**Introduccion a DevOps**

Definicion: es un conjunto de practicas destinadas a reducir el tiempo entre el compromisio de un cambio en una aplicación y el cambio que se coloca en la producción normal, al tiempo que garantiza una alta calidad.

En la industria del desarrollo es muy habitual que las aplicaciones o sistemas que funcionan en un determinado ambiente, no funcionen en otro distinto en el que se quiere desplegar o probar la misma. O sea existe una descoordinación entre equipos de desarrollos que no trabajan en manera conjunta y se genera malentendidos entre las partes.

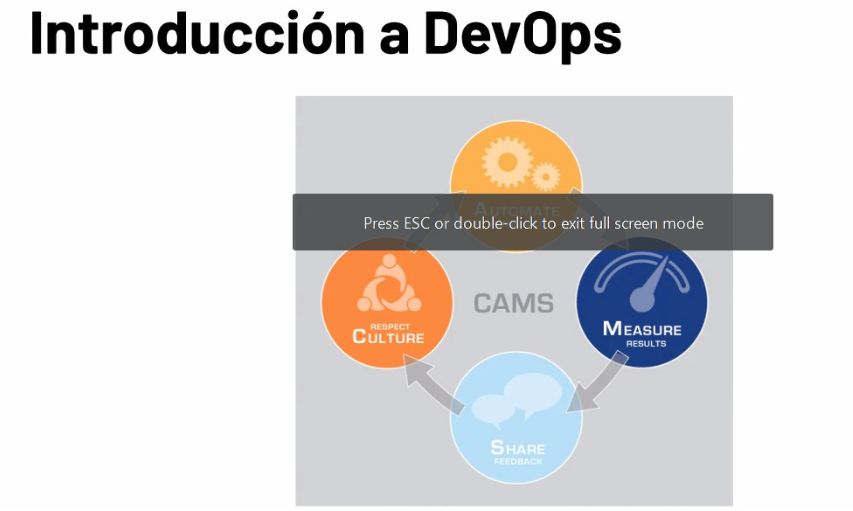
Podemos decir que tenemos una barrera entre las partes.

La idea del devops es funcionar en el medio del equipo de desarrollo y el equipo de operaciones para que no existan estos inconvenientes y funcionar como un coordinador para que estas dos partes trabajen uniformemente. Evitar esto significa evitar pérdida de tiempo y de costos entre otras cosas.

….

El devops está involucrado desde que nace el código, desde que se está pensando el código, hasta que queda operativo en la ultima etapa. Como devops debemos tener un conocimiento minimo de lo que ocurre en cada una.

**Pilares de DevOps**



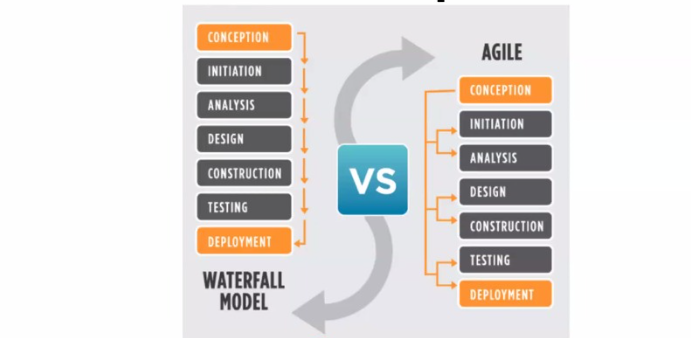
Es necesario que todo el equipo maneje el mismo mainset, de otra forma es muy difícil que se llegue a buen puerto, es necesario que todos manejen los mismos valores

**Primero automatizar lo mas posible**, por ejemplo un build o un deploy de una aplicación, un backup de una base de datos, un deploy de un servidor, que son cosas que van a ocurrir siempre

**Segundo trabajar orientado a resultado**, o sea buscando resultados. Esto no quiere decir que si estoy asignado a una tarea sea yo el encargado de realizarla si no que es un trabajo en equipo y es parte del equipo que el resultado se llegue a concretar. Definicion: es un conjunto de practicas destinadas a reducir el tiempo entre el compromisio de un cambio en una aplicación y el cambio que se coloca en la producción normal, al tiempo que garantiza una alta calidad.

Tercero, es necesario el **feedback continuo** o compartir feedback. Es importante compartir el feedback malo y también el bueno, siempre de manera constructiva y con el objetivo en mejorar.

Cuarto, **respetar la cultura** y todo lo mencionado anteriormente, sin eso nada va a funcionar. Todos debemos compartir los mismos valores y trabajar todos de la misma manera



El de la izquierda es un modelo de desarrollo en cascada, lo que significa que no se continua con la etapa siguiente hasta que la anterior no está finalizada. Es utilizado en proyectos donde solamente los requerimientos se encuentran bien identificados y no van a sufrir cambios constantemente.

Este modelo tiene el problema de que demanda mayor tiempo de desarrollo ya que mientras no se finaliza una etapa no se continua con la siguiente y además el cliente no ve el resultado hasta el final del desarrollo, lo que puede hacer que los resultados no coincidan con la necesidad del cliente.

Este ciclo de desarrollo tiene mucho sobre costo si se tiene que volver atrás por algún error detectado, no es fácil hacerlo

Los paradigmas orientados a DevOps no van de la mano con una modalidad de desarrollo en Cascada, ya que no son agiles.

Devops va de la mano de desarrollos agiles, que básicamente son pequeñas iteraciones de todas las partes del cascada que se van iterando entre si hasta tener un mínimo de producto viable. El cliente en cada iteración va ir viendo el resultado, lo que nos permite ir corrigiendo y asegurando lo que estamos haciendo. Los paradigmas orientados a devops se apoyan mucho en metodologías agiles.

**Ciclo de desarrollo**



1: planificación de las tareas a desarrollar

2: análisis de los requerimientos

3: diseño

4: implementación de la aplicación

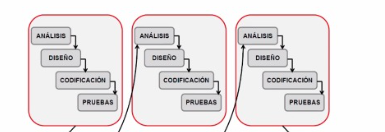
5: testing e integración

6: mantenimiento

La idea de metodologías agiles es que el ciclo del proyecto no quede estancado en mantenimiento y que siga iterando una y otra vez.

Metodologias afines

Incremental o iterativo



Se itera sobre todas las etapas, si se detecta algún problema o el cliente remarca algo, tendremos la posibilidad de en la próxima iteración mitigar el problema y arreglarlo.

Es utilizado en proyectos donde los requerimientos pueden ser cambiantes y se requiere obtener resultados a medida que se desarrolla.

Evolutivo o espiral

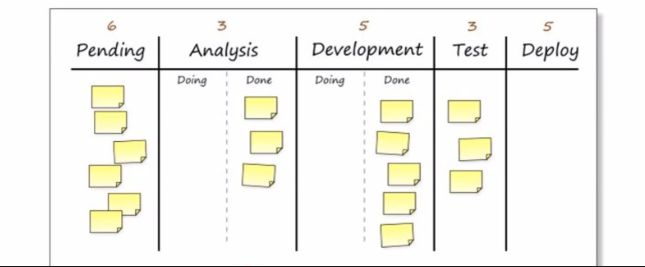


Es parecido al anterior pero agrega que al final de cada iteración se hará una evaluación de los riesgos

Posibilidad de volver hacia atrás en caso de detectar alguna anomalía.

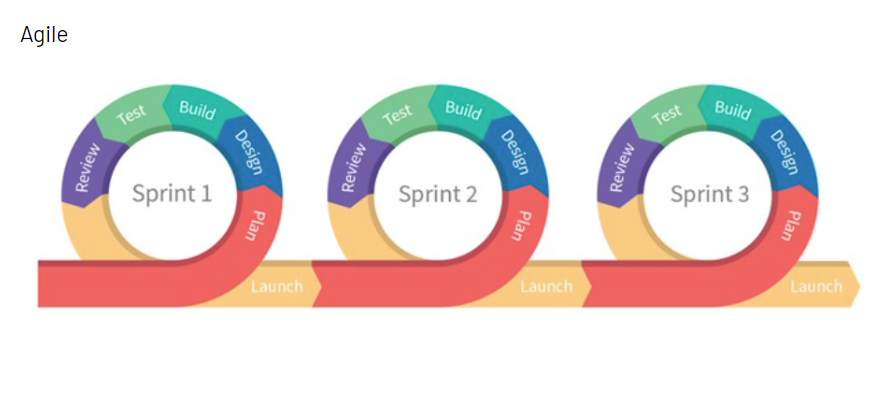
**Gestion de proyecto, tablero Kanban**

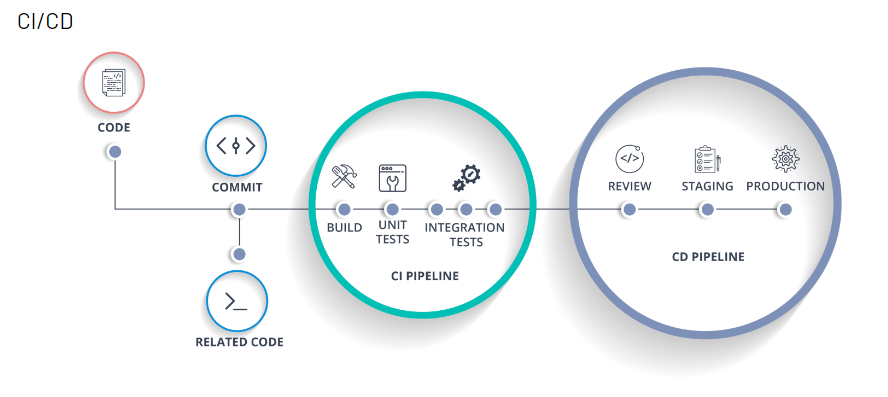
Es una manera rápida de ver como se encuentra la carga de tareas distribuidas por personas y por etapas de las tareas o del ciclo en el que estemos trabajando. Dependiendo de los equipos será la cantidad de columnas que va a tener el tablero, el equipo definirá la cantidad de ellas y la transacción que ser hará de una a otra.



Agile

Scrum se apoya mucho en esta modalidad, se





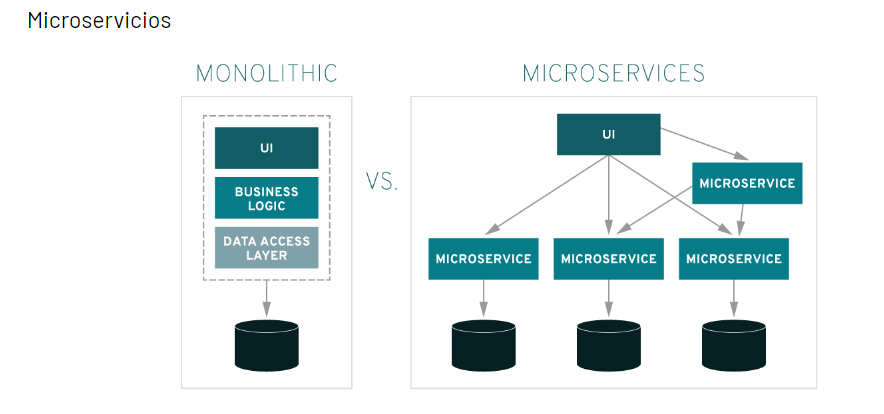
CI es la parte de continue integration, que va relacionado a todo lo que es el desarrollo, el build, el reléase, el testing

CD es la parte de continue development, es todo lo relacionado a operaciones o a la operativa, la revisación, el empaquetado, cambios en los archivos de configuración, puesta en producción.

En conjunto estas dos áreas se llaman continues delivery, es la integración de la etapa ci con cd, o sea la integración de continues integración con continue development.

Pipelines que nacen del ci cd los vamos a utilizar para tareas repetitivas y que sean necesarias automatizarlas.

El pipeline lo podemos ver como una cadena de producción.



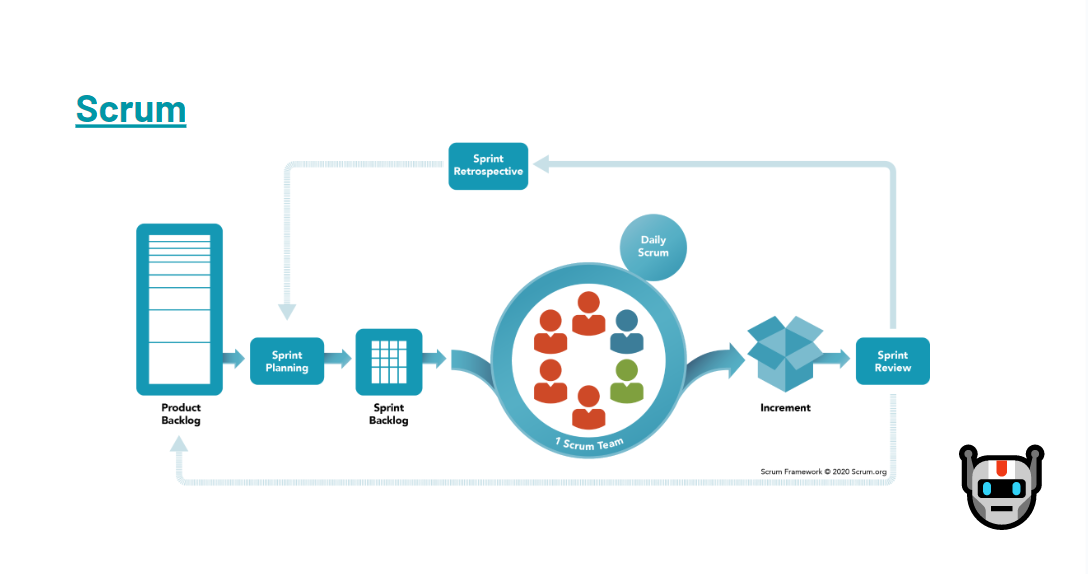
**Containers**

Dentro de cada conteiner voy a tener la aplicación y las librerías necesarias para levantar mi aplicación, o sea todo lo necesario para ejecutarla. Esto me da portabilidad, puedo configurar una imagen de conteiner en una maquina y pasarla a otra sin importar el sistema operativo que tenga esa imagen y va a funcionar si tiene un configurador de contenedores o algo así

**Kubernetes** es el orquestador o manejador de contenedores a mas alto nivel, el escalamiento, el sensado, el balanceo lo hace kubernetes. No es lo mismo que docker, se utilizan en conjunto. Los dos manejan los contenedores, pero el kubernetes maneja muchas instancias de docker al mismo tiempo, es como que sirve para manejar grandes cantidades de contenedores

Cuando tengo una gran cantidad de contenedores que manejar necesito alguien que maneje una gran cantidad de instancias y de manera automatizada, de eso se encarga kubernetes.

Procesos devops

**Agile frameworks**

Ayuda a las organizaciones a generar un valor que pueda verse transmitido a través de las soluciones que se terminan brindando en función de los problemas complejos que se trata de abordar.

Valores de scrum



**Auto organizado** para trabajar, no trabajando solo si no en el sentido de ser capaz de poder velar por las tareas que nos asignan y asegurar el cumplimiento o el acompañamiento de la misma hasta la resolución

**Co-Located:** cercanía entre los compañeros de equipo y poder trabajar en cojunto.

**Long-Lived:** que los equipos se mantengan unidos, si se rota mucho el equipo se tarda mucho en que los nuevos puedan aprender como se trabaja y todo lo relacionado al equipo.

**Cross-Functional:** que los equipos no tengan que depender de alguien de afuera para desarrollar las tareas, por eso se busca que sean multifuncionales y que se pueda solucionar dentro del equipo.

**Dedicated**: cada miembro del equipo debe ser asignado a tiempo completo a cada proyecto.

**Scrum team**

Todo el equipo debe adherirse a ciertas reglas y normas para perseguir un objetivo en común.

**Roles:**

**Product Owner o PO**: Es el que define todo lo que es inherente a aspectos y características del producto, además es el encargado de tener actualizado el product backlog y es responsable de que todo el equipo tenga visibilidad, claridad y que todos conozcan las prioridades. Se estila que sea alguien allegado al cliente, con cercanía al mismo.

**Development team:** va a decidir a nivel arquitectónico de que manera se va a realizar el trabajo establecido por el PO. El PO solo va a listar las tareas pero no tiene la potestad de decir como se va a hacer, de eso se encarga el equipo de desarrollo. Son recursos humanos autónomos, auto organizados y autosuficientes. Cuentan con diferentes know how entre sus miembros.

**Scrum Master:** ayuda al equipo a desempeñarse en su máximo nivel en función de que sigan los lineamientos pactados. Tiene responsabilidad sobre el equipo de desarrollo y sobre el product owner para que estos realicen lo que dijeron que iban a realizar en cada iteración. Además trata de mantener las mejores, correcciones y trata de solucionar problemas y bloqueos que puedan surgir. Es también el que interactúa con otros equipos asociados

**Scrum events**

**Sprint:** se toma como sprint a la carga de trabajo que se va a realizar durante un periodo de tiempo, las tareas que están en el backlog se van a ir desarrollando en distintos sprints y al final de cada sprints esas ideas van a ser convertidas en valores entregables. Las iteraciones tienen una duración fija y dependen de cada equipo. Equipos más longevos suelen tener duraciones de sprint más largas y equipos nuevos tratan de hacer iteraciones más cortas.

Una vez definido el sprint se intenta no realizar cambios, aunque siempre en la práctica aparecen nuevos requerimientos a lo largo del sprint. En estos casos se trata de intercambiar tareas en las que se estima la misma duración que la tarea agregada.

Las tareas no realizadas en el sprint volverán al product backlog. Si esto ocurre es porque existió una mala estimación o una explicación mala u otros factores variables.

**Sprint planning:** Es una instancia que genera el scrum master a la que todo el equipo va a asistir, en un lapso de tiempo de una hora y no más de dos horas, en las que el equipo decide en conjunto las tareas a priorizar, cuales son las tareas a realizar en el próximo sprint, los asignados a cada tarea y como se va a realizar el trabajo seleccionado para ese sprint. Cada uno de los miembros del equipo de desarrollo van a salir de la reunión con las tareas asignadas. Aquí se tendrá en cuenta las horas disponibles para realizar las tareas de todos los miembros para asignar las tareas. Una vez que finaliza esta reunión se intenta no cambiar los requerimientos.

**Daily scrum:** se hacen de manera diaria hasta el término del sprint, en las que se dice que hizo el día de ayer, que va a hacer en el día y si está trancado en algo específico. Teóricamente no deberían decirse más que estas cosas.

**Sprint review:** Ceremonia en donde se junta el equipo a hacer una puesta a punto con el cliente sobre el trabajo finalizado en el sprint. Es la ceremonia que se da al finalizar el sprint. Conocida también como la demo, básicamente es donde se va a presentar a las partes interesadas, stakeholders o quien sea, los avances y los resultados de ese sprint.

**Sprint restrospectiva**: se realiza una evaluación del trabajo realizado en la ultima iteración, se evalua que salió bien, que no salio bien, que aprendimos, que se debe seguir realizando, que deberíamos proponer para seguir mejorando. Una vez lsitas las cosas hechas bien y las que no se hicieron bien, se va a elegir un top 3 o un top 5 de las cosas buenas y malas

**Product backlog**

Es la lista ordenada de todo lo necesario para realizar el producto, a partir de acá obtendremos las posibles tareas que se van a realizar en un sprint. Suelen contar con todos los datos necesarios y de importancia para realizar la tarea, el product owner va a ser el encargado de que estas tareas se encuentren bien escritas, bien detalladas. Si esto se encuentra bien ordenado debería tener un ordenamiento de mayor a menor importancia.

Es normal que esta lista vaya sufriendo cambios a lo largo del desarrollo del producto.

Burndown chart: es una grafica que se utiliza bastante para mostrar el trabajo realizado y el trabajo pendiente.

DevOps Lifecycle y herramientas

El proceso de devops consta de 8 etapas, dependiendo de como se aborde el proceso, puede ocurrir que se junten las etapas en una sola.