

# Metodologías de Desarrollo de Software

AXEL RODRÍGUEZ ESPINOZA

NICOLAS OYARZUN HERNANDEZ

PEDRO SALAS VERGARA

Universidad Tecnológica Metropolitana

## Abstract

*Metodología de desarrollo de software se describe como el conjunto de herramientas, técnicas, procedimientos y soporte documental para el diseño de Sistemas de información.*

## I. INTRODUCCIÓN

Particularmente, como ingeniería de software, una metodología se basa en una combinación de los modelos de proceso genéricos para obtener como beneficio un software que soluciones un problema. Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades, junto con prácticas, técnicas recomendadas y guías de adaptación de la metodología al proyecto. Sin embargo, la complejidad del proceso de creación de software es netamente dependiente de la naturaleza del proyecto mismo, por lo que el escogimiento de la metodología estará acorde al nivel de aporte del proyecto, ya sea pequeño, mediano o de gran nivel.

## II. METODOLOGÍAS

### I. Metodología Cascada

También conocido como modelo en cascada, esta metodología se enfoca en generar un orden riguroso en torno a etapas temporales que se aplican en el proceso de desarrollo de un software, de tal forma que al finalizar una fase, se comienza la siguiente tomando como datos de entrada los resultados de la anterior. En cada etapa del modelo se va detallando y perfeccionando el desarrollo, con la finalidad

de obtener un código eficaz y eficiente que satisfaga las necesidades (requisitos previamente determinados).

1.- Análisis de requisitos -> 2.- Diseño del Sistema -> 3.- Diseño del Programa -> 4.- Codificación -> 5.- Pruebas -> 6.- Implantación -> 7.- Mantenimiento ->

### Análisis de requerimientos

Se analizan las necesidades del usuario final del software, con el objetivo de determinar los objetivos específicos de este. Es importante que se realice una documentación completa y consensuar estos los requerimientos de modo que la proyección del desarrollo sea lo correcto (ya que no puede sufrir modificaciones de este tipo en etapas futuras).

Se analizan los requisitos del usuario, donde su objetivo es determinar los servicios que debe ofrecer el sistema para satisfacer las necesidades de este, se crea un documento de requisitos de usuario (DRU) para ser validado al finalizar el desarrollo. También se analizan los requisitos del sistema, que es básicamente la construcción de un modelo lógico de software que describe las funciones necesarias para resolver el problema y sus relaciones (no se toma en cuenta el como se implementará).

### Diseño del sistema

Se desglosa el sistema para que pueda elaborarse por separado. Esto se plasma en un documento de diseño de software, que describe la estructura relacional del sistema y lo que debe hacer cada una de sus partes, como también sus combinaciones. Se define la estructura de la solución, con sus módulos y relaciones. Con esto se define la arquitectura de la solución. Una vez realizado se definen los algoritmos y la organización del código para comenzar la implementación.

### Codificación o Programación

Como lo dice el nombre de la etapa, es la fase de programación, donde se implementa el código fuente, dando paso a ensayos constantes para la corrección de errores. Se crean librerías y funciones que puedan ser reutilizadas para llevar a cabo procesos mas rápidos.

### Pruebas

Se arma el sistema y se comprueba el funcionamiento efectivo de este.

### Implantación

Se pone en marcha la implantación del software, tomando en cuenta la documentación en torno al software y hardware que componen el proyecto.

### Mantenimiento

El sistema -ya en funcionamiento- puede sufrir cambios en la medida que el usuario sienta la necesidad de solucionar errores o incorporar mejoras a este.

Dentro de las características del modelo, importante es entender que mientras se necesiten realizar modificaciones en cualquiera de las etapas, se deben volver a recorrer las siguientes para llegar a la nueva solución. También es

riguroso con respecto a la revisión del proceso para cada fase, esto determinará si pasa a la siguiente o no.

## II. Metodología Scrum

Scrum es una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo, fue desarrollado por Jeff Sutherland y elaborado más formalmente por Ken Schwaber. Se enfoca en el hecho de que procesos definidos y repetibles sólo funcionan para atacar problemas definidos y repetibles con gente definida y repetible en ambientes definidos y repetibles. Y se divide un proyecto en iteraciones (que ellos llaman carreras cortas) de 30 días. La literatura de Scrum se orienta principalmente en la planeación iterativa y el seguimiento del proceso.

### Características

- Enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos, prácticas de desarrollo, implementación y demás cuestiones técnicas.
- Hace uso de Equipos auto-dirigidos y auto-organizados.
- Puede ser aplicado teóricamente a cualquier contexto en donde un grupo de gente necesita trabajar junta para lograr una meta común.
- Desarrollo de software iterativos incrementales basados en prácticas ágiles.
- Iteraciones de treinta días; aunque se pueden realizar con más frecuencia, estas iteraciones, conocidas como Sprint.
- Dentro de cada Sprint se denomina al Scrum Master al Líder de Proyecto quien llevará a cabo la gestión de la iteración.
- Se convocan diariamente un Scrum Daily Meeting el cual representa una reunión de avance diaria de no más de 15 minutos

con el propósito de tener realimentación sobre las tareas de los recursos y los obstáculos que se presentan.

En cuanto al ciclo de vida del modelo Scrum es el siguiente:

1. Pre-Juego / Planeamiento: El propósito es establecer la visión, definir expectativas y asegurarse la financiación. Las actividades son la escritura de la visión, el presupuesto, el registro de acumulación o retraso Metodologías Ágiles (backlog) del producto inicial y los ítems estimados, así como la arquitectura de alto nivel, el diseño exploratorio y los prototipos. El registro de acumulación es de alto nivel de abstracción.
2. Pre-Juego / Montaje (Staging): El propósito es identificar más requerimientos y priorizar las tareas para la primera iteración. Las actividades son planificación, diseño exploratorio y prototipos.
3. Juego / Desarrollo: El propósito es implementar un sistema listo para entrega en una serie de iteraciones de treinta días llamadas corridas (sprints). Las actividades son un encuentro de planeamiento de corridas en cada iteración, la definición del registro de acumulación de corridas y los estimados, y encuentros diarios de Scrum.
4. Pos-Juego/ Liberación: El propósito es el despliegue operacional. Las actividades, documentación, entrenamiento, mercadeo y venta.

### III. Diferencias entre los modelos de proceso tradicional y ágil

#### Metodología ágil:

- Están basadas en heurística provenientes de prácticas de producción de códigos.
- Están preparadas para cambios durante el proyecto.
- Son impuestas internamente (por el equipo).
- Proceso menos controlado.
- No existe contrato tradicional.
- Son bastante flexibles.
- El cliente es parte del equipo de desarrollo.
- Grupos pequeños y trabajando en el mismo sitio.
- Menos énfasis en la arquitectura del software.

#### Metodología tradicional:

- Basadas en normas provenientes de estándares.
- Presentan cierta resistencia a los cambios.
- Impuestas externamente.
- Proceso mucho más controlado, con numerosas políticas.
- Existe un contrato prefijado.
- Son un poco rígidas.
- El cliente interactúa con el equipo de desarrollo mediante reuniones.
- Grupos grandes y posiblemente distribuidos.
- La arquitectura del software es esencial y se expresa mediante modelos.