

Cristóbal Sotelo Gallardo

Gerardo Hernández Hernández

14-011-0500

Introducción a la ingeniería de Software

# UACM

Universidad Autónoma  
de la Ciudad de México

---

*Nada humano me es ajeno*

## **SCRUM**

Es un proceso en el que se aplican de manera regular un conjunto de prácticas para trabajar en equipo y obtener un mejor resultado en un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de una manera de trabajar en equipos altamente eficiente.

Se realizan entregas parciales y regulares del producto final, dándole prioridad a las que aportan mayor beneficio al receptor del proyecto. Por eso, está especialmente indicado para proyectos en entornos complejos, donde es necesario obtener resultados pronto, donde los requisitos están en constante cambio o son poco definidos y donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Se utiliza para resolver situaciones en las que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable o cuando se requiere trabajar utilizando un proceso especializado en el desarrollo de producto.

### **Proceso**

El proyecto se ejecuta en ciclos temporales cortos y de duración fija iteraciones de 2-4 semanas. Cada iteración debe proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

La lista de objetivos/requisitos priorizada del producto actúa como plan del proyecto. Esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas.

### **Planificación de la iteración**

La primera parte de la iteración se divide a su vez en 2 partes:

1.-Selección de requisitos: el cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surjan y se seleccionan los requisitos más prioritarios que prevé se podrán completar en la iteración de manera que puedan ser entregados si el cliente lo solicita.

2.-Planificación de la iteración: el equipo elabora las listas de tareas de la iteración necesarias para desarrollar los requisitos seleccionados. La estimación del esfuerzo se hace de manera conjunta y los miembros del equipo se auto asignan las tareas y se organizan para trabajar incluso en equipos para compartir conocimientos o para resolver objetivos complejos.

### **Ejecución de la iteración**

Cada día se realiza una reunión de sincronización. El equipo inspecciona el trabajo que el resto está realizando para poder hacer adaptaciones necesarias que permitan cumplir con la previsión de objetivos a mostrar al final de la iteración.

Durante la iteración el Scrum Master se encarga de que el equipo pueda mantenerse enfocado para poder cumplir sus objetivos. Durante la iteración el cliente junto con el equipo refina la lista de requisitos y de ser necesario cambian o re planifican los objetivos del proyecto con el objetivo de maximizar la utilidad de lo que se está desarrollando y el retorno de inversión.

### **Inspección y adaptación**

El último día de la iteración se realiza una revisión de esta, la cual está dividida en 2 partes.

1.-Revision (demostración): el equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios en el contexto del proyecto, el cliente realiza adaptaciones de manera objetiva, desde la primera iteración, re planificando el proyecto.

2.-Retrospectiva: el equipo analiza cómo fue su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continúa su productividad. El Scrum Master se encargará de eliminar o sortear los obstáculos identificados que estén más allá del ámbito de acción del equipo.

## **Manifiesto por el desarrollo ágil de software**

Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

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.

El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

El software funcionando es la medida principal de progreso.

Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

## **Conclusiones**

Este manifiesto nos demuestra los puntos más importantes para el desarrollo de software dándole un papel central al de satisfacer al cliente mediante un producto que se entregue en tiempo y de manera continua. Además de mencionar cuales son las formas en que se trabajan tanto con el cliente como con los colaboradores.

Son la base fundamental en el desarrollo de software y se puede ver en la mayoría de ellos y también se puede ver los puntos que no tomaron muy en cuenta, es una manera interesante de trabajar ya que se trabaja directamente con todos los involucrados y se trata de tener un ambiente de trabajo lo más agradable posible para todos y donde todos cooperan en cuanto ideas y opiniones dándole preferencia al cliente de esta forma se mejora la competitividad en el mercado.

### **Extreme programming (Programación Extrema)**

Es el más destacado de los procesos ágiles de desarrollo de software. Se diferencia de las metodologías tradicionales principalmente porque pone más énfasis en la adaptabilidad que en la previsibilidad.

Los usuarios de dicha metodología consideran que los cambios en los requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable en el desarrollo de proyectos. Se puede considerar a la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo en el proyecto y aplicarlo de una manera dinámica.

La simplicidad es la base de este método, se simplifica el diseño para agilizar el desarrollo y facilitar su entendimiento. También se busca la simplicidad en la documentación comentándolo en su justa medida intentando que el código este autodocumentado.

La comunicación se realiza de diferentes formas. Para los programadores el código comunica mejor mientras más simple sea. El código autodocumentado es más fiable que los comentarios ya que estos se desfasan a medida que el código es modificado.

Las pruebas unitarias son otra forma de comunicación las cuales describen el diseño de las clases y los métodos al mostrar ejemplos de cómo utilizar su funcionalidad. Los programadores se comunican gracias a la programación por parejas. La comunicación con el cliente es fluida ya que esta forma parte del equipo de desarrollo. El cliente decide qué características tiene prioridad y siempre debe estar disponible para solucionar dudas.

Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real y al tener ciclos muy cortos para mostrar resultados, se minimiza el tener que rehacer partes que no cumplan con los requisitos y ayuda a los

programadores a centrarse en lo más importante. El código también funciona como retroalimentación gracias a las herramientas de desarrollo.

Sus características fundamentales son:

- Pruebas unitarias continuas
- Programación en parejas
- Frecuente integración del equipo de programación con el cliente o usuario
- Corrección de todos los errores antes de añadir una nueva funcionalidad
- Refactorización del código
- Propiedad del código compartida
- Simplicidad en el código

Los roles que se ocupan en esta metodología son:

- Programador:  
Produce el código del sistema
- Test developer:  
Produce el código de los test unitarios del sistema
- Cliente:  
Escribe las historias de usuario, pruebas funcionales, da prioridad a las historias de usuario y decide cuales se implementan en cada iteración
- Tester:  
Interpreta el pedido del cliente y ayuda al equipo de desarrollo a escribir las pruebas funcionales
- Tracker:  
Se encarga de dar seguimiento, proporciona realimentación al equipo
- Entrenador (coach):  
Responsable del proceso global
- Consultor:  
Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto
- Gestor (big boss):  
Dueño de la tienda y vínculo entre clientes y programadores

## **Agile Unified Process (Proceso Unificado Agil)**

Es una versión simplificada del Proceso Unificado Racional (RUP), el cual describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software usando técnicas ágiles y conceptos que se mantienen validos en RUP.

Tiene siete flujos de trabajo cuatro ingenieriles y tres de apoyo los cuales son:

### Ingenieriles

- Modelado
- Implementación
- Prueba
- Despliegue

### Apoyo

- Gestión de configuración
- Gestión de procesos
- Ambiente

El flujo del modelado agrupa los 3 primeros flujos de RUP (Modelamiento del negocio, requerimientos y análisis y diseño).

Dispone de 4 fases igual que RUP:

- Incepción o creación: identifica el alcance y la dimensión del proyecto, propone la arquitectura y el presupuesto al cliente
- Elaboración: confirmación idónea de la arquitectura
- Construcción: desarrollo incremental del sistema, siguiendo las prioridades funcionales de los implicados
- Transición: validación e implementación del sistema

Dentro de los flujos de trabajo ingenieriles se encuentran:

- El modelado es el flujo de trabajo que tiene el objetivo de entender el negocio de la organización.
- La implementación tiene como objetivo transformar el modelo en código ejecutable y realizar un nivel básico de las pruebas.
- La prueba realiza una evaluación objetiva para garantizar la calidad, buscando defectos y verificando que se cumplan los requerimientos.
- El despliegue cuyo objetivo es el plan para la prestación del sistema y la ejecución de dicho plan.

Las técnicas ágiles que aplica el Agile Unified Process son;

- Desarrollo dirigido por pruebas
- Modelo ágil
- Gestión de cambios ágil
- Refactorización de base de datos para mejorar la productividad

Los principios en los que se basa Agile Unified Process son:

- Simplicidad: todo se describe concisamente utilizando poca documentación
- Agilidad: el ajuste a los valores y principios del manifiesto ágil
- Centrarse en actividades de alto valor: la atención se centra en las actividades que en realidad lo requieren
- Herramienta de la independencia: se puede usar cualquier conjunto de herramientas que se desee
- Se quiera adaptar este producto para satisfacer nuestra propias necesidades: al ser un producto de fácil uso utilizando cualquier herramienta, no es necesario el uso de herramientas especiales para poder adaptar dicha metodología



## **Agile Modeling (Modelado Ágil)**

Es una metodología basada en la práctica para modelado efectivo de sistemas de software. Esta es guiada por una colección de prácticas, guiadas por principios y valores que pueden ser aplicados por profesionales de software. No es un proceso prescriptivo, ni define procedimientos detallados de cómo crear un tipo de modelado, en lugar de eso sugiere prácticas para ser un modelador efectivo.

Cuenta con las siguientes características:

- Satisface su propósito
- Es inteligible
- Es suficientemente preciso
- Es suficientemente consistente
- Es suficientemente detallado
- Aporta valor positivo
- Es lo más simple posible

El modelo ágil es una actitud, no un proceso prescriptivo.

Es un complemento de los métodos existentes, no es una metodología completa.

Es una manera efectiva de trabajar en conjunto para alcanzar las necesidades de las partes interesadas en el proyecto.

Es efectivo y se trata sobre ser efectivo.

Es algo que funciona en la práctica, no en la teoría académica.

Es para el desarrollador promedio, no es un sustituto de la gente competente.

No es un ataque a la documentación, de hecho aconseja la creación de documentos que tengan valor.

No es un ataque a las herramientas CASE (Ingeniería de software Asistida por Computadora).

No es para todos.

Entre los valores que conforman el Modelado Ágil están:

- Comunicación
- Coraje
- Retroalimentación
- Humildad
- Simplicidad

Además cuentan con valores del manifiesto por el desarrollo ágil de software:

- Individuos e interacciones más que procesos y herramientas.
- Software operante más que documentación completas.
- Colaboración con el cliente más que negociaciones contractuales.
- Respuesta al cambio más que apegarse a una rigurosa planificación.

Como principios centrales se encuentran:

- Asumir simplicidad
- Bienvenido el cambio
- Permitir el siguiente esfuerzo es el objetivo secundario
- Cambio incremental
- Maximizar la inversión de las partes interesadas en el proyecto
- Modelar con un propósito
- Múltiples modelos
- Trabajo de calidad
- Rápida retroalimentación
- El software es el objetivo primario
- Viaje con poco equipaje

Además de algunos principios adicionales:

- El contenido es más importante que la representación
- Todos podemos aprender de todos
- Conozca sus modelos
- Conozca sus herramientas
- Adaptación local
- Comunicación abierta y honesta
- Trabaje con el instinto de las personas

