

Clase 8:

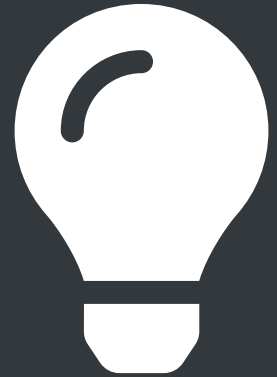
Arquitectura ágil

DigitalHouse >
Coding School

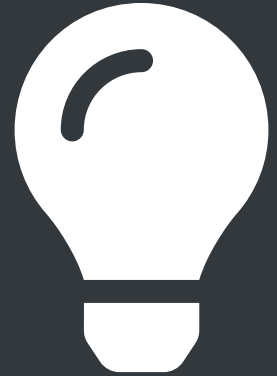


**Certified
Developer**
The Ultimate Tech Degree

¿Cómo estamos?

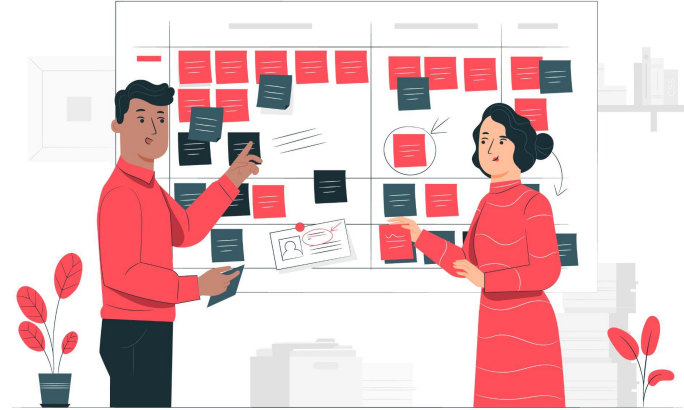


**¿Qué vimos en la
clase anterior?**



¿Que vimos la clase anterior?

1. Métricas Ágiles



ACTIVIDAD: REPASO GENERAL



Actividad 1

Ubicar las palabras en las diferentes etapas del desarrollo de software con la que se relacionan.

Arquitectura Ágil

Kanban de Ciclo de Vida

1- Kanban de Ciclo de Vida

Ubica las palabras en las diferentes etapas del desarrollo de software con la que se relacionan

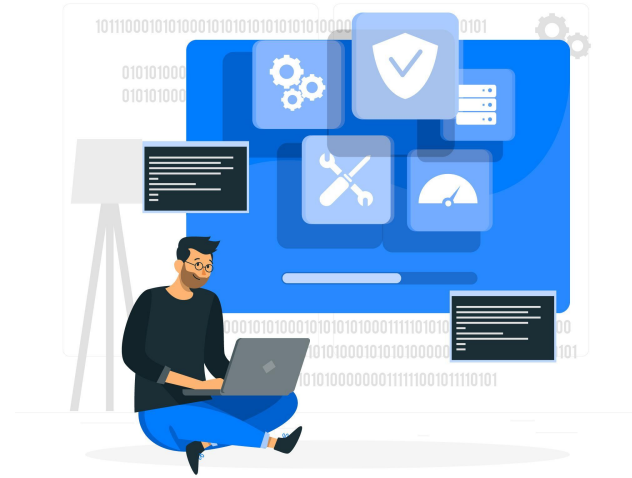
| | | | | | | | |
|-----------|-----------------|------------|--------------|---------------|-------------|---------|-----------------|
| Marketing | Hardware | Product | Organización | Product Owner | UI/UX | QA | DevOps |
| Marketing | Cloud Computing | API | Full Stack | Frontend | Mobile Apps | Backend | Cloud Computing |
| Cloud | API | Deployment | API | Deployment | API | API | API |
| API | API | API | API | API | API | API | API |
| API | API | API | API | API | API | API | API |

KANBAN

| 1- Análisis | 2- Diseño | 3 Construcción | 5-Pruebas | 6-Despliegue | 7- Mantenimiento |
|-------------------------------|-----------|----------------|-----------|--------------|------------------|
| | | | | | |
| Cross a todo el ciclo de vida | | | | | |

¿Que vamos a ver hoy?

1. Calidad del software.
2. Equipos en tecnología.



1 | Modelado de software y calidad

Construcción de una casa para “Fido”

Puede hacerlo una sola persona.

Requiere:

1. Modelado mínimo.
2. Proceso simple.
3. Herramientas simples.



Construcción de una casa

Construida eficientemente y en un tiempo razonable por un equipo.

Requiere:

1. Modelado.
2. Proceso bien definido.
3. Herramientas más sofisticadas..



¿Qué pasa cuando por
querer hacer algo rápido
lo hacemos mal?



“

Es preferible construir tu producto con calidad desde el inicio, que controlarla luego.

W. Edwards Deming



”

Las metodologías nos permiten gestionar la calidad en el desarrollo de un producto de software.



Evitan costos de corrección y retrabajo.



Mejoran satisfacción de cliente.



Controlan riesgos.



Reducen mantenimiento.

**Primero hacer
las cosas bien,
luego hacerlas
rápido.**

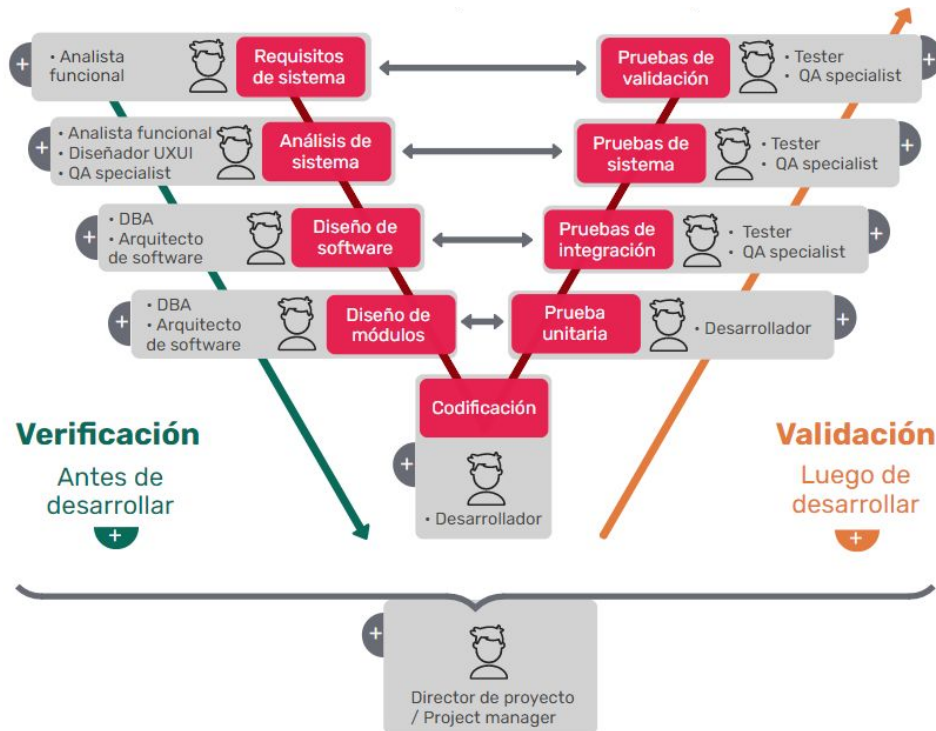


¿Cómo aseguramos la calidad?

- Usando metodologías probadas.
- Aplicando técnicas y herramientas de ingeniería.
- Asegurando alto nivel de consenso.
- Haciendo revisiones cruzadas en etapas tempranas.
- Priorizando la previsión más que la corrección.
- Estandarizando el proceso de optimización de calidad.
- Midiendo la calidad (queue size bugs).



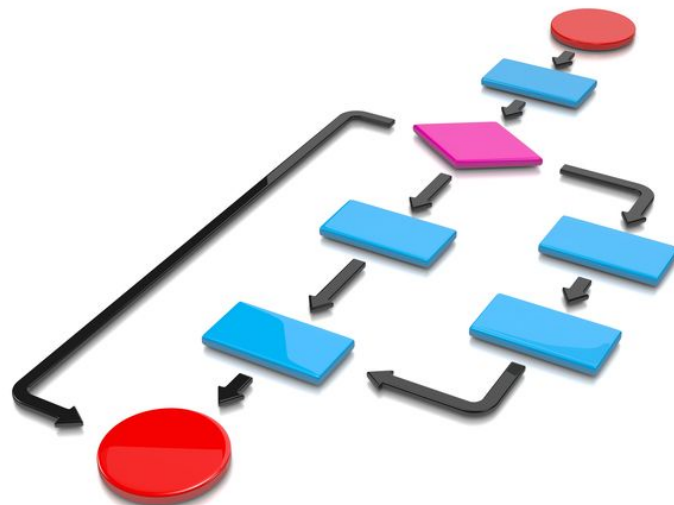
Calidad durante todo el ciclo de vida del desarrollo de software



DEBATE: TÉCNICAS DE MODELADO DE SOFTWARE

En grupos de Zoom, debatamos sobre RUP y UML.

¿Para qué sirve el modelado de software?
¿Dónde lo aplicarían? ¿Es importante para las empresas? UML en un #



ACTIVIDAD UML



Actividad 2

Son un equipo de desarrollo diseñando un software de cocina de pizza virtual. Deben realizar un **diagrama de flujo**

Proceso:

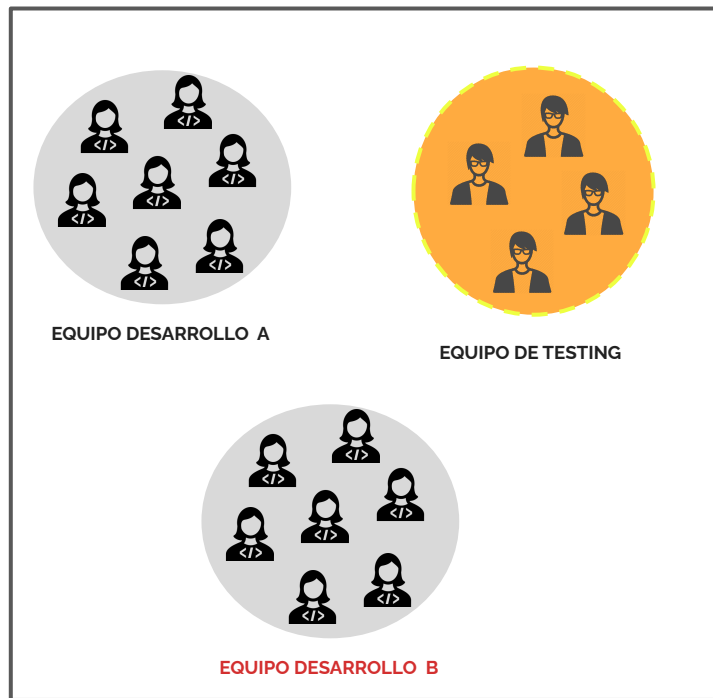
El usuario ingresa gusto de la pizza y el dinero

1. El sistema procesa el pedido.
2. Todos los gustos son válidos.
3. Si el dinero alcanza: cocina la pizza, sino rechaza el pedido
4. El sistema le devuelve al usuario la pizza, el recibo y si hay, vuelto

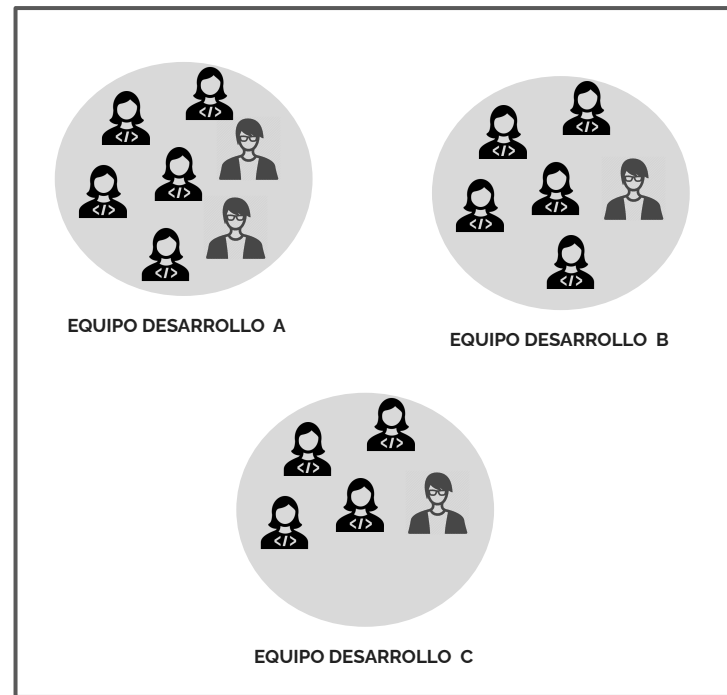


2 | Recapitulando

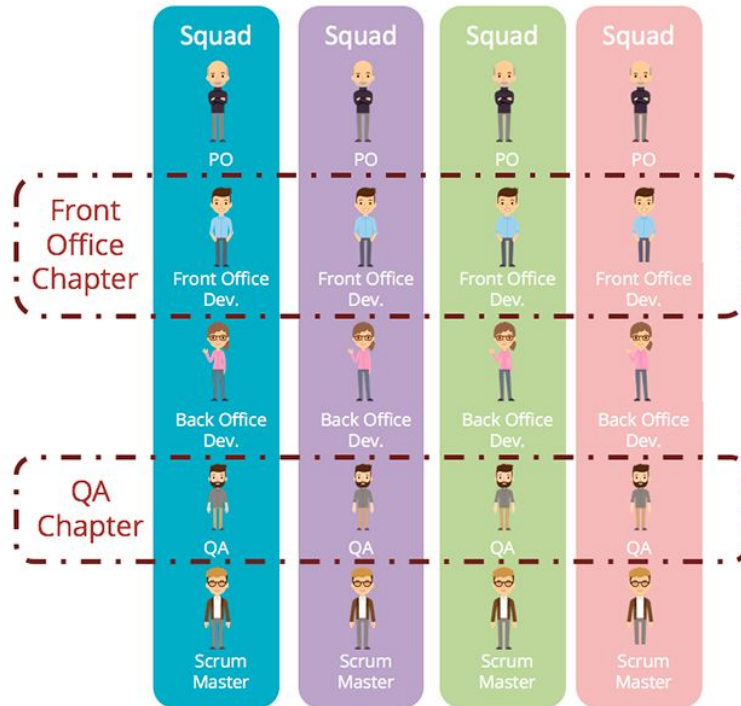
Organizaciones dev teams



VS.



Spotify Rules

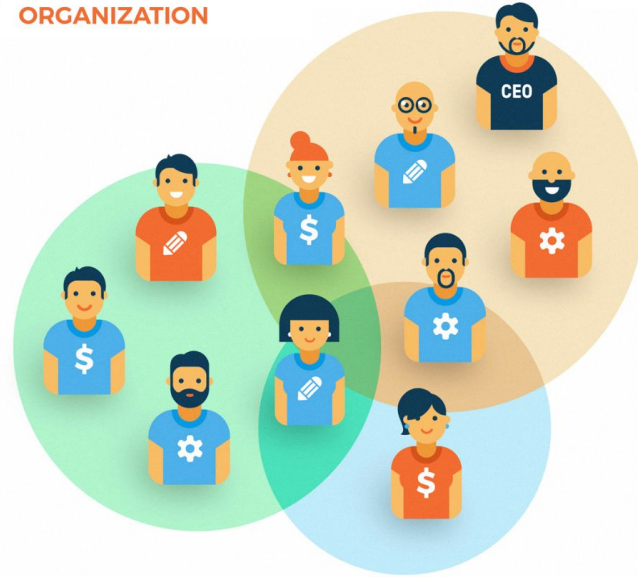


Fuente: [Deloitte](#).

HIERARCHICAL ORGANIZATION

VS

HOLACRATIC ORGANIZATION



Fuente: [Railware](#).





**¿Cómo llegamos
hasta acá?**

Lectura para la próxima clase:

Material asincrónico
Clase 9.



DigitalHouse>
Coding School