# METODOLOGIAS DE DESARROLLO DE SOFTWARE TRADICIONALES VS AGILES

# Metodologías Tradicionales y Agiles en el desarrollo de software

# Samil Leonel Sanchez Acevedo

# Corporación Universitaria Inpahu

# Facultad de Ingeniería

# Noviembre 2018

# Abstract

Desarrollar software implica muchas cosas, desde su planificación hasta la puesta en marcha se deben de seguir un sinnúmero de pasos o actividades. Hoy en día existen diversas metodologías para hacerlo, sin embargo, es necesario definir primero la naturaleza del software antes de elegir un determinado ciclo de vida. En este ensayo vamos a tratar de mirar las dos metodologías más utilizadas en el desarrollo de software, como son las tradicionales y las agiles las cuales nos dan unos pasos a seguir dependiendo del tipo de proyecto en su ciclo de vida.**Ciclo de vida del desarrollo**

Los primeros desarrolladores de software realizaban los proyectos de manera empírica, dado que no se llevaba un orden en el desarrollo optaron por incluir metodologías ya existentes y adaptarlas en los proyectos de software. La solución a esto fue la introducción de la Programación Estructurada, minimizando los varios factores que pueden conllevar al fracaso de un proyecto de desarrollo de software. Actualmente, se distinguen 2 grandes grupos: las metodologías tradicionales y las ágiles, estas distintas metodologías se han basado en un conjunto de actividades básicamente técnicas, perfectamente diferenciadas y relacionadas con: análisis, diseño, programación y prueba

***Metodología Tradicional***

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada

Entre las principales metodologías tradicionales tenemos los ya tan conocidos RUP y MSF entre otros, que centran su atención en llevar una documentación exhaustiva de todo el proyecto y centran su atención en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto.

modelo de cascada  
En [Ingeniería de software](http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software) el desarrollo en cascada, también llamado modelo en cascada, es el enfoque metodológico que ordena rigurosamente las etapas del [proceso para el desarrollo de software](http://es.wikipedia.org/wiki/Proceso_para_el_desarrollo_de_software), de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. Como se muestra en la figura 1.

Un ejemplo de una metodología de desarrollo en cascada es:

**Análisis de requisitos.**

**Diseño del Sistema.**

**Diseño del Programa.**

**Codificación.**

**Pruebas.**

**Implantación.**

**Mantenimiento.**

De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo. Dando así el modelo de las siguientes metodologías tradicionales como los son , RUP (Rational Unified Procces), MSF (Microsoft Solution Framework), Win-Win Spiral Model, Iconix



Figura 1.

***Metodología Ágil***

Las metodologías ágiles son un conjunto de técnicas para gestionar y desarrollar proyectos de software donde los requisitos evolucionan para adaptarse a las necesidades del proyecto. Algunos de los rasgos comunes entre las varias metodologías ágiles son:

* Considerar más importantes a los individuos e interacciones que a los procesos y herramientas
* Sobreponer que el software funcione a la documentación excesiva
* Colaborar con el cliente a la negociación contractual
* Priorizar la respuesta ante los cambios a seguir con un plan

Las metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas pragmáticas que puede que no curen todos los males, pero harán la entrega del proyecto menos complicada y más satisfactoria tanto para los clientes como para los equipos de entrega logrando sobrepasar los obstáculos de la planificación de un proyecto como se muestra en la figura 2. Entre las metodologías ágiles más destacadas hasta el momento se pueden nombrar:

• XP (Extreme Programming)

• Scrum

• Crystal Clear

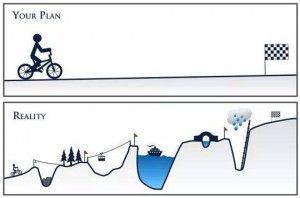
• DSDM (Dynamic Systems Developmemt Method)

• FDD (Feature Driven Development)

• ASD (Adaptive Software Development)

• XBreed

• Extreme Modeling

figura 2.

En resumen, ni cascada significa fracaso o arcaísmo, ni incremental significa éxito o ágil, cada equipo debe decidir cuál es el ciclo más conveniente para su proyecto. La elección del ciclo de vida del proyecto debe realizarse según las condiciones de cada proyecto.  Qué dice el PMBOK® al respecto ?  Primero mucha gente suele confundir los Procesos para la Dirección del proyecto con el Ciclo de vida de un proyecto.  Respecto de este último lo que menciona la Guía es lo siguiente: “*El ciclo de vida del proyecto define las fases que conectan el inicio de un proyecto con su fin. La transición de una fase a otra dentro del ciclo de vida de un proyecto generalmente implica y, por lo general, está definida por alguna forma de transferencia técnica. Generalmente, los productos entregables de una fase se revisan para verificar si están completos, si son exactos y se aprueban antes de iniciar el trabajo de la siguiente fase. No existe una única manera, que sea la mejor, para definir el ciclo de vida ideal de un proyecto. Asimismo, las prácticas comunes de la industria a menudo conducen a usar un ciclo de vida preferido dentro de dicha industria*”.

La metodología ágil de proyectos se utiliza básicamente para proyectos de IT y dentro de ellos esencialmente para el desarrollo de software.  Surgieron como respuesta a la dificultad de acomodar cambios, requerimientos no claros, entregas rápidas y adaptación. Las prácticas ágiles no están limitadas al uso de Scrum, XP, FDD u otra “marca” de métodos ágiles

Como simplificación podríamos decir que utilizando estas 3 técnicas estamos de alguna manera agilizando nuestro proceso:

1. Desarrollo Iterativo-Incremental: cada iteración del SDLC sigue el mismo proceso de desarrollo, entrega algo más de valor al cliente en etapas y espera su aprobación para la siguiente iteración, permite introducir cambios o mejoras al final de cada iteración, el cliente y el equipo colaboran para determinar el alcance de cada iteración (historias de usuario que definan la funcionalidad, lista de “features” agrupados, selección de requerimientos de una backlog list, etc). Esta aproximación de desarrollo apunta a otro concepto importante: el “time Boxing”, donde el PM pregunta cuanto del proyecto (en términos de funciones) puede ser desarrollado en un período de tiempo preestablecido (normalmente corto: dos o cuatro semanas). Cada desarrollo incremental otorga al cliente un paso más para llegar al producto final.
2. Equipos de alto rendimiento que trabajan junto con el cliente: muchas prácticas ágiles incluyen al cliente en fases tempranas del desarrollo y está siempre representado dentro del equipo de trabajo (on site, experto, product owner, etc.). Esta técnica (incluir al cliente dentro del equipo de trabajo) ofrece a menudo mucha resistencia dado que los clientes no quieren dedicar recursos full-time al desarrollo. Es cuestión de hacerle entender al cliente que la calidad del producto final se incrementa en proporción al tiempo que el cliente dedica su tiempo al equipo de desarrollo.
3. Mantener el desarrollo simple: un desarrollo ágil trata de que el proceso sea simple, el código menor posible, entregar solo lo que sea necesario y documentar solo lo imprescindible.  Se utiliza mucho la práctica de “refactoring” para simplificar el código y eliminar redundancias. Otra opción de mantener las cosas simples es la práctica de Lean Project Management que básicamente consiste en emplear las técnicas de Lean Manufacturing, Six Sigma y TOC en la gestión de proyectos (herramientas de control estadístico, procesos ágiles o rápidos y eficientes, value stream mapping, uso de cadena crítica, eliminación de desperdicios y utilización de los principios de “Poka Yone” entre otros.

Ya con estas tres técnicas básicas dadas por la academia Project Management podemos lograr identificar con mayor facilidad que metodología vamos a utilizar para la realización de nuestro proyecto, dando herramientas para la creación de nuevos métodos combinando los ya existentes como lo es La metodología Scrum está pensada para aumentar la productividad, y XP para agregar valor y calidad al proyecto. Por esto muchas organizaciones que utilizan Scrum adoptan el Pair Programming, el TDD y la refactorización como prácticas que mejoran calidad, consiguiendo al mismo tiempo una productividad alta.

También cabe recalcar que a menudo se utiliza el tablero de Kanban por los equipos de Scrum para realizar un seguimiento de las tareas que hay en cada sprint. Esto se basa sobre la experiencia recogía en los diferentes manejos de metodologías siguiendo los tres pasos anteriormente citados

# Referencias

1. Metodologías tradicionales y agiles del software <http://www.milenio.com/opinion/varios-autores/universidad-tecnologica-del-valle-del-mezquital/las-metodologias-tradicionales-y-el-desarrollo-agil-de-software>
2. Presentación de las diferentes metodologías del software https://prezi.com/xpo8pdv2ghqm/metodologias-tradicionales-vs-metodologias-agiles/
3. Metodologías de desarrollo de software <http://masteringenieriasoft.blogspot.com/2012/04/metodologias-de-desarrollo-de-software.html>
4. Metodología Tradicional

<https://pmqlinkedin.wordpress.com/about/metodologia-tradicional-o-agil/>

1. Metodologías Agiles scrum y Kanban

<https://www.quonext.com/blog/metodologias-agiles-scrum-kanban-xp/>

1. Metodología en cascada en el desarrollo

<http://modelo-cascada.blogspot.com/>