

## Explique las diferencias entre un ciclo de vida iterativo de duración fija y un ciclo de vida iterativo de alcance fijo

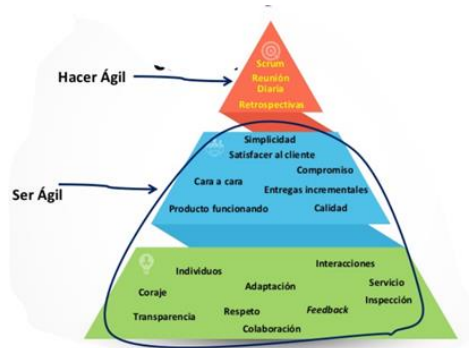
El ciclo de vida lo tenemos que entender como una abstracción. Nos guiará y nos otorga 2 elementos principales para que nosotros podamos aplicar un proceso. Los dos elementos son Fases y el orden de cada una de las fases.

Los ciclos de vida se diferenciarán según si estamos trabajando con un proyecto o producto. Y además contamos con 3 tipos de ciclos de vidas: Iterativo, Secuencial y Recursivo.

En los ciclos de vidas iterativos debemos de definir o marcar como serán las iteraciones. Es decir si estarán marcadas por un alcance fijo o por una duración fija. Hay diferencias entre los ciclos:

- Tiempo Fijo
  - Este ciclo de vida lo utilizará Ágil para poder determinar la duración de las iteraciones. Es importante porque de esta manera con un tiempo fijo podrá asegurarse de conseguir procesos de retroalimentación cortos.
  - El hecho de que el tiempo se fije significa que se “jugará” con el resto de las variables en el sentido de que podremos moverla para lograr la balanza. Los costos son fijos también por lo que debemos de ver el alcance
  - Tenemos que entender que la duración fija hace que no siempre seamos capaces de entregar una funcionalidad al cliente.
    - Entra ágil y define la US y con el modelo INVEST tratamos de asegurar que siempre en una iteración seamos capaces de entregar valor de negocio puesto que esta es nuestro principal objetivo
- Alcance fijo
  - En este ciclo de vida, utilizado por las metodologías tradicionales, tenemos que poner foco en que al final de la iteración se debe entregar una funcionalidad.
  - La idea de usar este enfoque es que el modelo tradicional tiene como punto de partida que podemos obtener los requerimientos desde un principio. Entonces en base a eso podemos determinar el alcance.
  - Los requerimientos no son fijos pueden variar.
  - Aparece el triángulo de hierro. En donde al dejar el alcance fijo nos encontramos con que tanto el tiempo y el costo serán variables. Es decir que una iteración puede durar más que otra pero al final de la misma ambas entregarán funcionalidad.

Explique detalladamente lo que representa la siguiente imagen:



Entre ser ágil y hacer ágil hay una diferencia que va más allá de aplicar lo que dice el manifiesto. Ser ágil lo podemos comprender cuando un equipo trabaja con frameworks ágiles, estos frameworks le facilitan el trabajo aplicando algunos conceptos del agilismo, pero si nosotros no aplicamos de manera Intrínseca la forma de pensar ágil con sus valores y sus principios nunca podremos decir que estamos haciendo ágil.

La filosofía ágil, como bien lo dice su nombre, es una filosofía, una forma de vivir y de trabajar que supera las herramientas propias que se pueden generar con ella. Hay que aprender de los principios y evolucionar como equipo junto a ellos.

El manifiesto se lo plantea como un contrato, pero no es un contrato de que tenemos que hacer, si no más bien de lo que podemos hacer frente a situaciones específicas del desarrollo, independientemente de nuestras practicas individuales.

Es acá cuando nosotros vemos que hay una diferencia entre ser ágil y hacer ágil. Porque nosotros debemos de ser capaces de aplicar los conocimientos y las técnicas ágil. Un ejemplo muy claro es el siguiente: Un equipo que trabaja con herramientas ágiles y desarrolla software con un ciclo de vida iterativo aplicando procesos empíricos está trabajando con una empresa la cual por alguna razón no tiene un PO o algun representante del negocio. Además la empresa no tiene tiempo para las reuniones y sólo quiere que le hagan el software que ellos esperan. Claramente vemos un equipo que está aplicando algunos principios del agile pero no está haciendo ágil para nada. No esta cumpliendo con ciertos valores y no podemos plantear un equipo donde no haya colaboración con el cliente o negocio como ágil.

## **Gestión de Configuración de Software en ambientes ágiles: ¿Es necesaria? ¿Se hace? ¿Cómo?, explique justificando sus respuestas:**

La SCM es una disciplina de soporte del ISW, es sumamente necesaria no solamente en ambientes ágiles sino también en ambientes tradicionales.

Para entender la necesidad de la SCM debemos partir de que existe una facilidad inherente en el software con respecto a los cambios. La idea de que los cambios son fáciles de hacer en software hace referencia no a cualquier tipo de cambio, si no a que con un solo click nosotros podemos eliminar el trabajo de 6 meses o el trabajo entero.

Allí es donde entra en juego la necesidad de tener una gestión de la configuración de software. La misma nos ayudará a realizar una administración de los Items de configuración mediante el versionado y la estructurización de un repositorio.

El objetivo principal del SCM es mantener la integridad del producto de software. Integridad en el sentido de que:

- Satisfaga necesidades del cliente
- Satisfaga criterios de performance
- Fácil y reestructurable
- Cumplir con las expectativas de costo

A eso nos referimos con integridad. Como podemos ver hasta acá no hay nada que no indique porque es importante para tradicional y porque no lo es para ágil.

En ágil nosotros trabajamos con un producto, un producto que queremos mantener íntegro para asegurar la calidad. Debemos acordarnos del principio ágil sobre la excelencia técnica que facilitará ser más ágil.

Bajo todas estas premisas es muy claro que la necesidad del SCM está también en ágil.

Como agregado podemos decir que si bien la necesidad está, no todo del SCM coincide con Ágil. Hablamos de 4 elementos principales del SCM:

- Identificación de IC
- Control de cambios
- Auditorías
- Informes de estados

Las auditorías son justamente lo que no va con el principio de los mejores diseños y arquitectura salen de un equipo autoorganizado. De hecho ágil se niega a las auditorías porque estas son realizadas por un 3ro el cual tiene que ser independiente al equipo de trabajo.