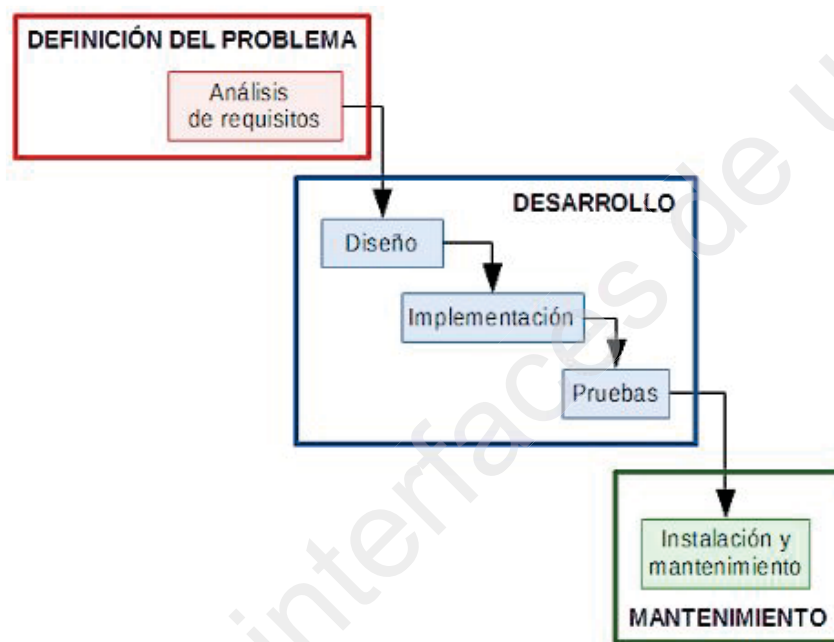


TEMA 1. METODOLOGÍA SCRUM

Antecedentes

Existen una gran variedad de metodologías para administrar proyectos de desarrollo de software. Uno de los más utilizados fue el modelo “en cascada”. Se trata de un modelo secuencial donde una etapa de desarrollo no puede iniciarse hasta que la etapa anterior haya finalizado. En particular, las etapas de desarrollo de este modelo son:



1. **Análisis de requisitos:** donde se analizan los requisitos y necesidades del cliente con el fin de identificar los objetivos del proyecto a desarrollar. Es importante destacar que en esta etapa se determinan TODOS los requisitos que tendrá el sistema y que, aunque es posible que se puedan identificar nuevos requisitos en etapas posteriores, éstos no podrán añadirse al proyecto
2. **Diseño:** donde se diseñan los módulos del sistema, se diseñan las bases de datos, se definen la arquitectura de los datos, etc. Asimismo, el sistema final se descompone y organiza en elementos que puedan desarrollarse por separado con el fin de aprovechar las ventajas del desarrollo en equipo. En este punto es importante distinguir entre *diseño de alto nivel o arquitectónico* y *diseño detallado*. Así, el *diseño de alto nivel o arquitectónico* tiene como objetivo definir la estructura de la solución (una vez que la fase de análisis ha descrito el problema) identificando grandes módulos (conjuntos de funciones que van a estar asociadas) y sus relaciones. Con ello se define la arquitectura de la solución elegida. Por el contrario, el *diseño detallado* es el encargado de definir los algoritmos empleados y la organización del código para iniciar la implementación.

3. **Implementación:** donde se codifican cada uno de los módulos, implementando el código fuente y haciendo uso de prototipos, pruebas y ensayos para corregir errores. Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.
4. **Pruebas e integración:** donde se integran los módulos y se realizan pruebas unitarias, modulares, integrales y de aceptación.
5. **Verificación:** donde el usuario final ejecuta el sistema. A esta etapa se llega después de que los programadores hayan realizado exhaustivas pruebas para comprobar que el sistema no falla.
6. **Mantenimiento:** es una de las etapas más críticas del desarrollo, ya que se destina un 75% de los recursos. Su objetivo es el mantenimiento del software desarrollado, ya que puede ser que no cumpla con todas nuestras expectativas.

No obstante, esta metodología era demasiado rígida. De hecho, raramente se introducían cambios en la planificación inicial, puesto que eso implicaba el rediseño e implementación de todas las partes afectadas, aumentando los costes y duración del desarrollo.

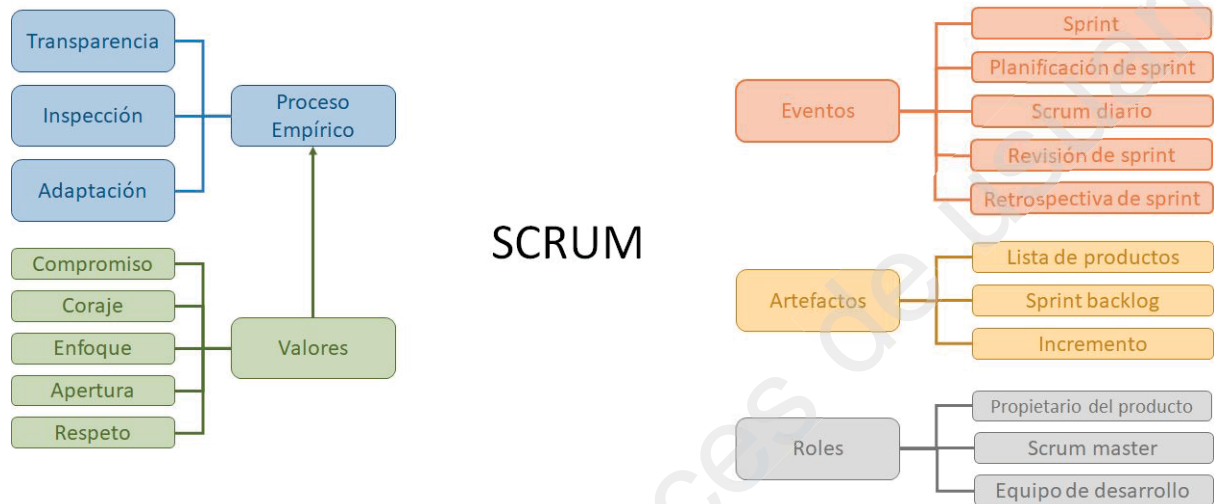
Como solución, en febrero de 2001, 17 expertos en la industria del software crearon la alianza Ágil y publicaron el manifiesto ágil en el que declararon que estaban descubriendo formas mejores de desarrollar software tanto por su propia experiencia como ayudando a terceros. A través de este trabajo aprendieron a valorar cuatro aspectos:

1. Individuos e interacciones versus procesos y herramientas
2. Software funcionando versus documentación
3. Colaboración con cliente versus negociación contractual
4. Respuesta ante el cambio versus a seguir el “plan”

Aunque se siguen valorando los aspectos de la derecha, se valoran más los aspectos de la izquierda. A partir de este trabajo surgió el enfoque ágil, donde las soluciones iterativas cambian y evolucionan, y el equipo que lo desarrolla es multidisciplinar, auto-organizado, colaborativo y con capacidad de tomar decisiones a corto plazo. En este sentido, han surgido distintas metodologías, técnicas o frameworks ágiles como pueden ser RUP, LEAN, 3x3, Kanban o SCRUM.

SCRUM

Scrum puede definirse como un marco de trabajo por el cual las personas pueden abordar problemas complejos adaptativos y entregar productos del máximo valor posible de manera productiva y creativa. Así pues, **SCRUM no es una técnica, sino un marco de trabajo** que puede resumirse de la siguiente forma:



SCRUM está basado en la experiencia y la observación. Por lo tanto, el conocimiento procede de la experiencia y las decisiones se basarán en aquello que se conoce. Esta teoría de control de procesos empíricos se sustenta en tres pilares:

1. **Transparencia**, garantizando la visibilidad a todos los responsables del resultado y la definición de un standard común
2. **Inspección** del progreso y artefactos de SCRUM sin interferir en el trabajo
3. **Adaptación** en cada iteración, identificando cualquier anomalía que pudiera interferir en el resultado esperado y adaptarlo para que se logren los objetivos

Cabe destacar que el proceso base de SCRUM es el ciclo inspección-adaptación:

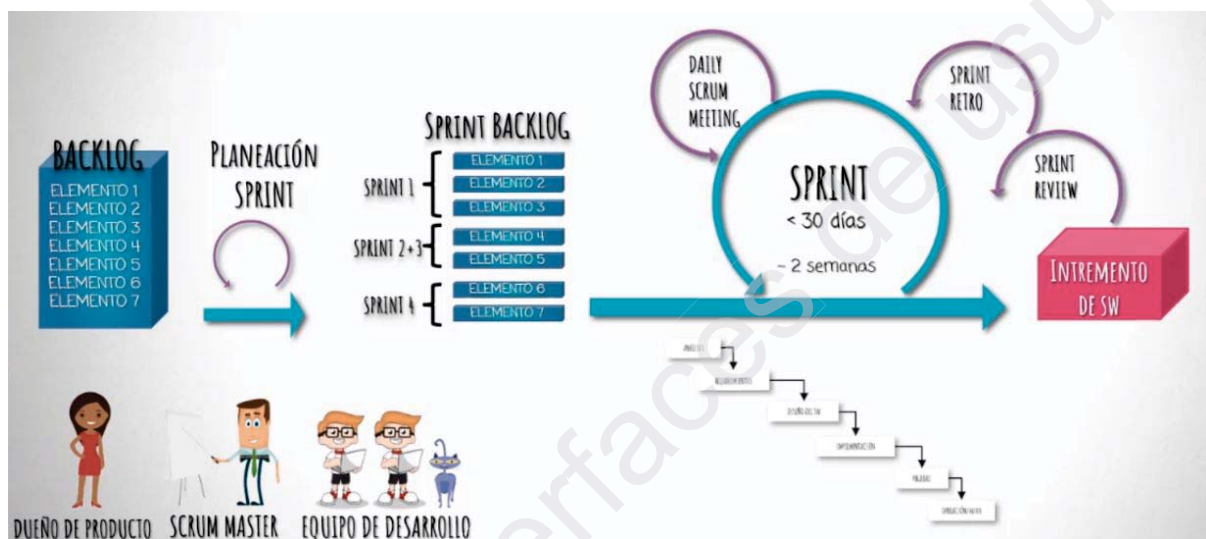
1. Aprender del pasado
2. Mejorar
3. Tomar nuevas decisiones
4. Adaptarse

Por otra parte, los tres componentes principales de SCRUM son:

1. Roles: quién hace qué
2. Eventos: bloques de tiempo para la comunicación
3. Artefactos: que nos ayudan a tener transparencia en el proceso

¿Cómo funciona SCRUM?

SCRUM es un proceso iterativo que permite desarrollar, entregar y mantener productos funcionales complejos, que puede representarse como sigue:



Un aspecto fundamental en este proceso son los roles. Como puede observarse, existen tres roles distintos:

1. El **propietario del producto**, que es el responsable de maximizar el valor del producto resultante del trabajo del equipo de desarrollo. Así pues, es el que da una visión del producto, gestiona el **backlog** y conecta con el producto final
2. El **scrum master**, que es el responsable de promover y apoyar SCRUM. Para ello, ayudan a todos a entender la teoría, práctica, reglas y valores de SCRUM
3. El **equipo de desarrollo**, está formado por los profesionales que realizan el trabajo de entregar un incremento de producto "terminado". Por lo tanto, es el encargado de transformar la visión del producto del propietario del producto en realidad. Se caracteriza por ser auto-organizado y multifuncional, abarcando distintas competencias y tomando decisiones colectivas. Así pues, el equipo de trabajo se compromete a realizar pequeñas cantidades de metas en períodos cortos de tiempo

El proceso se inicia con el **backlog**. El **backlog** es la descripción del producto final que se quiere obtener de manera que, mediante una lista, se indican las funcionalidades, requisitos, mejoras, defectos, solicitudes y riesgos que tendrá el producto final que se desea desarrollar. Este elemento nunca está completo y es dinámico, evolucionando con el

producto y su entorno (requisitos de negocio, condiciones de mercado, tecnología). Cada uno de sus ítems tiene como atributos su descripción, su prioridad en el desarrollo, su estimación y su valor.

A continuación, tiene lugar la **planificación del sprint**. Es importante tener en cuenta que, a diferencia del modelo “en cascada”, en SCRUM el ciclo completo se realiza en un **sprint**. De hecho, el **sprint** es el corazón de todo el desarrollo. Podría definirse como un contenedor que ayuda a generar y coordinar cada uno de los eventos (que no son más que oportunidades formales para la inspección y adaptación del desarrollo). Cada **sprint** puede considerarse como un proyecto a corto plazo que tiene un objetivo, un diseño, un plan flexible, un trabajo en equipo y un incremento (entendido como una parte del código que está listo para ser liberado al finalizar el sprint). Su duración no superará los 30 días (1 mes) y no suele ser menor de 15 días (2 semanas). Es importante tener en cuenta que, una vez que comienza un **sprint**, su duración es fija y no puede modificarse. No obstante, un **sprint** puede cancelarse antes de que termine el bloque de tiempo para el que estaba planificado siempre y cuando su objetivo se haya quedado obsoleto o desfasado. Dicha cancelación únicamente puede realizarla el propietario del producto y resultará en una revisión de aquel desarrollo que ya está finalizado, una revisión y estimación de las partes no finalizadas, y una nueva planificación del **sprint**.

La **planificación del sprint** es donde se decide el trabajo que se va a realizar durante dicho sprint. La duración de la planificación va a depender de la duración que tendrá el **sprint** (4 horas como máximo cuando el sprint dure 15 días u 8 horas como máximo cuando el sprint dure 30 días). El objetivo de esta tarea es contestar qué y cómo se valoran los objetivos que se definen en el **sprint**. Para ello, el propietario del producto genera una propuesta de **sprint backlog**¹ sobre la que trabaja y discute la junta (el scrum master y el equipo de desarrollo) para elaborar el **sprint backlog** final, el cual define el objetivo del **sprint**. Después se definen las distintas tareas y se inicia el desarrollo, que el scrum master se asegurará de que se lleven a cabo, que el equipo de desarrollo entienda su propósito y que se cumpla el bloque de tiempo.

Además del **sprint** y la **planificación del sprint**, en SCRUM existen otros tres eventos: el **scrum diario**, la **revisión del sprint** y la **retrospectiva del sprint**. Todos los eventos son bloques de tiempo de tal modo que todos tienen una duración máxima con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas. Del mismo modo, los eventos aseguran que haya transparencia, inspección y adaptación.

El **scrum diario** se realiza cada día y a la misma hora y se utiliza para planificar el trabajo de ese día. Para ello, el equipo de desarrollo responde a dos preguntas: qué se hizo ayer y qué se va a hacer hoy. También se indican los impedimentos² que hayan podido surgir. En este

¹ Lista de los elementos pendientes de cada **sprint**. Básicamente, se subdivide en bloques que se van a realizar en cada uno de los sprints. Esos bloques tienen cada uno de los elementos necesarios para alcanzar el producto final

² Un impedimento es básicamente cualquier cosa que se interponga en el objetivo como la falta de claridad en los requisitos, las malas estimaciones o la estimación de tiempos y/o presupuestos irreales

proceso el *scrum manager* no interviene, aunque sí que se asegura que se realice y es el encargado de moderarla.

Al finalizar un sprint se realiza la *revisión del sprint*. Se trata de una reunión informal en la que se ve el incremento del producto resultante del *sprint* que finaliza y se presentan los problemas que surgieron durante el desarrollo así como la forma en la que se resolvieron. Tiene una duración máxima de 4 horas cuando el *sprint* ha sido de un mes. Es la única reunión en la que interviene personal externo como CEO, stakeholders, etc. cuyas dudas y preguntas son respondidas por el propietario del producto, excepto las relativas al desarrollo que son respondidas por el equipo de trabajo. Además, se determina cuál es el siguiente paso en el desarrollo, se revisa la línea temporal, el presupuesto y las capacidades potenciales y prevista del producto.

Tras la *revisión del sprint*, se realiza la *retrospectiva del sprint*. En esta reunión se comenta qué funcionó durante el sprint y qué se debe mejorar. En este caso, el scrum manager sí que interviene en la reunión como un miembro más del equipo. El resultado de este evento, que no durará más de 3 horas para un *sprint* de 30 días, es un plan para implementar las mejoras a la forma en la que se desarrolla el producto.