



# Introducción a DevOps

## LECTURA 1

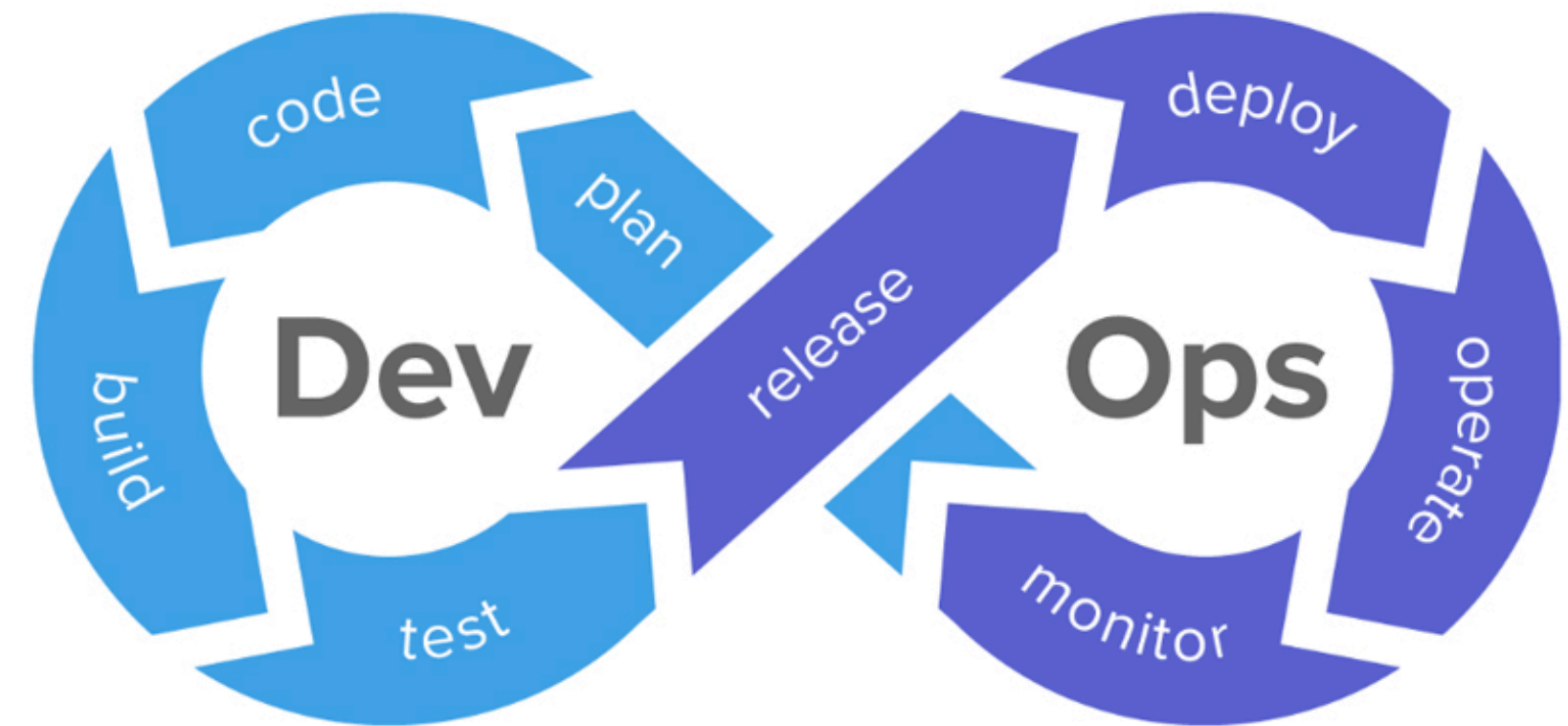
### INTEGRANTES:

Gerson Zuñiga

Akira Garcia

Joel Seminario

# INTRODUCCIÓN A DEVOPS



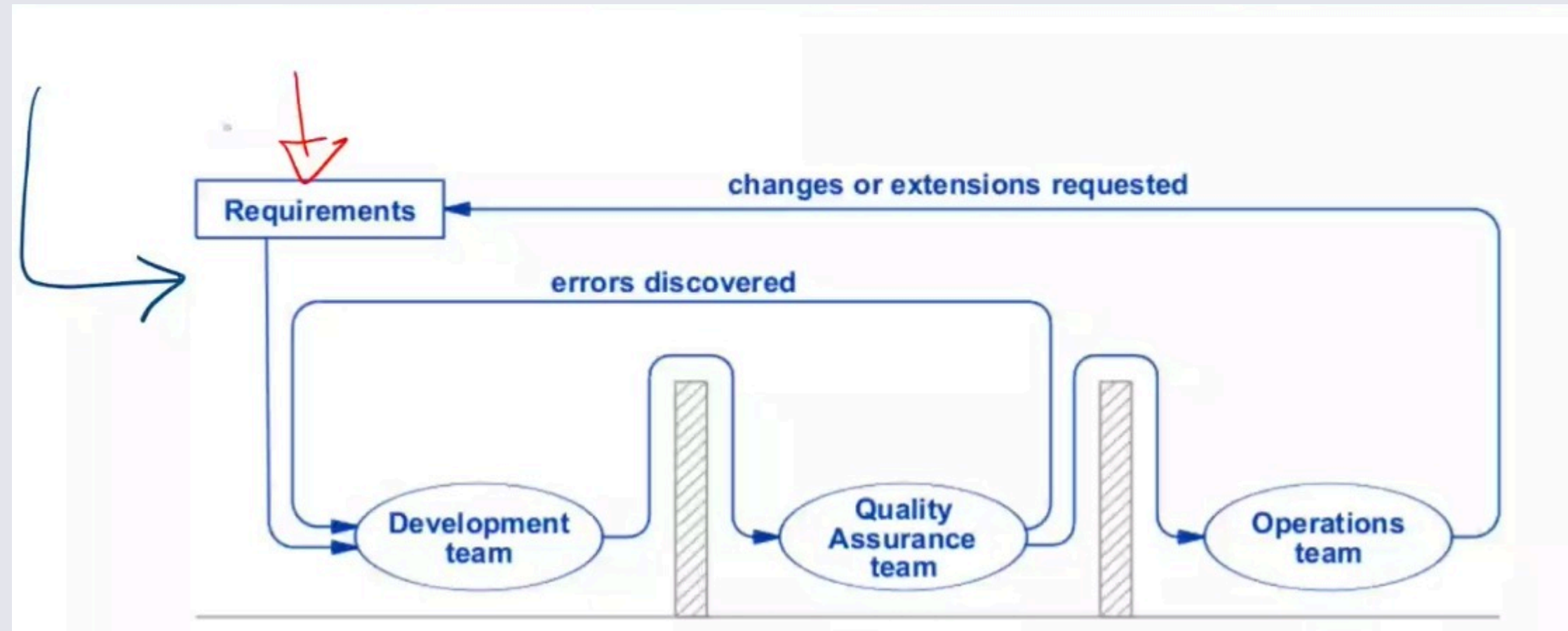
# 1. ¿Por qué surgió la necesidad de DevOps en el desarrollo de software?

Equipos  
tradicionales de  
Software

- 
- Equipo de desarrollo
  - Equipo de aseguramiento de calidad (QA)
  - Equipo de operaciones

Tradicionalmente, los equipos trabajaban por separado, esto ocasionaba que las actualizaciones tardaran mucho y **era difícil adaptarse rápido a los cambios tecnológicos y a las necesidades del negocio.**

2. Explica cómo la falta de comunicación y coordinación entre los equipos de desarrollo y operaciones en el pasado llevó a la creación de DevOps.



Antes de DevOps, Desarrollo buscaba rapidez en entregar nuevas funciones, mientras que Operaciones priorizaba la estabilidad. La falta de comunicación entre ambos causaba conflictos, fallos en producción y retrasos en la entrega.

DevOps surgió para integrar ambos equipos, mejorar la colaboración y automatizar procesos

3. Describe cómo el principio de mejora continua impacta tanto en los aspectos técnicos como en los culturales de una organización.

Técnico	Cultural
Implementación de CI/CD	Comunicación y transparencia
Automatización de procesos	Enfoque en el desarrollo profesional

#### 4. ¿Qué significa que DevOps no se trata solo de herramientas, individuos o procesos?



- No solo se basa en técnicas de automatización, sino una mentalidad de DevOps.
- No solo son individuos, sino actitudes en equipo
- No solo son procesos aislados, sino un cambio de infraestructura completo.



## 5. Según el texto, ¿cómo contribuyen los equipos autónomos y multifuncionales a una implementación exitosa de DevOps?

- Rompen con los silos organizacionales
- Facilitan la comunicación
- Aceleran los ciclos de desarrollo



# INFRAESTRUCTURA COMO CÓDIGO

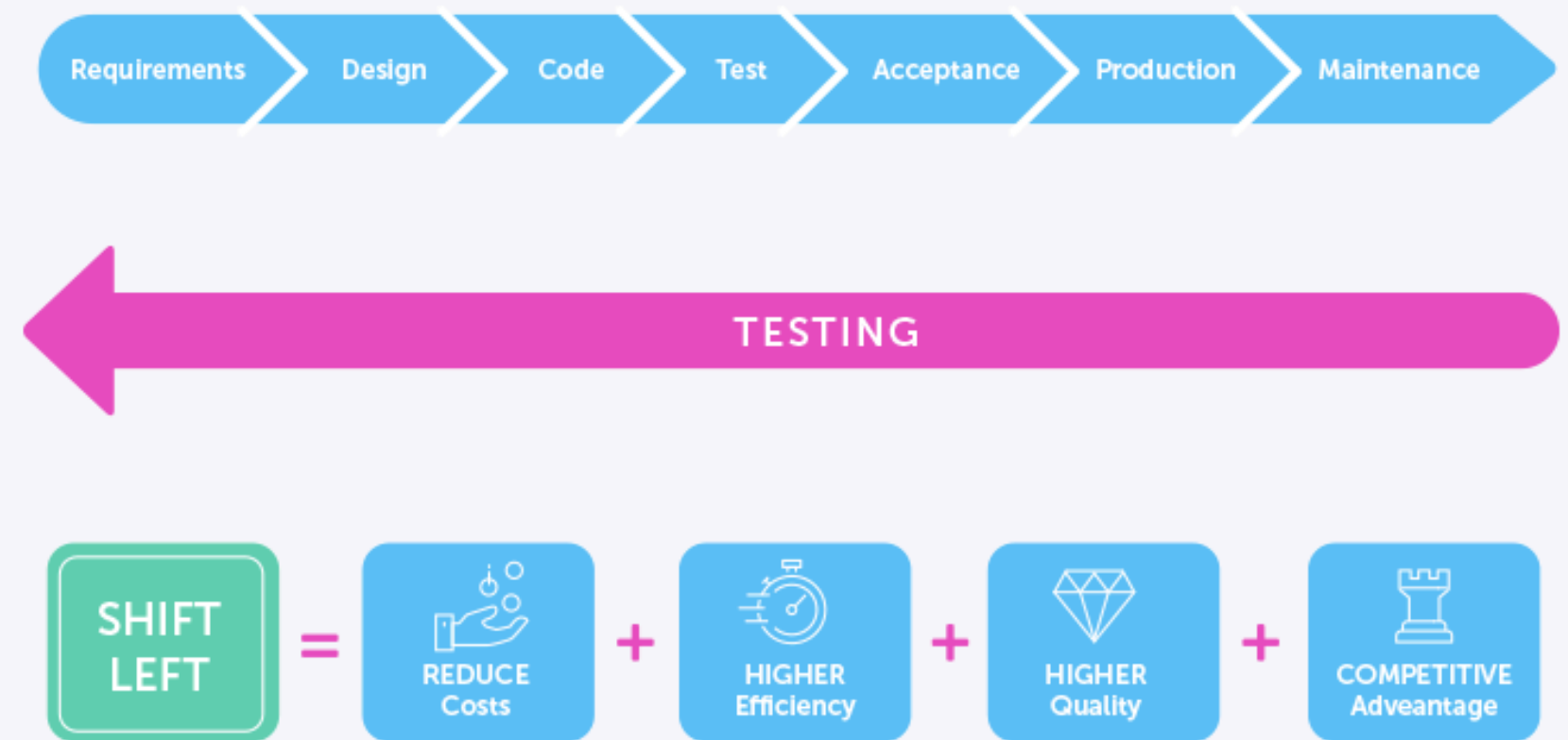




# Preguntas de reflexión

## ¿Qué significa “desplazar a la izquierda”?

Las pruebas se realizan con mayor anticipación en el desarrollo, moviéndolas “a la izquierda” en la línea de tiempo del desarrollo.



# ¿Cómo mejora IaC a la consistencia y escalabilidad?



01

Mejora la consistencia porque la infraestructura es fácilmente reproducible con archivos de configuración.

02

Mejora la escalabilidad porque los recursos pueden ser fácilmente asignados o desasignados como módulos.

## ¿En qué se diferencian el monitoreo y la observabilidad?

- El monitoreo es parte de la observabilidad.
- El monitoreo recolecta información de los sistemas involucrados.
- La observabilidad es un análisis más completo del comportamiento del software.

## ¿Por qué es crucial la observabilidad en sistemas complejos?

- Los equipos de desarrollo necesitan analizar la información del sistema de forma eficiente.
- La observabilidad proporciona herramientas para esto sin tener que conocer a profundidad cada componente del sistema.

# ¿Qué impacto tiene un desarrollador experimentado?



## Cuellos de botella

Un desarrollador experimentado puede identificar mejor las etapas que ralentizan o toman más esfuerzo entre los desarrolladores, aplicando herramientas o mejores configuraciones según sea la necesidad.

## DevSecOps

Los desarrolladores con mayor experiencia son más conscientes de buenas prácticas para mantener una alta seguridad y calidad en el producto final, apoyándose de las técnicas y herramientas de DevSecOps.



# ¿Qué es InnerSource y cómo reduce los silos organizacionales?

- InnerSource es una metodología de trabajo en la que el software en desarrollo abre su código a toda la organización que la desarrolla.
- Esta metodología reduce los silos organizacionales al permitir que cualquiera puede leer, auditar, colaborar y usar partes del código desarrollado.

