

Actividad 1 - Comparación de Metodologías de Desarrollo

Samuele Tonarini

CIFP Francesc de Borja Moll

14 Noviembre 2021

Tabla de Contenidos

Introducción	ii
Metodologías de Desarrollo	1
Metodologías Tradicionales	1
Metodologías ágiles	1
Cascada	2
Incremental	4
Prototipado	6
Espiral	8
SCRUM	10
Conclusión	12
Bibliografía	12

Introducción

En esta practica se describen y comparan diferentes métodos de desarrollo de software. Se verán algunos entre los mas conocidos, entre cuales están el muy popular método **scrum** y el método **cas-cada**. Se comparan características como tiempo para finalizar el trabajo, efectividad, ventajas y desventajas en confronto a otras metodologías y flujo de cada una.

Las metodologías de desarrollo de software son unas técnicas organizadas que se utilizan para desarrollar software informático, normalmente en grupo. El objetivo es el de agilizar el tiempo en que se desarrolla software y mejorar la calidad del mismo.

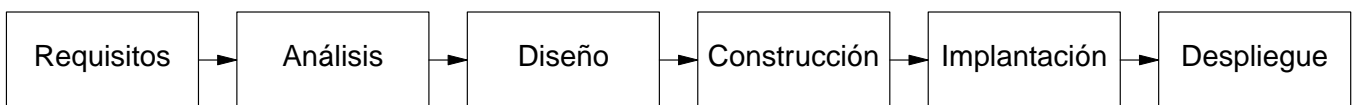
Estas metodologías determinan la rutina que hay que seguir como equipo, teniendo en cuenta las herramientas utilizadas, dificultad del proyecto/tarea, presupuesto, tamaño del equipo, etc... Así que la función de una metodología de desarrollo es disminuir la dificultad, aumentar la comunicación y la participación del grupo, aumentar la velocidad de desarrollo y mejorar la calidad del producto final.

Metodologías de Desarrollo

Metodologías Tradicionales

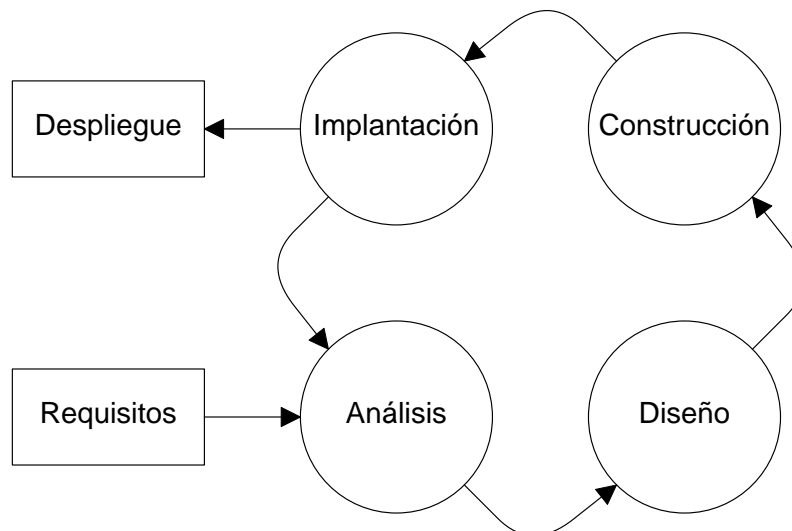
Estas Metodologías están basadas en pasos preestablecidos que tienen que completarse en orden secuencial; Desde requisitos a diseño, de diseño a implementación, etc... Cada fase tiene uno requisitos que tienen que completarse y estar comprobados ante de pasar a la fase siguiente.

Este tipo de metodologías se suelen usar en situaciones donde el objetivo esta bien establecido y los requisitos y funcionamiento son claros. Debido a eso, estas metodologías pueden proponer muchas desventajas y inconvenientes que concretaremos mas adelante en detalle.



Metodologías ágiles

Las metodologías ágiles son las mas utilizadas hoy en día debido a su flexibilidad y a su continua revisión del proyecto por parte del cliente y los desarrolladores. Los métodos ágiles proponen un enfoque incremental de desarrollo, creando el programa paso a paso y revisando los avances cada poco tiempo, cambiando objetivos si necesario.



Aquí abajo se detallan algunas de estas metodologías, tanto tradicionales como ágiles.

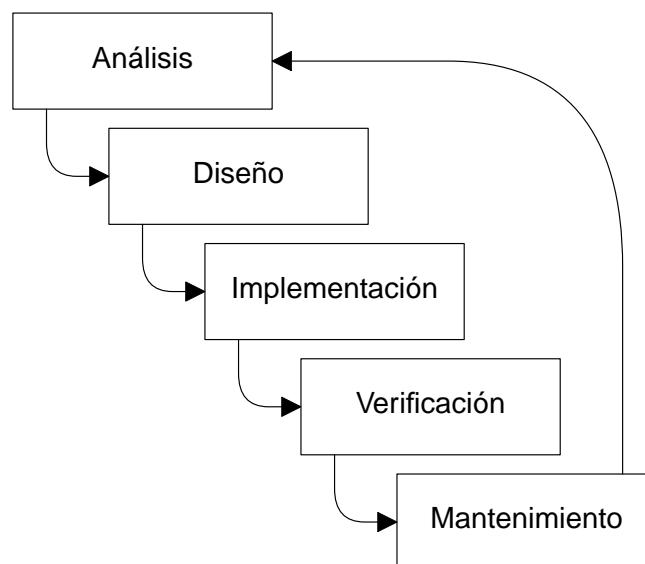
Cascada

El modelo en cascada es una metodología de desarrollo de tipo tradicional. Se basa en seguir unas fases de desarrollo una después de otra. Su nombre viene de que las diferentes fases se ponen una encima de otra y se empieza desde la de arriba, y luego se va ejecutando cada una de arriba a abajo, recordando al flujo de una cascada.

Se suele usar en entornos de industria y construcción, donde normalmente es muy difícil y caro cambiar algo una vez que este implementado.

Se utilizó en el mundo del software como uno de los primeros métodos de desarrollo, pero fue porque en su día aun no se habían generado muchas otras opciones y esta fue heredada de otros sectores. El desarrollo de un programa de software se diferencia mucho al de un edificio, y por eso la poca flexibilidad de cambios que viene con este tipo de metodología y el desarrollo de software no se casan muy bien.

Fases del Modelo en Cascada



Análisis

En la fase de análisis (y/o requisitos), se preparan y describen los objetivos, presupuesto, tiempo de entrega del proyecto, herramientas, riesgos, etc... Todo esto debe estar muy detallado y comprendido por los desarrolladores, ya que una vez que se haya empezado a trabajar con el modelo, no se podrá cambiar en el medio del ciclo de desarrollo.

Diseño

En esta fase es cuando tenemos que diseñar la estructura del software, identificando las varias funciones que va a tener nuestro programa. También deberemos identificar sus elementos y la relación que tiene el uno con el otro.

Implementación

En la fase de implementación deberemos pasar a código los elementos, relaciones y funciones establecidas en la tapa anterior. El lenguaje, tecnologías, paradigmas y técnicas se han establecido en la fase de análisis.

Verificación

Solo se entra en la fase de verificación cuando el software **ya cumple su objetivo**. Osea que esta fase sirve para refactorizar código, optimizar algoritmos y detectar y remover bugs y posibles vulnerabilidades. También es la fase donde el proyecto se evaluá con los objetivo iniciales y se presenta al cliente final.

Mantenimiento

En la fase de mantenimiento es donde solo esperamos para correcciones de problemas nuevos y la introducción de otras funcionalidades del software. En el caso de tener que modificar el software, se entraría en un nuevo ciclo de cascada que empieza de cero con los nuevos objetivos.

Ventajas y Inconvenientes

Ventajas

En un mundo ideal, donde los objetivos y ideas no cambian y donde cliente y desarrolladores se entiende perfectamente, el método de cascada proporciona diferentes ventajas:

- Al ser estructurado, con objetivos claros, es fácil entender el proyecto
- La documentación del proyecto es muy extendida, lo que hace mas fácil para un externo saber que se esta haciendo y como funciona
- La previa planeación de los objetivos aumenta el rendimiento ya que evita que se generen problemas al trabajar
- Ayuda a tener claro donde estas en el proyecto y que tienes que hacer

Inconvenientes

Muchas veces la comunicación entre el cliente y los desarrolladores no es clara, o hay un cambio de idea en el medio del proyecto, o simplemente no se pudo prevenir los problemas de un proyecto debido a su complejidad o tamaño.

Todos esos imprevistos requieren flexibilidad, cosa que no esta permitida en el método de cascada; Eso nos lleva a los inconvenientes de este método:

- Como previamente dicho, no se pueden prevenir todos los problemas, así que esta casi garantizado de encontrar grandes desafíos en medio del desarrollo que sera complicado superar debido a la poca flexibilidad del método
- El cliente no vera el producto hasta que estemos en casi la ultima fase, lo que crea inseguridad del cliente, ademas de una posible insatisfacción con el producto final
- Muchos problemas no se detectaran hasta la fase de verificación, y algunos de ellos podrá ser muy critico y se necesitara pasar otra vez a la fase de diseño o incluso de análisis

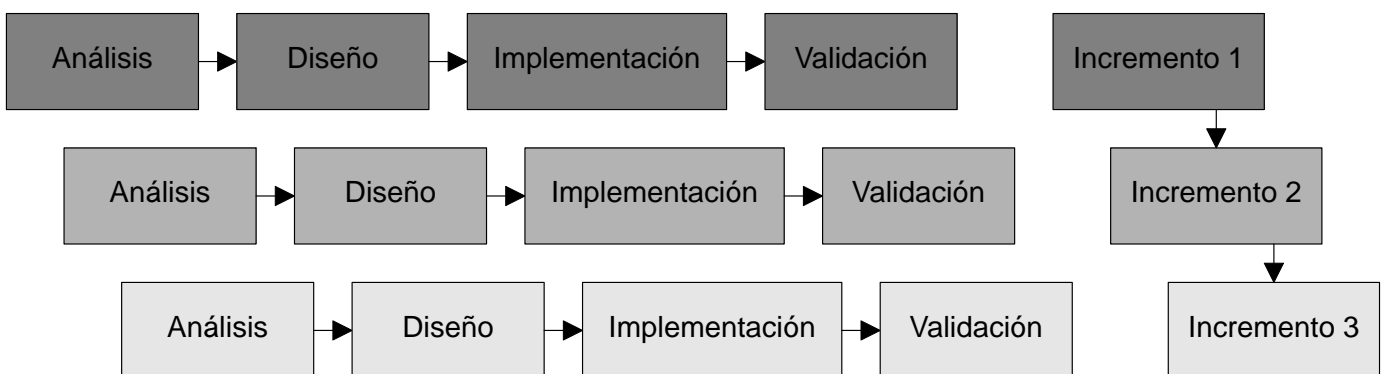
Incremental

El método incremental es una metodología ágil de desarrollo que se creó en el sector de desarrollo de software debido a los límites impuestos por las metodologías tradicionales como el método de cascada.

En este método los objetivos son separados en metas más pequeñas. Una vez separadas se determina la importancia de cada meta y luego se aplicará el ciclo de incremento a cada una de ellas. Este ciclo consiste en analizar los objetivos, diseñar la solución para esos objetivos, implementar la solución en código y por último validar que el código funcione sin problemas.

Una vez acabado el ciclo el incremento vendrá añadido al resto del proyecto, y este bucle se repite hasta que se alcancen los objetivos iniciales. Al finalizar cada incremento habrá una revisión que determinará el siguiente incremento. En la revisión también participará el cliente, pudiendo ver como el programa se crea paso a paso.

Fases del Modelo Incremental



Como se puede observar las fases son similares a las del modelo de cascada, la única pero importante diferencia es que las metas son sub-objetivos del objetivo inicial, y que en cada incremento repetimos el proceso de análisis. Así que para explicar mejor las fases y para no repetir lo que ya hemos descrito en otros apartados, podemos dividir las fases de la siguiente manera:

Análisis Inicial

Aquí es donde separamos el objetivo en metas y donde planeamos de manera general las herramientas. También es donde haremos una visión general de como será todo el proceso y los incrementos, presupuesto y tiempo estimado.

Iteración de Incrementos

Aquí es donde aplicaremos la iteración de los incrementos establecidos. Al acabar cada incremento (después de la incorporación en el proyecto principal) se hará un nuevo análisis, menos profundo que el inicial, pero que determinará el siguiente incremento. Principalmente se verá si hay algún cambio que hacer o si surge algún problema para resolver.

Entrega del Producto

Una vez que todas las metas se hayan alcanzado, el proyecto se entregara al cliente. Este no recibirá normalmente ninguna gran sorpresa porque habrá estado participando a los análisis de los incrementos y proponiendo cambios donde lo veía oportuno.

Ventajas y Inconvenientes**Ventajas**

Este método es mucho mas flexible que los métodos tradicionales, y, debido a eso tiene unas cuantas ventajas:

- El coste inicial para tener un producto funcional suele ser mas bajo
- El cliente influye y opina durante todo el proceso de desarrollo y puede concretar mas su idea
- Al tener validación cada incremento, el código tiene menos problemas y los que tienen serán de menor importancia
- Es mas rápido a generar un producto funcional, ya que se enfoca a desarrollar un producto mínimo viable y luego añadir pequeños incrementos.

Inconvenientes

Aunque tenga muchas ventajas, este modelo sigue teniendo inconvenientes, aunque se pueden considerar de menor importancia:

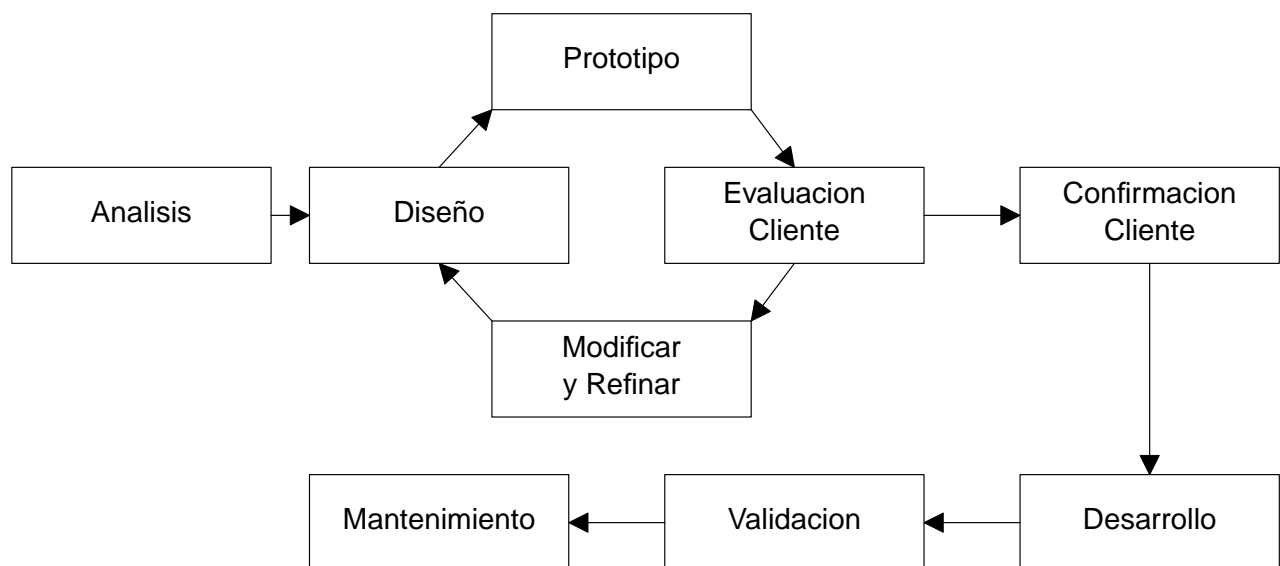
- Si se presenta algún cambio radical durante el desarrollo, puede que haya que replantear las metas y su importancia
- Se pueden ir sumando problemas, que cambiarían la prioridad de los sub-objetivos a la refactorización
- El cliente puede que no este muy decidido, y así retrasar y empeorar el desarrollo, pidiendo cambios constantes

Prototipado

El modelo de prototipado es un modelo ágil que se divide en dos partes clave. La parte de ciclo de prototipos, y la parte de desarrollo y mantenimiento. Este modelo consiste en crear prototipos y presentarlos al cliente/usuario final, que nos dará una valoración de estos y, basándonos en estas valoraciones, refinar el prototipo y presentarlo otra vez, hasta que el cliente este satisfecho y que todos los objetivos estén cumplidos.

Una vez que el cliente este satisfecho, se pasa a la fase de desarrollo donde se escribe el programa, y, una vez acabado se valida y finaliza, entrando así en la fase de mantenimiento.

Fases del Modelo de Prototipado



Como dicho anteriormente, este método se puede separar en dos fases, una (como sugiere el nombre del método) de prototipado, y otra de desarrollo. Estas fases vienen separadas por la confirmación del cliente, que es el que determina que el prototipo este listo o si le falta trabajo. Con eso en mente podemos separar este método en 3 fases:

Análisis y Requisitos

En esta fase determinamos los requisitos para hacer un primer prototipo del programa que cumpla con unos requisitos básicos. También es donde examinamos posibles dificultades y elegimos herramientas.

Fase de Prototipado

En esta fase es donde entramos en una iteración de diferentes pasos. Primero se diseña una solución para los objetivos que hemos declarado en la fase de análisis, después se crea un prototipo funcional basándonos en el diseño. Una vez que el prototipo esta listo, se presenta al cliente/usuario final para que nos de su opinión.

Cuando el cliente pruebe el prototipo pueden pasar dos cosas: la primera es que no lo encuentre listo, en ese caso se apuntan los fallos y se refina y mejora el prototipo, volviendo al paso de diseño. La segunda es que el prototipo se encuentre listo, en este caso se seguiría a la siguiente fase.

Desarrollo

Una vez que nuestro prototipo este listo, debemos crear el programa basándonos en este. En esta fase se trata de crear un sistema que sea igual que el prototipo pero funcional, además de refinar el código y de prepararlo para la implementación final.

Validación

Cuando el código este acabado, tenemos que validarlo para que no haya ningún problema una vez que se use. Esta fase soluciona errores y optimiza el código presente sin añadir funcionalidades nuevas.

Mantenimiento

Una vez validado, el código esta listo para el despliegue. En ese momento solo hará falta el mantenimiento del mismo, hasta que haya que implementar nuevas funcionalidades o resolver errores.

Ventajas y Inconvenientes

Ventajas

Este método se parece mucho al método incremental porque el cliente esta activamente participando en el desarrollo. Además el cliente esta a contacto directo y prueba de primera mano el software. Eso nos trae diferentes beneficios:

- El cliente obtiene lo que quiere
- El cliente puede entender mejor el funcionamiento del software y que se puede hacer con el
- Es mas fácil identificar lo que falta, lo que no esta muy claro (hasta para el cliente), y lo que sera difícil implementar

Inconvenientes

Este método tienes bastante inconvenientes que pueden retrasar el desarrollo del programa:

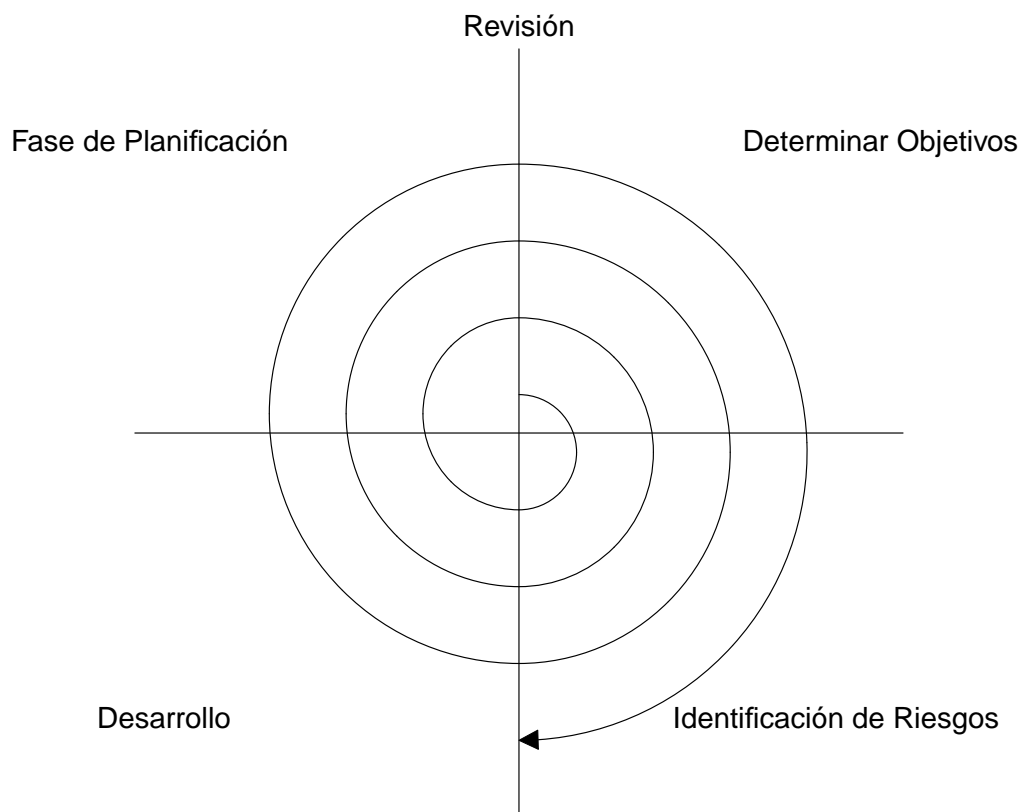
- Si el cliente no colabora, el software no podrá avanzar efectivamente
- Se tarda mas tiempo en acabar el proyecto, ya que antes hay que crear el prototipo, y luego empezar a desarrollar sobre eso
- Puede que haya complicaciones a la hora de implementar la funcionalidad en el prototipo

Espiral

El método de espiral es una mezcla entre el método de cascada y el método incremental. Este combina los dos métodos, siguiendo unas fases preestablecidas, pero en manera dinámica y flexible, recibiendo feedback del cliente. Es un método diseñado para grandes proyectos y para minimizar riesgos.

Como el modelo incremental, este modelo comienza con definir los requisitos mínimos mas importantes y trabaja sobre ellos. Después presenta el resultado al cliente, y basándonos en su opinión, se planteara la siguiente fase. Es aun mas flexible que el modelo incremental ya que nos deja libertad en cada fase y no solo al momento de comprobación con el cliente.

Fases del Modelo de Prototipado



Determinar Objetivos

Al comienzo de cada ciclo, se identifica y analiza los objetivos a partir de los requisitos. Después se elabora el plan a seguir durante este ciclo, junto con el cliente.

Identificación de Riesgos

Durante esta fase se identifican riesgos y problemas y se resuelven. Comparado con los otros métodos aquí buscamos antes los errores y luego implementamos una solución. Una vez eliminados los posibles errores de este ciclo se sigue a la siguiente fase.

Desarrollo

En esta fase se desarrolla una versión funcional del producto basado en el objetivo de este ciclo. Normalmente en el primer ciclo se desarrolla un prototipo, así que se podría decir que este método comprende metodologías inspiradas también por el método de prototipado.

Fase de Planificación

En esta ultima fase se planifica las siguientes fases y se evaluá la finalización del proyecto. Gracias a la flexibilidad de este modelo se pueden saltar/añadir fases según lo que vea el equipo de desarrollo, y esta es la fase donde principalmente se introducen dichos cambios.

Ventajas y Inconvenientes**Ventajas**

Este modelo tiene unas cuantas ventajas debido a que coge las mejores partes de modelos anteriores:

- Permite el control de riesgos ya que trabaja poco a poco, buscando riesgos en todas las pequeñas partes del programa
- El cliente obtiene lo que quiere ya que implementamos un modelo de prototipado en los primeros pasos del desarrollo
- Tenemos flexibilidad a la hora de elegir la siguiente fase
- Hay muchas fases donde recopilamos información, gracias a eso tendremos una documentación muy detallada del proyecto

Inconvenientes

Este método tienes unos inconvenientes relacionados con la estimación de tiempo y presupuesto:

- Debido a la cantidad de información a recopilar y examinar, puede retrasar el desarrollo. Lo que lo lleva a nos ser tan recomendable para proyectos pequeños
- No se puede estimar exactamente la fecha de finalización
- Al examinar tanto los riesgos y lo optimo para el desarrollo, se requiere experiencia para que salga rentable el tiempo invertido. Si no puede pasar que perdamos tiempo en cosas no relevantes

SCRUM

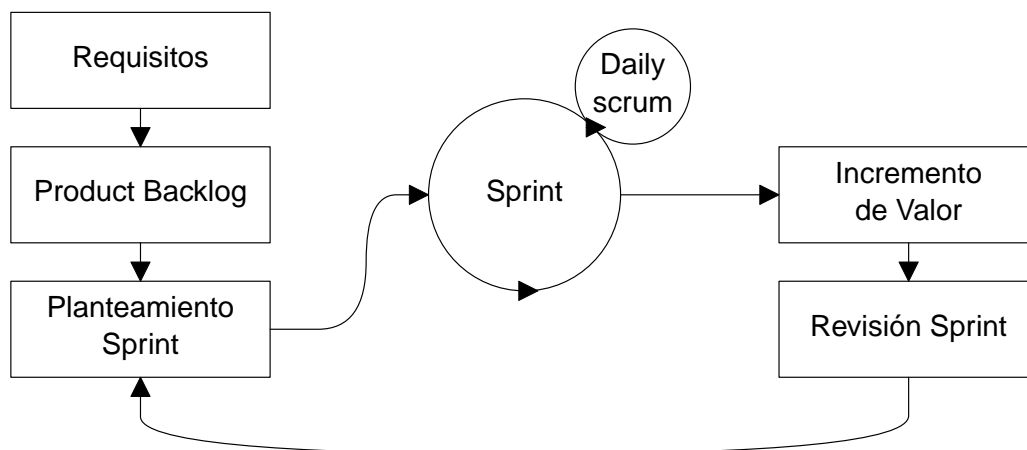
SCRUM es uno de los métodos de desarrollo más famosos actualmente. Los creadores de SCRUM lo describen como un framework ligero para encontrar soluciones a problemas complejos. SCRUM es un método relativamente complejo y profundo a primera vista, pero se pueden detallar solo los pasos más importantes para una descripción más comprensible.

En pocas palabras SCRUM se basa en tener un **SCRUM Master** que tiene que gestionar un entorno donde:

- Un **Product Owner** genera y organiza trabajo a partir de un objetivo en un documento llamado **Product Backlog**.
- El **SCRUM Team** con el SCRUM Master eligen una pequeña parte del Product Backlog y empiezan a trabajar sobre algo que se denomina como incremento de valor (un diseño, una función, una solución de un problema, etc...). El SCRUM Team trabajará sobre este incremento de valor durante este **Sprint**. Un Sprint es un periodo de tiempo preestablecido para tener un incremento listo.
- Al final del Sprint el SCRUM Team junto con el cliente analizan los resultados y preparan el siguiente Sprint.

En SCRUM se usa también un método de reunión ágil conocido como **Daily SCRUM**, que es una reunión de 15 minutos que el SCRUM Team hace cada día para hablar de problemas encontrados y de cómo va el proyecto en general.

Fases del Modelo SCRUM



Requisitos y Product Backlog

El Product Owner crea el Product Backlog basándose en los requisitos del proyecto, en conjunto con el SCRUM Team. El Product Backlog es donde el objetivo final es distribuido en pequeñas metas más sencillas.

Planteamiento Sprint

Desde el Product Backlog se eligen las metas para el Sprint que se va a hacer. También se determina el tiempo que se va a dar para completar estas metas. Se genera un **Sprint Backlog**, que son metas mas pequeñas de las metas, para simplificar aun mas el trabajo.

Sprint

En el Sprint el SCRUM Team trabaja para completar el Sprint Backlog. Cada día se hace el Daily SCRUM, para confrontar el avance del proyecto y los problemas encontrados.

Incremento de Valor y Revisión del Sprint

Una vez acabado el Sprint, se incorpora al producto, y, junto con el cliente se hace una revisión del Sprint, y basándonos en los resultados se plantea el siguiente Sprint; Esto se repite hasta que el Product Backlog no este acabado.

Ventajas y Inconvenientes

Ventajas

Debido a la estructura de SCRUM, este tiene las siguiente ventajas:

- Un uso efectivo de tiempo y recursos
- Una división de proyectos grandes fácil de gestionar gracias a los Sprint
- El cliente puede dar su opinión al final de cada Sprint
- Los Daily SCRUM hacen claros los avances de cada miembro del equipo y evidencian problemas
- Permite la rápida implementación de una función al finalizar cada Sprint

Inconvenientes

Por cada ventaja que tiene el método SCRUM, tiene algunos inconvenientes que no es difícil que ocurran en su implementación:

- No hay una visión clara de una fecha limite
- Si los miembros del equipo no participan se pierde mucha efectividad
- Adoptarlo en grupos grandes es difícil. Seria mejor tener mas grupos pequeños
- Si no se utiliza validación agresiva del código, es probable que la calidad se vea afectada

Conclusión

Hemos visto diferentes metodologías, y en conclusión creo que podemos coincidir que en el ámbito de desarrollo de software, las metodologías ágiles hacen el proceso mucho mas flexible y dan un mejor resultado comparadas con las tradicionales. Dentro de las metodologías ágiles, deberemos mirar nuestro caso en concreto para determinar cual usar, aunque podemos observar que todas siguen la misma idea de desarrollar poco a poco y valuando la opinión del cliente.

Lo que se puede hacer sabiendo las desventajas de cada método es juntar métodos como hace el método de espiral para intentar disminuir desventajas. Esto también dependerá de caso a caso.

Bibliografía

Metodologías

<https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>

<https://www.kpipartners.com/blog/traditional-vs-agile-software-development-methodologies>

Cascada

<https://www.whatissixsigma.net/development-methodologies/>

<https://www.crehana.com/mx/blog/desarrollo-web/modelo-en-cascada/>

Incremental

<https://www.guru99.com/what-is-incremental-model-in-sdlc-advantages-disadvantages.html>

<https://www.educba.com/incremental-model/>

<https://www.obsbusiness.school/blog/caracteristicas-y-fases-del-modelo-incremental>

Prototipado

<https://www.hostingplus.cl/blog/modelo-de-prototipos-que-es-y-cuales-son-sus-etapas/>

<https://www.javatpoint.com/prototype-model>

<https://www.educba.com/prototype-model/>

Espiral

<https://www.tecnologias-informacion.com/metodo-espiral.html>

https://en.wikipedia.org/wiki/Spiral_model

<https://www.guru99.com/what-is-spiral-model-when-to-use-advantages-disadvantages.html>

SCRUM

<https://www.scrum.org/resources/what-is-scrum>

<https://www.simplilearn.com/scrum-project-management-article>