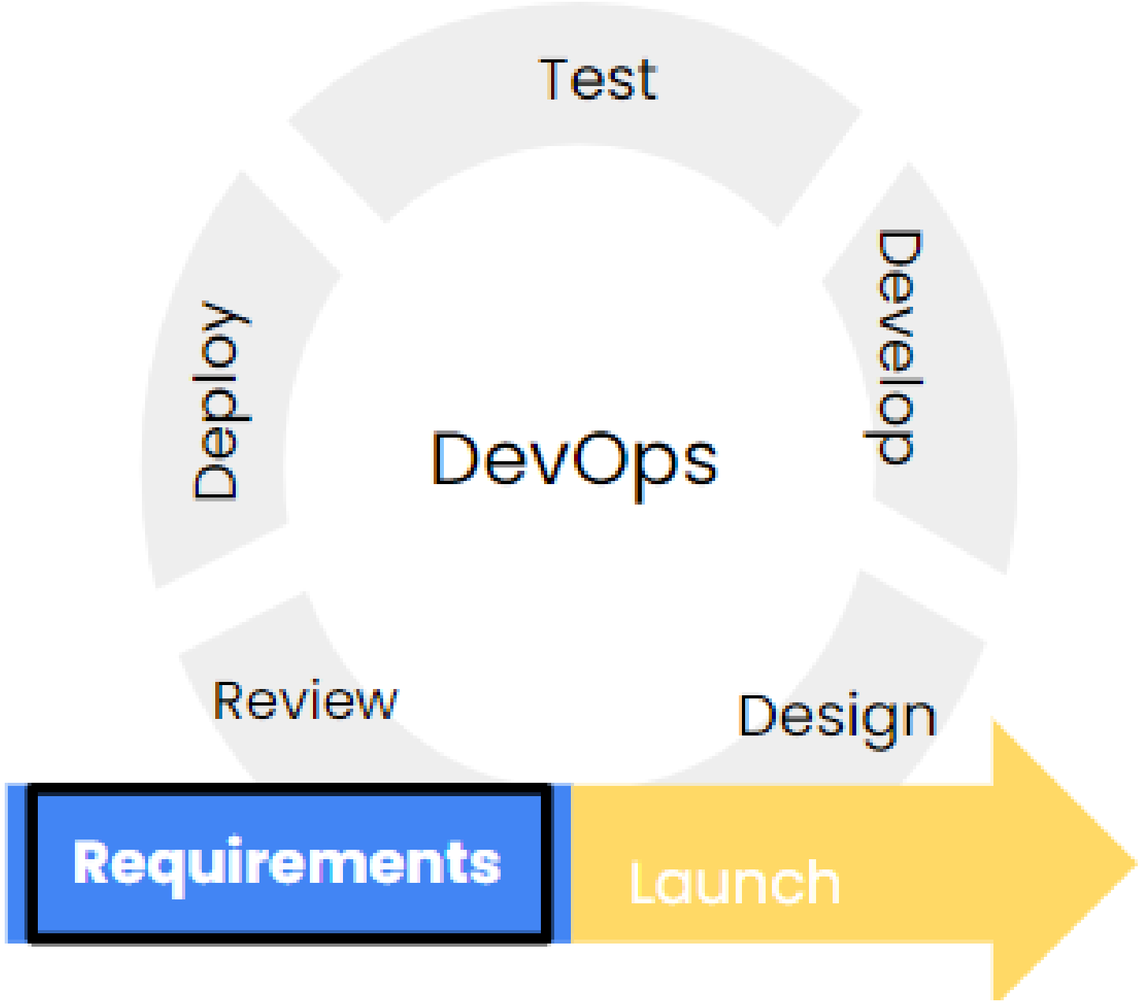
The DevOps Change Management Model

INTRODUCTION TO DEVOPS

Cem Sakarya
DevOps Risk Advisor

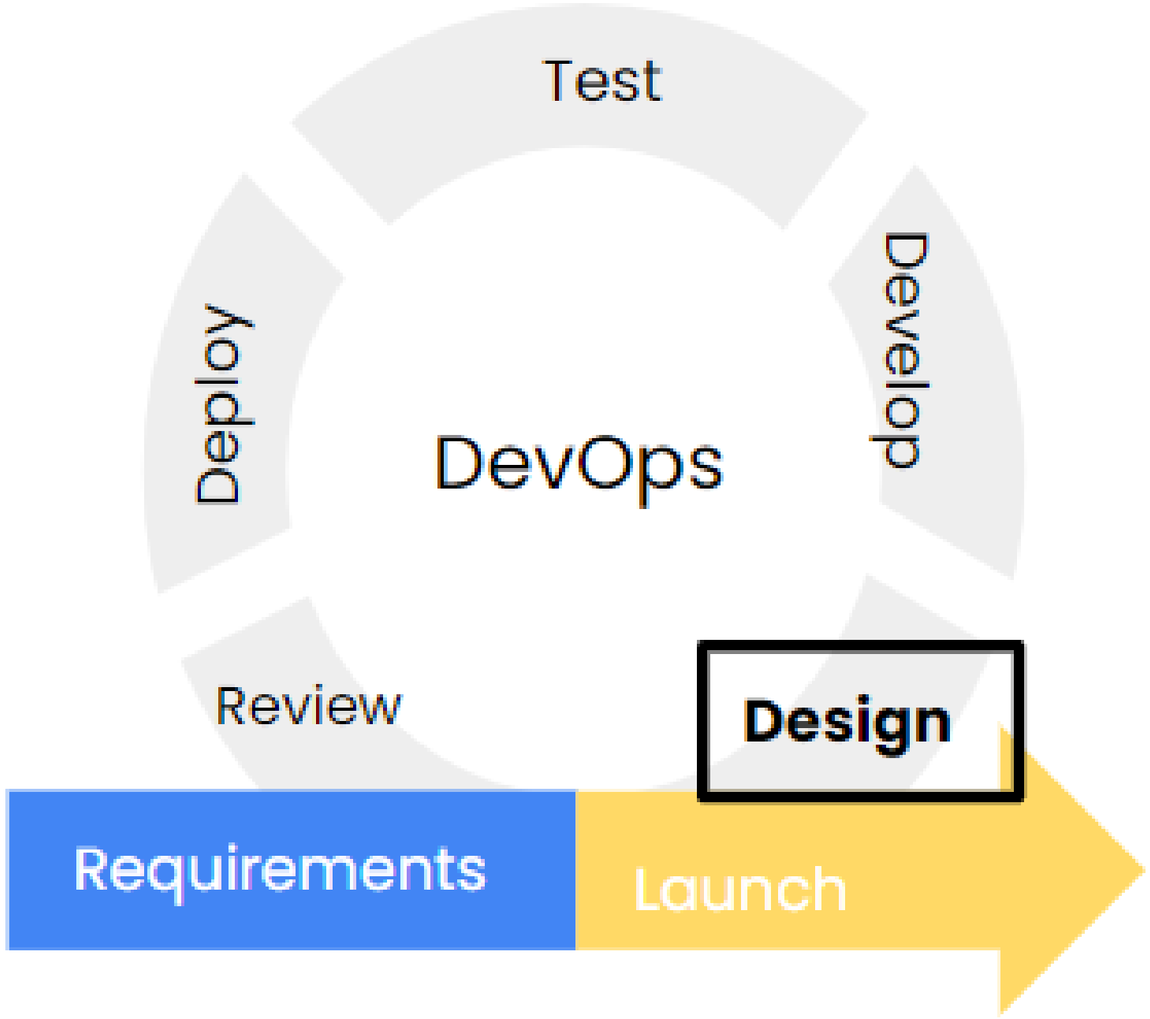
Requirements



Developing a new product

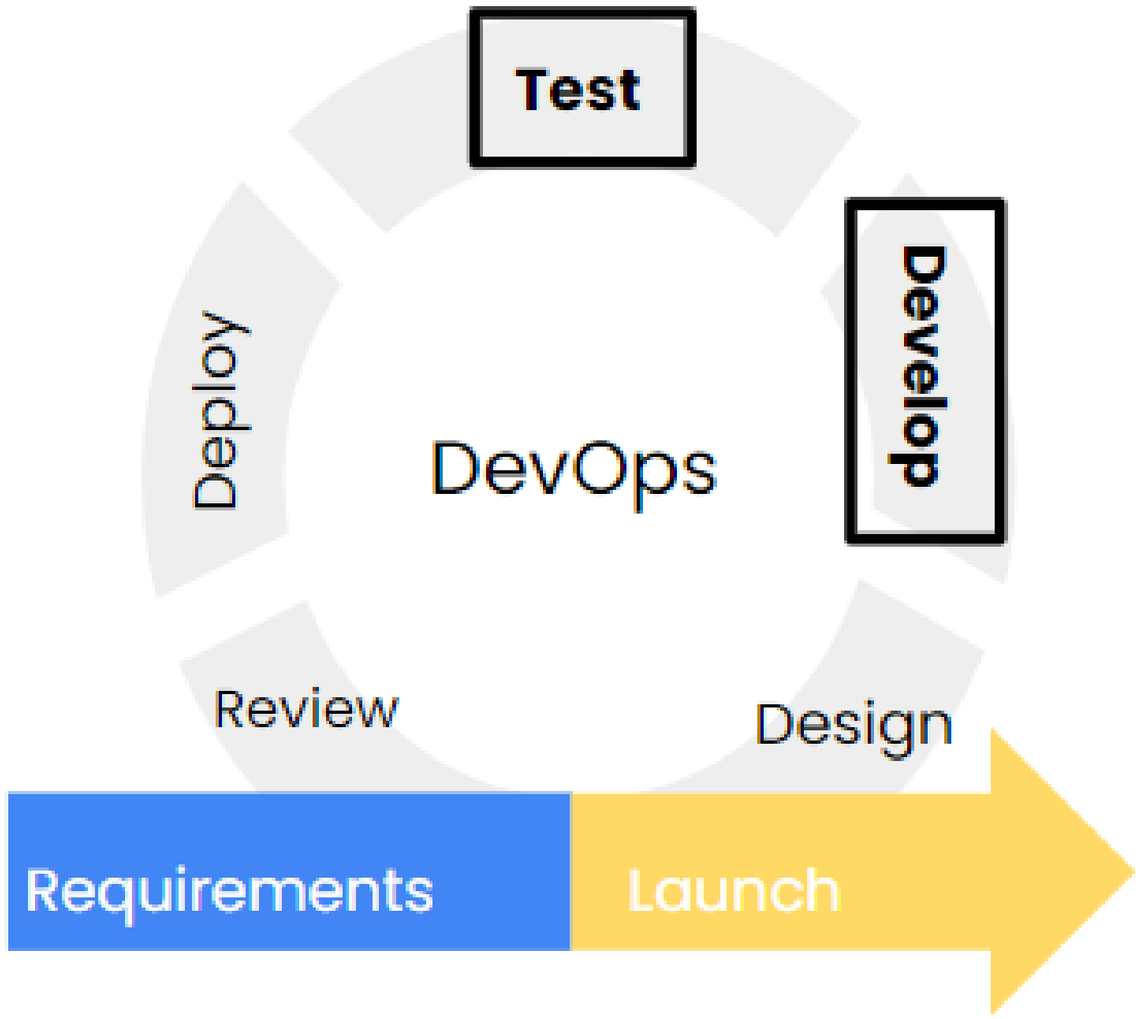
- Leadership: Sets business expectations
- Product Experts: Define user experience requirements

Design



- Product Engineers: Designing the customer experience
- Infra Engineers: Designing the parts that product engineers will use
- Data Engineers: Designing the data pipelines for the new product

Develop and test



Product Engineers

- Develop for the customer experience
- Customer interaction

Infra Engineers

- Supporting the product
- Providing testing environment

Data Engineers

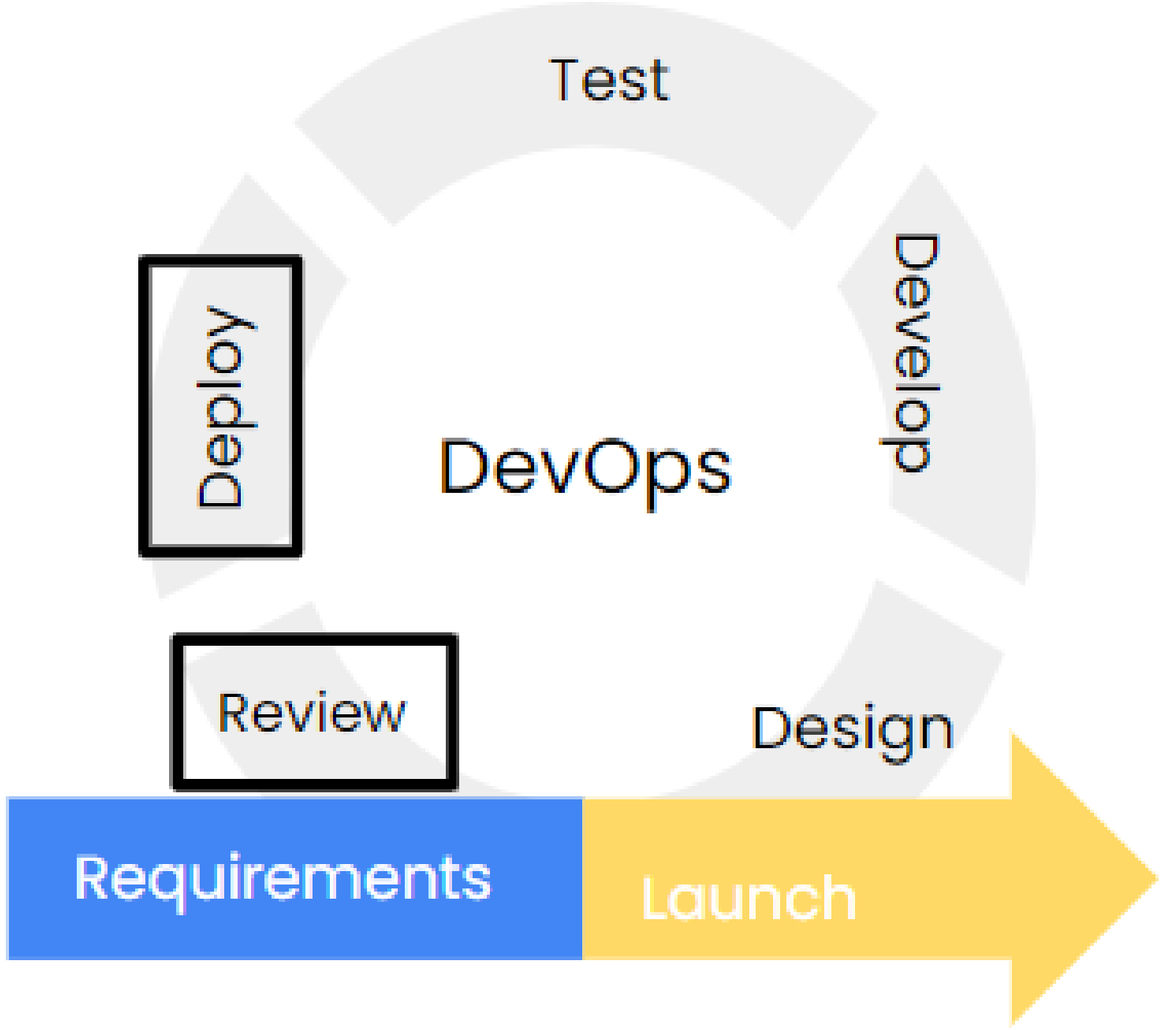
- Providing test data

Testing

Quality Checks

Security Checks

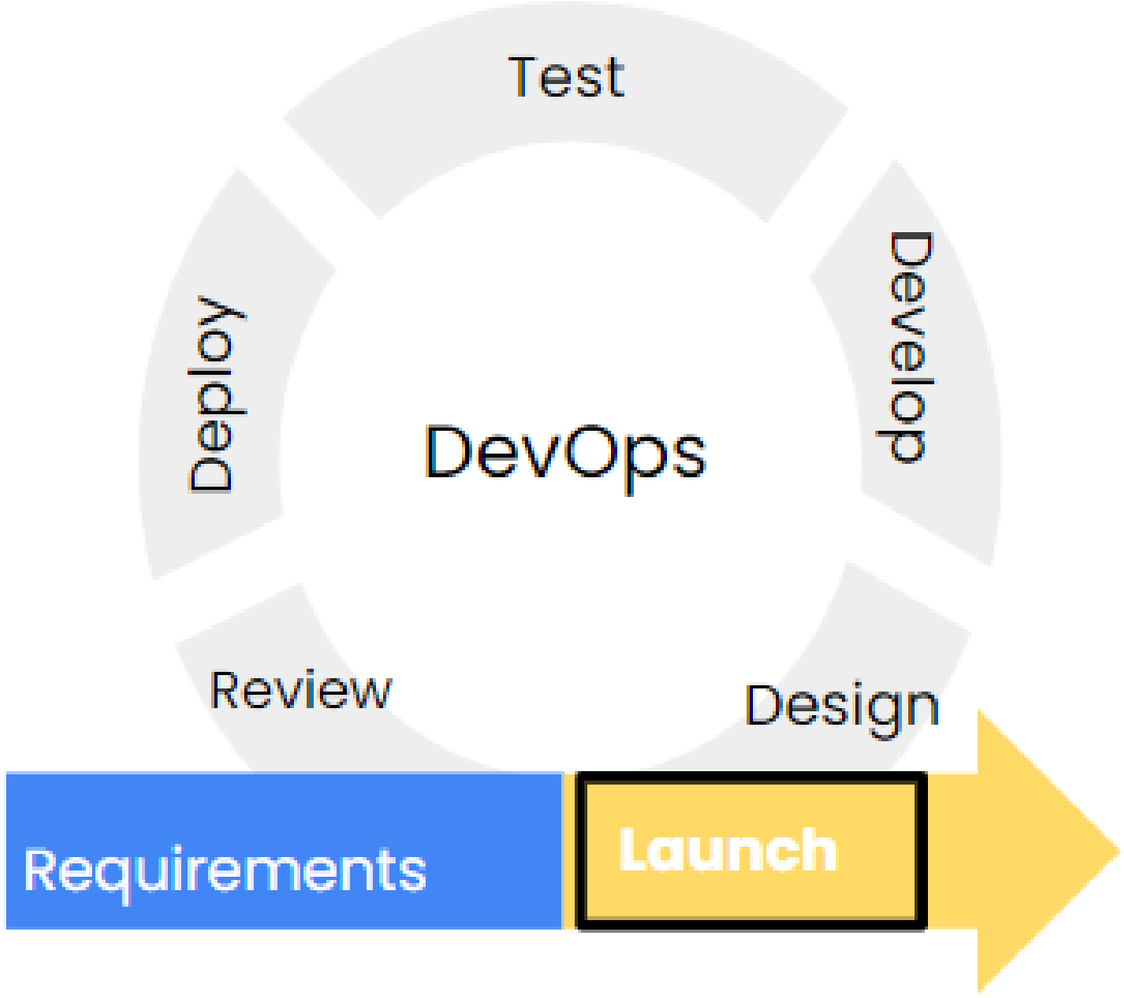
- Testing and developing at the same time
- Testing before customer interaction
- Otherwise poor user experience and security issues



Experimentation

- Limited set of users
- Early user interactions
- Observe if the users like the product
- Real user feedback

Cyclical development



Customers

Let's practice!

INTRODUCTION TO DEVOPS

Main Software Architecture Systems

INTRODUCTION TO DEVOPS

Cem Sakarya
DevOps Risk Advisor

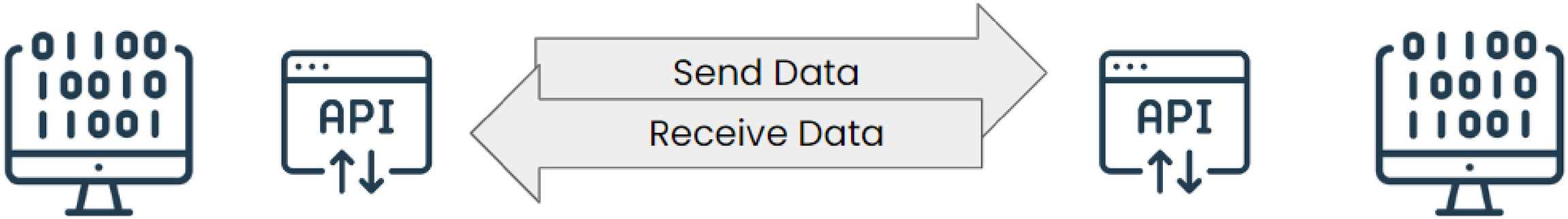
datacamp

Software architecture

> Software architecture refers to the fundamental structures of a software system and the discipline of creating such structures and systems. Each structure comprises software elements, relations among them, and properties of both elements and relations.

[1] https://en.wikipedia.org/wiki/Software_architecture

Application Programming Interfaces (APIs)



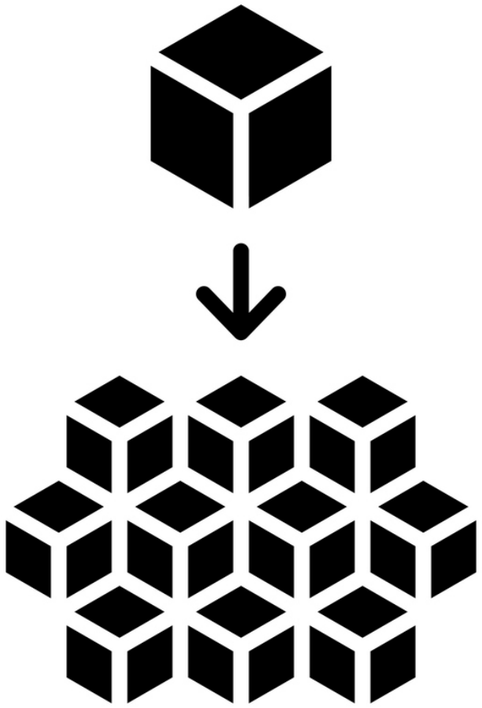
- Data flows through APIs
- They are essential for software and architecture

Various architectural systems

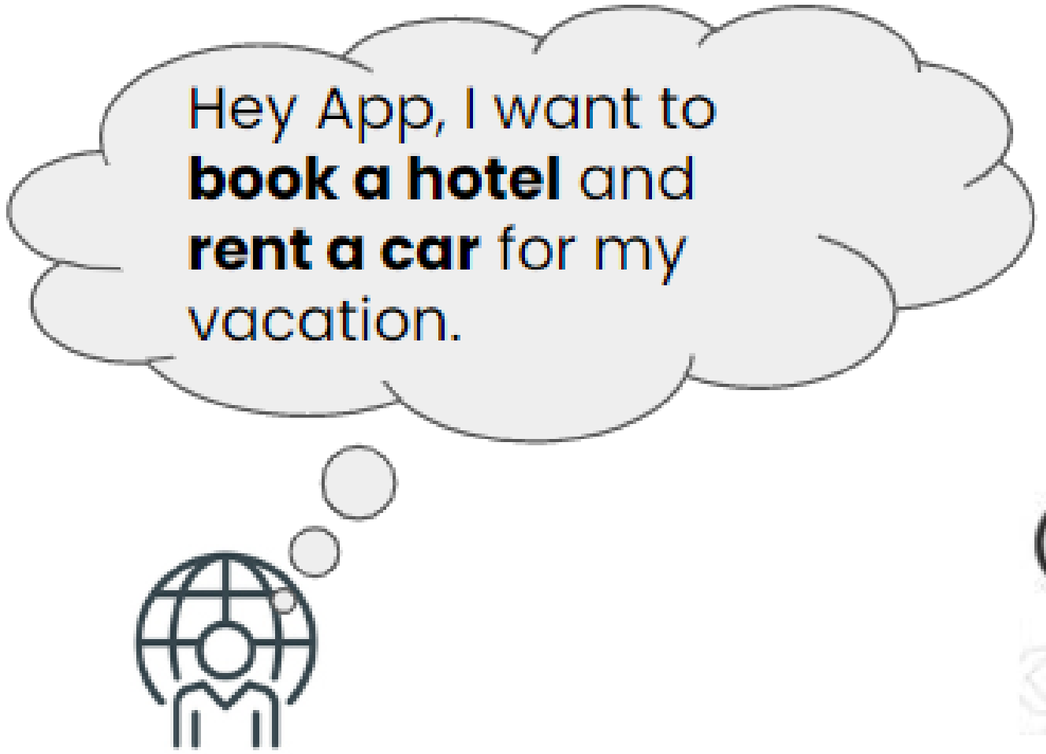
Factors to consider

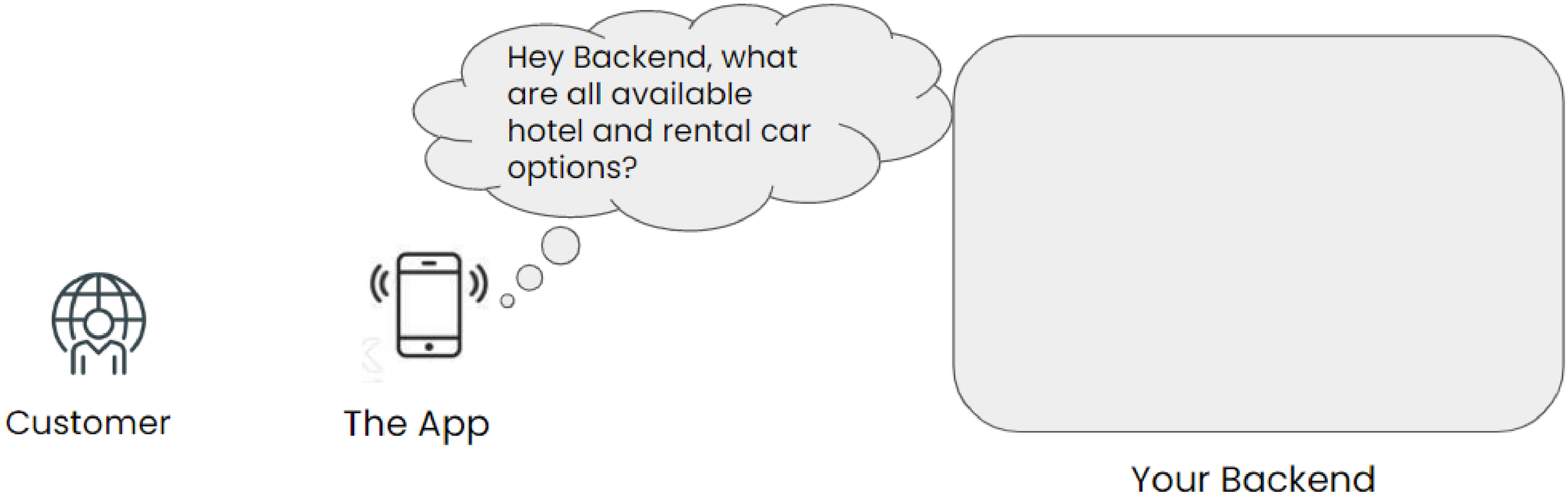
- Type of software
- In machine or in the browser
- Complexity of the software
- Simpler software parts can be treated as a single unit
- Separate the pieces for more complex software

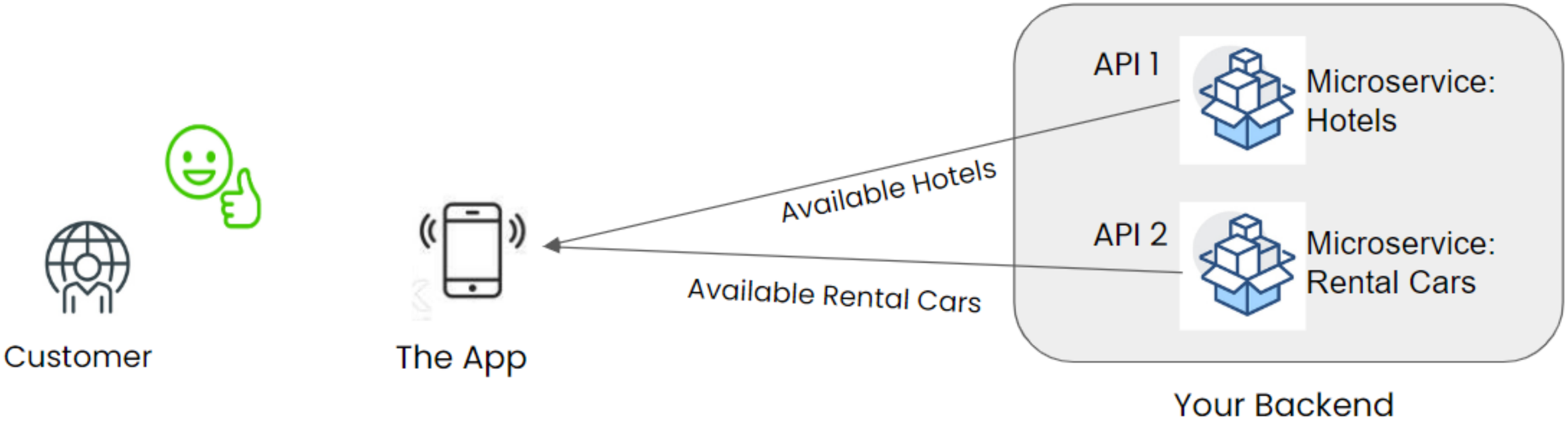
Microservices



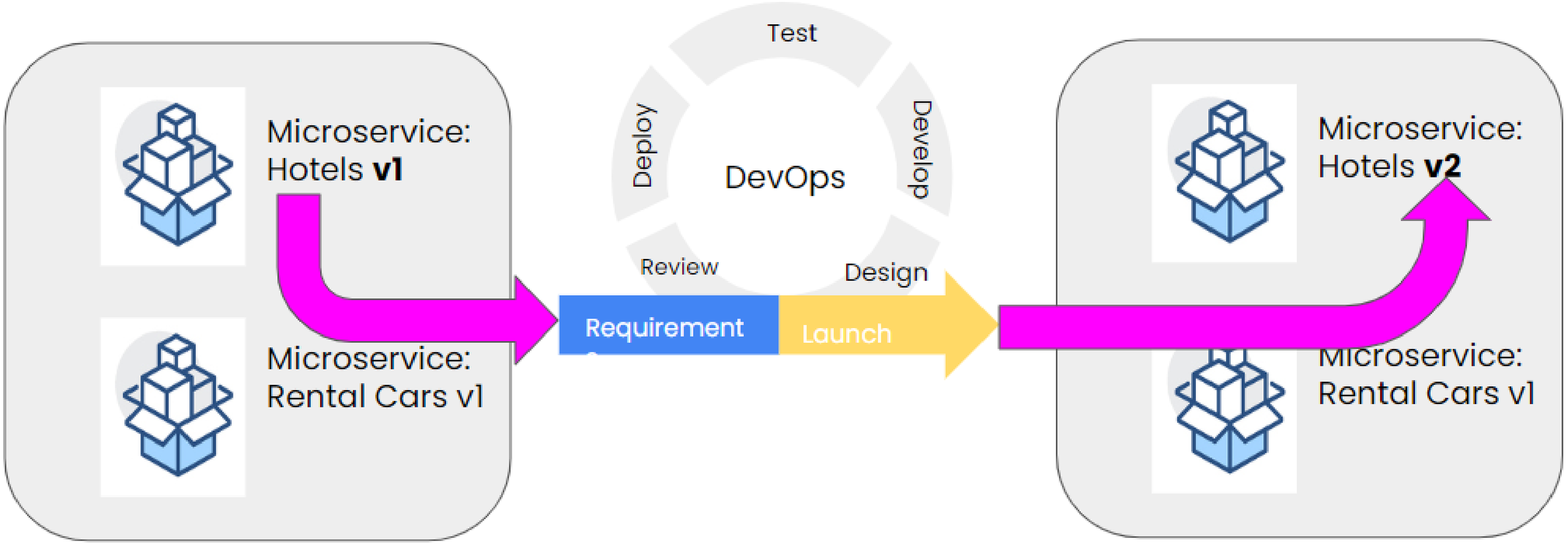
- Most important architectural system for complex software
- Common in modern organizations
- Rather than a big single unit, have many smaller independent units
- Smaller units are called the microservices
- They communicate to each other via APIs


Hey App, I want to **book a hotel** and **rent a car** for my vacation.
Customer
The App


Hey Backend, what are all available hotel and rental car options?
Customer
The App
Your Backend


API 1
Microservice:
Hotels
API 2
Microservice:
Rental Cars
Available Hotels
Available Rental Cars
Customer
The App
Your Backend

DevOps for microservices architecture



Benefits of DevOps for microservices

- Software is changed constantly in large organizations
- Change them one at a time
- Microservices are more secure and reliable
- If anything goes bad with one microservice, the rest will still function well

Let's practice!

INTRODUCTION TO DEVOPS

Modern IT Infrastructure Concepts

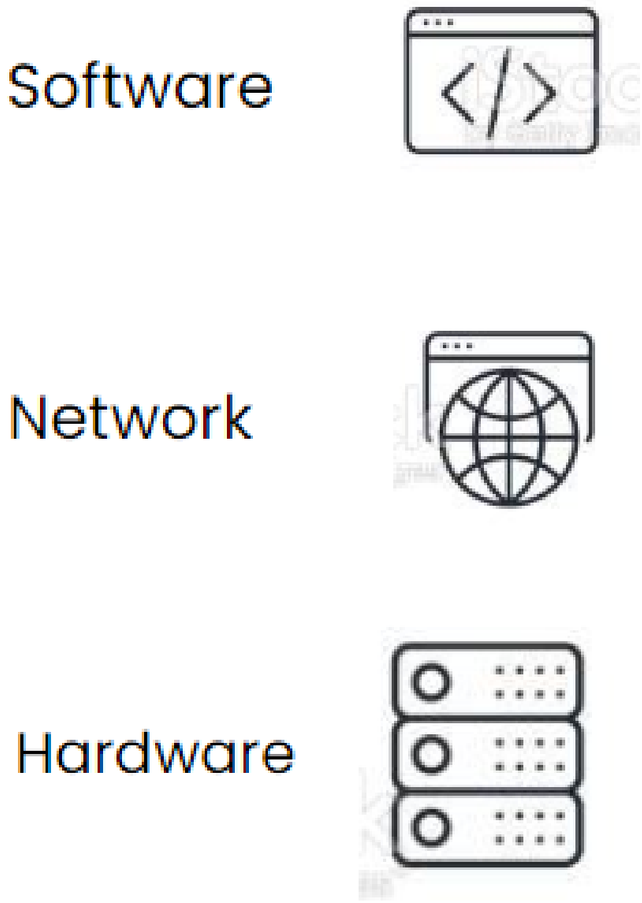
INTRODUCTION TO DEVOPS

Cem Sakarya
DevOps Risk Advisor

datacamp

Main components of IT infrastructure

- Combination of various components
- Customized for organizations specific needs



Hardware

- Can be hosted on organization's own premises
- Or it can be accessed through a cloud service provider

Network

- Communication with other parts of the software
- Communication with the users through the Internet

Software

We need software to:

- Manage the hardware and network
- Configure them for the organization's use

The goal of infrastructure software:

- enable the product teams to develop and serve customers.

Change management

- Software needs constant maintenance and improvement
- These changes could be very expensive and disruptive to the services of the organization
- Infrastructure Engineering provides the tools that's necessary to build products

Developer platform

Developer Platform

Developer changes part of the codebase

Codebase & Version Control

Code is merged to the rest of the codebase securely

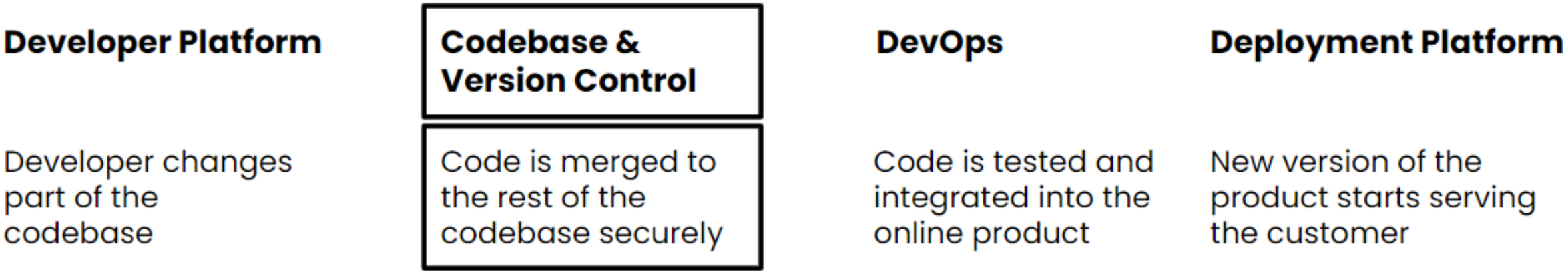
DevOps

Code is tested and integrated into the online product

Deployment Platform

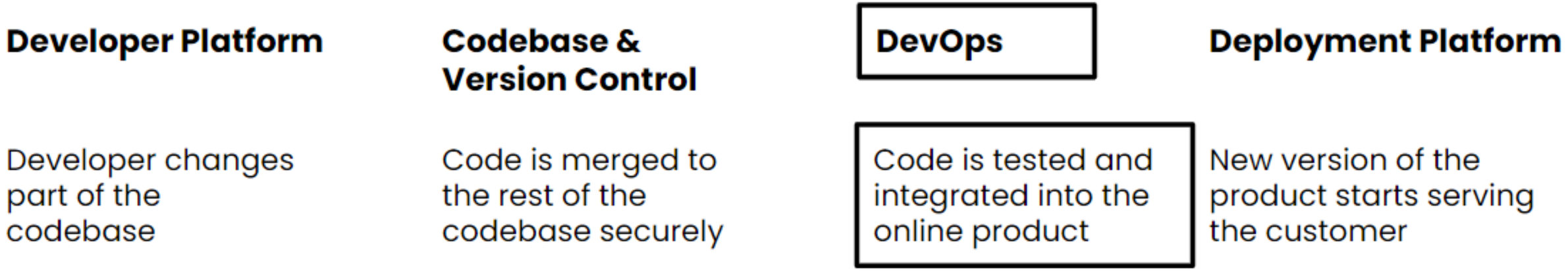
New version of the product starts serving the customer

Codebase & version control



- Version Control: Git

DevOps CI/CD pipelines

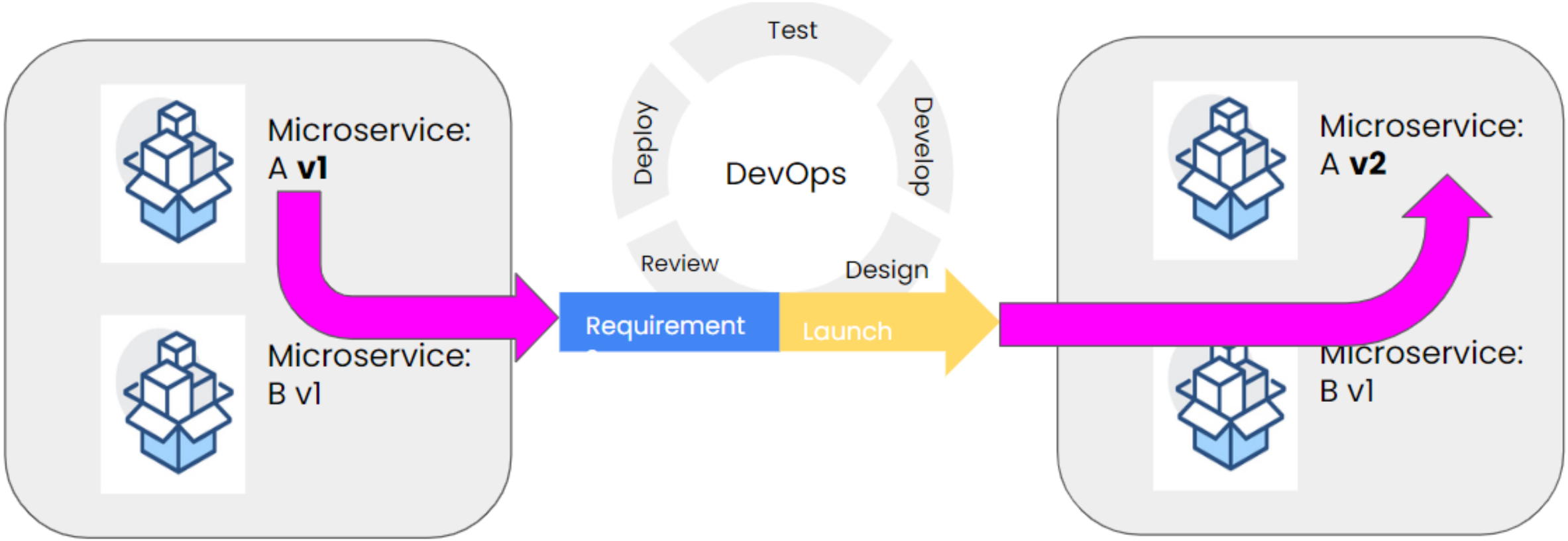


- DevOps in narrow meaning - CI/CD
- Code needs to be built
- Turn code file into machine executable file

Deployment platform

Developer Platform	Codebase & Version Control	DevOps	**Deployment Platform**
Developer changes part of the codebase	Code is merged to the rest of the codebase securely	Code is tested and integrated into the online product	New version of the product starts serving the customer

Change management in microservices



Let's practice!

INTRODUCTION TO DEVOPS

Elements of DevOps

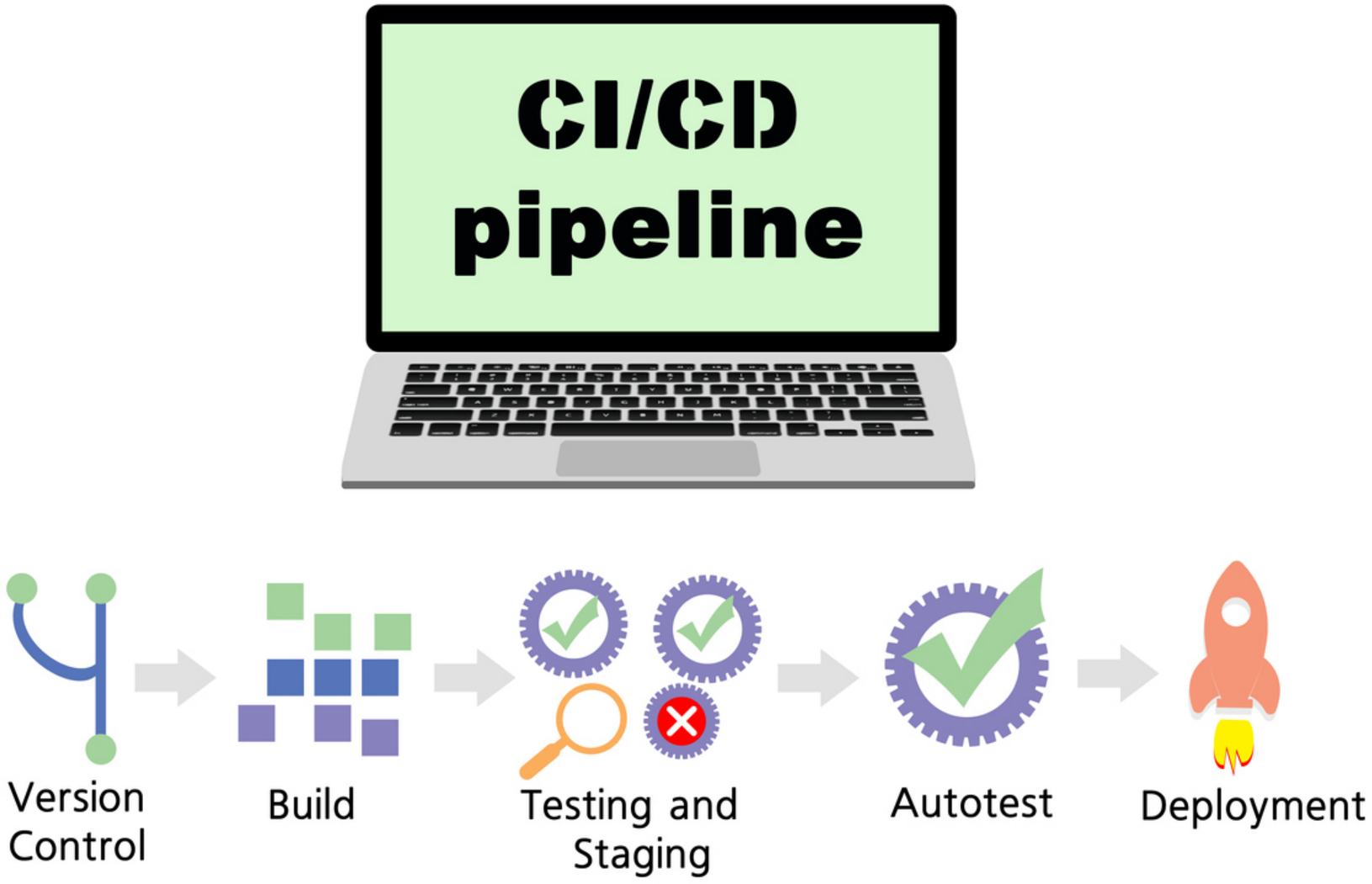
INTRODUCTION TO DEVOPS

Cem Sakarya
DevOps Risk Advisor

DevOps change management model

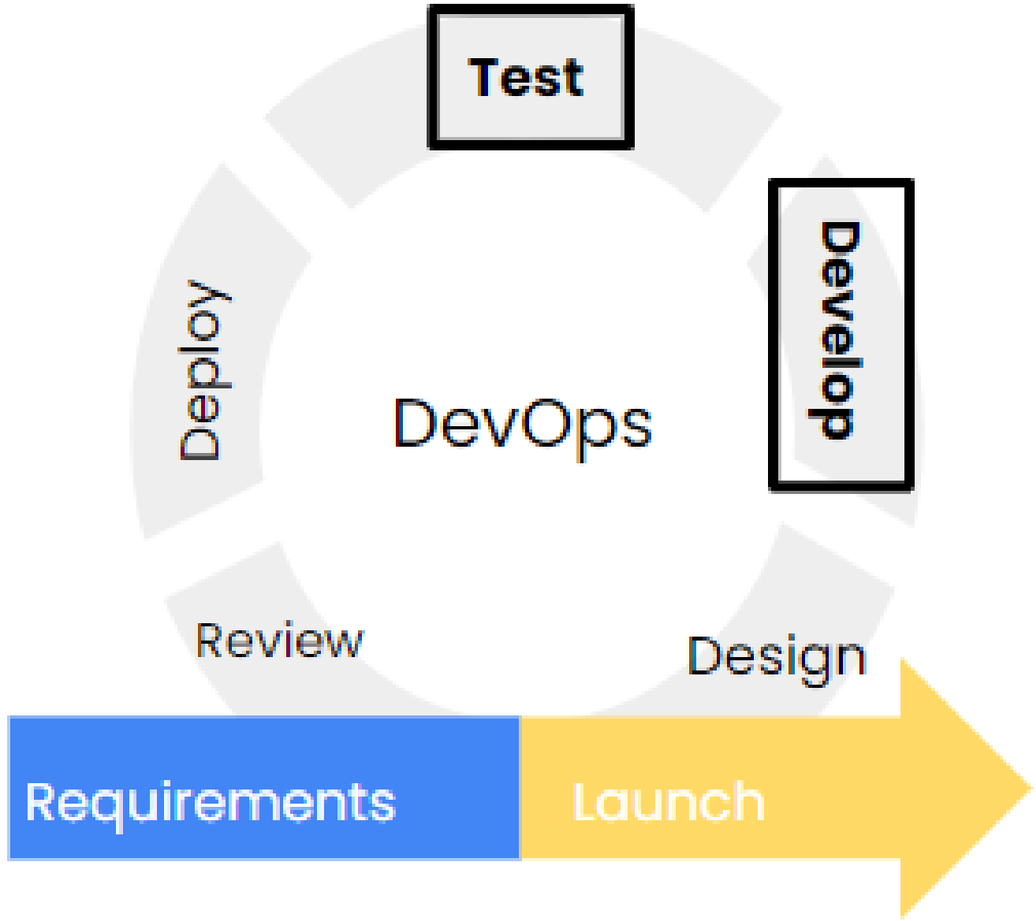
Developer Platform	Codebase & Version Control	DevOps	Deployment Platform
Developer changes part of the codebase	Code is merged to the rest of the codebase securely	Code is tested and integrated into the online product	New version of the product starts serving the customer

- Continuous Integration & Continuous Delivery

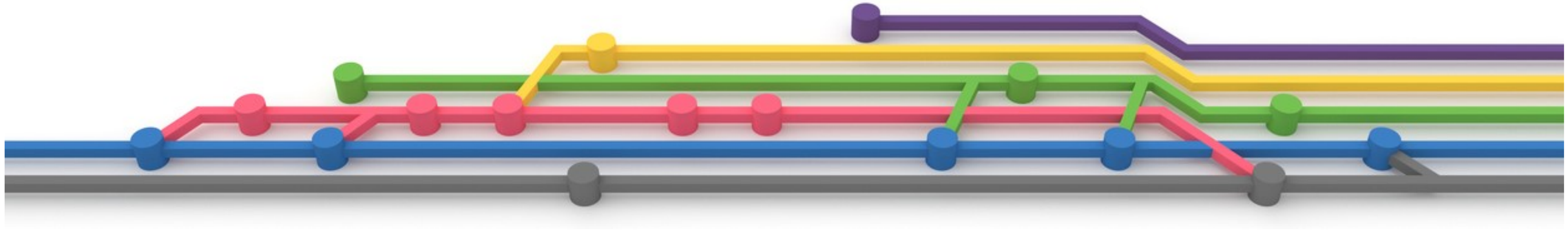

CI/CD
pipeline
Version
Control
Build
Testing and
Staging
Autotest
Deployment

Principles of CI

1. Version Control Software
2. Frequent integrations



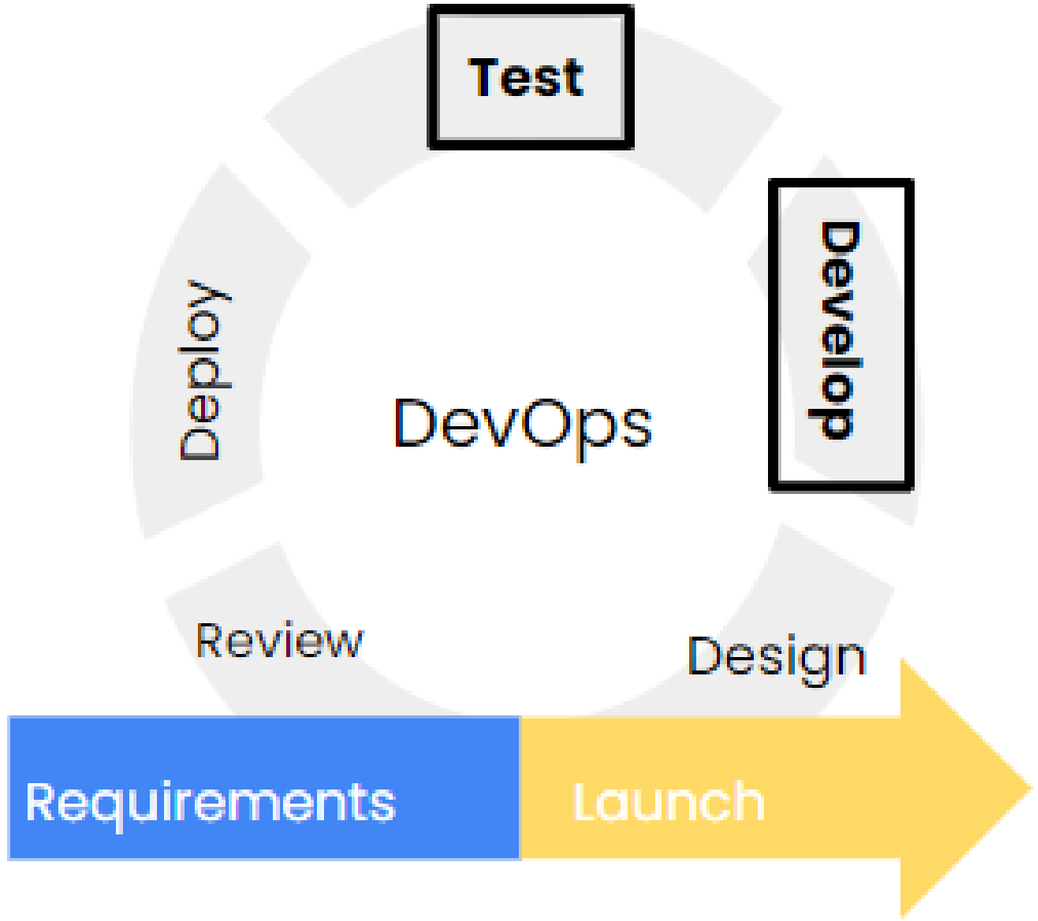
Version control



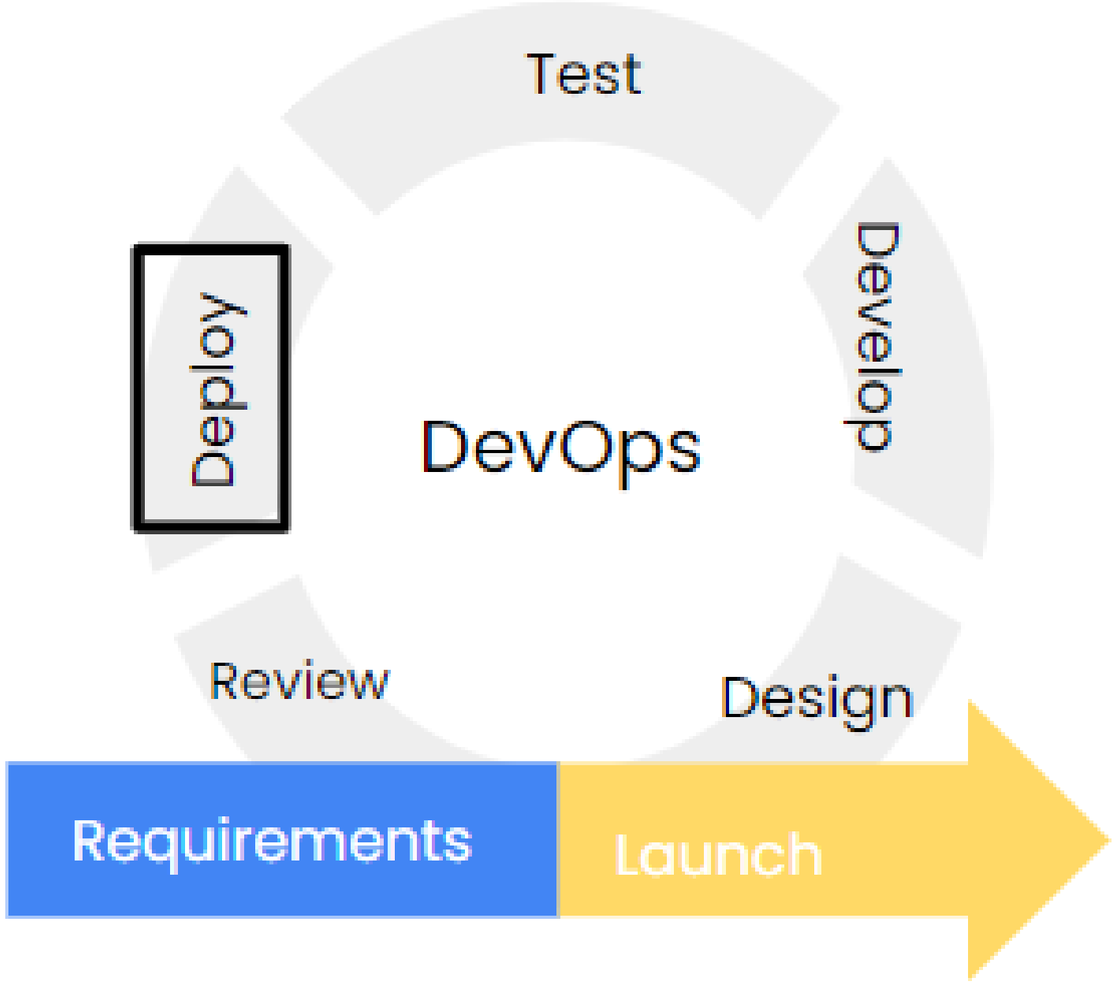
- Main branch as the source-of-truth for the codebase
- Multiple teams working on the same software
- Peer Review
- Testing in every step

Principles of CD

1. Automated builds
2. Automated tests
3. Predictable and short change times

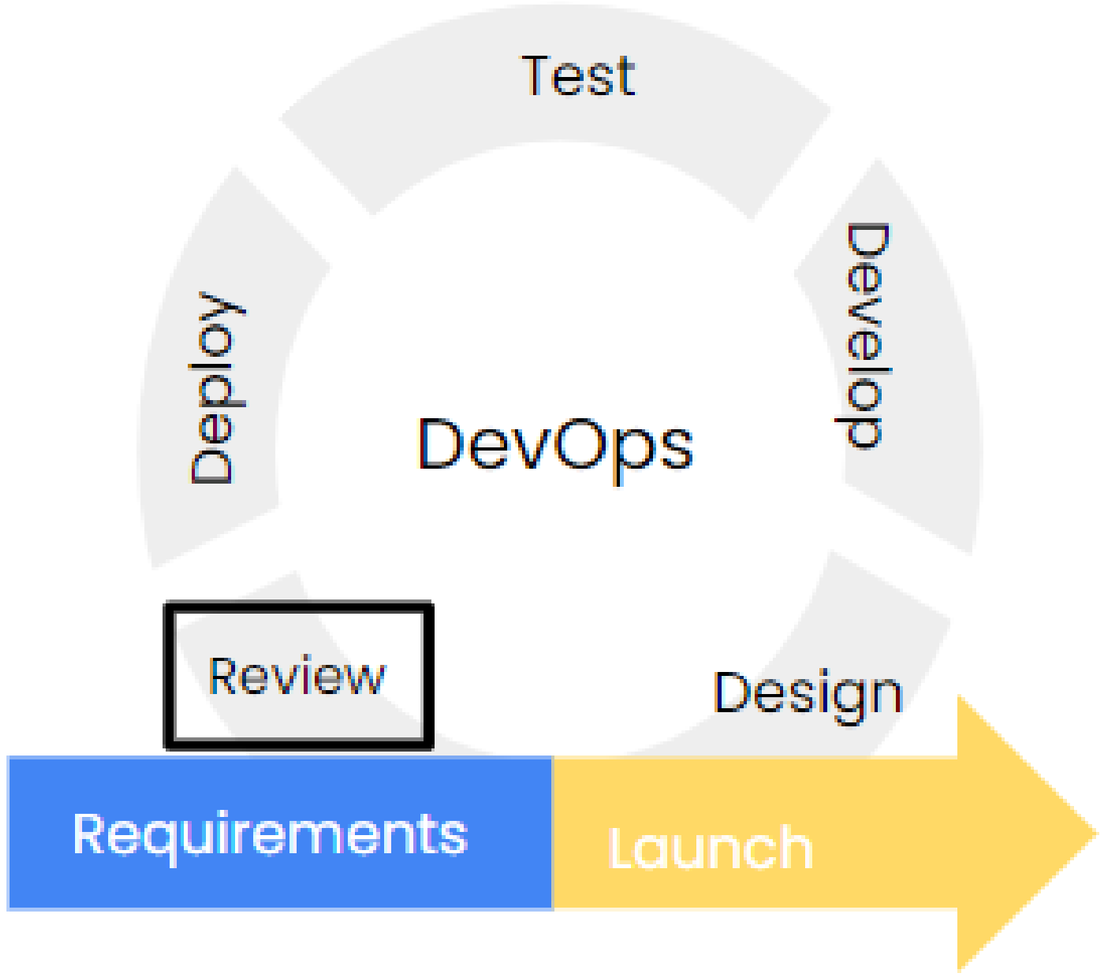


Experimentation



- Limited set of users
- Collect their feedback
- Happens once a change is deployed

Feedback loops



- Continuously improving the product
- Use the review stage for the feedback loop
- Turn the feedback into action
- Change the software if necessary

Let's practice!

INTRODUCTION TO DEVOPS