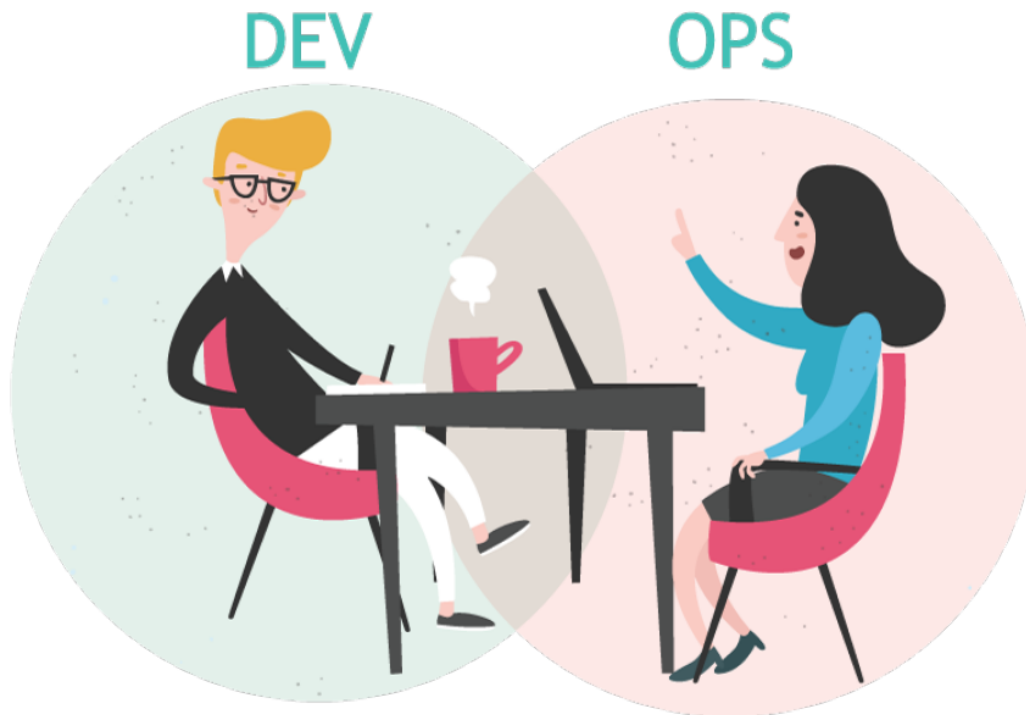


## ¿Qué es DevOps?



La palabra **DevOps** es una contracción de “Desarrollo” (Development) y “Operaciones” (Operations).

## ¿Por qué DevOps?

Comunicación tradicional entre desarrollo y operaciones.





## Definiciones de DevOps



No hay un acuerdo claro y universal sobre su definición.

Hay varias opiniones sobre qué es y qué no es **DevOps**, generalmente se define como una nueva forma de organización, una cultura o incluso una nueva forma de pensar.



## ¿Qué no es DevOps?



DevOps no es una estrategia para todos.

Hay gran diversidad de tecnologías empresariales y drivers a ser considerados para establecer la estrategia de adopción para **DevOps**.

DevOps no es automatización.

**DevOps** implica automatización. **DevOps** es más que automatización.

## ¿Qué no es DevOps?

DevOps no es una herramienta implementada.

Aunque hay herramientas que son usadas en **DevOps**, no deberíamos limitar su alcance a herramientas específicas como **Chefs** o **Jenkins**. Esto limita el amplio alcance como si una sola herramienta de automatización se equiparara con **DevOps**.

DevOps no es equipo de trabajo nuevo y separado de las demás áreas de TI.

Tener un equipo **DevOps** separado, anula el propósito de evitar las posibles fricciones y falta comunicación entre los desarrolladores y operadores de **TI** ya que crea un silo más.

### Año 2009

**John Allspaw** y **Paul Hammond**, dos empleados en jefe de **Flickr**, presentaron una charla clave, titulada *“Diez despliegues en el día: cooperación de Dev y Ops en Flickr”*. En esta charla **Allspaw** y **Hammond** señalaron poderosamente como el conflicto llevó a *“apuntarse con el dedo”* entre los desarrolladores y operadores al culparse entre ellos.

Ellos señalaron que la única manera de construir y desplegar un software viable era hacer a operaciones y desarrollo integrados y transparentes.

Inspirado por esto, **Debois** organizó su propia conferencia (*DevOpsDay*). El nombre de este movimiento fue reducido a **DevOps** después de esta convención.

El primer evento **DevOps** fue organizado por **Debois** en Ghent, Bélgica.

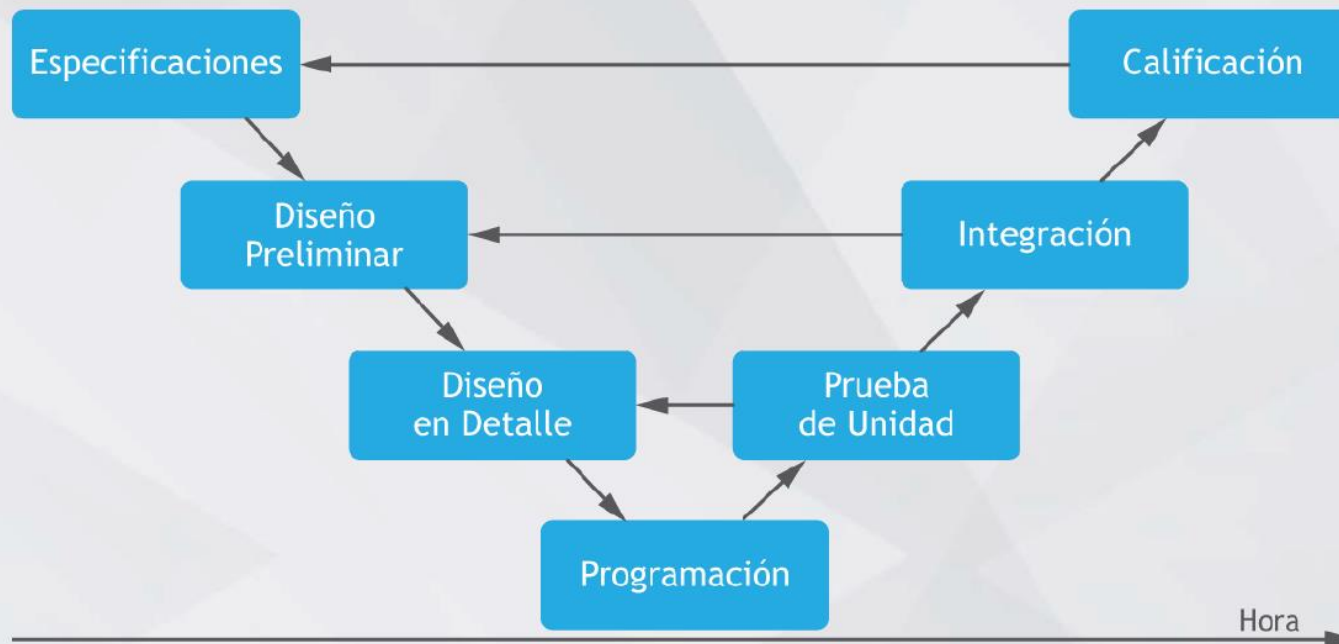
## Modelo en Cascada

El modelo de ciclo de vida en cascada se comenzó a diseñar en 1966 y se terminó alrededor de 1970.

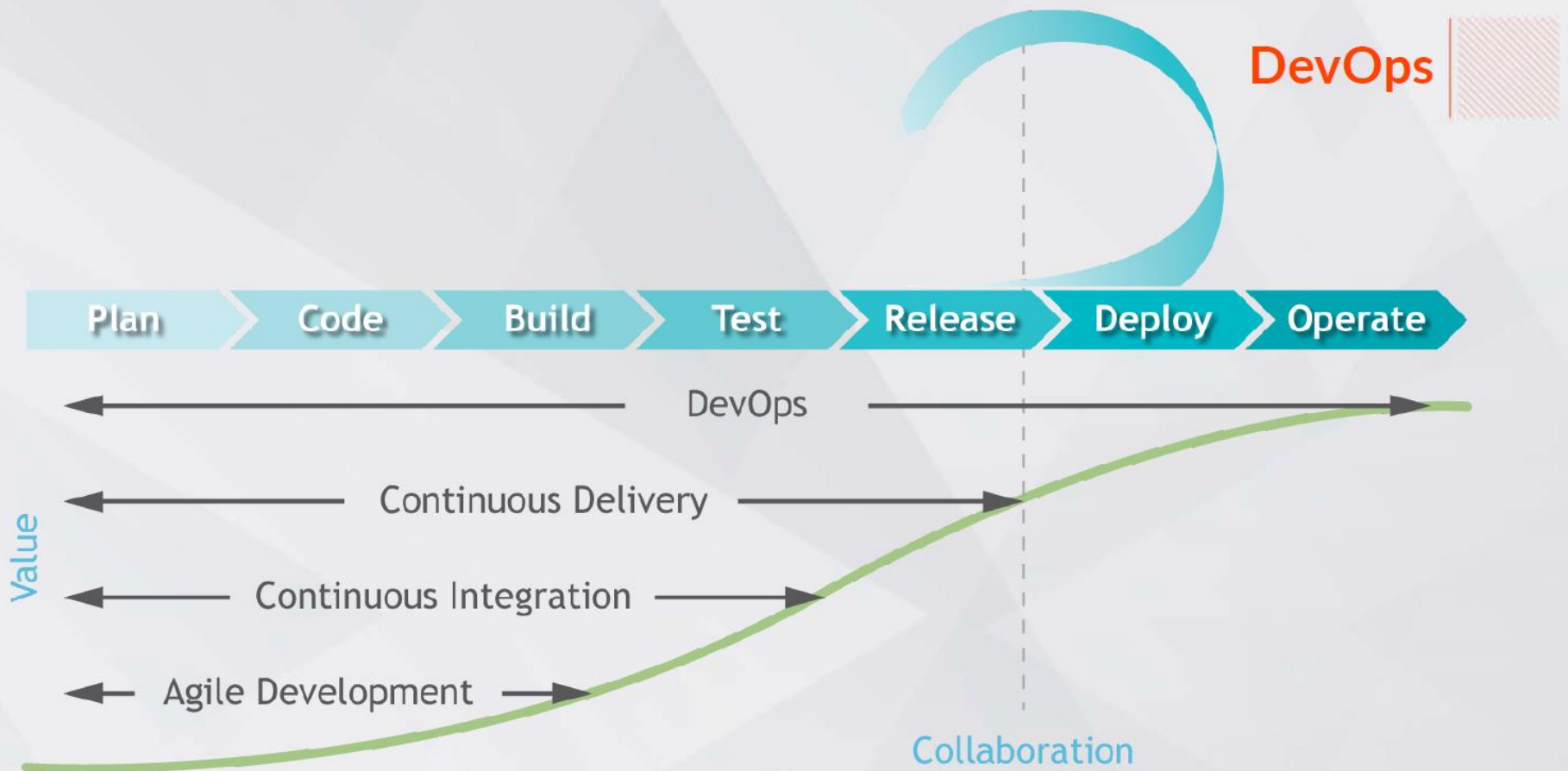
Se define como una secuencia de fases donde al final de cada una de ellas se reúne la documentación para garantizar que cumple las especificaciones y los requisitos antes de pasar a la fase siguiente.



## Modelo V



El modelo de ciclo de vida **V** proviene del principio que establece que los procedimientos utilizados para probar si la aplicación cumple las especificaciones ya deben haberse creado en la fase de diseño.



## Herramientas DevOps

El uso de herramientas no es hacer DevOps.

Hacer **DevOps** sin el uso de herramientas es complejo si entendemos que **DevOps** tiene componentes de automatización. **DevOps** es y será un tema cultural y no será tan solo el conjunto de herramientas.

Miremos un ejemplo, las herramientas no hacen cultura, pensemos en una empresa que desea hacer que sus reuniones internas inicien a tiempo. Si la compañía compra un software de gestión de reuniones, ¿logrará que las reuniones inicien a tiempo?

Si la cultura interna es iniciar reuniones tarde, se deberá trabajar en cambiar la cultura interna primero. La herramienta por más buena que sea no logrará hacer que la cultura cambie.

*¿Qué herramientas para DevOps o llamadas DevOps conocen los candidatos a la Certificación DevOps Essentials? ¿Qué hacen puntualmente esas herramientas?*

