

Metodología de desarrollo software

Conjunto de métodos, procedimientos, técnicas, herramientas y soportes documentales que definen las reglas para realizar las transformaciones internas de las actividades de un modelo de Ciclo de Vida, y que permiten a los desarrolladores implementar nuevos productos de software.

Ciclo de vida

IEEE 1074: "Una aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software" ISO 12207: "Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso"

Proceso

Conjuto de pasos realizados para un fin determinado. (IEEE) Un conjunto de actividades interrelacionadas que transforman entradas en salidas (ISO 12207 / UNE 77104) Un proceso, entendido de manera general, es una serie de pasos que incluyen actividades, restricciones y recursos que resultan en un producto determinado con ciertas características.

Diferencias

Ciclo de Vida: indica qué es lo que hay que obtener a lo largo de desarrollo del proyecto, pero no cómo.

Metodología: dota de contenido a los Ciclos de Vida. Una metodología puede seguir uno o varios modelos de Ciclo de Vida. **Proceso:** pasos de ingeniería necesarios para alcanzar un producto. Una metodología incluye varios procesos, como por ejemplo, proceso de gestión, proceso de análisis, etc.

Metodología de desarrollo software

Conjunto de métodos que se utilizan en una determinada actividad con el fin de formalizar y optimizarla. Determina los pasos a seguir y cómo realizarlos. Optimiza el proceso y el producto software. Métodos que guían en la planificación en el desarrollo del software. Define qué hacer, cómo y cuándo durante todo el desarrollo y mantenimiento de un proyecto. Define una estrategia global para enfrentarse con el proyecto. Entre los elementos que forman parte de una metodología:

- **Fases:** tareas a realizar.
- **Productos:** E/S de cada fase.
- **Procedimientos y herramientas:** apoyo a la realización de cada tarea.
- **Criterios de evaluación:** del proceso y del producto. Saber si se han logrados los objetivos.

Ventajas uso Metodologías

Gestión:

- Facilita la tarea de planificación.
- Facilita la tarea de control y seguimiento de un proyecto.
- Mejorar la relación coste/beneficio.
- Optimizar el uso de recursos disponibles.

- Facilitar la evaluación de resultados y cumplimiento de los objetivos
- Facilitar la comunicación efectiva entre usuarios y desarrolladores. Ingenieros de software:
- Ayudar a la comprensión del problema.
- Optimizar el proceso de desarrollo.
- Facilitar el mantenimiento del producto.
- Permitir la reutilización de partes del producto. Cliente o usuario.
- Garantía de calidad en el producto final.
- Confianza en los plazos fijados en la definición del proyecto.

Tipos de Metodologías

Elegir la metodología adecuada para un determinado proyecto es trascendental para el éxito del producto. Según la filosofías del desarrollo:

- Metodologías tradicionales.
- Metodologías ágiles. Según el paradigma:
- Metodologías estructuradas.
- Metodologías orientadas a objetos.

Metodologías tradicionales

Cumplir con un plan de proyecto definido en la fase inicia del desarrollo. Llevar una documentación exhaustiva de todo el proyecto. Altos costes ante cambios y falta de flexibilidad en proyectos donde el entorno es volátil. Se focalizan en la documentación, planificación y procesos.

Metodologías ágiles

Nacen como respuesta a los problemas que puedan ocasionar las metodologías tradicionales. Se basan e la adaptabilidad de los procesos de desarrollo: retrasar las decisiones y planificación adaptativa. La capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. Es un proceso Incremental, Cooperativo y Adaptativo.

Principios

1. Se encarga de valorar al individuo y las interacciones del equipo más que a las herramientas o los procesos utilizados.
2. El cliente está en todo momento colaborando en el proyecto.
3. Es más importante crear un producto software que funcione, que escribir mucha documentación.
4. Es más importante la capacidad de respuesta ante un cambio realizado que el seguimiento escrito de un plan.

¿Tradicionales o Ágiles?

Metodologías Tradicionales	Metodologías Ágiles
Especialización	Equipo multidisciplinar
Fases	Solapamiento
Requisitos detallados	Visión del producto

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

Críticas al desarrollo ágil

Falta de estructura y documentación. Enorme coste para los clientes, encuentros a intervalos muy frecuentes. Muy complicado obtener estimaciones realistas del esfuerzo para proporcionar un presupuesto. Puede ser muy ineficiente si los requisitos de un área cambian durante varias iteraciones, misma programación varias veces.

¿Tradicionales o Ágiles?

características relevantes para decidir la metodología más adecuada:

1. Principal prioridad de negocio.
2. Estabilidad de los requisitos.
3. Rigidez del producto.
4. Coste de prototipado.
5. Criticidad del sistema.
6. Tamaño del sistema.

Metodología tradicional Métrica versión 3

MÉTRICA es una metodología para la simplificación, desarrollo y mantenimiento de sistemas de información (SI). Promovida por el antiguo Ministerio de Administraciones Públicas del Gobierno de España. Sistematización de actividades del ciclo de vida de los proyectos software en el ámbito de las administraciones públicas. Cubre el Proceso de Desarrollo y de Mantenimiento de SI. Abarca el desarrollo completo de SI sea cual sea su complejidad y magnitud. Debe adaptarse y dimensionarse de acuerdo a las características particulares de cada proyecto. Enfoque orientado al proceso. Descompone cada uno de los procesos en actividades, y éstas en tareas. Establece un conjunto de técnicas a utilizar y productos a obtener. Desarrollar SI de mayor calidad, productividad y satisfacción de los usuarios. Facilitar el mantenimiento posterior.

Métrica V3. Procesos

Planificación de Sistemas de información (PSI): Proporcionar un marco estratégico de referencia para los SI de un determinado ámbito de la Organización. **Desarrollo de Sistemas de Información:** Contiene todas las actividades y tareas que se debe llevar a cabo para desarrollar un sistema, cubriendo desde el análisis de requisitos hasta la instalación del software. **Mantenimiento de Sistemas de Información (MSI):** Obtener una nueva versión de un SI a partir de las peticiones de mantenimiento. **Estudio de Viabilidad del Sistema (EVS):** Analizar un conjunto concreto de necesidades, con la idea de proponer una solución a corto plazo. **Análisis del Sistema de Información (ASI):** Especialización detallada del SI, a través de un catálogo de requisitos y una serie de modelos. **Diseño del Sistema de Información (DSI):** Obtener la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del SI. **Construcción del Sistema de Información (CSI):** Construcción y prueba de los distintos componentes del SI. **Implantación y Aceptación del Sistema de Información (IAS):** Entrega y aceptación del sistema, incluyendo actividades para el paso a producción del sistema. Cada proceso e Interfaz está compuesto por **Actividades**, a su vez, las Actividades están compuestas por **Tareas**.

Proceso o Interfaz	# Actividades	# Tareas
Planificación de SI (PSI)	9	24
Estudio de Viabilidad del Sistema (EVS)	6	18
Análisis del SI (ASI)	11	34
Diseño del SI (DSI)	12	44
Construcción del SI (CSI)	9	19
Implantación y Aceptación del SI (IAS)	10	24
Mantenimiento de SI (MSI)	4	10
Gestión de Proyectos	16	29
Aseguramiento de la Calidad	25	38
Seguridad	31	34
Gestión de Configuración	5	8
Total	138	282

Métrica V3. Técnicas y Prácticas

Se utilizan para la elaboración de productos. Mejoran la productividad y aseguran la calidad de los productos. Una **técnica** es un conjunto de heurísticas y/o procedimientos que utiliza una notación específica. Una **práctica** representa un medio para la consecución de unos objetivos específicos de manera rápida, segura y precisa. Se distinguen 3 tipos:

- Técnicas de desarrollo.
- Técnicas de gestión de proyectos.
- Prácticas.

Métrica V3. Participantes

- Directivo
- Jefe de Proyecto
- Consultor
- Analista
- Programados

Metodología ágil Scrum

Scrum: Formación de rugby. Scrum es el término que define la formación más reconocible del rugby, en la que ambos equipos, agazapados y atezados entre sí empujan para obtener el balón y sacarlo de la formación sin tocarlo con la mano. A finales de los 80, Nonaka y Takeuchi compararon la nueva forma de trabajo en equipo que estaba identificando en las empresas tecnológicas más innovadoras con el avance en formación de Scrum. A finales de los 90, Ken Schwaber y Jeff Sutherland presentaron de forma conjunta un artículo describiendo el método Scrum. Características del método Scrum:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación u ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento táctico de las personas en equipos autoorganizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizarlas una tras otra en un ciclo secuencial o de cascada. **Sprint**: Iteración, núcleo central que proporciona la base de desarrollo iterativo e incremental. Cada sprint finaliza con la entrega de una parte operativa del producto: **incremento**. Se monitoriza la evolución de cada sprint en reuniones breves diarias.

Roles

Propietario del producto: decide en última instancia cómo será el resultado final y el orden en el que se van construyendo los sucesivos incrementos: qué se pone y qué se quita de la pila del producto y su prioridad.

Equipo de trabajo: Equipo **multifuncional**, en el que todos los miembros trabajan de forma solidaria con responsabilidad compartida. **Scrum Master**: Es el responsable del cumplimiento de las reglas de un marco de scrum. Proporciona la asesoría y formación necesaria al propietario del producto y al equipo.

Interesados: Usuarios, expertos, etc. Cualquiera que esté involucrado e interesado en el proyecto.