|  |  |  |
| --- | --- | --- |
| **FICHA: 2143152** | | |
| **ASIGNATURA** | **NOMBRE DEL ALUMNO** | **FECHA** |
| Metodologías Agiles | María Alejandra Montoya Marín |  |

METODOLOGÍA CASCADA Y MARCO SCRUM. VENTAJAS Y DESVENTAJAS.

HISTORIA DEL SCRUM

ROLES DEL SCRUM

PRODUCT BACKLOG

\* HISTORIAS DE USUARIO

\* PRODUCTO MÍNIMO VIABLE (PRODUCTOR MINIMISIMO VIABLE- PRAGMA)

**INCEPCIÓN**

**SPRINT PLANNING**

**\* ESTIMACIÓN EN SCRUM**

SPRINT BACKLOG

PRODUCT BACKLOG REFINEMENT

DAILY SCRUM

\* TABLERO KANBAN

REVIEW

RETROSPECTIVE

VALORES DE SCRUM

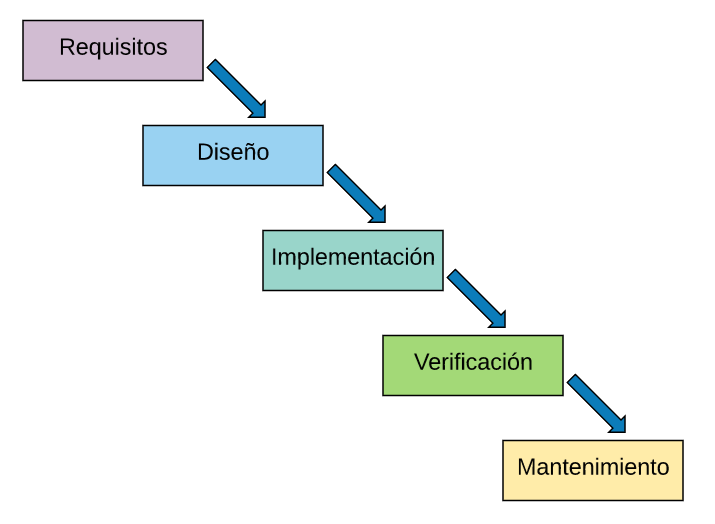
PRINCIPIOS DE SCRUM

MANIFIESTO AGIL

**METODOLOGIA EN CASCADA**

#### ¿QUE ES?

es un proceso de desarrollo secuencial. que se caracteriza por dividir los procesos de desarrollo en sucesivas fases de proyecto, en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra.



**REQUISITOS DEL SOFTWARE**

En esta fase se hace un análisis de las necesidades del cliente para determinar las características del software a desarrollar, y se especifica todo lo que debe hacer el sistema sin entrar en detalles técnicos. Por lo tanto, esta es la etapa en la que se lleva a cabo una descripción de los requisitos del software, y se acuerda entre el cliente y la empresa desarrolladora lo que el producto deberá hacer.  Los desarrolladores deben comprender de forma clara el producto que van a desarrollar. Esto se consigue teniendo una lista detallada de los requisitos, y con una comunicación fluida con el cliente hasta que termine el tiempo de desarrollo.

#### DISEÑO

Descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

#### IMPLEMENTACIÓN

En esta fase se programan los requisitos especificados haciendo uso de las estructuras de datos diseñadas en la fase anterior. La programación es el proceso que lleva de la formulación de un problema de computación, a un programa que se ejecute produciendo los pasos necesarios para resolver dicho problema.

#### VERIFICACIÓN

Como su propio nombre indica, una vez se termina la fase de implementación se verifica que todos los componentes del sistema funcionen correctamente y cumplen con los requisitos.

El objetivo de las pruebas es el de obtener información de la calidad del software, y sirven para: encontrar defectos o bugs, aumentar la calidad del software, refinar el código previamente escrito sin miedo a romperlo o introducir nuevos bugs, etc.

#### INSTALACIÓN Y MANTENIMIENTO

Una vez se han desarrollado todas las funcionalidades del software y se ha comprobado que funcionan correctamente, se inicia la fase de instalación y mantenimiento. Se instala la aplicación en el sistema y se comprueba que funcione correctamente en el entorno en que se va a utilizar.

A partir de ahora hay que asegurarse de que el software funcione y hay que destinar recursos a mantenerlo. El mantenimiento del software consiste en la modificación del producto después de haber sido entregado al cliente, ya sea para corregir errores o para mejorar el rendimiento o las características.

#### VENTAJAS

* El tiempo que se pasa en diseñar el producto en las primeras fases del proceso puede evitar problemas que serían más costosos cuando el proyecto ya estuviese en fase de desarrollo.
* La documentación es muy exhaustiva y si se une al equipo un nuevo desarrollador, podrá comprender el proyecto leyendo la documentación.
* Al ser un proyecto muy estructurado, con fases bien definidas, es fácil entender el proyecto.
* Ideal para proyectos estables, donde los requisitos son claros y no van a cambiar a lo largo del proceso de desarrollo

------------------------------🡪

* En muchas ocasiones, los clientes no saben bien los requisitos que necesitarán antes de ver una primera versión del software en funcionamiento. Entonces, cambiarán muchos requisitos y añadirán otros nuevos, lo que supondrá volver a realizar fases ya superadas y provocará un incremento del coste.
* No se va mostrando al cliente el producto a medida que se va desarrollando, si no que se ve el resultado una vez ha terminado todo el proceso.  Esto cual provoca inseguridad por parte del cliente que quiere ir viendo los avances en el producto
* Los diseñadores pueden no tener en cuenta todas las dificultades que se encontrarán cuando estén diseñando un software, lo que conllevará rediseñar el proyecto para solventar el problema.
* Para proyectos a largo plazo, este modelo puede suponer un problema al cambiar las necesidades del usuario a lo largo del tiempo. Si, por ejemplo, tenemos un proyecto que va a durar 5 años, es muy probable que los requisitos necesiten adaptarse a los gustos y novedades del mercado.
* Los pequeños cambios o errores que surgen en el software completo puede causar mucho problema.
* La mayor desventaja del modelo de cascada es que hasta la etapa final del ciclo de desarrollo se ha completado, un modelo de trabajo del software no está en las manos del cliente. Por lo tanto, es difícil en condiciones de mencionar si lo que se ha diseñado es exactamente lo que había pedido.
* sucede que el cliente no es muy claro de lo que exactamente quiere del software. Cualquier cambio que se menciona en el medio puede causar mucha confusión.

**MARCO SCRUM**

#### ¿QUE ES?

Es un marco de trabajo para el desarrollo de software ágil iterativo incremental. Este marco está orientado a la gestión de procesos o bien para grupos de mantenimiento de software. El Scrum se basa en la definición de roles para la correcta utilización de los grupos de personas dentro del ámbito de trabajo, describiendo sus prácticas y tareas, también describe la utilización de documentos para la administración y recopilación de la documentación.

El desarrollo del proyecto esta descrito y separado, los elementos y actividades que integran el Scrum es: el sprint que divide el tiempo entre una y cuatro semanas para el desarrollo de un entregable, se tiene también el documento llamado Producto Backlog que describe los requerimientos de la aplicación. Este documento se implementa y se consulta durante la reunión llamada Sprint Planning en el cual el ScrumMaster organiza y consulta los avances en el proyecto.

#### ROLES

**PRODUCT OWNER**

Se encarga de obtener exactamente el producto requerido por el cliente, también es el encargado de anotar los avances y demás detalles de parte del usuario final que serán utilizados como metas y objetivos para el desarrollo del proyecto.

**SCRUMMASTER**

Definido como el administrador del proyecto permitiendo que el grupo se auto-organice, también es el encargado de que el equipo de trabajo siga las reglas que esta metodología establece.

**TEAM**

Está conformado del grupo de personas que desarrollan la aplicación las cuales tienen distintos tipos de habilidades, como por ejemplo análisis, diseño, desarrollo, pruebas y documentación entre otros.

**ROLES AUXILIARES**

Se encuentran los Stakeholders haciendo referencia a los clientes, proveedores, vendedores. Los Stakeholders son las personas que hacen posible el proyecto y que intervienen únicamente en las revisiones del sprint. También se pueden mencionar a los Administradores los cuales se encargan de proporcionar el ambiente adecuado para el desarrollo del proyecto.

#### REUNIONES

**DAILY SCRUM**

Esta reunión consiste en obtener información del estado del proyecto, en el cual el ScrumMaster deberá de realizar una serie de preguntas o recordarles a los integrantes del grupo de trabajo que mencionen determinados aspectos del proyecto, así también está definido que se debe de realizar la reunión en un lugar específico, a una hora determinada con castigos para las personas que llegan tarde o no se presentan y entre otros detalles, la reunión deberá de durar 15 minutos todos los días sin importar el tamaño del grupo de trabajo.

El **SCRUM DE SCRUM** se realiza después del Daily Scrum permite a los equipos de trabajo discutir su trabajo en la cual, es elegido un representante de cada grupo y cada integrante de la reunión deberá de contestar preguntas para obtener un mejor detalle del estado del proyecto.

**LA REUNIÓN DE PLANIFICACIÓN DEL SPRINT** se lleva a cabo cada 15 o 30 días en la cual se especificará el trabajo a desarrollar dentro del periodo sprint, se organizará el equipo de trabajo teniendo un límite de 8 horas para obtener la mayor cantidad de detalles del proyecto en desarrollo.

Al finalizar del ciclo sprint es necesario realizar dos reuniones, una de ellas es la reunión de revisión del sprint en el cual se presenta el trabajo completado mostrándoselo al cliente como un demo, así también se demuestra y se explica el trabajo no completado y tiene un tiempo límite de cuatro horas. También se tiene la retrospectiva del sprint la cual tiene como objetivo mostrar las opiniones de cada integrante del equipo para obtener una mejora en el trabajo y tiene como tiempo límite cuatro horas.

#### DOCUMENTOS

Los documentos que se deben de realizar en esta metodología son el **PRODUCT BACKLOG,** el cual especifica los requerimientos del proyecto, convirtiéndolo de suma importancia para obtener exactamente lo que el cliente necesita. Este documento es abierto para el equipo de trabajo para que cada uno pueda realizar tanto modificaciones como anotaciones.

Este método describe una grafica para indicar el avance en cada requisito llamada **BURN DOWN CHART**. En esta los integrantes del equipo de trabajo podrán observar el avance del proyecto a través de líneas, estas tendrán un comienzo e indicaran su finalización en el momento en que llegan a su eje horizontal.

#### VENTAJAS

* **Gestión de las expectativas del usuario.**Los usuarios pueden participar en cada una de las etapas del proceso y proponer soluciones. De hecho, el proceso en su conjunto está pensado para un tipo de evaluación conjunta.
* **Resultados anticipados.**Cada etapa del proceso arroja una serie de resultados. No es necesario, por tanto, que el cliente espere hasta el final para ver el resultado.
* **Flexibilidad y adaptación a los contextos.**Se adapta a cualquier contexto, área o sector de la gestión. Es decir, no es una técnica exclusiva de ninguna disciplina.
* **Gestión sistemática de riesgos.**Del mismo modo, los problemas que aparecen durante los procesos de gestión que pueden afectar a un proyecto son gestionados en el mismo momento de su aparición. Esto es posible debido a que la intervención de los equipos de trabajo puede ser inmediata.

#### DESVENTAJAS:

* **Funciona más que nada con equipos reducidos.**Las empresas grandes, por ejemplo, deben estar sectorizadas o divididas en grupos que tengan objetivos concretos. De lo contrario, en la práctica, el efecto de la técnica se perderá.
* **Requiere una exhaustiva definición de las tareas y sus plazos.**Cuando estos dos aspectos no se definen adecuadamente, Scrum se desvanece. Recuerda que la división del trabajo en cada etapa (y de éstas en tareas específicas) son la esencia de esta metodología.
* **Exige que quienes la utilicen cuenten con una alta cualificación o formación.**No es una modalidad de gestión propia de grupos junior o que apenas estén en proceso de formación. Gran parte del éxito de Scrum radica en la experiencia que aportan los profesionales de los equipos, quienes por lo general acumulan años de experiencia.

**En conclusión, la metodología Scrum es una metodología ágil que hace énfasis en** **el trabajo en equipo** **donde** **la claridad de los objetivos es crucial para avanzar hacia una versión cada vez mejor.**Scrum es parte del desarrollo de software ágil, aunque, a día de hoy, se trata de un método que muchas compañías de diferentes sectores han incluido como parte de su estrategia.

Scrum **es reconocida por ser la metodología ágil más prestigiosa internacionalmente en el sector empresarial**y ha tenido una gran aceptación desde su creación, en el año 1992, cuando el teórico norteamericano Jeff Sutherland sentó las bases para su posterior desarrollo.

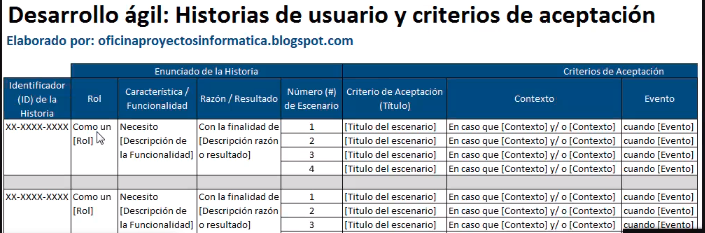
**HISTORIA DEL SCRUM**: El modelo fue identificado y definido por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los 80. En su estudio, compararon las nuevas formas de trabajo en equipo con el avance en formación de melé (SCRUM en inglés) de los jugadores de rugby, a raíz de lo cual quedo acuñado el término (SCRUM) para referirse a ella.

**PRODUCT BACKLOG:** Es el listado de las tareas que se hacen durante el desarrollo de un proyecto estas se deben listarse en el product backlog, para que estén visibles ante todo el equipo y se pueda tener una visión panorámica de todo lo que se espera realizar.

#### COMO SE REDATAN LOS ELEMENTOS DEL PRODCT BACKLOG: La forma predominante en un equipo que usa Scrum, es expresar las características en forma de user stories (historias de usuario), que son breves descripciones de la funcionalidad que se desea, contadas desde la perspectiva del usuario. Un ejemplo es. "Como comprador, yo puedo revisar los productos que están en mi carrito de compras antes de confirmar mi compra, y así estar seguro de lo que he seleccionado"

#### **PRODUCTO MÍNIMO VIABLE:** solo las características básicas suficientes para lanzar el producto (CONJUNTO DE HITORIAS LO PRIMERO QUE VA SALIR)

**\* HISTORIAS DE USUARIO:** Las historias de usuario son descripciones cortas y simples de una característica contada desde la perspectiva de la persona que desea la nueva capacidad, generalmente un usuario o cliente del sistema. Por lo general, siguen una plantilla simple:



* Como <Usuario>
* Quiero <algún objetivo>
* Para que <motivo>
* **Como** usuario de poder, **quiero** especificar archivos o carpetas **para** realizar copias de seguridad en función del tamaño del archivo, la fecha de creación y la fecha de modificación.
* **Como** usuario, **quiero** indicar carpetas que no deben respaldarse **para que** mi unidad de respaldo no esté llena de cosas que no necesito guardar.

**\* PRODUCTO MÍNIMO VIABLE (PRODUCTOR MINIMISIMO VIABLE- PRAGMA)** Un producto viable mínimo tiene solo las características básicas suficientes para lanzar el producto, y no más. Los desarrolladores típicamente lanzan el producto para un subconjunto de los posibles clientes, como los "primeros seguidores", que son más tolerantes, más propensos a dar retroalimentación y capaces de captar la visión de producto a partir de un prototipo temprano. Esta estrategia va enfocada a evitar la construcción de productos que los clientes no quieren y busca maximizar la obtención de información sobre el cliente con respecto a los gastos.

**INCEPCIÓN: SON NECESIDADES QUE SE EVALUAN SOBRE EL PROYECTO A REALIZAR** La primera actividad del Inception es una breve introducción del equipo y del objetivo del proyecto. Scrum Master (SM) presenta cada integrante del equipo y sus roles. Recordar que en el equipo de scrum, todos los integrantes son definidos como developers ya que más allá de que hay skills determinados, todos deben contribuir con el objetivo común planteado para el sprint.

SM define marco de trabajo: roles, horario, importancia de la fase, objetivo, interacciones necesarias dentro del equipo. Mención de restricciones del proyecto, en el caso que las hubiese. Por ejemplo: no se puede gastar más de X presupuesto, o el proyecto no puede ir más allá de X fecha.

**SPRINT PLANNING:** es el primer evento de Scrum en dónde se planifican las tareas a realizar en el Sprint en curso. En esta reunión participan, de manera colaborativa, todo el equipo Scrum: Scrum Master, Product Owner y Equipo de Desarrollo. En esta parte, el Equipo de desarrollo pronostica su capacidad de desarrollo en el Sprint. El Product Owner explica el objetivo de la iteración, y los ítems del Backlog que se deberían hacer para conseguir el objetivo final. Todo el equipo trabaja de manera colaborativa para comprender el trabajo a realizar.

**ESTIMACIÓN EN SCRUM** El proceso de estimación se puede hacer utilizando una técnica llamada planning póker (póker de planificación). El objetivo del planning póker es obtener una medida de tamaño relativo de todas las historias respecto a sí mismas. “A es más grande que B y que C” saber cuál es la fecha estimada de finalización del proyecto, y en qué iteración estará lista determinada funcionalidad. Un beneficio adicional que nos brinda es que, de existir complicaciones severas, que pongan en juego la factibilidad del proyecto, éstas generalmente se ven expuestas bien temprano

**Planning Póker es una técnica para calcular una estimación basada en el consenso, en su mayoría utilizada para estimar el esfuerzo o el tamaño relativo de las tareas de desarrollo de software.**