



Metodologías de Desarrollo Software

Ingeniería del Software

Raquel Martínez España

Facultad Politécnica

Grado en Ingeniería Informática

Miércoles, 10 de febrero de 2015

Universidad Católica San Antonio de Murcia - Tlf: (+34) 968 27 88 00 info@ucam.edu - www.ucam.edu

<i>Introducción</i>	3
<i>Conceptos</i>	3
<i>Visión histórica del desarrollo de metodologías</i>	4
Desarrollo Convencional.....	4
Desarrollo Estructurado	5
Desarrollo Orientada a Objetos	6
<i>Características principales de las metodologías</i>	6
<i>Clasificación de las metodologías</i>	7
Metodologías Estructuradas	7
Metodologías orientadas a procesos.....	8
Metodologías orientadas a datos jerárquicos.....	9
Metodologías orientadas a datos no jerárquicos.....	9
Metodologías Orientadas a Objetos	9
<i>Principales metodologías de desarrollo</i>	10
Metodología MERISE.....	10
Metodología SSADM.....	11
Metodología Métrica	11
<i>Referencias</i>	12

Introducción

Para desarrollar un proyecto de software es necesario establecer un enfoque disciplinado y sistemático. Las metodologías de desarrollo se elaboran a partir del marco definido por uno o varios ciclos de vida. A continuación vamos a definir algunos conceptos generales y vamos a dar una perspectiva acerca de las metodologías de desarrollo software.

Conceptos

Antes de empezar a desarrollar y analizar las diferentes metodologías de desarrollo de software, vamos a definir un conjunto de conceptos básicos.

Metodología: conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información. En otras palabras, una metodología de desarrollo de software define un estilo o una forma de guiar el proceso de desarrollo de un proyecto informático. Para este fin, las metodologías definen una estructura jerárquica de pasos a seguir, organizados en fases, módulos, actividades, etapas, etc.

Tarea: Actividades elementales en que se dividen los procesos.

Procedimiento (método): forma en la que se ejecuta la tarea y se obtienen los productos.

Técnica: determinan el formato de los productos resultantes de cada tarea. Se utiliza para aplicar un procedimiento y se pueden utilizar una o varias.

Herramienta: utilizada para realizar una técnica, se pueden utilizar herramientas software que automatizan su aplicación (por ejemplo herramientas CASE).

Producto: resultado obtenido de un procedimiento.

Metodologías vs Ciclo de Vida vs método: Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. Una metodología es un concepto más amplio que método, puesto que una metodología puede considerarse como un conjunto de métodos. La metodología indica cómo hay que obtener los distintos productos parciales y finales. Así una metodología es un conjunto de componentes que especifican:

- Cómo se debe dividir un proyecto en etapas según el ciclo o ciclos de vida seguidos.
- Qué tareas se llevan a cabo en cada etapa.
- Qué salidas se producen y cuando se deben producir
- Qué restricciones se aplican

Asignatura

4

- Qué herramientas se van a utilizar
- Cómo se gestiona y controla un proyecto.

De forma general podemos identificar tres necesidades principales que se intentan cubrir con una metodología:

- Aplicaciones de mejor calidad.
- Un proceso de desarrollo que identifica los productos intermedios de cada fase, de forma que se puede planificar y controlar el proyecto en todo momento.
- Un proceso estándar en la empresa.

Dependiendo de las metodologías seleccionadas, éstas pueden tener objetivos diferentes.

- Registrar los requisitos de un sistema de información de una forma acertada
- Proporcionar un método sistemático de desarrollo de forma que se pueda tener controlado su progreso.
- Construir un sistema de información dentro de un tiempo prefijado y ajustado a unos costes.
- Construir un sistema bien documentado y de fácil mantenimiento.
- Ayudar a identificar, dentro de lo posible, cualquier cambio que sea necesario.
- Proporcionar un sistema que sea adecuado a todas las personas afectadas, ya sean clientes, directivos, etc.

Visión histórica del desarrollo de metodologías

Los enfoques metodológicos han ido evolucionando a lo largo del tiempo. El primer periodo que se identifica es un periodo de desarrollo convencional durante el cual las prácticas de desarrollo eran totalmente artesanales sin ninguna metodología definida. Después de este periodo comenzó a verse la necesidad de definir una metodología para disponer de unas aproximaciones más sistemáticas en el desarrollo. Después del desarrollo convencional surgió el desarrollo estructurado, partiendo de la programación estructurada a los que siguieron los métodos de análisis y diseño estructurados, hasta llegar a las metodologías estructuradas que cubren el ciclo de vida completo. Después apareció el paradigma de la orientación a objetos como un nuevo enfoque en la ingeniería del software.

Desarrollo Convencional

Comenzó en los años 50 cuando los sistemas se asociaban principalmente a las aplicaciones de desarrollo científico. Estas aplicaciones se basaban en funciones básicas de proceso de datos, como copias, recuperaciones, etc. Las personas que desarrollaban tales sistemas eran programadores, más orientados a la tarea de codificar que a la de recoger y comprender las

necesidades de los clientes. De ahí surgió la necesidad del análisis y el diseño en el desarrollo de los sistemas y comenzó a hablarse de analistas-programadores y de analistas de sistemas, pero aun así este enfoque convencional tiene serios problemas, donde los más relevantes son:

- Los resultados finales son impredecibles: No se sabe la fecha exacta de finalización de un proceso, ya que el proceso de desarrollo no depende del método sino de la capacidad y experiencia de los desarrolladores.
- No hay forma de controlar lo que está sucediendo en el proyecto: Esto sucede porque no hay fases preestablecidas, ni productos intermediarios sobre los que realizar verificaciones, de ahí, que la detección de los defectos sea bastante tardía con este tipo de desarrollo.
- Los cambios organizativos afectan negativamente al proceso de desarrollo: Esto sucede porque no suelen existir documentos estandarizados y si existen no suelen estar actualizados.

Desarrollo Estructurado

El nacimiento de técnicas estructuradas se puede considerar el punto de partida en el que se pasa de la construcción de programas de forma artesanal a una construcción de programas que sigue unos métodos de ingeniería, sentando las bases para un desarrollo automatizado. La programación estructurada estableció unas normas para la aplicación de las estructuras de datos y de control. Además el enfoque estructurado también se extendió posteriormente a la fase de diseño, donde se utilizaba el módulo de programa como componente básico de construcción. En la gran mayoría de los proyectos de desarrollo se hacía una especificación narrativa de los requisitos. Estas especificaciones estaban afectadas por diversos problemas, por ejemplo:

- Eran monolíticas: había que leer toda la especificación para llegar a comprenderla.
- Eran redundantes: se solía repetir información, por lo que el cambio de cualquier requisito, suponía modificar en varias partes del documento.
- Eran ambiguas: los requisitos eran interpretados de diferente forma por diferentes personas.
- Imposibles de mantener: cuando se acababa el proyecto la especificación estaba obsoleta.

Mientras se intentaban resolver estos problemas y se realizaban avances en programación y diseño estructurado, apareció un movimiento gradual hacia las especificaciones funcionales:

- Gráficas: compuestas por una variedad de diagramas.
- Particionadas: de forma que se puedan leer partes independientes de la especificación.
- Evitando la redundancia: dentro de lo posible se intenta minimizar la redundancia en la especificación de requisitos.

Este enfoque se conoce como análisis estructurado o como análisis descendente (top-down), se utiliza principalmente en organizaciones de desarrollo de sistemas de gestión. Desde entonces, se han contemplado diversos cambios:

- Dar menor importancia a la construcción de modelos físicos y lógicos.

- Diferenciar más los modelos físicos y los lógicos, conceptos algo confusos dentro del análisis estructurado clásico.
- Ampliar las técnicas para poder modelar sistemas de tiempo real. Para ello principalmente se amplía la notación de la técnica de los diagramas de flujo de datos (DFD) y se añade la técnica de los diagramas de transición de estados (DTE).
- Enfocarse en el modelo de datos.
- Estudio de los eventos: el proceso de análisis estructurado cambia añadiendo la lista de eventos.

Desarrollo Orientada a Objetos

El paradigma orientado a objetos a diferencia del enfoque estructurado, trata los procesos y los datos de forma conjunta. La orientación a objetos empieza con los lenguajes de programación orientados a objetos (LOO), tales como SIMULA o Smalltalk. En estos lenguajes se daba énfasis a la abstracción de datos y los problemas del mundo real se representaban como un conjunto de objetos de datos para los que se adjuntaba un conjunto de operaciones. Así se sentó la base para establecer una mayor importancia a la abstracción, a la ocultación de la información y a la modularidad.

Características principales de las metodologías

Los entornos de desarrollo incluyen un conjunto de componentes que condicionan la construcción de software. Cualquier cambio que se realice dentro del entorno de desarrollo puede tener un efecto inmediato sobre la productividad del personal de desarrollo, pero la productividad en sí no basta y debe estar asociada a la calidad de los productos finales. La metodología de desarrollo es el corazón de este entorno e influye directamente en estos dos factores. Por eso las características deseables que una metodología debería tener son:

- Existencia de reglas predefinidas: La metodología debería de definir unas reglas que definan sus fases.
- Cobertura total del ciclo de desarrollo: La metodología debería de indicar los pasos a realizar desde el planteamiento hasta su mantenimiento.
- Verificaciones intermedias: La metodología debería de contemplar las verificaciones de los productos intermedios.
- Planificación y control: La metodología debería proporcionar una forma de desarrollar software de manera planificada y organizada.
- Comunicación efectiva: La metodología debería de proporcionar un medio de comunicación efectivo.
- Utilización sobre un abanico amplio de proyectos. La metodología debería de ser flexible para poder ser aplicada a un amplio abanico de proyectos.

- Fácil formación: La metodología debería ser utilizada por todo el personal de la organización.
- Herramientas CASE. La metodología debería estar soportada por herramientas automatizadas que mejoren la productividad del equipo de desarrollo.
- La metodología debe contener actividades que mejoren el proceso de desarrollo.
- Soporte al mantenimiento: La metodología debería tener en cuenta las posibles modificaciones de los sistemas ya existentes.
- Soporte de la reutilización de software: La metodología debería proporcionar mecanismos para la reutilización de componentes software.

Clasificación de las metodologías

Para realizar una clasificación de las metodologías debemos de atender a tres puntos de vista, el enfoque, el tipo de sistema y la formalidad. Así de forma esquemática la tabla muestra como dependiendo del enfoque tenemos un tipo de sistema y una formalidad diferente.

ENFOQUE	TIPO DE SISTEMA	FORMALIDAD
Estructuradas <ul style="list-style-type: none"> - Orientadas a procesos - Orientadas a Datos <ul style="list-style-type: none"> - Jerárquicos - No jerárquicos - Mixtas 	Gestión	No formal
Orientadas a objetos	Tiempo real	Formal

Metodologías Estructuradas

Las metodologías estructuradas proponen la creación de modelos del sistema que representan los procesos, los flujos y la estructura de los datos de una manera descendente (top-down). Se pasa de una visión más general del problema hasta llegar a un nivel de abstracción más sencillo. Dentro de las metodologías estructuradas podemos encontrar los siguientes tipos:

- Orientadas a procesos
- Orientadas a datos
 - Orientadas a estructuras de datos jerárquicas
 - Orientadas a estructuras de datos no jerárquicas

- Mixtas: Engloban las dos modalidades orientadas a procesos y a datos.

Veamos un poco más de detalle algunas de ellas.

Metodologías orientadas a procesos

La ingeniería del software está fundada sobre el modelo básico de entrada/proceso/salida de un sistema. Los datos se introducen en el sistema y el sistema responde transformándolos para obtener salidas. Estas metodologías orientadas a procesos se enfocan fundamentalmente en la parte del proceso. Se apoyan en técnicas gráficas dando lugar a un nuevo concepto que es la especificación estructurada. Esta especificación se compone de:

- Diagramas de Flujo de Datos (DFD). Estos diagramas representan los procesos que debe de llevar a cabo un sistema.
- Diccionario de Datos (DD). Es el conjunto de definición de todos los datos que aparecen en un DFD.
- Especificaciones de proceso. Describe con más detalle lo que ocurre dentro de un proceso.

Vamos a describir los pasos de las principales metodologías orientadas a procesos.

Los pasos de la metodología **De Marco** son los siguientes:

1. Estudio del entorno físico actual. Se comienza haciendo un modelo del sistema actual y el sistema resultante será verificable por el usuario realizando un conjunto de DFD físicos.
2. Derivación del correspondiente modelo lógico actual. Se obtiene un modelo derivado del anterior pero sin connotaciones físicas (por ejemplo, los lugares de la empresa donde se realiza el proceso).
3. Derivación del nuevo modelo lógico. En esta etapa se toman en consideraciones las nuevas necesidades de los usuarios, estableciendo un modelo que describe aquello que hacer, pero no cómo. El resultado es una especificación estructurada formada por los DFD, el diccionario de datos y las especificaciones del sistema.
4. Crear un conjunto de modelos físicos alternativos. A partir del modelo lógico se establecen diferentes alternativas de las que se escogerá posteriormente la más conveniente.
5. Valorar cada opción. Se estudian los costes y los beneficios de los modelos físicos.
6. Seleccionar una opción. Se selecciona el modelo físico a partir de los datos derivados del paso anterior.
7. Empaquetar la especificación. Recopilar toda la documentación en un documento de especificación.

La metodología de análisis y diseño estructurado de **Yourdon/Constantine** se divide en los siguientes pasos:

1. Realizar el DFD del sistema.
2. Realizar el diagrama de estructuras, obteniéndolo a partir de los DFD, mediante dos técnicas: el análisis de transformación y el análisis de transacción.

3. Evaluación del diseño, midiendo la calidad de la estructura resultante mediante el acoplamiento y la cohesión.
4. Preparación del diseño para la implantación, es decir, dividirlo en unidades físicas de implantación denominadas cuadernos de carga.

Metodologías orientadas a datos jerárquicos

Dentro del modelo básico entrada/proceso/salida de un sistema, estas metodologías se centran más en las entradas y salidas. En este enfoque debemos de destacar que:

- La estructura de control del programa debe de ser jerárquica y debe de derivar de la estructura de datos del programa.
- El proceso de diseño consiste en definir primero las estructuras de los datos de entrada y salida, mezclarlas todas en una estructura jerárquica del programa y después ordenar detalladamente la lógica para que se ajuste a la estructura.
- El diseño lógico debe preceder y estar separado del diseño físico.

Metodologías orientadas a datos no jerárquicos

Las metodologías basadas en la información se centran en la creencia de que los datos constituyen el corazón del sistema de información. Los procesos también se estudian en este tipo de metodologías, pero vienen derivados de una definición inicial de los datos. Este tipo de metodología se divide en cuatro etapas con los siguientes objetivos:

- Planificación: construir una arquitectura de la información y una estrategia que soporte los objetivos de la organización.
- Análisis: comprender las áreas de negocio y determinar los requisitos del sistema.
- Diseño: establecer el comportamiento del sistema deseado por el usuario y que sea alcanzable por la tecnología.
- Construcción: construir sistemas que cumplan los tres niveles anteriores.

Metodologías Orientadas a Objetos

Con las metodologías orientadas a objeto se produce un cambio “filosófico” entre las metodologías clásicas de análisis y las de diseño estructurado. En las metodologías orientadas a objetos tiene más importancia el aspecto de modelado del sistema, examinando el dominio del problema como un conjunto de objetos que interactúan entre sí. Estas metodologías eliminan las fronteras entre fases debido a la naturaleza iterativa del desarrollo orientado al objeto. Con esta metodología aparece una nueva forma de concebir los lenguajes de programación y su uso al incorporarse bibliotecas de clases y otros componentes reutilizables. Además, hay un alto grado de iteración y

solapamiento, lo que lleva a una forma de trabajo muy dinámica. Las ventajas de utilización de este tipo de desarrollo es que es interactivo e incrementación, que aporta facilidad para dividir el sistema en varios subsistemas y además se fomenta la reutilización de componentes.

Principales metodologías de desarrollo

Metodología MERISE

Las bases de la metodología MERISE comenzaron a desarrollarse en 1972 en el seno de un equipo universitario de ingenieros. El proyecto para crear esta metodología partió de las necesidades para cubrir las necesidades tanto de la administración francesa como de las empresas. Las mayores aportaciones de la metodología son:

- Un ciclo de vida más largo, con la inclusión de una etapa de planificación, previa al desarrollo, denominada esquema director.
- Introducción de dos ciclos complementarios: ciclo de abstracción y ciclo de decisión. El ciclo de abstracción se basa en la percepción de tres niveles de abstracción: conceptual, organizativo y físico u operativo. Se definen dos modelos para cada nivel: un modelo de datos y un modelo de tratamiento.

NIVELES	DATOS	TRATAMIENTO
Conceptual	Modelo conceptual de Datos	Modelo conceptual de tratamientos
Organizativo	Modelo lógico de datos	Modelo organizativo de tratamientos
Físico	Modelo físico de datos	Modelo operativo de tratamientos

Esta metodología se divide en cuatro fases:

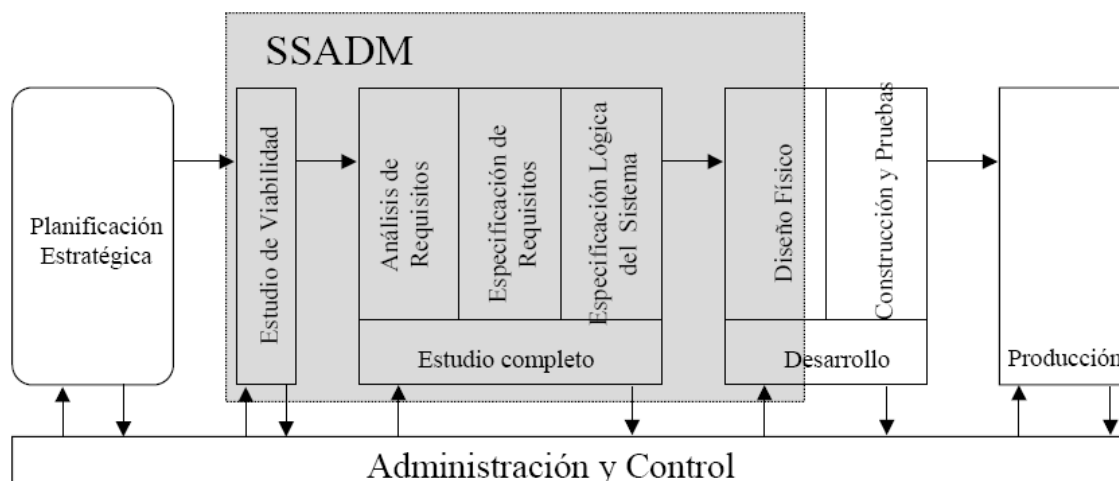
1. Estudio preliminar. Se analiza la situación existente para proponer una solución global.
2. Estudio detallado. El objetivo de esta fase es definir, a nivel funcional, la solución que hay que implementar.
3. Implementación. El objetivo es realizar los programas y se dividen en dos partes: Estudio técnico, donde se distribuyen los datos en ficheros y la producción de programas que permite codificar y verificar los programas.
4. Realización y puesta en marcha. Se efectúa la implementación de los medios técnicos y organizativos, así como la recepción definitiva por parte del usuario.

Metodología SSADM

Esta metodología SSADM (Structured Systems Analysis and Design Method) surge en los años ochenta cuando el gobierno británico tiene la necesidad de crear una metodología de desarrollo en un intento de estandarizar los diferentes proyectos gubernamentales. Desde su aparición ha ido evolucionando para ir corrigiendo las debilidades del método. Los aspectos claves de esta metodología son:

- Énfasis en los usuarios: sus requisitos y participación.
- Definición del proceso de producción: qué hacer, cuándo y cómo.
- Tres puntos de vista: datos, eventos y procesos.
- Máxima flexibilidad en herramientas y técnicas de implementación.

SSADM proporciona un conjunto de procedimientos para llevar a cabo el análisis y el diseño, pero no cubre aspectos como la planificación estratégica ni entra en la construcción de código, tal y como se muestra en el siguiente esquema que muestra la cobertura de SSADM al ciclo de vida.



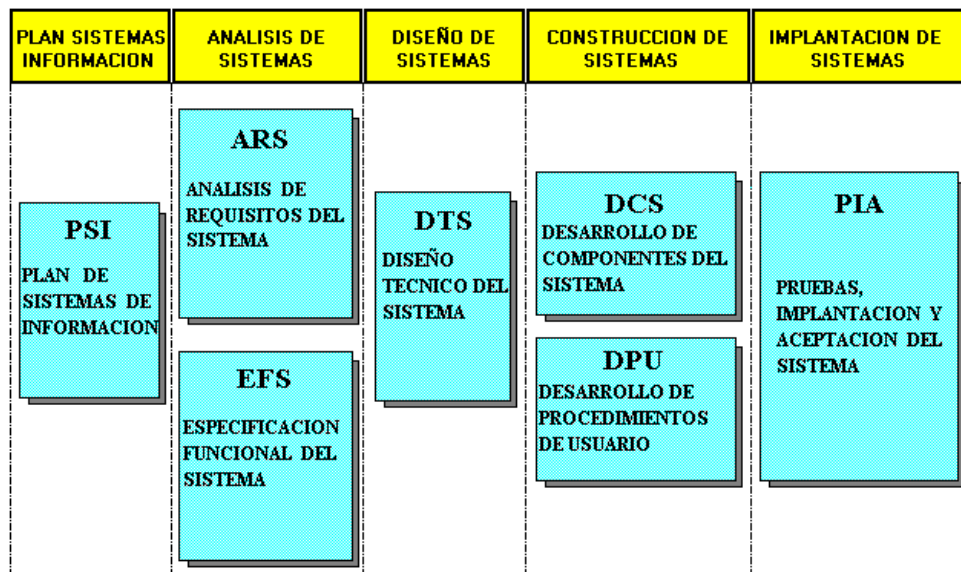
Metodología Métrica

El consejo superior de Informática fue creado en 1983 con el objetivo de elaborar y desarrollar las directrices en materia de tecnologías de la información y comunicaciones para la administración española. En 1989 se acuerda la realización del proyecto Métrica, cuyo objetivo fue la creación de un marco metodológico común para la planificación y el desarrollo de sistemas de información de la administración pública española.

Métrica versión 2 aparece en 1993 y se encuentra estructurada mediante una sucesión de fases, módulos, actividades y tareas que hay que seguir para el desarrollo de sistemas, e indica los

productos que se obtienen en cada una de las tareas. Algunos de los productos finales y otros productos intermedios que servirán de base para la realización de tareas posteriores. La metodología está dividida en las siguientes fases:

- FASE 0: Plan de sistemas de información
- FASE 1: Análisis de sistemas
- FASE 2: Diseño de sistemas
- FASE 3: Construcción de sistemas
- FASE 4: Implantación de sistemas



Actualmente la metodología Métrica se encuentra en su versión 3, la cual analizaremos en detalle en la última unidad de la asignatura.

Referencias

García., S.; Morales, E. Desarrollo de Aplicaciones Informáticas. Análisis y Diseño detallado de aplicaciones informáticas de gestión. Paraninfo. 2003

Piattini, M.; Calvo, J.; Cervera J., Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión, Ed. Ra-Ma, 2007.