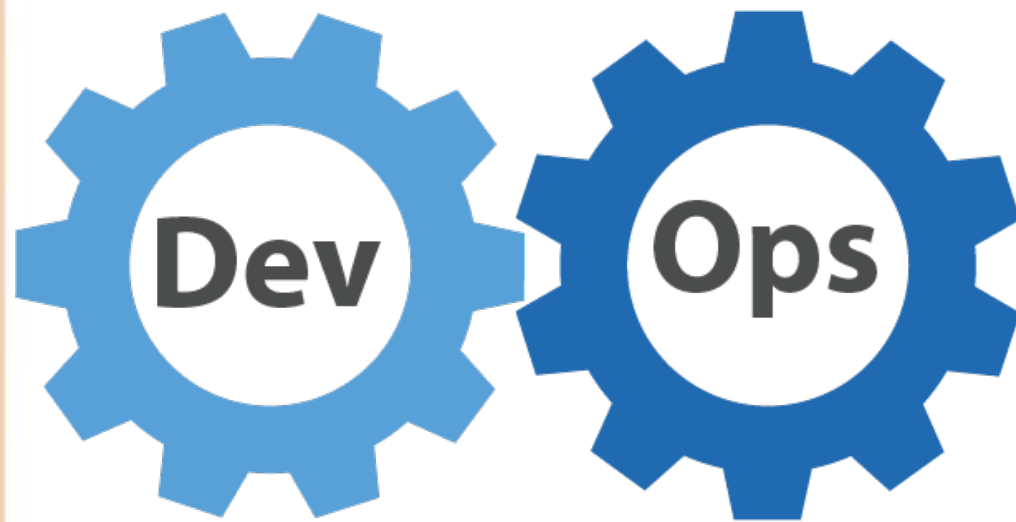


# Types of Constantly Changing Challenges



# DevOps Approach to the Challenges

# DevOps Way to Overcome Challenges



Reduction in the code size delivery results in increased productivity

Smaller batch sizes, dedicated teams, and automated processes make scheduling simpler to operate

Batch sizes are divided into small cells. Each cell gathers its own data, reducing the size of the reports

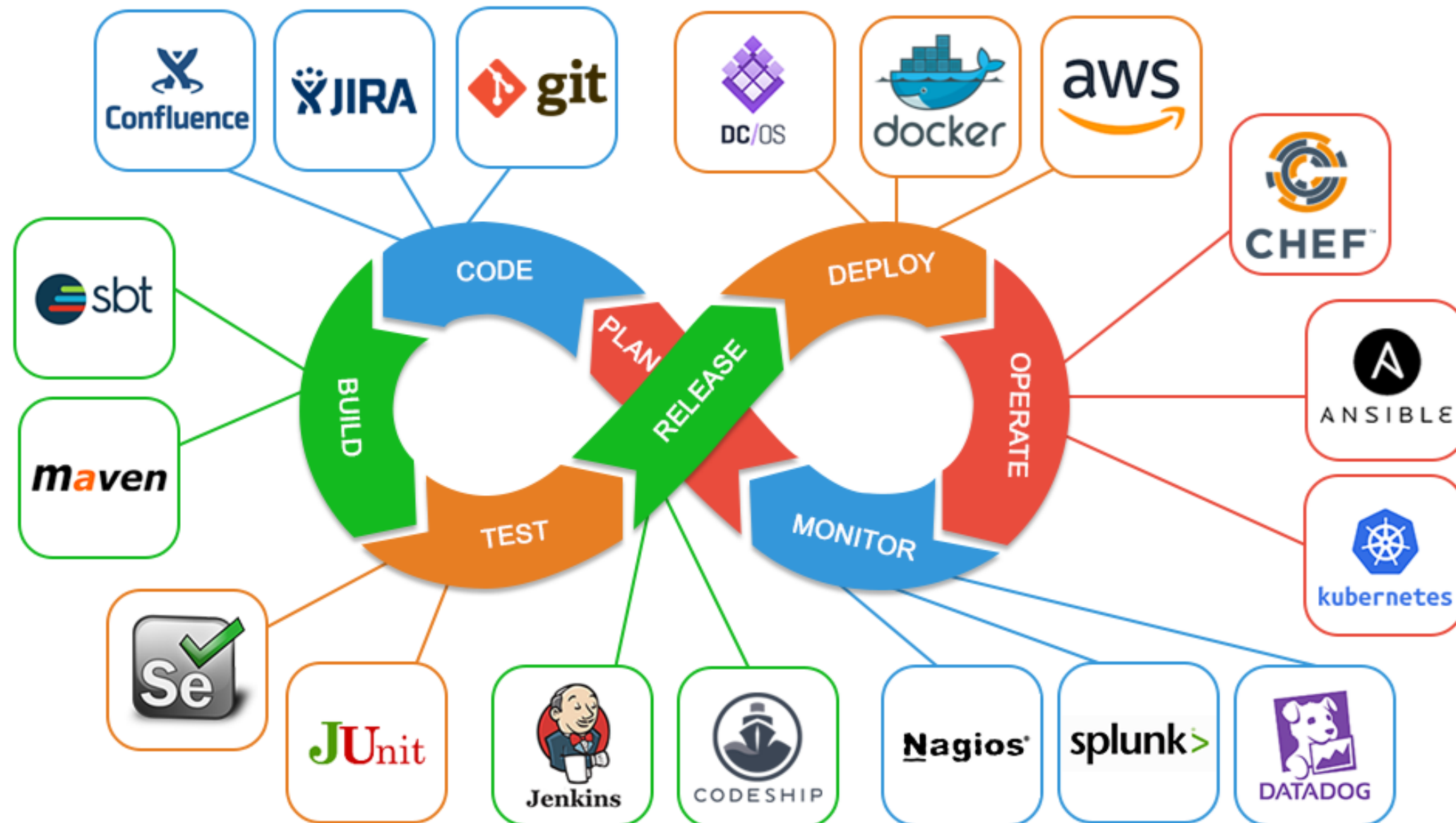
Identifies productive and loss areas in the process. As a result, an organization can focus more on their goals

# Overview of DevOps Tools



# DevOps Tools

To implement DevOps and work within the DevOps setup, the various tools required are:



# Best Practices for DevOps



# DevOps and Cloud Computing Training

Proper training in DevOps and cloud computing will lead to better understanding and help resolve challenges in DevOps environment.



# Security

Security should be part of the automated testing. It should be built into continuous integration and deployment processes during the migration to cloud-based platform.



# Proper DevOps Tools Selection

The Applications should be deployable on different clouds. In this way, you can pick and choose the best public or private cloud for the job.

Open source and licensed tools

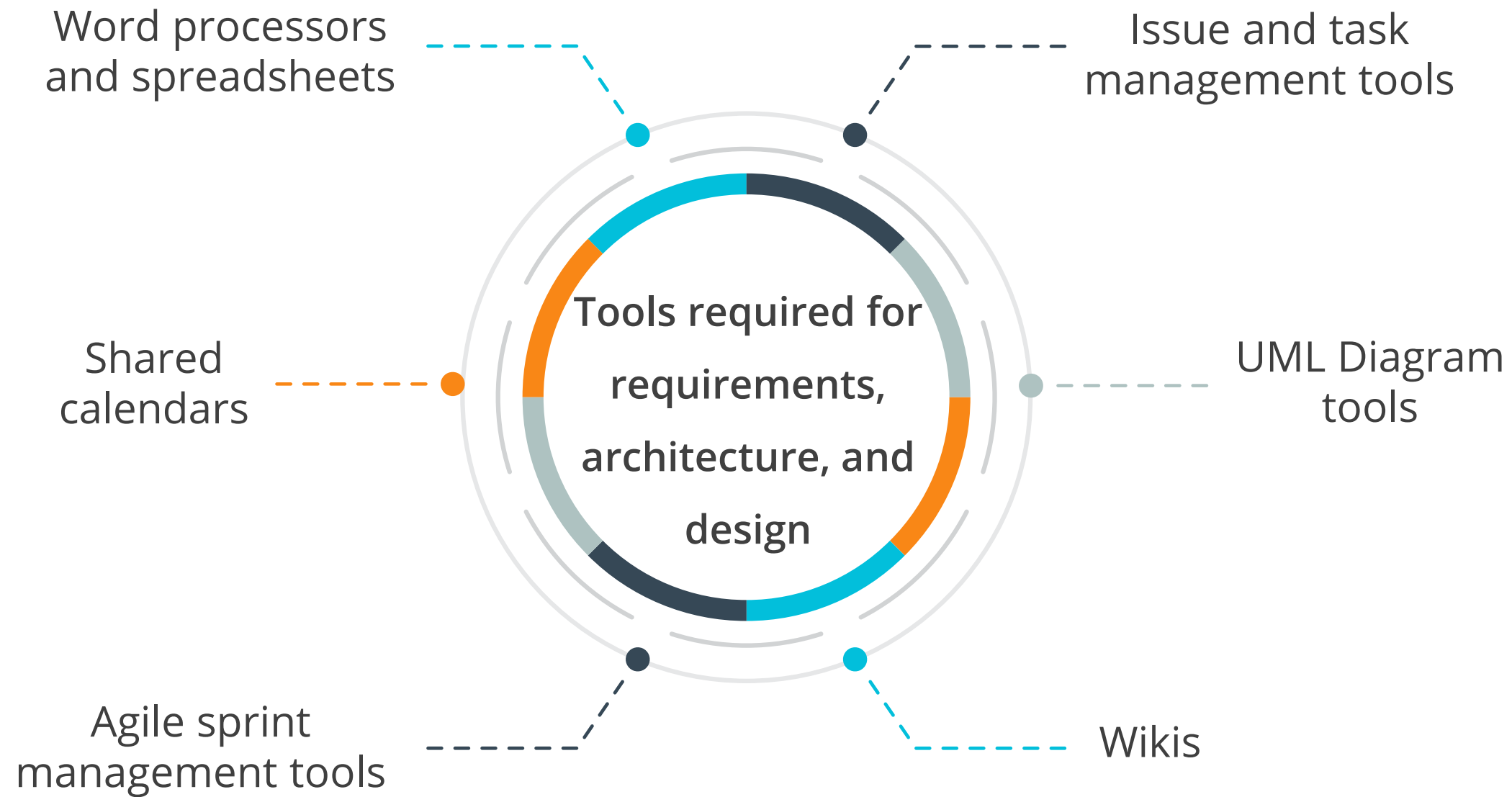
Tools that work together

Tools for different SDLC phases

Tools that perform similar functions

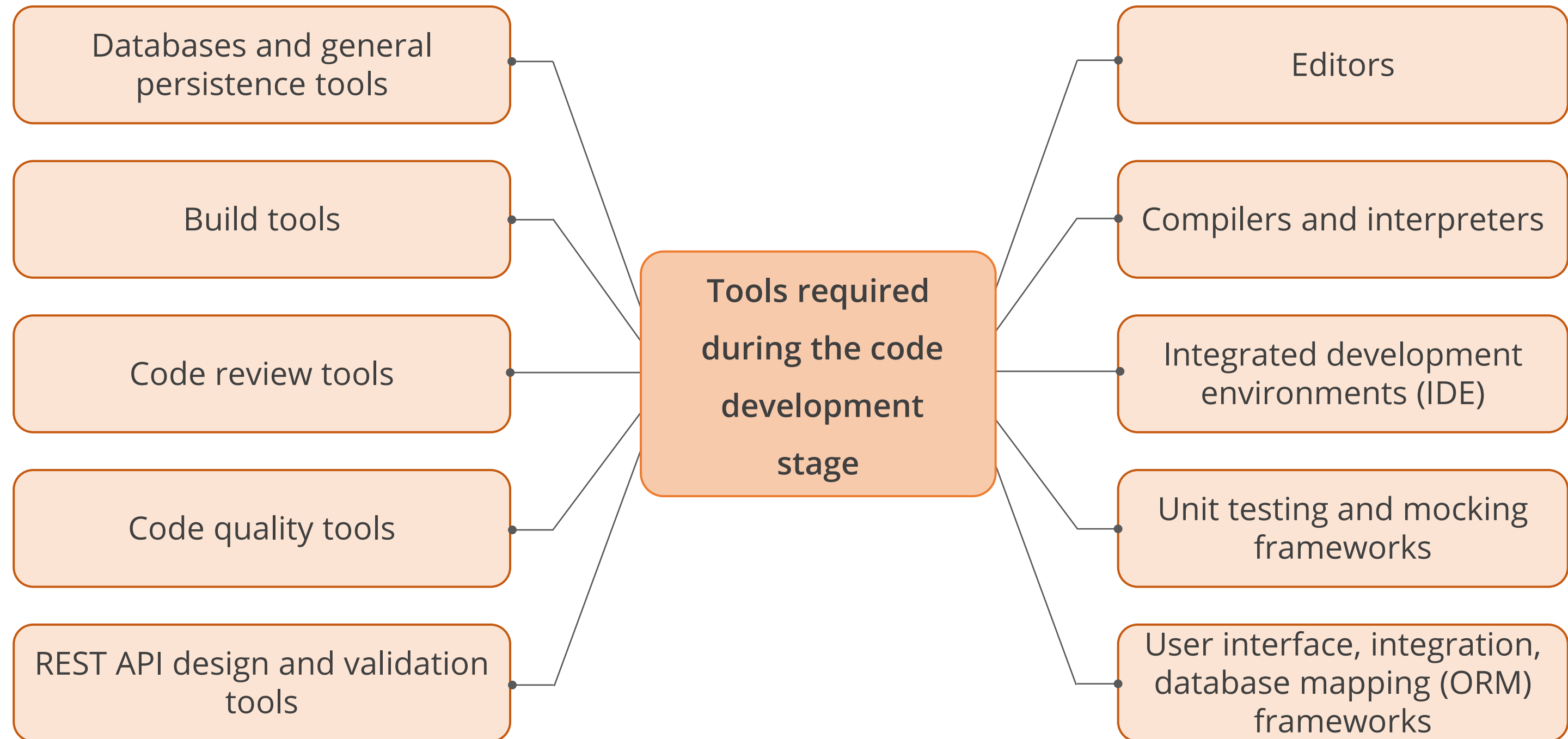
# Requirements Tools

Tools are used to share the files and communicate within the team and other teams.

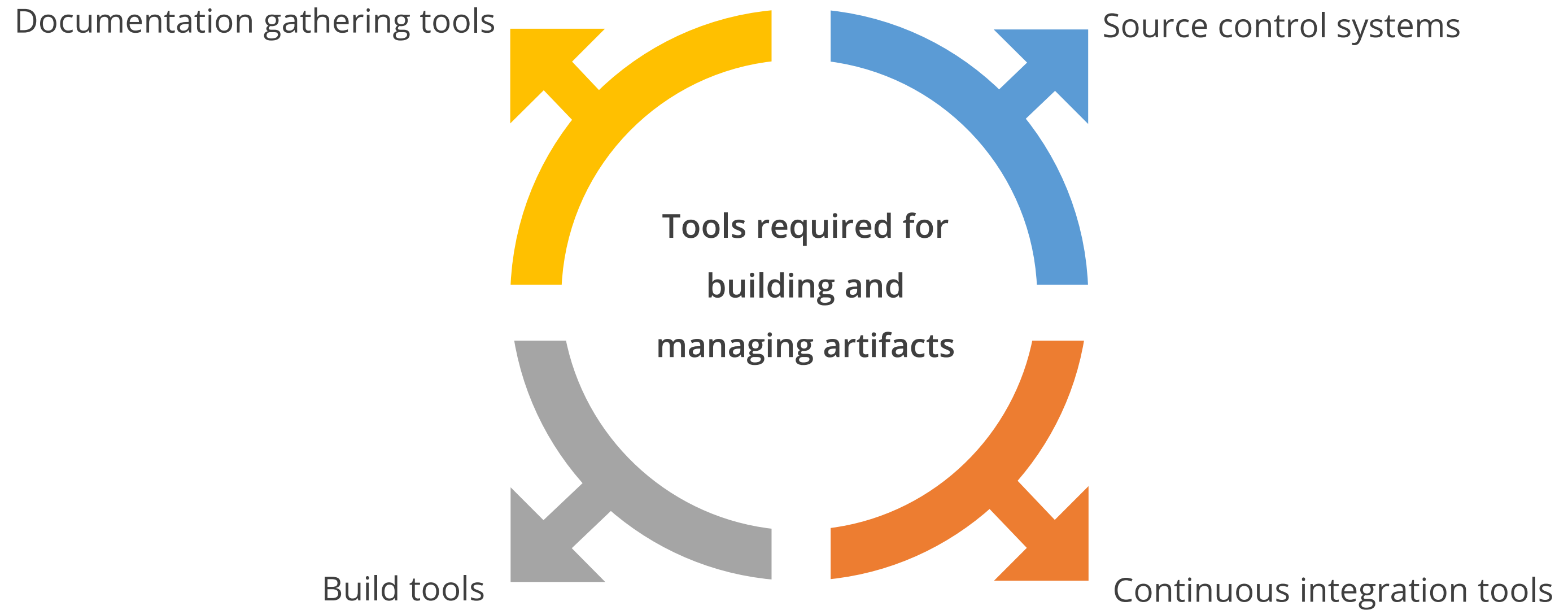




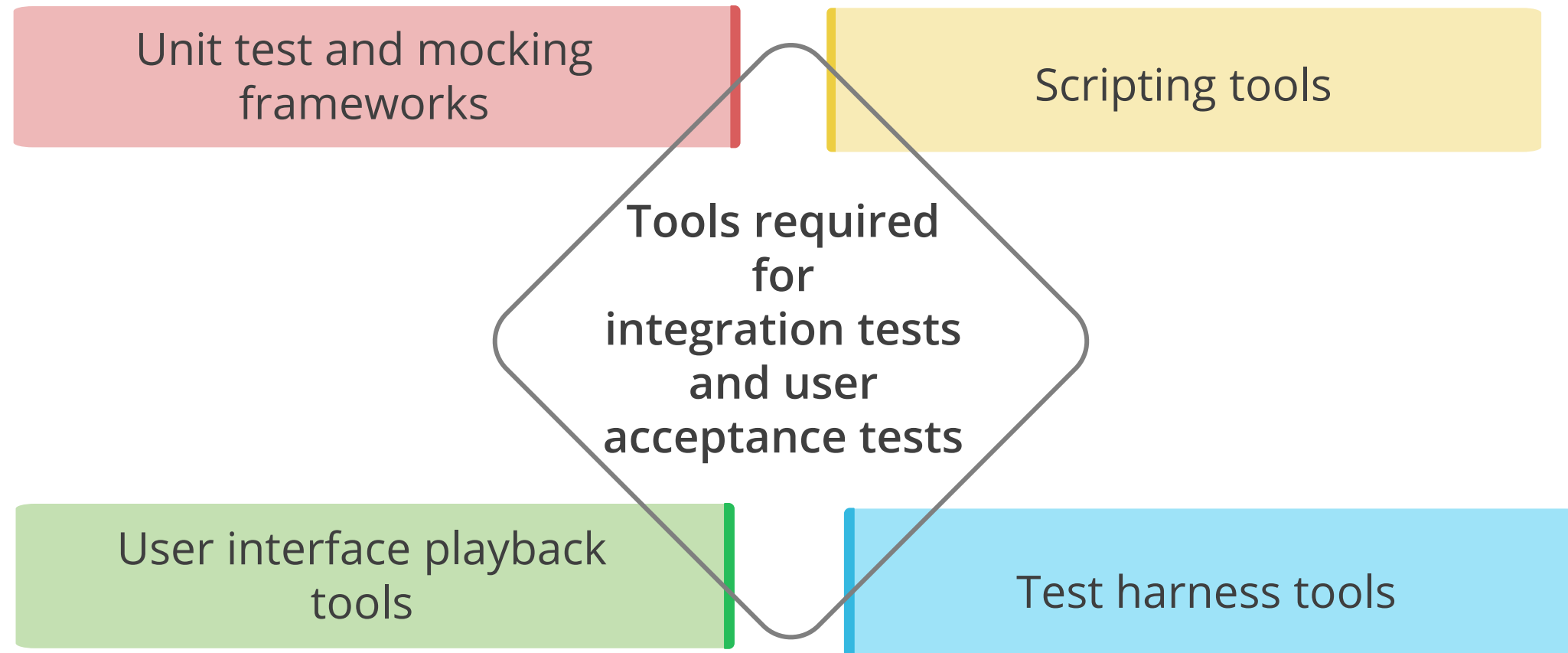
# Code Development Tools



# Artifact Creation Tools



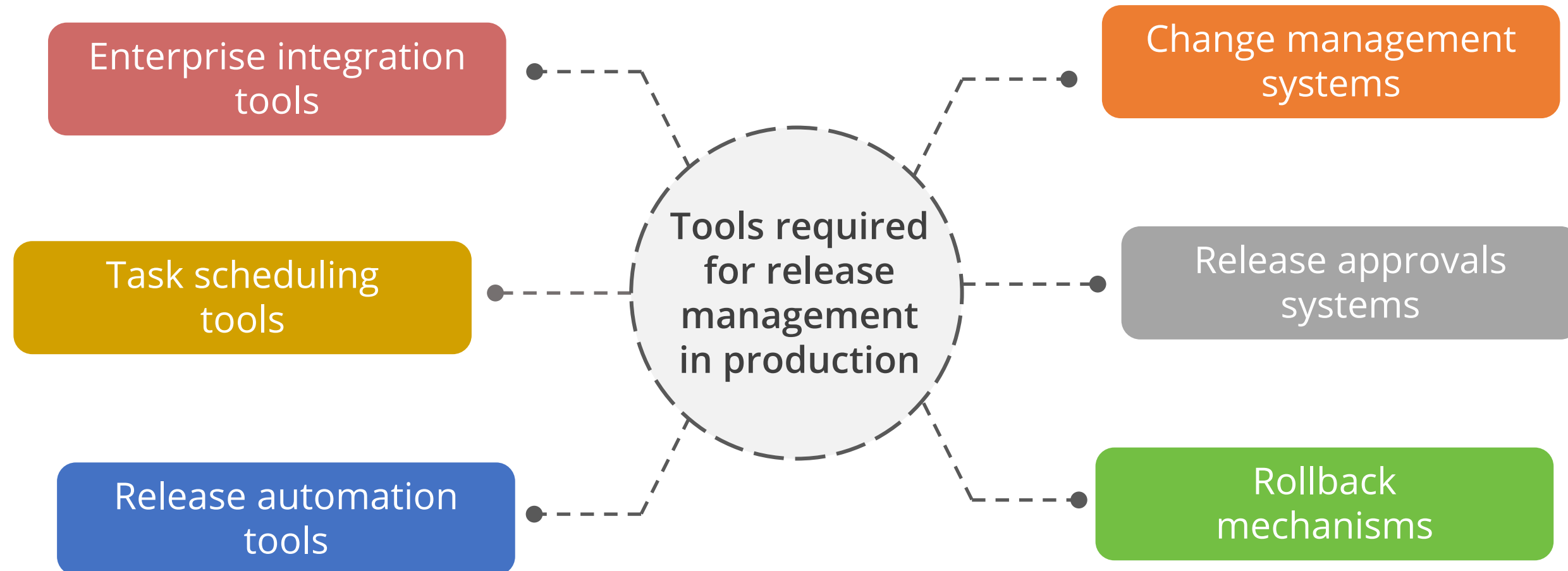
# Testing Tools



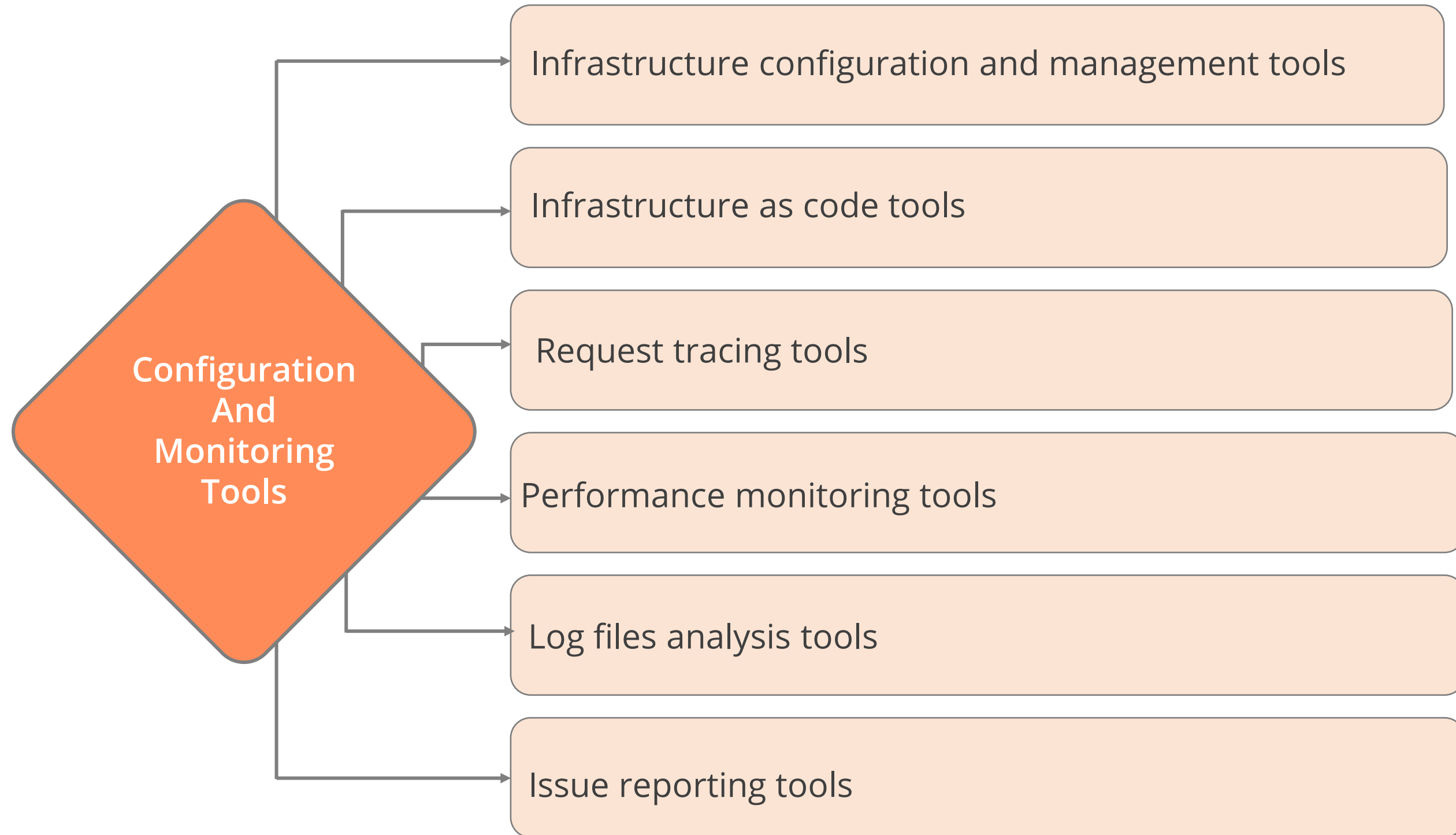
# Packaging Tools



# Release Management Tools



# Configuration and Monitoring Tools

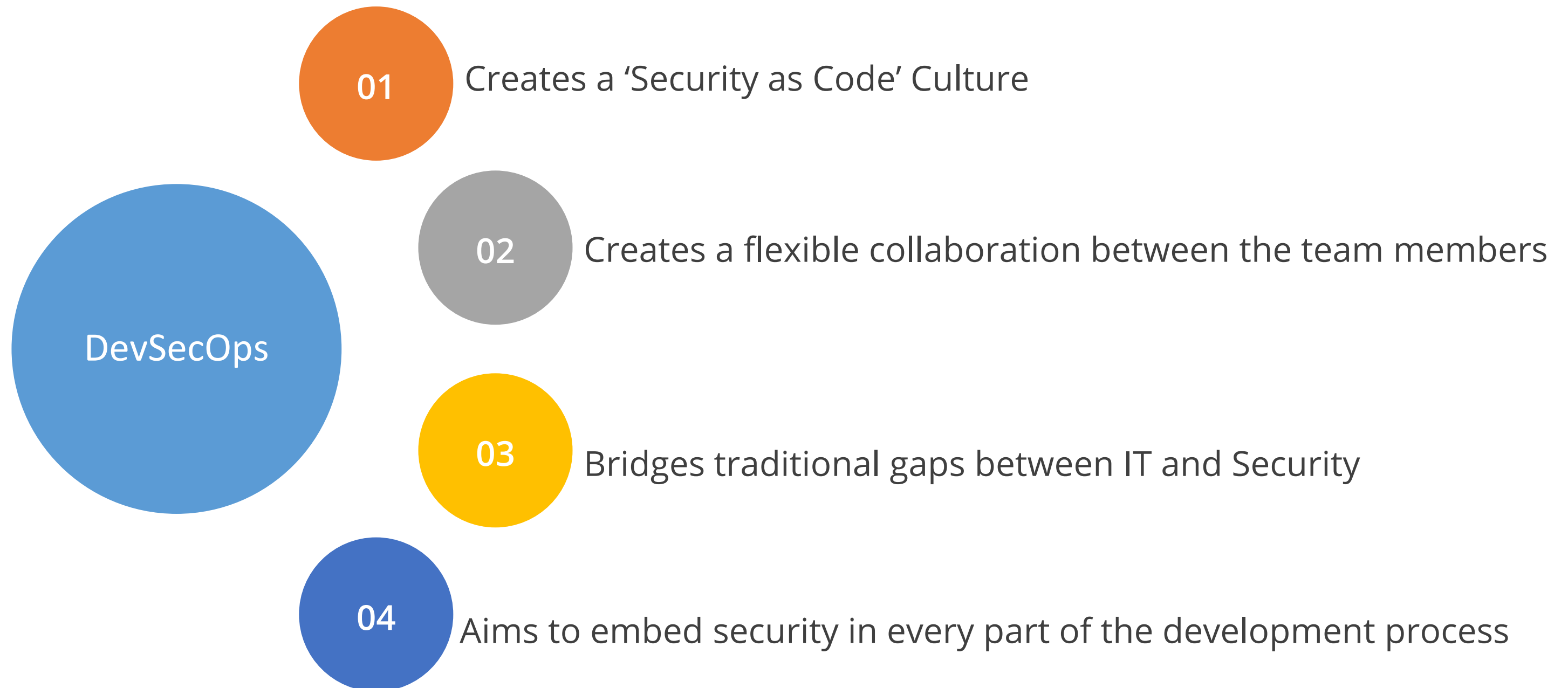




# DevSecOps and It's importance

# DevSecOps

DevSecOps is the ideology of integrating security practices within the DevOps culture.



# DevSecOps Manifesto

---

**Leaning in** over Always Saying “no”

**Data & Security Science** over Fear, Uncertainty and Doubt

**Open Contribution & Collaboration** over Security-Only

Requirements

**Consumable Security Services with APIs** over Mandated Security

Controls & Paperwork

**Business Driven Security Scores** over Rubber Stamp Security

**Red & Blue Team Exploit Testing** over Relying on Scans & Theoretical Vulnerabilities

**24\*7 Proactive Security Monitoring** over Reacting after being informed of an incident

**Shared Threat Intelligence** over Keeping Info to Ourselves

**Compliance Operations** over Clipboards & Checklists

# Workflow of DevOps and DevSecOps

---

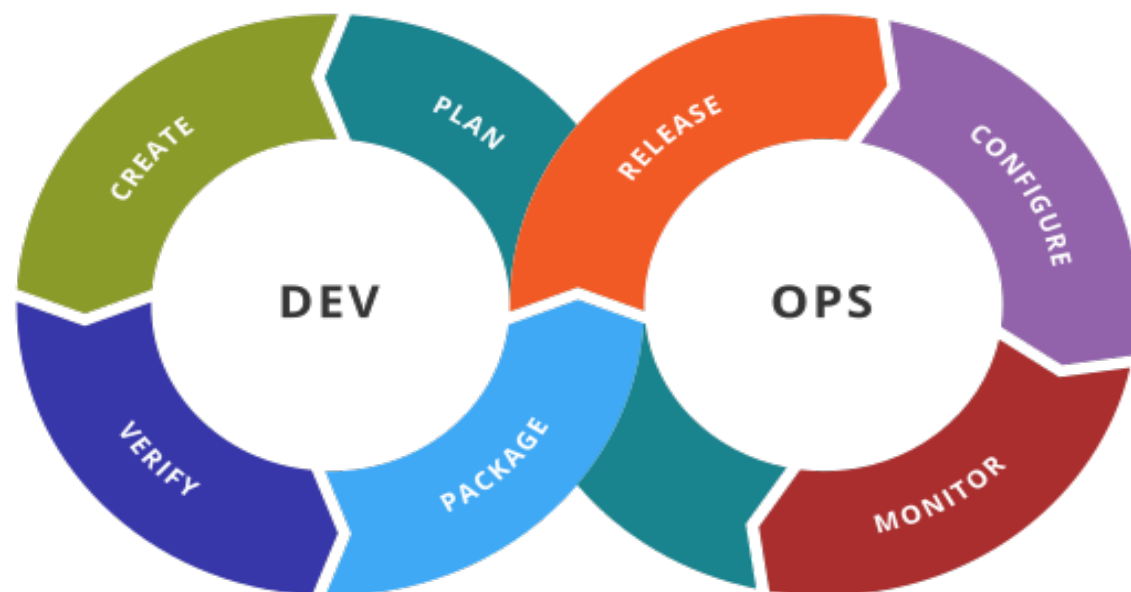
- A Developer writes a code for a feature within the version control management system
- The required changes are committed to the version control management system
- Another Developer retrieves the code from the version control management system and performs analysis of the code to identify security threats
- An environment is created using any management tool such as puppet or chef. The application is deployed and security configurations are applied to the system.

## Workflow of DevOps and DevSecOps (Contd.)

---

- A test automation suite is executed against the deployed application on factors like UI, integration, security tests, and APIs
- If the application passes the test, it is deployed to the production environment
- The newly production environment is monitored continuously to identify any active security threats

# DevOps vs. DevSecOps



## Features:

Speed

Rapid delivery

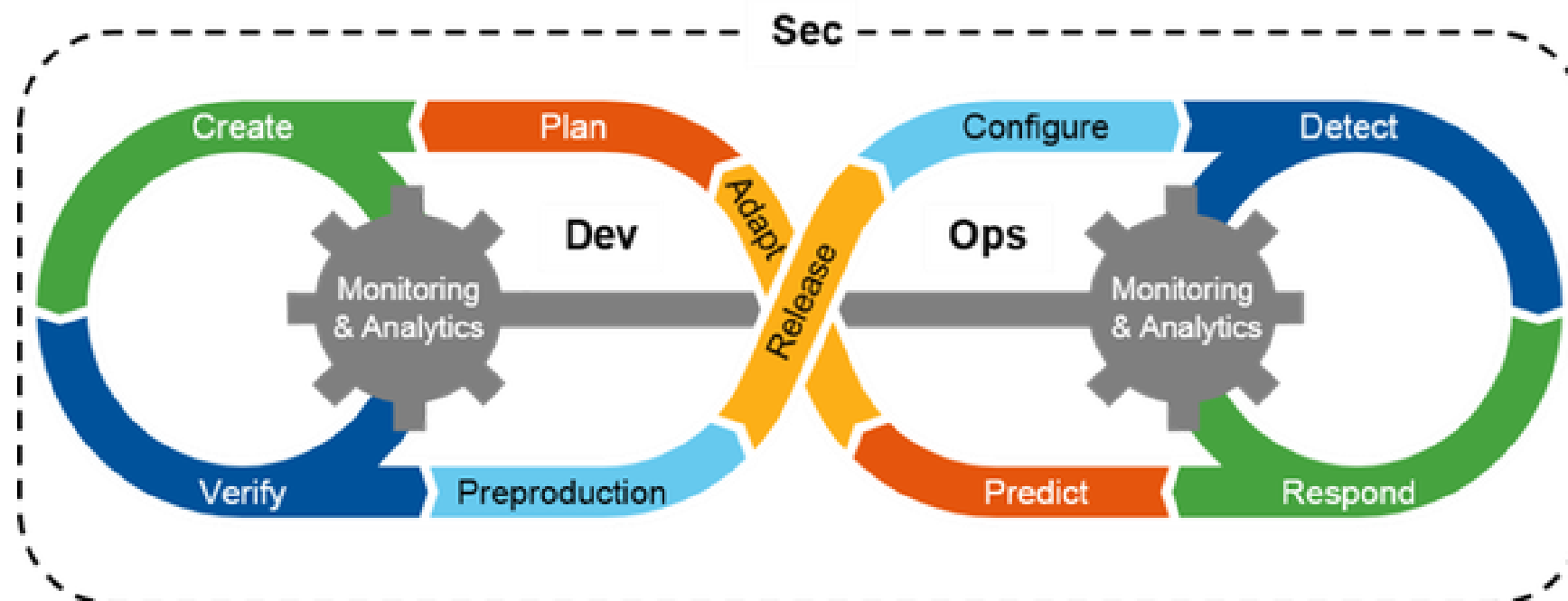
Scaled and less secured

## Features:

Agility

Security automation

Security as code





# Key Takeaways

You are now able to:

- State the principles of DevOps
- Explain the challenges in the traditional approach
- Describe how DevOps helps in overcoming challenges faced in traditional approach
- Describe the DevOps tools
- Explain DevSecOps and its importance





## Knowledge Check

## Knowledge Check

1

Which one of the following methodologies has least impact on DevOps methodology?

- A. Lean Manufacturing
- B. Agile Software Delivery
- C. Waterfall Software Delivery
- D. Continuous Software Delivery



## Knowledge Check

1

Which one of the following methodologies has least impact on DevOps methodology?

- A. Lean Manufacturing
- B. Agile Software Delivery
- C. Waterfall Software Delivery
- D. Continuous Software Delivery



The correct answer is **C**

**Waterfall methodology** has the least impact on DevOps methodology as it is time consuming.



## Knowledge Check

2

Which one of the following techniques makes DevOps a successful methodology to develop and deliver software?

- A. DevOps helps in organizing the teams and organizational mission
- B. DevOps helps in creating the software with built-in quality and monitoring
- C. DevOps helps to quickly identify, fix and learn from errors
- D. All of the above



## Knowledge Check

2

Which one of the following techniques makes DevOps a successful methodology to develop and deliver software?

- A. DevOps helps in organizing the teams and organizational mission
- B. DevOps helps in creating the software with built-in quality and monitoring
- C. DevOps helps to quickly identify, fix and learn from errors
- D. All of the above



The correct answer is **D**

**All of the above mentioned techniques make DevOps a successful methodology to develop and deliver software.**



## Knowledge Check

3

Which is the best technique to convert normal changes into standard changes?

- A. Use the existing track record of successful automated deployments with standard changes
- B. Negotiate with release managers
- C. Publicly complain about bureaucracy and make everyone be aware of it
- D. Make sure normal changes are very carefully deployed to the production



## Knowledge Check

3

Which is the best technique to convert normal changes into standard changes?

- A. Use the existing track record of successful automated deployments with standard changes
- B. Negotiate with release managers
- C. Publicly complain about bureaucracy and make everyone be aware of it
- D. Make sure normal changes are very carefully deployed to the production



The correct answer is **A**

**Using existing track record of successful automated deployments with standard changes is the best solution in long run.**

## Knowledge Check

4

Which is the widely reusable asset to reinforce information security?

- A. Data Storage Systems
- B. Tools handling the logging of sensitive client information
- C. Transfer of data between clients and software
- D. All of the above



## Knowledge Check

4

Which is the widely reusable asset to reinforce information security?

- A. Data Storage Systems
- B. Tools handling the logging of sensitive client information
- C. Transfer of data between clients and software
- D. All of the above



The correct answer is **D**

**All of the above mentioned options are widely used as reusable assets to reinforce information security.**

## Knowledge Check

5

**What is the benefit of using feature toggles embedded in configuration of the application during fixing errors?**

- A. Easiest way to fix a problem
- B. Don't have to correct erroneous pieces immediately during deployment
- C. DevOps team can take time to properly identify root cause of an issue and improve the techniques
- D. All of the above





**Knowledge  
Check**

**5**

**What is the benefit of using feature toggles embedded in configuration of the application during fixing errors?**

- A. Easiest way to fix a problem
- B. Don't have to correct erroneous pieces immediately during deployment
- C. DevOps team can take time to properly identify root cause of an issue and improve the techniques
- D. All of the above



The correct answer is **D**

**All of the above mentioned options are the benefits of using feature toggles embedded in configuration of the application during fixing errors.**