

# **Tendencias en Metodologías de trabajo de Desarrollo Ágil y Metodologías Híbridas**

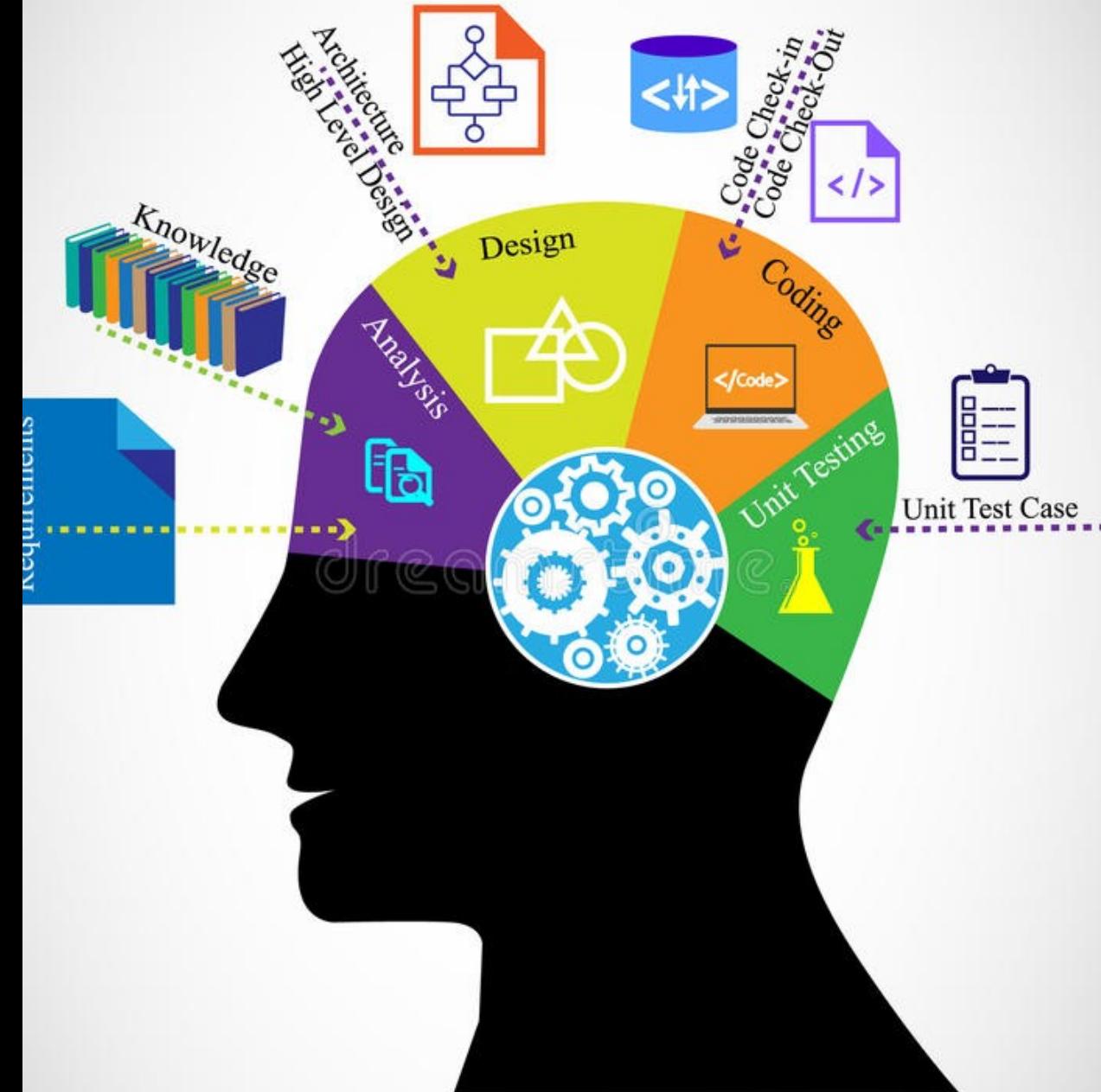
---

**UNIDAD 2**



# Metodologías de desarrollo de software

- Una metodología de desarrollo de software se refiere a un marco de trabajo que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información o proyectos de software.
- *El marco de trabajo utiliza herramientas, modelos y métodos para asistir al proceso de desarrollo de software.*





## 2.1

# DevOps

---

- 2.1.1 Marco de trabajo e importancia
- 2.1.2 Métodos de DevOps
- 2.1.3 Cadena de herramientas
- 2.1.4 Prácticas de DevOps
- 2.1.5 Ventajas

*DevOps es una metodología de desarrollo de software basada en la integración entre desarrolladores y administradores de sistemas, que permite acelerar la integración de nuevas funcionalidades, testear de forma automática y solventar errores de forma ágil.*



# DevOps

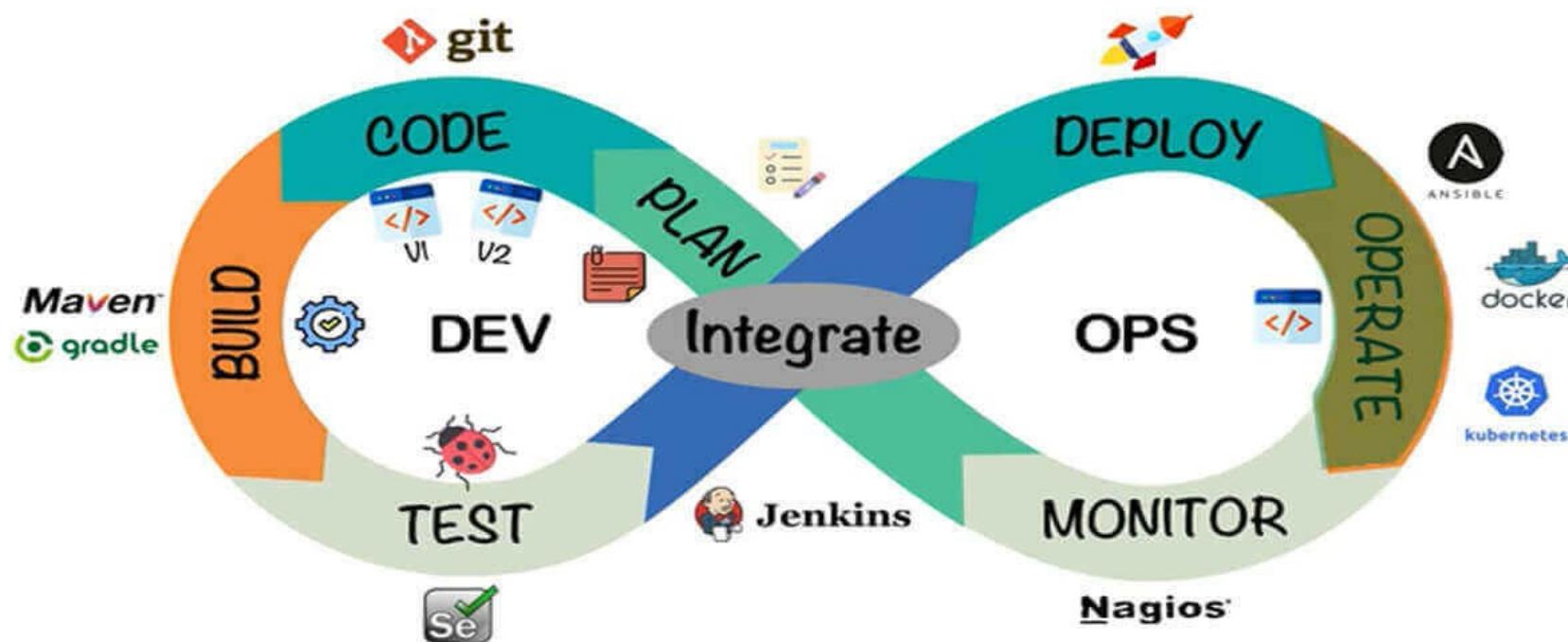


- El término DevOps, que es una combinación de los términos **development (desarrollo)** y **operations (operaciones)**, designa la unión de personas, procesos y tecnología para ofrecer valor a los clientes de forma constante.

# CULTURA DevOps

- DevOps es un **conjunto de prácticas** que enfatizan la **colaboración** y la comunicación entre los desarrolladores de software y los profesionales de operaciones de TI, automatizando el proceso de entrega de software y los cambios de infraestructura.

The DevOps culture is implemented in several phases with the help of several tools



# DEVOPS

DevOps

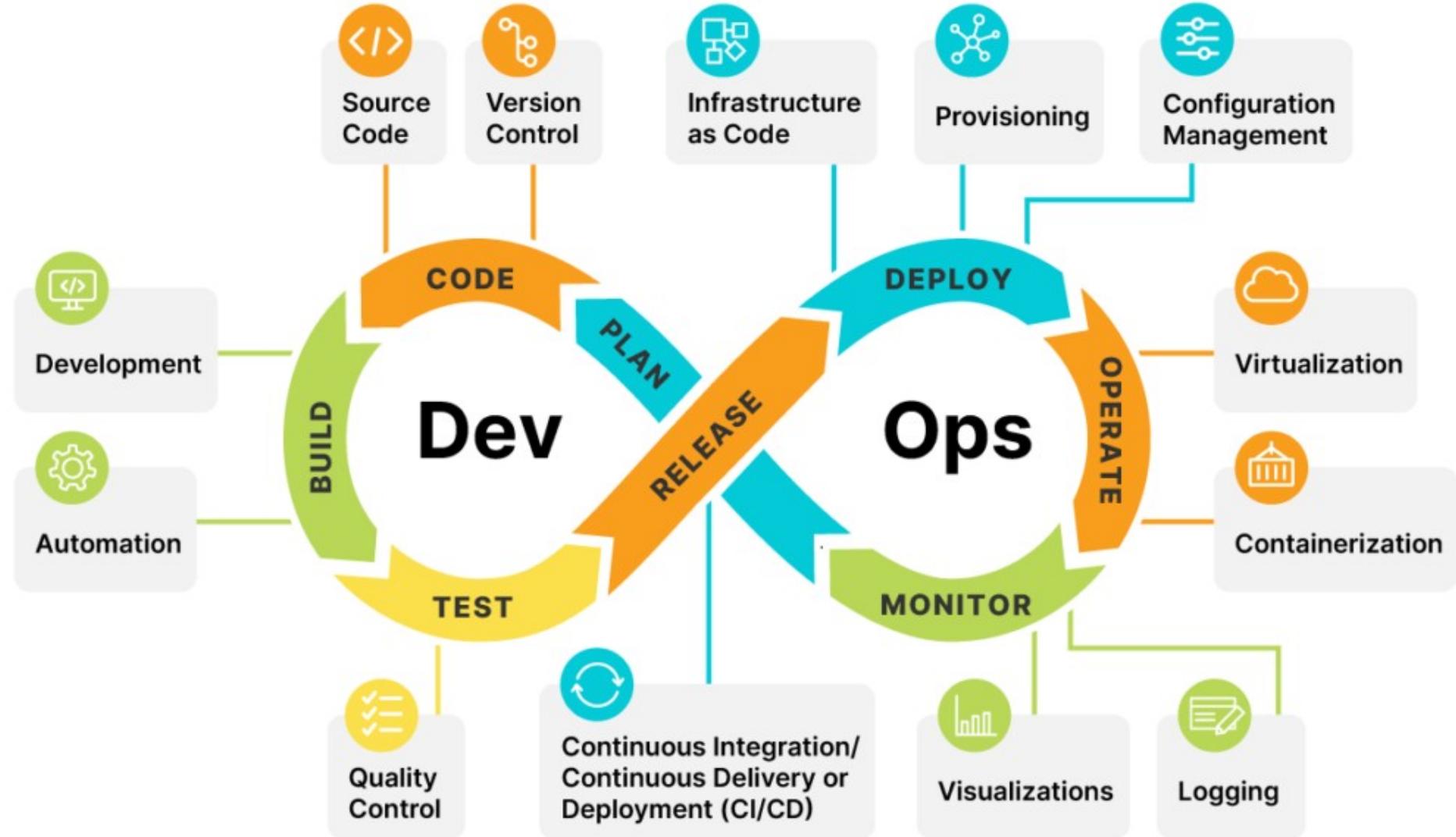
Solución

Implementación  
Desarrollo

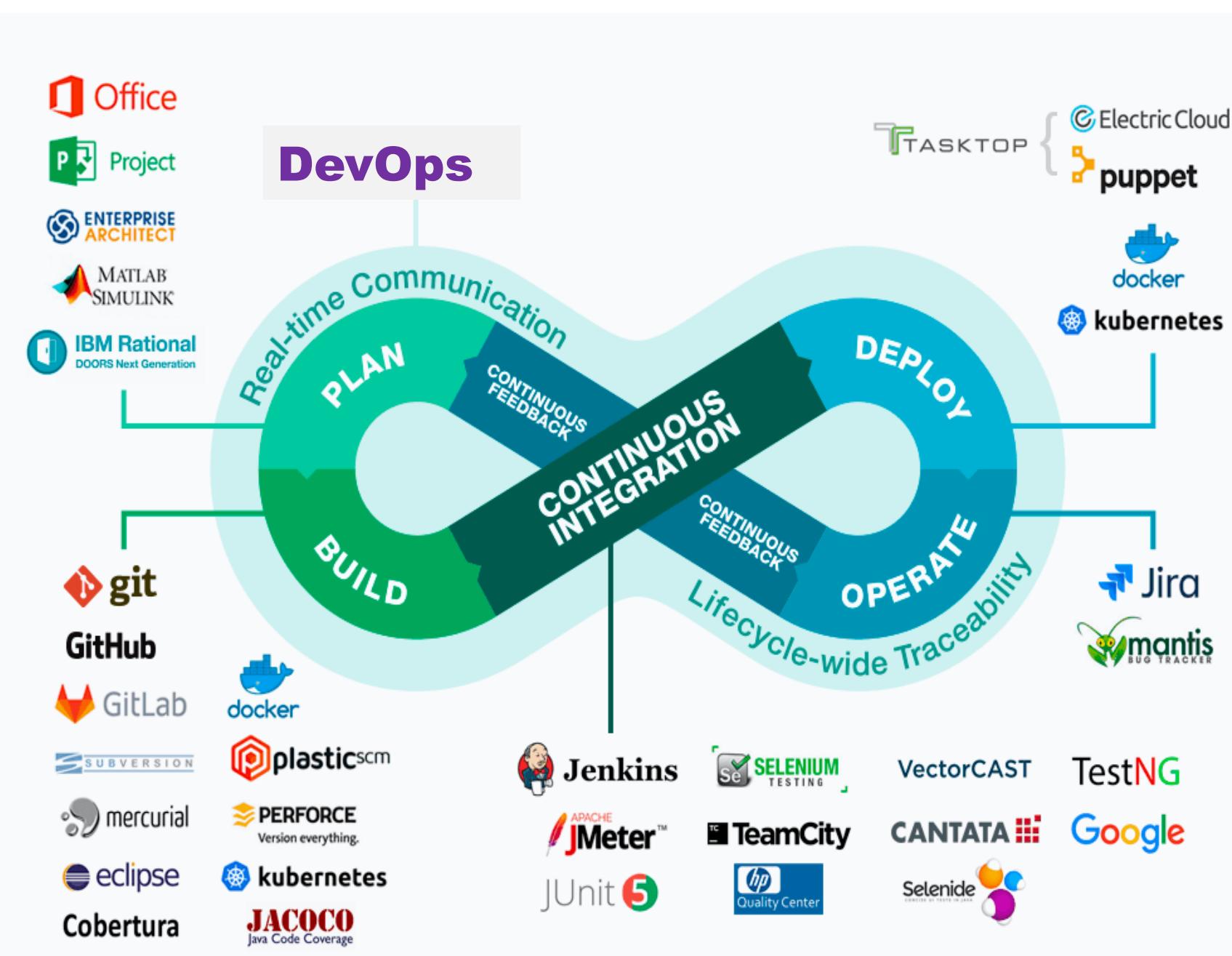
Testing

Despliegue

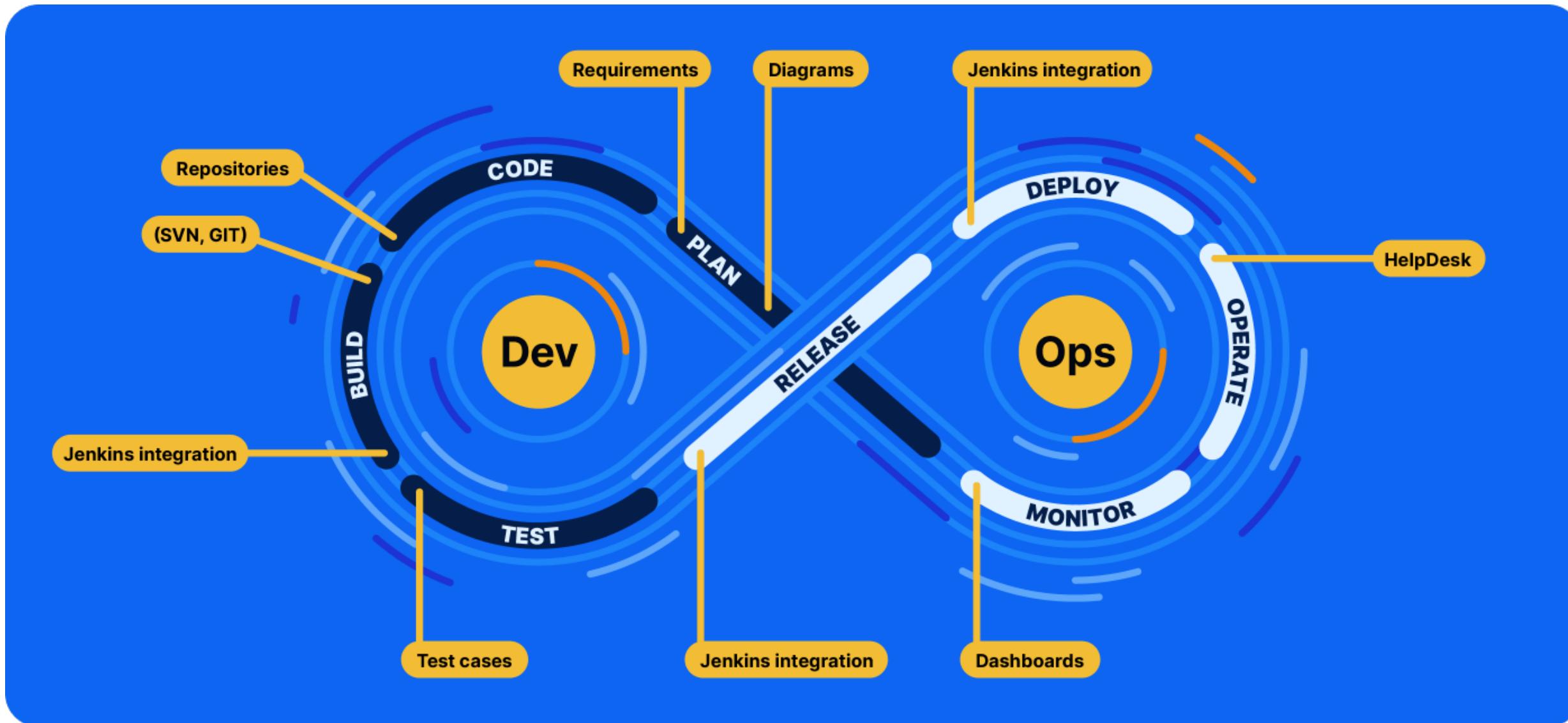




# Cadena de herramientas



# Conjunto de Herramientas



# HERRAMIENTAS Y FASES

- **Planificación.** En esta fase se definen los requisitos y valores empresariales. Algunas herramientas de muestra son Jira o Git, con las cuales se puede hacer un seguimiento de los problemas conocidos y llevar a cabo la gestión de los proyectos.

# HERRAMIENTAS Y FASES

**Codificación.** Esta fase implica el diseño del software y la creación del código. Algunas herramientas de muestra son GitHub, GitLab, Bitbucket o Stash.

# HERRAMIENTAS Y FASES

- **Compilación.** En esta fase se gestionan las versiones y las compilaciones del software, y se utilizan herramientas automatizadas que ayudan a compilar y crear paquetes de código para publicarlos después para la producción.
- Se utilizan repositorios de código fuente o repositorios de paquetes que también «empaquetan» la infraestructura que se necesita para el lanzamiento del producto.
- Algunas herramientas de muestra son Docker, Ansible, Puppet, Chef, Gradle, Maven o JFrog Artifactory.

# HERRAMIENTAS Y FASES

- **Prueba.** Esta fase incluye la realización de pruebas continuas (manuales o automatizadas) para garantizar la calidad de la programación. Algunas herramientas de muestra son JUnit, Codeception, Selenium, Vagrant, TestNG o BlazeMeter.

## 2.2. DESIGN THINKING



# DESIGN THINKING >

Metodología basada en conocer las necesidades de las personas, para así poder resolverlas de forma innovadora y obtener resultados excepcionales.

## CARACTERÍSTICAS



Utiliza el pensamiento divergente



Modifica para obtener resultados optimos



Genera lluvia de ideas



Se trabaja en equipo

## ¿CÓMO FUNCIONA?



### 1 Empatizar

Ponerse en el lugar del cliente y conocer su verdadero problema.



### 2 Definir

Recopilar la información más importante y además identificar el problema.



### 3 Ideas

Todo el equipo aporta ideas para llegar a una solución.



### 4 Prototipos

Seleccionar la idea más indicada y aterrizarla.



### 5 Testea

Solución planteada y evaluada.

## BENEFICIOS PARA TU EMPRESA



Incrementa la capacidad de los empleados



Mejora la productividad



Maneja el trabajo en equipo

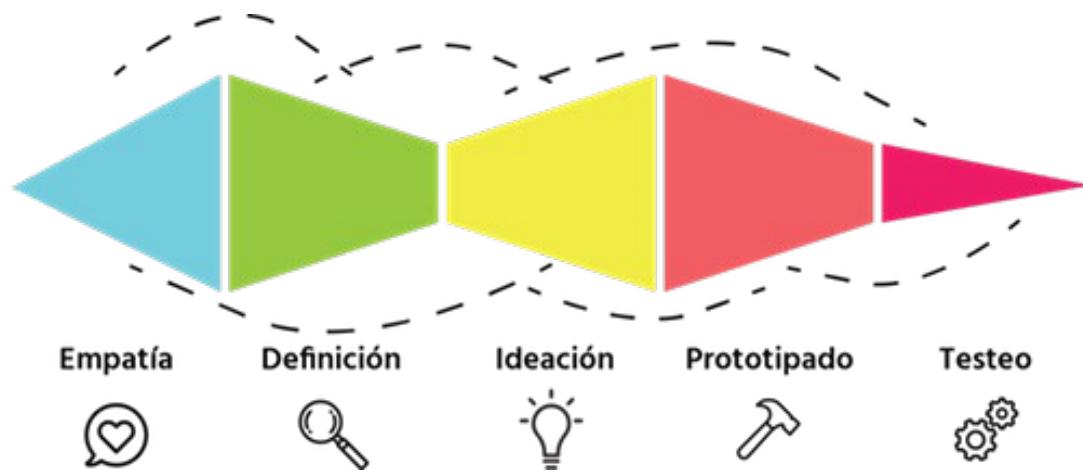


Refuerza el clima laboral



Aumenta el valor del producto o servicio

# FASES DE LA METODOLOGÍA



## Fase 1: Tener empatía

- Investigar las necesidades de tus usuarios. Se trata de entender empáticamente el problema que está tratando de resolver. Por lo tanto, esta fase suele comenzar por la investigación del usuario o consumidor.

## Fase 2: Definir

- Identificar las necesidades y los problemas de sus usuarios. Es hora de poner sobre la mesa la información recopilada durante la primera etapa. Se analizan todas las observaciones y se sintetizan para definir los problemas centrales que el equipo ha identificado.

# FASES DE LA METODOLOGÍA

## Fase 3: Idear:

- En esta fase, se trata de generar cuantas ideas sea posible. Se trata de «pensar fuera de la caja», buscar formas alternativas de ver el problema e identificar soluciones innovadoras para el planteamiento del problema.

## Fase 4: Prototipar

- Construir soluciones. Inicialmente se identifica la mejor solución para cada problema encontrado. A continuación, el equipo tiene que construir versiones económicas y muy primitivas de lo que sería el producto final, aunque sea en papel.



# FASES DE LA METODOLOGÍA

## Fase 5: Probar

- Probar los prototipos es la última de las fases del proceso de Design Thinking. Existe una figura -la de los evaluadores-, que se encarga de probar rigurosamente los prototipos.
- Aunque esta es la fase final, la metodología del design Design Thinking se basa en un modelo de trabajo iterativo. A raíz de los resultados obtenidos, los equipos a menudo utilizan ese como punto de partida. Por lo tanto, es posible volver a las etapas anteriores para realizar más iteraciones, alteraciones y refinamientos.



# MODELOS DE DESARROLLO ÁGIL

A man in a dark blue suit and glasses is seen from behind, reaching up towards a large circular graphic. The graphic features the words 'AGILE DEVELOPMENT' in bold, dark blue capital letters. The background of the slide is a collage of various business-related icons, including gears, charts, arrows, and people, all in shades of blue and grey.

## AGILE DEVELOPMENT

- SCRUM
- KANBAN

# METODOLOGÍAS ÁGILES

Son metodologías de desarrollo flexible de un proyecto.

Se utilizan para ayudar a los usuarios a responder más rápidamente a los requisitos cambiantes y permite a los usuarios evaluar de forma regular el rendimiento.

Tienen como principal objetivo desarrollar productos de calidad que respondan a necesidades cambiantes de los clientes.

Los proyectos se adaptan al día a día del trabajo y a las urgencias que puedan surgir.

# 12 PRINCIPIOS ÁGILES



Satisfacer a  
los clientes



Requisitos  
cambiantes



Entrega de  
valor frecuente



Colaboración diaria



Equipos motivados



Comunicación  
cara a cara



Medición del progreso



Ritmo de trabajo  
sostenible



Excelencia técnica



Simplicidad



Equipos  
autoorganizados



Mejora continua

# VENTAJAS Y DESVENTAJAS

Permite al equipo dividir el proyecto en etapas

Permiten adaptar el proyecto a medida que avanza.

Retroalimentación más rápida de los usuarios finales.

Flexibilidad para definir funciones prioritarias y establecer objetivos.

La estrecha interacción entre el equipo y el cliente garantiza un progreso constante.

- Al inicio del proyecto, es difícil determinar con precisión la cantidad de tiempo y dinero que se necesitará para completarlo, debido a los requisitos en constante cambio.
- El equipo necesita tener una base sólida y habilidades.
- Se requiere un alto nivel de interacción entre el cliente y los desarrolladores.
- Falta de atención a la documentación
- Existe una falta de límites del proyecto lo que provoca una expansión descontrolada.

# SCRUM

- Generalidades de Scrum
  - Roles
  - Reuniones
  - Artefactos
  - Gestión de un proyecto
- Scrum**

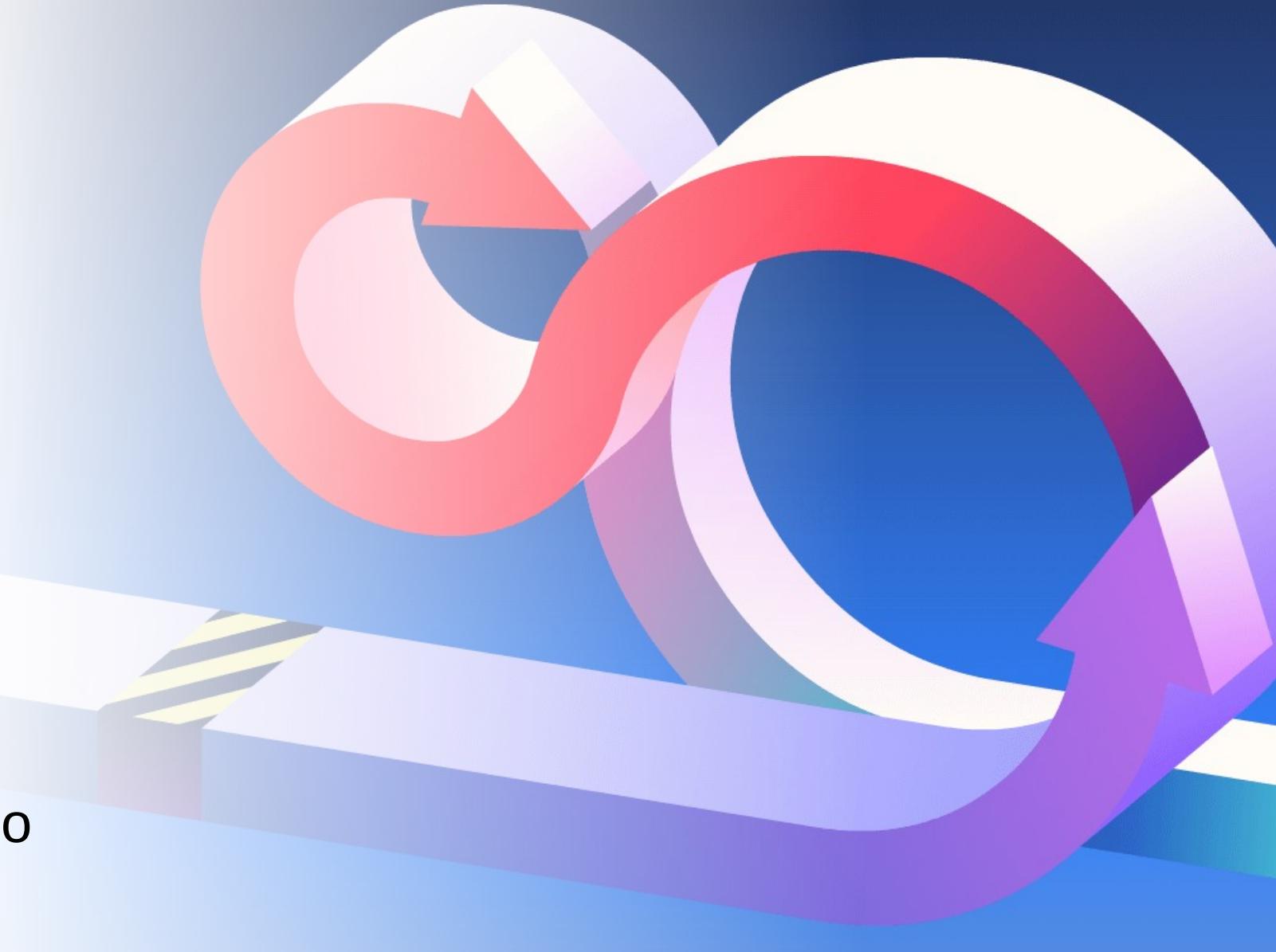
<https://www.youtube.com/watch?v=XDOayQluC78>



- Scrum es un proceso de gestión que ***reduce la complejidad*** en el desarrollo de productos para satisfacer las necesidades de los clientes, ***promueve la colaboración en los equipos.***



- 
- Scrum está basado en un modelo de proceso empírico, con respeto a las personas y ***basado en la auto-organización de los equipos*** para lidiar con lo imprevisible y resolver problemas complejos inspeccionando y adaptando continuamente.



# Scrum

## El equipo de SCRUM

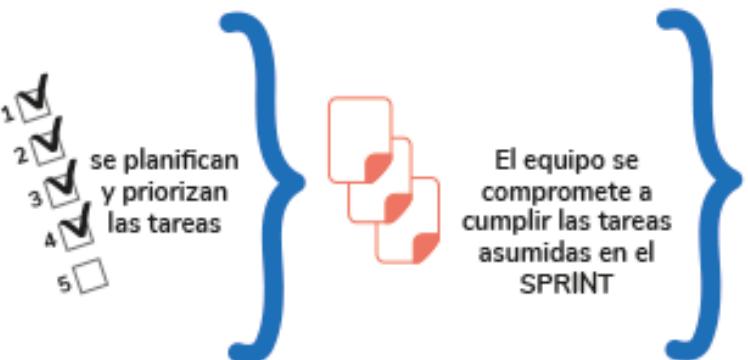


PRODUCT OWNER  
Captura Requerimientos  
Selección del equipo.

EQUIPO DE  
DESARROLLO

SCRUM MASTER

- 1
- 2
- 3
- 4
- 5



PRODUCT BACKLOG



SPRINT PLANNING  
MEETING



SE DECLARA  
EL ESTADO  
DE LAS TAREAS

## Proceso de la metodología Ágil



CONTROL DE PROYECTO  
GRÁFICA:  
Velocidad de Avance



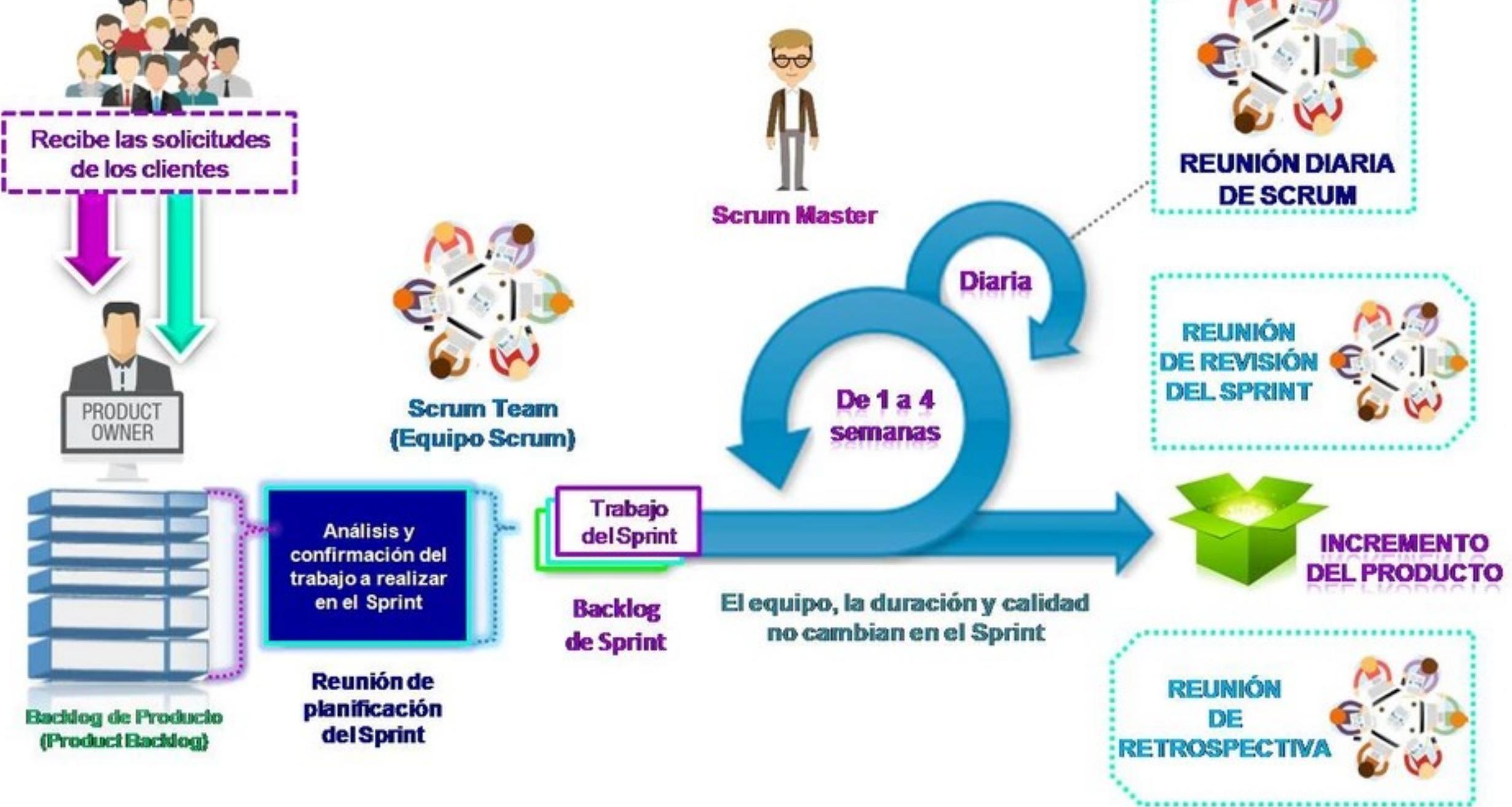
SPRINT REVIEW  
2 a 4 h.



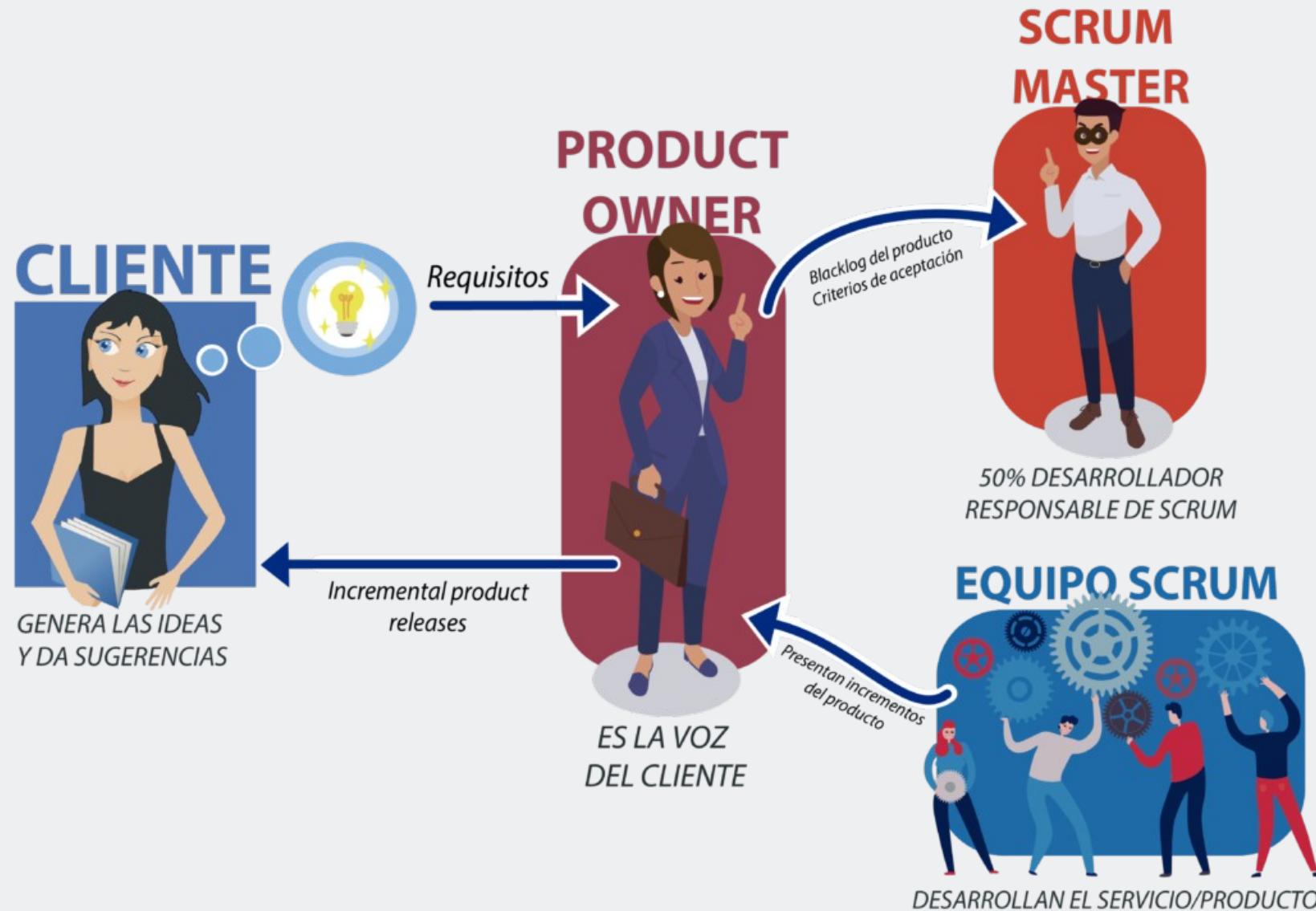
TRABAJO REA-  
LIZADO



SPRINT RETROSPECTIVE  
1 a 3 h.



# ROLES EN SCRUM





### CLIENTE

Es la persona que toma las decisiones finales, es quien selecciona o limita los objetivos y requisitos del proyecto.



### PRODUCT OWNER

Es la persona que toma las decisiones dentro del proyecto, es quien tiene el conocimiento del negocio y su visión del producto.



### SCRUM MASTER

Es la persona que se encarga de asegurar que el modelo Scrum utilizado sea el correcto.



### TEAM SCRUM

Es el grupo de persona o equipo del proyecto que estará trabajando con el proyecto.

# ROLES DE SCRUM



PRODUCT OWNER

# PRODUCT OWNER

- Su rol es similar a un líder de proyecto.
- Básicamente son los ojos del cliente, será la persona encargada del proyecto y de vigilar que se lleve a cabo de tal forma que cumpla las expectativas de lo que se espera.



# SCRUM MASTER

- En cada reunión que se realice, siempre debe estar un líder, este es el Scrum Master y ayudará en los problemas que hayan surgido.
- Es un “facilitador” el cual minimizará obstáculos, sin embargo no los omitirá.
- El Scrum Master debe ser una persona empapada de conocimientos sobre el lenguaje o lenguajes bajo los cuales se llevará a cabo el proyecto, de lo contrario no tendría como ayudar a solucionar problemas.



# EQUIPO SCRUM

- Básicamente es el núcleo de la metodología Scrum, pues es el equipo de desarrollo, encargado de lo que es la codificación del software y de cumplir los objetivos o metas propuestas por el Product Owner.

# CLIENTE

- El cliente también forma parte del equipo.
- En la metodología Scrum, el cliente tiene la capacidad para influir en el proceso, debido a que siempre estará empapado de él, ya sea que proponga nuevas ideas o bien haciendo algún tipo de comentario.



# **PROCESOS DE SCRUM**

---

**Product Backlog**

---

**Sprint Backlog.**

---

**Sprint Planning Meeting.**

---

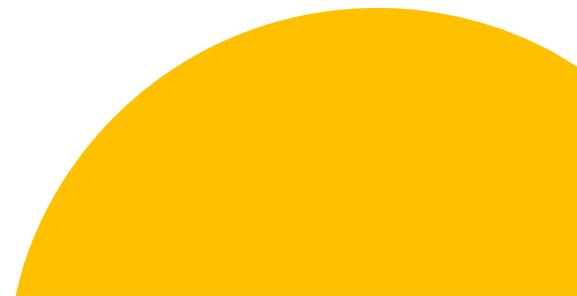
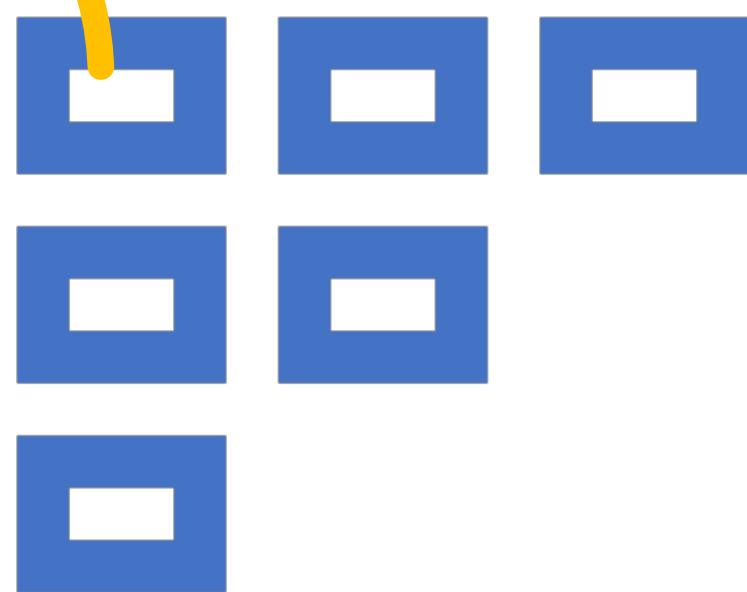
**Daily Scrum o Stand-up Meeting.**

---

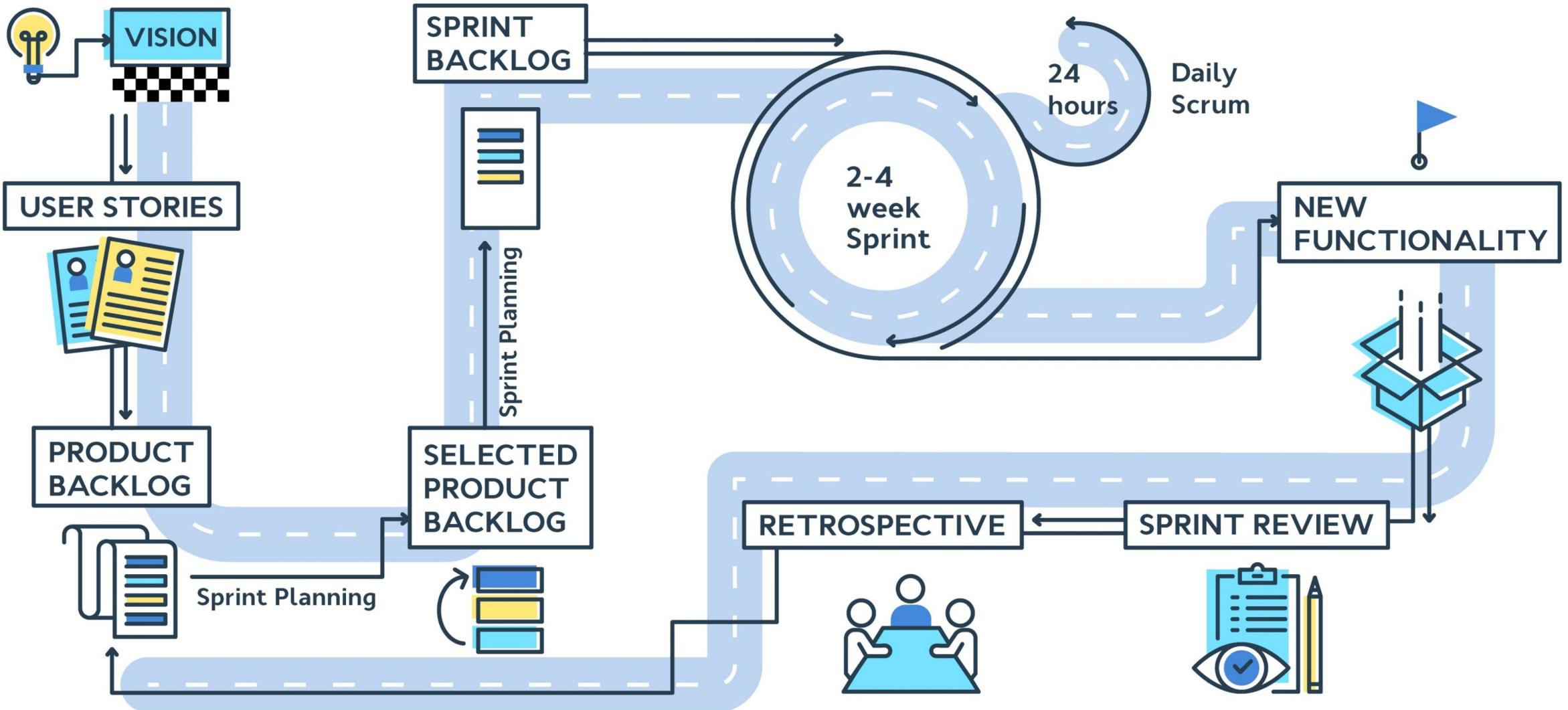
**Sprint Review.**

---

**Sprint Retrospective.**

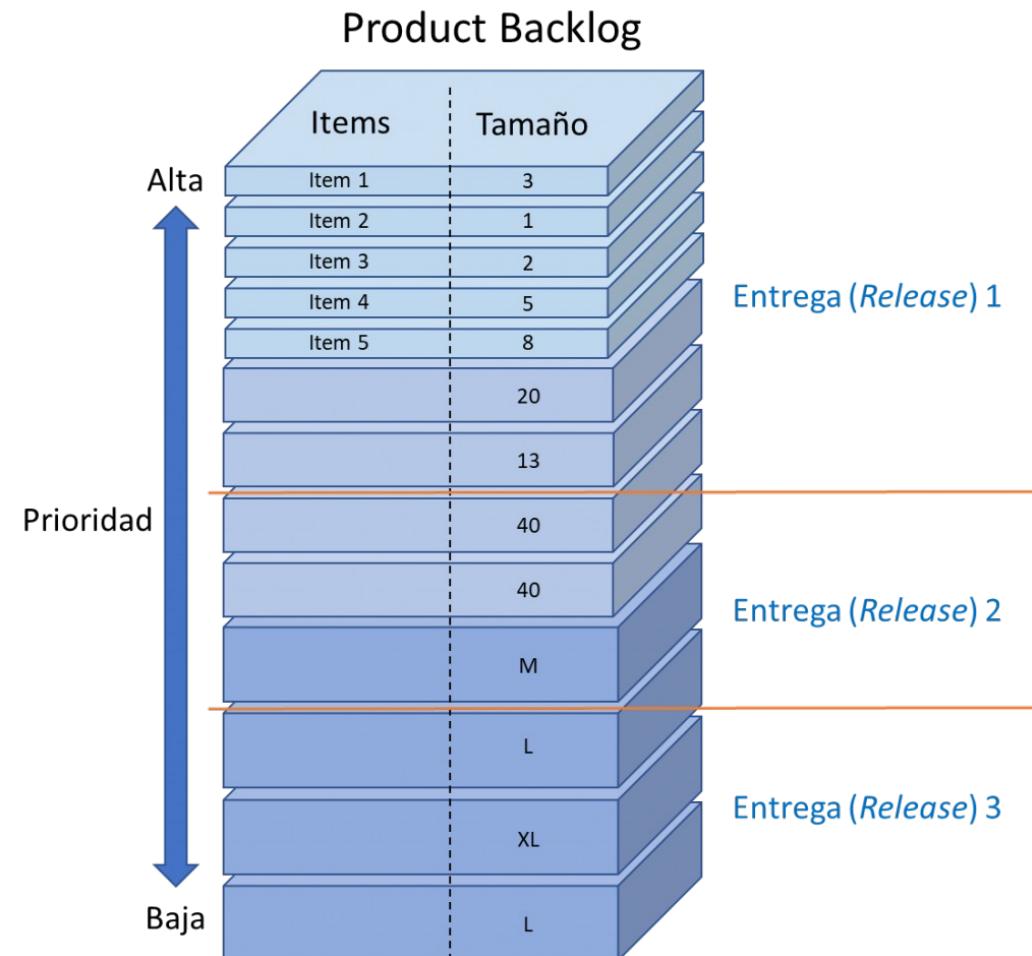


# SCRUM PROCESS



# PRODUCT BACKLOG

- El Product Backlog no es más que una lista de las funcionalidades del producto a desarrollar.
- Este debe ser elaborado por el **Product Owner**
- El Product Backlog debe estar ordenado de acuerdo a las prioridades del sistema de mas a menos, con la idea de que las cosas con mayor prioridad sean las que se realicen antes de cualquier cosa.
- De forma concreta, digamos que el objetivo base del Product Owner, es que nos de respuesta a la pregunta “¿Qué hay que hacer?”.



# SPRINT BACKLOG

Sprint Backlog

Producto Web de Compra de Libros	Pendiente	En Progreso	Finalizado
Nº Sprint: 04	Historia #1	Tarea 1.2 Tarea 1.3	Tarea 1.1 Tarea 1.4
Objetivo del Sprint <i>El objetivo de este Sprint es que el usuario pueda completar una búsqueda de libros por autor y ordenarla por precio de compra ascendente y descendente, así como por año de publicación.</i>			Historia #2 Tarea 2.1 Tarea 2.2 Tarea 2.3
	Historia #3 Tarea 3.5	Tarea 3.3 Tarea 3.4	Tarea 3.1 Tarea 3.2
	Tarea Técnica #04 Subtarea 4.1	Subtarea 4.2	
	Spike #1 Bug 001 Bug 005 Bug 012		

- Una vez que se tiene el Product Backlog terminado, entonces aparece el primer **Sprint Backlog**.
- ¿Qué es el **Sprint Backlog**? Consiste básicamente en seleccionar algunos de los puntos escritos en el Product Backlog, los cuales procederán a ser realizados.
- Sin embargo en este punto el Sprint Backlog tiene como requisito marcar el tiempo en que se llevará a cabo el Sprint.

# SPRINT PLANNING MEETING.

- Antes de iniciar un Sprint, el cual es la fase de desarrollo, se realiza lo que es un Sprint Planning Meeting.
- Es una reunión que se realiza para definir plazos y procesos a efectuarse para el proyecto establecido en el Product Backlog.
- Cada Sprint, se compone de diversos features, que son procesos o subprocesos que se deben realizar, puede ser la creación de un logo, la gestión de contenido, el diseño visual, etc.



# **DAILY SCRUM O STAND-UP MEETING.**

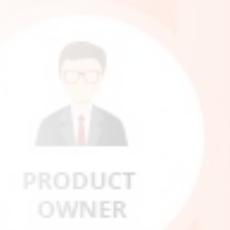
- Son reuniones diarias mientras se está llevando a cabo un Sprint, para responder las siguientes preguntas: ¿Que hice ayer?, ¿Qué voy a hacer hoy, ¿Qué ayuda necesito?.
- El Scrum Master será el encargado de determinar la solución de los problemas y cada complicación que suceda.



# SPRINT REVIEW MEETING

## SPRINT REVIEW

- ❑ El Sprint Review es una reseña de lo que fue el Sprint.
- ❑ Consiste específicamente en la revisión del Sprint terminado y para este punto ya tendría que haber algo que mostrarle al cliente, algo realmente visual o tangible para que se pueda analizar un cierto avance.

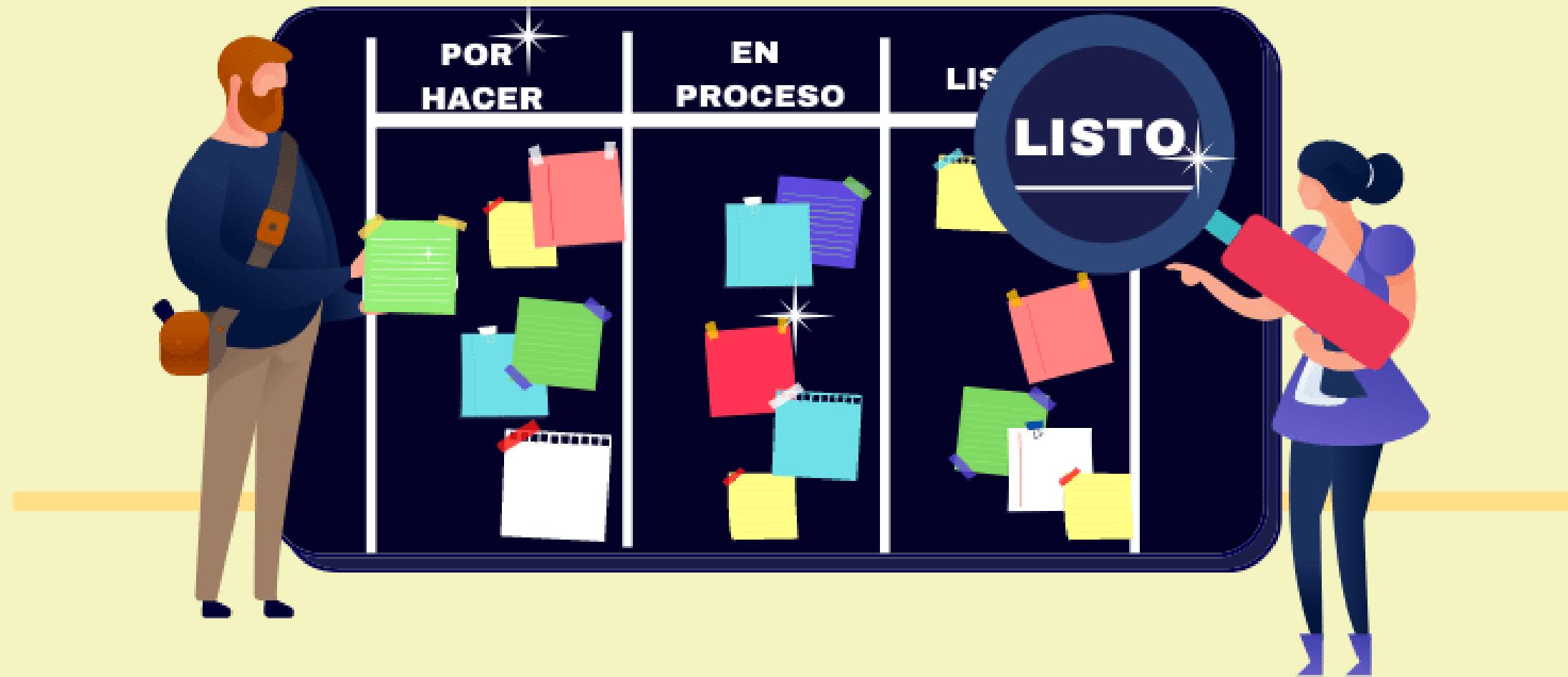


# SPRINT RETROSPECTIVE



- Permite al equipo analizar los objetivos cumplidos, si se cometieron errores, visualizarlos y tratar de no cometerlos nuevamente mas adelante.
- También sirve este proceso para lo que son la implementación de mejoras.

# KANBAN



<https://www.youtube.com/watch?v=WP6Nt5XV980>

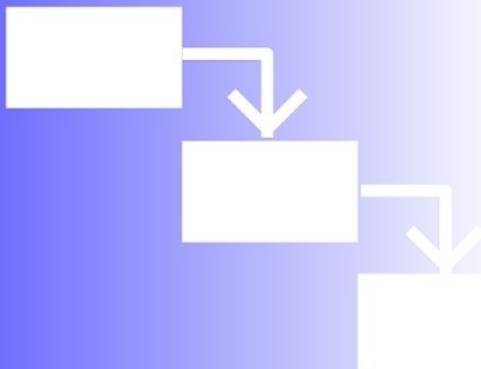
# Híbrido

## Metodologías híbridas

Ágil



Cascada

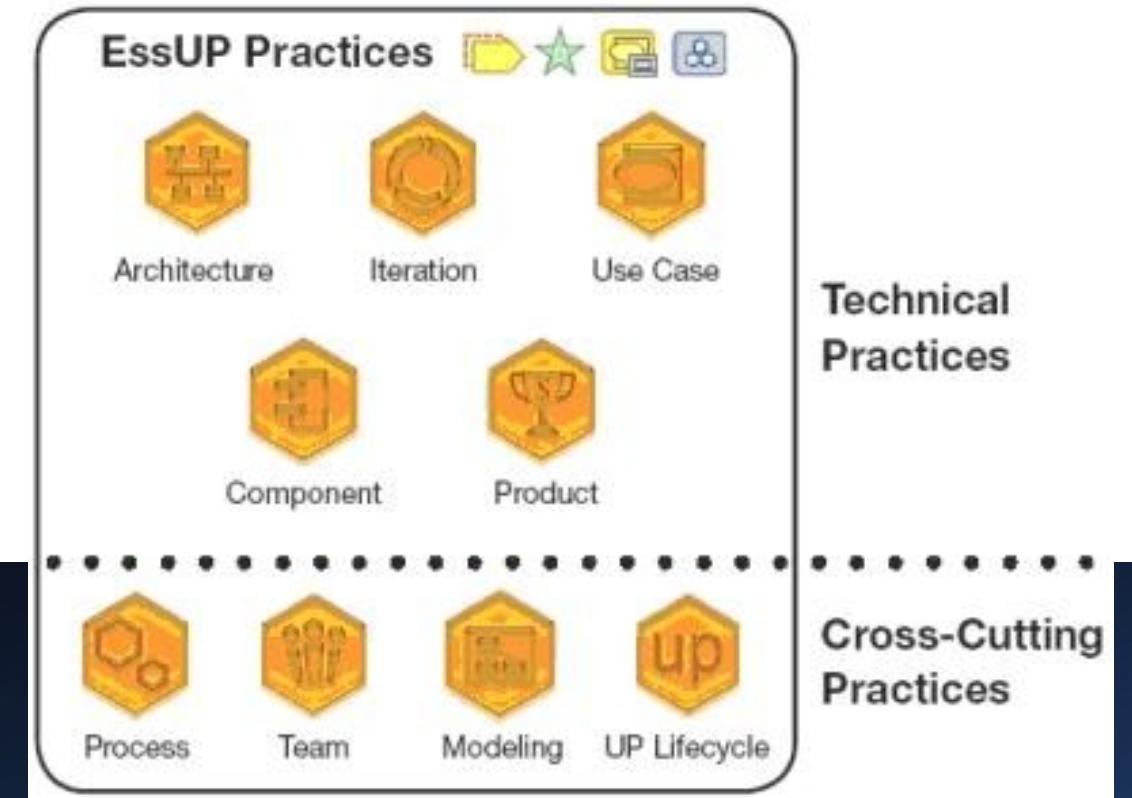


- Ess UP

# Ess UP

Metodología híbrida que combina dos metodologías : RUP y SCRUM

Creada por Ivan Jacobson





# **Ess UP : QUE ES?**

---

- El Proceso Unificado Esencial (EssUP) es ágil, iterativo, ligero y un proceso adecuado para usar en la mayoría de los tipos de proyecto.
- Se centra en proporcionar un conjunto mínimo de prácticas esenciales en una forma que sea atractivo, fácil de usar y muy flexible.

# **ESS UP : SOLO LO ESENCIAL**

---

- Todo lo que no es esencial se ha eliminado, las prácticas que se mantienen han mejorado sustancialmente y algunas ideas nuevas e importantes se han añadido para simplificar y fortalecer el proceso.



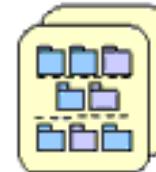
# PRÁCTICAS ESENCIALES

Las prácticas esenciales de esta metodología están divididas en:

- 4 prácticas transversales
- 5 prácticas técnicas

## Technical Practices

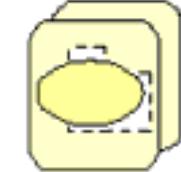
### EssUP Practices



Architecture



Iterative



Use Case



Component



Product

## Cross-Cutting Practices



Process



Team



Modeling

# PRÁCTICAS TRANSVERSALES

- ***Process Essentials***

Asegura una mejora continua de los procesos y ayuda a adoptar y afianzar las nuevas formas de trabajar.

- ***Team Essentials***

Enfocada a conseguir un equipo de trabajo colaborativo más efectivo.

- ***Modeling Essentials***

Describe un enfoque ágil para modelado que le permite adoptar un nivel apropiado de detalle, mejorar la comunicación del equipo y reducir el riesgo del proyecto.

- ***Unified Process Lifecycle Essentials***

Ofrece un conjunto de fases e hitos que ayudan a planificar y monitorizar proyectos iterativos.

# PRÁCTICAS TÉCNICAS

- ***Architecture Essentials***

Basado en architecture driven development permite asegurar que la arquitectura elegida para el proyecto es adecuada a su propósito.

- ***Iterative Essentials***

Adopción de un desarrollo iterativo mediante timeboxing para administrar y monitorizar el proyecto y sus riesgos.

- ***Use-Case Essentials***

Le permite capturar sus necesidades en de manera ágil y utilizarlos para impulsar la desarrollo y la prueba de la solución.

- ***Component Essentials***

Desarrollo del software de forma simple, escalable y aplicandotest-driven.

- ***Product Essentials***

Captura la esencia de la gestión de productos para que pueda estar más cerca de sus clientes e identificar sus versiones principales