



**UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE  
MÉXICO**

**ASIGNATURA: INTRODUCCIÓN A LA INGENIERÍA DE  
SOFTWARE**

**PROFESOR: GERARDO HERNÁNDEZ HERNÁNDEZ**

**ALUMNO: JOSÉ LUIS GONZÁLEZ MARTÍNEZ**

**PROYECTO: METODOLOGÍAS ÁGILES**

## SCRUM

Scrum es un método de regulación en las empresas para el desarrollo de proyectos. Se dice metodología ágil porque en el grupo de trabajo se han de desarrollar tareas en un período de tiempo determinado, conocidos como iteraciones.

Dichas iteraciones deben entregarse en un tiempo establecido con el cliente. Al principio de esta metodología se redacta un registro sobre las funcionalidades que debe tener el producto, estas deberán ir ordenadas de mayor a menor importancia para el negocio. Lo cual permitirá que se realicen más rápido.

Además, scrum trabaja con ciclos cortos y se caracteriza porque hay reuniones de forma periódica con el cliente, en donde está muy involucrado y sabe cómo avanza el proyecto.

## Extreme programming (XP)

Se concentra en potenciar las relaciones interpersonales como clave para el éxito en desarrollo del software, generando el trabajo en equipo.

XP se basa en retroalimentación continua entre cliente y el equipo de desarrollo, también adecuada para proyectos con requisitos imprecisos y muy cambiantes. Algunas características:

- *Se valora al individuo y las interacciones del equipo* de desarrollo sobre el proceso y las herramientas.
- *Desarrollar software que funciona* más que conseguir una buena documentación.
- *La colaboración con el cliente.* Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo.

- *Responder a los cambios.* La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto determina también el éxito o fracaso del mismo. La planificación debe ser flexible y abierta.

## *Feature-Driven Development (FDD)*

Esta metodología se enfoca en iteraciones cortas y de calidad que permiten entregas tangibles del producto en un periodo corto de tiempo, de como máximo dos semanas. Algunas características son:

- Se preocupa por la calidad, por lo que incluye un monitoreo constante del proyecto.
- Ayuda a contrarrestar situaciones como el exceso en el presupuesto, fallas en el programa o el hecho de entregar menos de lo deseado.
- Propone tener etapas de cierre cada dos semanas. Se obtienen resultados periódicos y tangibles.
- Se basa en un proceso iterativo con iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorear.
- Define claramente entregas tangibles y formas de evaluación del progreso del proyecto.
- No hace énfasis en la obtención de los requerimientos sino en cómo se realizan las fases de diseño y construcción.

## *Test Driven Development (TDD)*

Este tipo de metodología hace un código más robusto, más seguro, más mantenible y una mayor rapidez en el desarrollo. La idea principal es realizar de forma inicial las pruebas unitarias para el código que tenemos que implementar. Es decir, primero codificamos la prueba y, posteriormente, se desarrolla la lógica de negocio. Necesitamos cumplir con:

- Tener bien definidos los requisitos de la función a realizar. Debemos saber qué se quiere y qué posibles implicaciones puede tener en el código a desarrollar.
- Criterios de aceptación, contemplando todos los casos posibles, tanto exitosos como de error.
- Cómo vamos a diseñar la prueba. Para realizar un buen test unitario debemos ceñirnos únicamente a testear la lógica de negocio que queremos implementar, abstrayéndonos en cierto modo de otras capas o servicios que puedan interactuar con nuestra lógica, simulando el resultado de dichas interacciones.
- Qué queremos probar. Cada caso para cada criterio de aceptación debería llevar su prueba asociada.
- ¿Cuántos test son necesarios? Tantos casos que encontremos. De esta manera aseguramos que nuestra cobertura de pruebas es lo suficientemente fuerte como para asegurar el correcto funcionamiento del código desarrollado.