

TEMA 3: METODOLOGÍAS ÁGILES

El modelo Waterfall (va en cascada, un paso después del otro)

1. Definición de requerimientos: ¿qué quiere el cliente o qué es lo que yo creo que quiere el cliente?
2. Análisis y diseño del software: ¿qué vamos a hacer y cómo lo vamos a hacer?
3. Implementación: el ingeniero (nosotros) hacemos lo que creemos que el analista dice sobre lo que quería el cliente (se va perdiendo un poco lo que quería con lo que hacemos)
4. Prueba: probamos que hace lo que pensaba que tenía que hacer
5. Implantación: instalamos el programa en el ordenador del cliente
6. Mantenimiento: solucionamos los pequeños problemillas que puedan surgir

Sin embargo, pueden surgir **problemas**:

- El cliente debe ser capaz de especificar sin ambigüedades los requisitos del sistema que quiere. La incertidumbre afecta al resultado final
- Detección tardía de errores: Los contactos del cliente con el desarrollo son al principio y al final (en la fase de implantación y pruebas). Es por esto que la mayoría de los errores se detectan en estas últimas etapas

La documentación no se actualiza

Problemas:

- Comunicación cliente-analista
- Comunicación analista – programador
- Modelo jerárquico poco integrador del equipo
- Detección tardía de errores

- Desarrollo de elementos que ya no son útiles tal y como fueron concebidos originalmente (documentación no se actualiza :?)

Llegan los Superhéroes del agile y crean una nueva metodología:

Kent Beck: “No soy un gran programador, solo soy un buen programador con muy buenos hábitos”

Robert C.Martin: “La única forma de ir rápido es ir bien”

“El papel más importante de un manager es ayudar a la gente a hacer su trabajo. Ponles un objetivo y déjalos trabajar. Elimina cualquier impedimento en su camino y dales lo que necesitan para trabajar. Haz lo que sea para que sean más productivos y efectivos. Entonces la organización podrá sacar el máximo rendimiento de su trabajo”.

Martin Fowler: “Cualquier tonto puede escribir código que un ordenador comprenda. Pero los buenos programadores escriben el código que los humanos puedan entender”.

“Cuando sientes la necesidad de escribir un comentario, antes intenta modificar el código para que cualquier comentario sea superfluo (no sirva para nada)”

Jim Hightsmith: “La agilidad no es una bala de plata. La agilidad es la habilidad de reaccionar pronto ante un problema”

Alistair Cockburn: “Antes de conseguir las cosas bien, las conseguimos mal”

Esta nueva metodología se resume en un manifiesto ágil de 4 postulados y 12 principios:

Postulados

- **Individuos e interacciones sobre procesos y herramientas:**
Hablar entre personas antes que tener mucha tecnología
- **Software funcionando sobre documentación extensiva**

Hay que enseñar el software funcionando y no solo decir lo que hace

- **Colaboración con el cliente sobre negociación contractual**

El cliente es quien te dirá si vas por buen camino, por lo que hay que colaborar mucho con él

- **Respuesta ante el cambio sobre seguir el plan:**

Si vas por mal camino haz un cambio, aunque esté planificado de otra forma

Principios

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua del software (hay que entregarle cosas para que las vea) (2)
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente. (4)
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible (2)
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto (1)
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo (siempre tienen que estar motivados y para ello le damos un buen entorno y buenas herramientas) (1)
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara. (1)
7. El software funcionando es la medida principal de progreso (debe mostrarse poco a poco progresando, enseñando cosas nuevas) (2)
8. Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida (1)
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad (los trabajadores deben saber hacer bien su trabajo) (1)

10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados (sin jefes) (1)
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia. (1)

Scrum

En 1995, Ken Schwaber presentó Scrum Development Process, un marco de reglas para el desarrollo de software. Uno de los avals que él ponía encima de la mesa es que él lo había utilizado con éxito.

El objetivo del Scrum es realizar una serie de iteraciones, cada una de las cuales incrementa la funcionalidad del producto, mientras que en método Waterfall se obtiene el producto al final.

Pilares/características del Scrum

- Reuniones diarias
- El cliente define las prioridades
- Definición a bajo nivel progresiva
- El equipo Scrum
- Transparencia

➤ Reuniones diarias

Con ellas, se debe comprobar la evolución del proyecto y detectar de forma temprana cualquier impedimento que impida la consecución de los logros previstos (problema)

Estas reuniones suponen un avance interesante en los conceptos de control de riesgos de desvío de los plazos previstos, por lo menos dentro del subproyecto que se está realizando en cada momento.

Bases de las reuniones diarias:

- Todos hablan el mismo lenguaje
- Todos entienden lo mismo por “terminado”
- Transparencia en la inspección (la gente debe ser clara cuando le preguntan que qué le queda)
- Adaptación ante retrasos o falta de calidad (si hay retrasos no hay problema, nos adaptamos)
- Es preferible ajustar desviaciones cuanto antes

➤ **El cliente define las prioridades**

- Maximiza el valor de negocio: se prioriza la parte del proyecto que aporta mayor valor de negocio.
- Maximiza el retorno de la inversión: se desarrolla primero aquellos elementos que con menor coste producen mayores beneficios.

➤ **Definición a bajo nivel progresiva**

Solo se detallan los elementos a desarrollar de manera inminente (próximamente). Se evita que posibles cambios en los objetivos hagan inservibles las horas dedicadas a diseñar en detalle partes del proyecto que finalmente no va a ser desarrollado como inicialmente estaba previsto

Dificulta cálculos de costes globales

➤ **El equipo Scrum**

Los equipos Scrum se focalizan en un objetivo muy concreto. No dispersan su trabajo en todos los requisitos del proyecto sino que se centran en aquellas partes priorizadas en una reunión anterior

En proyectos software, el coste personal suele ser la parte más importante del proyecto. El equipo debe estar y ser:

- Preparado
- Comprometido
- Capaz de obtener conocimientos
- Capaz de autogestionarse

Observaciones:

- Debemos elegir al personal adecuado
- Dispones de personas, imprescindible para hacer equipo
- Cada grupo es diferente y especial
- En gestión de equipos, 2+2 puede ser igual a 3 o a 5

Nadie es mejor que todos juntos

Cada componente del equipo debe sentirse miembro del mismo

Deben apoyarse y ayudar a sus compañeros, sabiendo que así los retos que se pueden superar son mucho mayores

El equipo Scrum tiene mucha libertad para desarrollar su trabajo. Esa libertad tiene como contraprestación el compromiso del equipo para lograr ese éxito que se espera de él.

➤ Transparencia

La transparencia no es un objetivo difuso a conseguir, es parte intrínseca del Scrum. No se oculta nada, la sinceridad es un valor a destacar. Debe ser sencillo conocer cuál es la situación del proyecto en cada momento

ROLES DEL SCRUM ¿QUIÉN?

- **Product Owner:**

El product owner debe conocer y validar los requerimientos del producto. Debe definir prioridades (habla con el cliente y decide lo que se va a hacer). Es solo una persona, por lo que asume la responsabilidad de determinar qué es ese producto (con 2 personas puede haber ambigüedad, así no).

- **Scrum Master**

El scrum master es el responsable de que el equipo pueda trabajar según las reglas Scrum y también es el responsable de que lo haga

- ❖ Favorece la auto-organización del equipo impidiendo interferencias y/o distracciones externas
- ❖ Resuelve problemas e impedimentos que puedan surgir (ya sean informáticos o de intendencia) (si faltan herramientas)
- ❖ Debe mantener visibles los artefactos Scrum
- ❖ Debe conseguir que la lista de producto sea clara y no tenga ningún tipo de ambigüedad

- **Scrum Team:**

Se basa en el trabajo colaborativo

Es fundamental el trabajo en equipo, abandonando el concepto de francotirador solitario.

- Auto-organizados y dirigidos desde el mismo equipo
- Multifuncionales, miembros con diferentes habilidades, desde desarrollo puro, a pruebas incluso análisis
- Autónomo, no depende de nadie externo al equipo a la hora de tomar decisiones, es el mismo equipo el que toma las decisiones que afectan al proyecto que están desarrollando.

- **Stakeholders**

Hacen posible el proyecto en el sentido de la financiación o solicitud, así como aquellos para quienes el proyecto producirá el beneficio acordado. Solo participan directamente durante las revisiones del producto que va finalizando el equipo de desarrollo (gente que se encarga de buscar lo que necesitan los desarrolladores: dinero o lo que sea. Y, por tanto, no están en todas las reuniones).

Reuniones y eventos Scrum

El Scrum define claramente el tipo de reuniones que el equipo va a tener, las cuales forman parte implícita en la filosofía de Scrum

El evento principal del Scrum es el Sprint.

Fija el incremento de funcionalidad del producto

Las reuniones giran alrededor de la preparación, evaluación o el seguimiento de éste

- El Scrum es un espacio de tiempo razonable para poder ser capaz de finalizar una serie de objetivos/funcionalidades.
- Es lo suficientemente corto para prever posibles desfases
- En algunas empresas los períodos mensuales son utilizados también para otro tipo de controles de facturación.

Un mes puede ser un período de tiempo para la realización de un Sprint (aunque puede cambiar según el proyecto)

Reuniones durante cada Sprint:

- **Una reunión de planificación del Sprint (Sprint Planning Meeting)**
 - Se realiza al principio del Sprint
 - Se decide qué se va a finalizar en este Sprint
 - Se decide cómo se va a organizar para finalizarlo

El objetivo es fijar cuáles son las tareas más prioritarias de todas las pendientes que se pueden realizar durante la duración del Sprint

El tiempo estimado para el Sprint es fijo (solo debería cambiar por motivos excepcionales). Es imprescindible que quede claro qué día se fija la finalización. Se debe cuantificar la lista de miembros del equipo y su porcentaje de dedicación al proyecto que van a trabajar en el Sprint.

En esta reunión se realizan varios pasos:

- **Propuesta de requisitos:** el product owner determina las funcionalidades que deberían completarse en ese Sprint. El equipo debe resolver preferentemente en este momento cualquier duda que surja sobre los requisitos.
- **Lista de tareas:** El equipo determina la lista de tareas a realizar (cada tarea no debería tener una duración mayor a una jornada) para completar los requisitos.
- **Prerrequisitos:** Se deben determinar orden de realización en el caso de que existan condicionantes de realización entre las diversas tareas

Fijar tiempos para cada tarea. El equipo debe determinar el coste de cada una de las tareas

Los mismos miembros del equipo se asignan inicialmente las tareas que van a realizar. Obviamente se deben tener en cuenta la capacidad en horas de los miembros del equipo y el coste en horas estimado de cada una de las tareas.

- **Reuniones diarias durante el Sprint, llamadas Scrums Diarios (Daily Scrums)**

El objetivo específico de cada reunión de Scrum Diario es el de comprobar el trabajo avanzado desde la última reunión y definir el trabajo que se va a finalizar hasta la siguiente sesión.

Ley básica: NO se deben sobrepasar en ningún caso los 15 minutos de la reunión

Tras cada reunión debe quedar totalmente claro

- Funcionalidades completas o parciales realizadas en la jornada previa (lo que hemos hecho)
- Funcionalidades completas o parciales que van a finalizar en la jornada de hoy (lo que vamos a hacer)
- Impedimentos o problemas que han surgido o se prevén que puedan afectar el objetivo del Sprint (problemas)

Estas reuniones marcan un inicio común de jornada laboral, mejoran el conocimiento del estado de todo el proyecto entre los miembros del equipo, fomenta cierta competitividad y permiten detectar impedimentos o problemas de manera casi inmediata.

Permiten conocer el estado del proyecto al día, elimina la necesidad de realizar otras reuniones no planificadas o informales ya que se procura trasladar el tratamiento de cualquier asunto a la reunión diaria y ayuda a crear vínculos entre los miembros del equipo.

Tenemos que responder a estas tres preguntas:

- ¿Qué hice desde la última reunión diaria hasta ahora?
- ¿Qué me comprometo a hacer hasta la próxima reunión diaria?
- ¿Qué problemas he tenido o considero que puedo tener en el normal desarrollo de mis tareas?

- **Una reunión de evaluación del Sprint (Sprint Review) - Revisión del Sprint**

Se presentan las nuevas funcionalidades terminadas (el software funcionando) y los Stakeholders y el product owner deben validar el trabajo realizado por el equipo de desarrollo. Se suele dedicar media jornada.

- Si el formato de la reunión es una presentación se debe hacer una descripción por parte del Scrum Master de las funcionalidades realizadas en un orden coherente y, posteriormente si se considera oportuno, una presentación por parte de los desarrolladores de las funcionalidades finalizadas en detalle.

- Si el formato es una validación esta debe llevarse a cabo por parte del product owner permitiendo marcar como terminadas las funcionalidades permitiendo marcarlas como terminadas o no y, posteriormente, en caso de ser necesario, un planteamiento por parte del product owner de mejoras o modificaciones a las funcionalidades entregadas.

Esta revisión posee diversas ventajas:

- Permite al desarrollador mostrar su trabajo y obtener validación y/o realimentación por parte el Product Owner
 - Permite detectar desviaciones entre lo que el Product Owner deseaba obtener y lo que le muestran
 - Permite preparar la replanificación del proyecto con la máxima información
- **Una reunión de retrospectiva del Sprint (Sprint Retrospective)**

Autoevaluar la forma de trabajar. Se suele dedicar media jornada.

(Aquí se suele decir por ejemplo si las tareas que pensábamos que iban a durar 10h han durado 15)

Artefactos Scrum

Permiten dar a conocer a los miembros del equipo cuál es la situación del proyecto. Dan máxima transparencia al proceso de desarrollo.

- Lista de producto (Product Backlog)

La lista de producto es una lista ordenada de todos los requisitos del proyecto. Es dinámica y no está completa con todos los requisitos del proyecto al principio.

Cada elemento de la lista se corresponde con una tarea que tiene una descripción donde se especifican cuáles son sus objetivos. Debe especificarse en lenguaje natural en vez de las especificaciones formales utilizadas en las metodologías tradicionales. Y esta descripción viene dada desde el punto de vista del analista de las especificaciones formales.

Ejemplo: almacén

Cada tarea debe estar estimada. La estimación es el número de horas o puntos estimados para completar la tarea. Los criterios de validación indican los aspectos que deben tenerse en cuenta para que la tarea se considere como terminada

➤ Pendientes del Sprint (Sprint Backlog)

En la reunión de planificación de Sprint, el Scrum Team, bajo el liderazgo del Product Owner debe seleccionar aquellos PBI del Product Backlog que van a formar parte del siguiente Sprint.

Esta elección depende de:

- La capacidad del equipo de desarrollo
- El tiempo asignado al siguiente Sprint
- Las prioridades fijadas por el Product Owner
- Las dependencias de las tareas a realiza

Para cada uno de los Sprint se eligen un conjunto de PBIs o tareas de la Lista de Producto. La realización de dichas tareas son el objetivo del Sprint. Todos y cada uno de los elementos que forman el Sprint deben estar detallados para que sean comprensibles en las reuniones de Scrum Diario.

Una vez se ha iniciado el Sprint no se deben modificar ninguno de los elementos que forman parte del mismo, salvo que se hayan convertido en obsoletos debido a que los objetivos hayan cambiado.

Cuando la tarea es finalizada, se quita de la lista de pendientes del Sprint y pasa a la Lista del Incremento o de Terminados

➤ Lista de incremento

El incremento es la suma de los elementos terminados. Un producto terminado debe añadir funcionalidad al sistema. El product Owner es el responsable de decidir si se incorpora a los sistemas en producción.

Una de las expresiones más escuchadas cuando preguntas por el estado en el que se encuentra el desarrollo de un programa o funcionalidad es la de “está casi terminado”. El concepto de casi terminado puede suponer en la mayoría de los casos un número de horas suficientemente considerable

En gestión de proyectos hay un dicho que reza “Los proyectos progresan rápidamente hasta que están completos en un 90% de sus objetivos, para luego permanecer eternamente en un 90%”. En proyectos software cuando se pregunta al programador sobre el avance de su desarrollo suele utilizar dos respuestas, tengo hecho un 90% o me falta sólo por hacer un 10%, todo ello cuando en ocasiones el tiempo utilizado no alcanza ni siquiera la mitad del previsto

En función de que un proyecto se acerca a su fin, las personas que toman las decisiones tienen más probabilidades de ocultar la información negativa que pudiera ponerlo en peligro

Si el proyecto supone un 90% de avance, el 81% no mencionó nada negativo, en comparación con sólo el 37.5% si el proyecto estaba en 10% de avance.

Semanas de 1 a la 5 - 50% del proyecto terminado

Semanas de la 6 a la 8 - 75% del proyecto terminado

Semanas de la 9 a la 10 - 85% del proyecto terminado

¿Cuánto queda para acabar el proyecto, una semana y media o dos semanas?

Semanas de 11 a la 12 - 90% del proyecto terminado

Semanas de la 13 a la 14 - 94% del proyecto terminado

¿¿Cuanto falta???

