Paradigmas metodológicos

Sitio: Agencia de Habilidades para el Futuro
Curso: Práctica Profesional 1: Aproximación al mundo laboral 1º D
Libro: Paradigmas metodológicos

Imprimido por: Eduardo Moreno
Día: lunes, 5 de mayo de 2025, 17:49

Tabla de contenidos

Introducción

Un modelo de desarrollo de software es una abstracción de un proceso real, desde una perspectiva específica. Este ofrece un marco de trabajo para controlar el proceso de desarrollo de software.

Estos modelos no representan cómo se debe desarrollar el software sino que son enfoques enfoque comunes de los pasos a seguir, de qué cosas describir. Cada modelo puede ser modificado y adaptado de acuerdo con las necesidades del software en proceso de desarrollo.

Hay varios modelos para perfilar el proceso de desarrollo, cada uno de los cuales cuenta con pros y contras. Por lo que se opta por el más apropiado para el proyecto a desarrollar. En ocasiones puede que una combinación de varios modelos sea lo mejor.

Existen tres paradigmas de las metodologías de desarrollo de software:

- Paradigma tradicional
- Paradigma orientado a objetos
- Paradigma de desarrollo ágil

A continuación realizaremos un recorrido por las características principales de cada uno.

Paradigma tradicional (hegemónico)

¿Qué es?

- Se lo llama, de forma despectiva, como metodologías pesadas.
- 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 centra especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada.
- No se adapta adecuadamente a los cambios, por lo que sus métodos no son convenientes cuando se trabaja en un entorno donde los requisitos no pueden predecirse o bien pueden variar.
- Las etapas realizadas no son autónomas de las siguientes, creando una dependencia estructural y en el caso de un error se atrasa todo el proyecto.

Paradigma orientado a objetos (hegemónico)

¿Qué es?

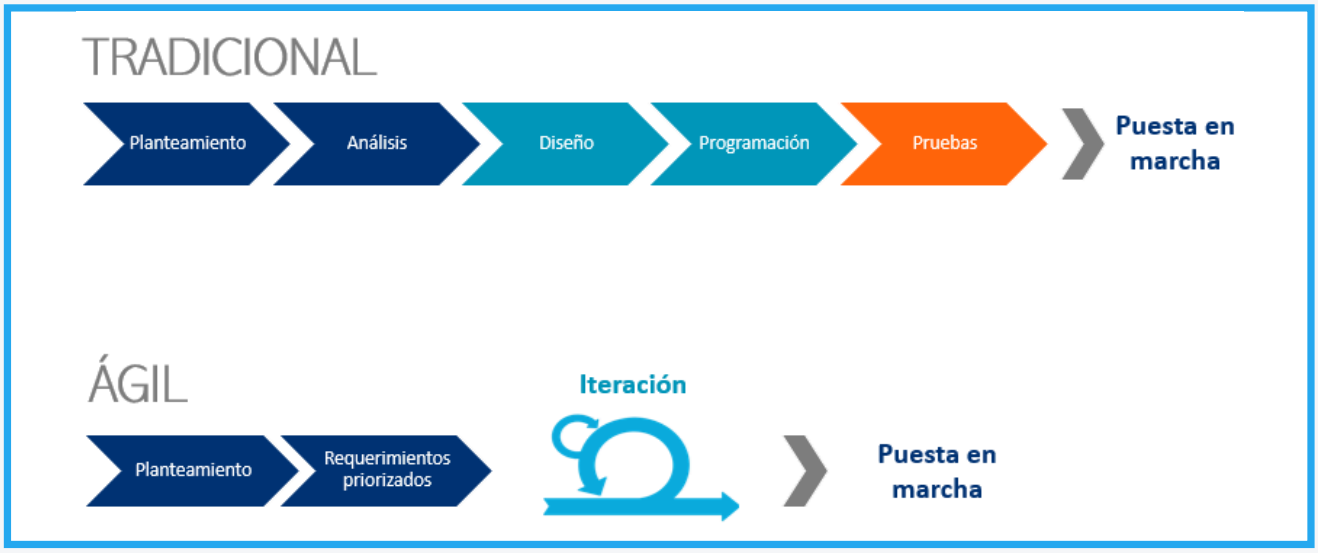
- Se basa en la programación orientada a objetos; por lo tanto, se refiere al concepto de clase, el análisis de requisitos y el diseño.
- Permite la reutilización de software.
- Facilita el desarrollo de herramientas informáticas de apoyo, el cual es simple al implementarla en una notación orientado a objetos llamado UML: Unified Modeling Language.

Paradigma de desarrollo ágil

Características:

- Es un paradigma de las metodologías de desarrollo basado en procesos ágiles.
- Estos intentan evitar los tediosos caminos de las metodologías tradicionales enfocándose en las personas y los resultados.
- Usa un enfoque basado en el valor para construir software, colaborando con el cliente e incorporando los cambios continuamente.

En la siguiente imagen se observan las diferentes etapas de trabajo de la metodología tradicional y la metodología ágil.



¿Cuál pensás que es la principal diferencia entre ambas?

Seguí leyendo el contenido para conocer más sobre estas metodologías.

Metodologías

Antes de continuar, es importante es importante saber qué es un método.

Para ello debemos reconocerlo como:

- Modo ordenado y sistemático de proceder para llegar a un resultado o fin determinado.
- Procedimiento que se sigue para conseguir algo.
- Un método se compone por diversos aspectos que nos permiten conseguir una meta o lograr un objetivo.

La metodología hace referencia al conjunto de procedimientos racionales utilizados para alcanzar un objetivo que requiera habilidades y conocimientos específicos. La metodología es una de las etapas específicas de un trabajo o proyecto que parte de una posición teórica y conlleva a una selección de técnicas concretas o métodos acerca del procedimiento para el cumplimiento de los objetivos. Es el conjunto de métodos que se utilizan en una determinada actividad con el fin de formalizarla y optimizarla. Determina los pasos a seguir y cómo realizarlos para finalizar una tarea.

- Quién debe hacer
- Qué debe hacer
- Cuándo debe hacerse
- Cómo debe hacerse
- Con qué debe hacerse

Manifiesto por el desarrollo ágil del software

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

Individuos e interacciones	⇨	sobre procesos y herramientas
Colaboración con el cliente	⇨	sobre negociación contractual.
Respuesta ante el cambio	⇨	sobre seguir un plan

Aunque valoremos los elementos de la derecha, tendremos más en cuenta los de la izquierda.

Te invitamos a ver el siguiente video donde se desarrollan los 12 principios de manifiestos ágiles.

[Clase 03] Los 12 principios del manifiesto ágiles

Para más información podés ingresar en:

https://agilemanifesto.org/iso/es/manifesto.html

Metodologías ágiles vs. tradicionales

En la página anterior empezamos a conocer las diferencias entre ambas metodologías, ahora profundizamos en ese eje.

Leé el siguiente cuadro comparativo para conocer más.

Metodologías ágiles	Metodologías tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.
Poca documentación.	Documentación exhaustiva.
Muchos ciclos de entrega.	Pocos ciclos de entrega.

Metodologías ágiles

Este enfoque está orientado a dar respuesta a los problemas que pueden ocasionar las metodologías tradicionales. Por lo que se centra en dos aspectos:

- retrasar las decisiones,
- planificación adaptativa.

Se destaca que la base de sus fundamentos es la adaptabilidad de los procesos de desarrollo.

Características:

- Proceso Incremental (entregas frecuentes con ciclos rápidos),
- Cooperativo (clientes y desarrolladores trabajan constantemente con una comunicación muy fina y constante),
- Sencillo (el método es fácil de aprender y modificar para el equipo)
- Adaptativo (capaz de permitir cambios de último momento).

Estas metodologías combinan una serie de pautas y principios junto a técnicas pragmáticas que hacen que la entrega del proyecto sea rápida y efectiva para quienes participan en el proceso.

Evitan instancias burocráticas e impulsan una capacidad de respuesta que posibilite el cambio. Evitando planes de trabajos extensos, centrados en la documentación y métodos formales.

Al aplicar este tipo de metodologías podemos dar cuenta de cómo organizar los contenidos para diseñar un cualquier proyecto en relación de dependencia o de manera autónoma.

A nivel laboral la implementación de estas metodologías presenta las siguientes características:

- **Definen de manera organizada** qué labora cumplirá cada trabajador.
- **Mejoran el trabajo** en equipo a través de la planificación de tareas.
- **Optimizan los tiempos** y se adaptan al desarrollo de sistemas grandes y pequeños.
- **Promueven la agilidad** y el sentido práctico.

Para tener en cuenta:

Podemos decir que un proyecto es un plan de trabajo. Y para ello es necesario contar con una serie de preguntas que nos orienten en la organización. Te invitamos a descargar esta ficha (accedé haciendo clic aquí) y ponerla en práctica.

A continuación veremos un ejemplo de metodología ágil e interactiva.

RAD - Desarrollo rápido de aplicaciones

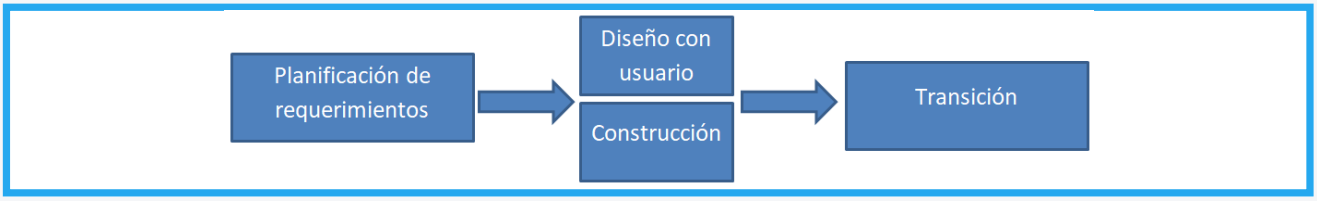
Características de la RAD

- La metodología de desarrollo conocida como diseño rápido de aplicaciones RAD (*rapid application development*) ha tomado gran auge debido a la necesidad que tienen las instituciones de crear aplicaciones funcionales en un plazo de tiempo corto.
- RAD es un ciclo de desarrollo diseñado para crear aplicaciones de computadoras de alta calidad.
- El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering).
- Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.
- Hoy en día se suele utilizar para referirnos al desarrollo rápido de interfaces gráficas de usuario tales como Glade, o entornos de desarrollo integrado completos.
- Algunas de las plataformas más conocidas son Visual Studio, Lazarus, Gambas, Delphi, Foxpro, Anjuta, Game Maker, Velneo o Clarion.
- En el área de la autoría multimedia, software como Neosoft Neoboo y MediaChance Multimedia Builder proveen plataformas de desarrollo rápido de aplicaciones, dentro de ciertos límites.

Fases de RAD 1

1. Planificación de requerimientos: durante esta etapa inicial, los diseñadores, desarrolladores y usuarios llegan a un acuerdo aproximado sobre el alcance del proyecto y los requisitos de la aplicación, para que puedan comenzar las etapas futuras con creación de prototipos.
2. Diseño con el usuario: los comentarios de los usuarios se recopilan con gran énfasis en la determinación de la arquitectura del sistema. Esto permite crear modelos y prototipos iniciales. Este paso se repite tantas veces como sea necesario a medida que el proyecto evoluciona.
3. Construcción: una vez que ha comenzado el diseño básico del usuario y del sistema, la fase de construcción es donde se lleva a cabo la mayor parte de la codificación, las pruebas y la integración reales de la aplicación. Junto con el diseño del usuario, la fase de construcción rápida se repite tantas veces como sea necesario, a medida que se requieran nuevos componentes o se realicen modificaciones para satisfacer las necesidades del proyecto.
4. Transicción: la etapa final de Transicción (o Cutover) le permite al equipo de desarrollo tiempo para mover los componentes a un entorno de producción en vivo, donde se pueden llevar a cabo todas las pruebas necesarias o la capacitación del equipo.

En la siguiente imagen vemos la información de forma esquemática.



¿Qué capacidades y habilidades vas a desarrollar?

Es un método de desarrollo ágil de software creado por Jeff Sutherland y su equipo de desarrollo a principios de la década de 1990. Los principios Scrum son congruentes con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega. Dentro de cada actividad estructural, las tareas del trabajo ocurren con un patrón del proceso llamado sprint. El trabajo realizado dentro de un sprint (el número de éstos que requiere cada actividad estructural variará en función de la complejidad y tamaño del producto) se adapta al problema en cuestión y se define y con frecuencia se modifica en tiempo real por parte del equipo Scrum (Pressman, R. 2010).

IMPORTANTE

Es una una metodología que le permite a las personas concentrarse en problemas adaptativos complejos de manera productiva y creativa a la vez que permite entregar productos con el mayor valor posible. Se puede decir que es una metodología ágil que utiliza secuencias de trabajo acumulativas e iterativas.

Al utilizar Scrum, el equipo de trabajo puede visualizar cómo avanza el proyecto poniendo en juego tres categorías: **"Por hacer", "Haciendo" y "Hecho".** Además, retomando lo trabajado en la semana anterior, es posible mantener una participación activa con retroalimentaciones sobre cada acción que vayan realizando.

Te invitamos a recorrer este **link** donde te encontrarás con un ejemplo de iteración Scrum.

Fuente: Pressman, R. 2010. Ingeniería del Software Un Enfoque Práctico. 7ma ed. University ofConnecticut. McGraw-Hill Interamericana Editores, S.A.