



Universidad Autónoma del Estado de Hidalgo

Instituto de Ciencias Básicas e Ingeniería

Licenciatura en Sistemas Computacionales

**METODOLOGÍAS PARA EL DESARROLLO DE SISTEMAS.
CASOS DE ESTUDIO: MÉTRICA II Y MERISE**

Monografía

Que para obtener el Título de Licenciado en Sistemas
Computacionales

P R E S E N T A

Alejandro Pérez García

Director del trabajo

Lic. Norma Laura Salazar Viveros.

Pachuca de Soto Hgo., Septiembre de 2006
México.

Agradecimientos

@ mis Padres:

*Por darme el mejor apoyo, la motivación, e inculcar en mí
los valores y el ejemplo que fueron fundamentales
para poder hacer de mí la persona que soy.*

@ mis Hermanos:

*Por su comprensión, cariño y sabios
consejos*

@ Ely:

*Por estar allí cuando siempre la necesite, darme el
valor para seguir adelante y por el apoyo
incondicional que siempre me dio*

@ mis maestros y en especial a mi asesor:

*Por darme el conocimiento que es necesario para
poder desenvolverme como profesional y tener la
paciencia suficiente para poder adquirirlo.*

Por todo eso y mucho mas, Muchas

GRACIAS

Alejandro

Contenido

OBJETIVOS	III
JUSTIFICACIÓN	V
INTRODUCCIÓN	VIII

CAPÍTULO 1.- INGENIERÍA DE SOFTWARE

1.1.- Concepto de Ciclo de Vida	2
1.2.- Ciclos de vida Estructurados	3
1.2.1.- Análisis estructurado	3
1.2.2.- Ciclo de Vida en Cascada	3
1.2.3 Ciclo de vida en Espiral	4
1.2.4.- Ciclo de vida Incremental	6
1.3.- Ciclos de vida Orientados a Objetos	6
1.3.1.- Análisis Orientado a Objetos (AOO)	6
1.3.2 Rational Unified Process (RUP)	7
1.3.3. - Dynamic Systems Development Method (DSDM)	9
1.3.4.- Desarrollo de BOOCH	11
1.3.5.- XP (Programación Extrema)	13
1.3.5.1.- Qué es XP	13
1.3.5.2.- Antecedentes	13

CAPÍTULO 2.- METODOLOGÍA MÉTRICA

2.1.- Antecedentes	17
2.1.1.- Objetivos	17
2.2.- Estructura de la Metodología MÉTRICA II	19
2.3.- FASE 0: Plan de Sistemas de Información	23
2.4.- FASE 1: Análisis de Sistemas	25
2.5.- FASE 2: Diseño de Sistemas	27
2.6.- FASE 3: Construcción de Sistemas	28

2.7.- FASE 4: Implantación de Sistemas	29
2.8.- Organización de la Metodología MÉTRICA II	30
2.9.- Gestión de Proyectos	32
2.10.- Implantación de la Metodología MÉTRICA II	34

CAPÍTULO 3.- METODOLOGÍA MERISE

3.1.- Antecedentes	38
3.1.1.- Objetivos	38
3.2.- Estructura de la Metodología MERISE	39
3.3.- Estudio Preliminar	39
3.4.- Estudio Detallado	40
3.4.1.- Análisis	41
4.4.1.1.- Identificar Necesidades	41
4.4.1.2.- Estudio de Viabilidad	42
4.4.1.3.- Estudio Económico y Técnico	43
4.4.1.4.- Modelado de la Arquitectura del Sistema	44
4.4.1.5.- Especificaciones del Sistema	44
3.4.2.- Diseño	45
4.4.2.1.- Diseño de la Salida	46
4.4.2.2.- Diseño de Archivos	47
4.4.2.3.- Diseño de Interacciones con la B. de D	47
3.5.- Implementación	48
3.6.- Puesta en Marcha	50
3.6.1.- Transferencia del producto	50
3.6.2.- Evolución	51
CONCLUSIONES	54
REFERENCIAS	57

OBJETIVOS

Objetivo General:

Realizar un estudio exhaustivo de las características y utilización de las metodologías MERISE y MÉTRICA II, para el desarrollo de sistemas.

Objetivos Específicos:

- ✦ Realizar una investigación sobre el análisis estructurado y el orientado a objetos, para el desarrollo de sistemas.
- ✦ Realizar un análisis comparativo sobre MÉTRICA II Y MERISE.
- ✦ Conocer más sobre los campos de utilización de estas herramientas.
- ✦ Hacer una fuente de investigación para los interesados en el tema.

JUSTIFICACIÓN

Justificación

Debido al gran desarrollo tecnológico que la civilización ha tenido desde la aparición de las primeras computadoras, la mayoría de las empresas u organizaciones se han visto en la necesidad de estar en constante cambio, ya sea para mejorar o para mantener un buen nivel competitivo en sus respectivos ramos.

Pero no solo es suficiente contar con un buen equipo de cómputo o con el paquete más sofisticado, sino que también es necesario un buen sistema que sea capaz de manejar la mayor cantidad de las operaciones de una organización.

Para el desarrollo de sistemas existen varias metodologías, pero desafortunadamente no todas se conocen en nuestro País. Para poder hacer alguna especialidad referente a estas es obligatorio salir al extranjero según sea el caso de la metodología. Ya que en México solo se conocen los clásicos ciclos de vida de desarrollo de software, como lo son *ESPIRAL*, *RAD*, *RUP*, por mencionar algunos.

La utilización que tienen estas metodologías, es que nacen para resolver los problemas de los desarrolladores de sistemas, que no cuentan con una metodología de desarrollo, que les proporcionen la documentación necesaria para llevar una buena interacción con los usuarios finales.

Hoy en día las organizaciones más grandes e importantes del mundo contratan personal con certificación de alguna de las ya tan mencionadas metodologías.

Muy pronto en México, las empresas se verán en la necesidad de contratar personas con el conocimiento o certificación de alguna de estas herramientas.

Por todo lo antes mencionado surge el interés para indagar un poco más a fondo acerca de algunas metodologías, si no todas al menos las más conocidas desarrolladas y existentes en otros países. Lo que son MÉTRICA II en España y MERISE en Francia.

INTRODUCCIÓN

INTRODUCCIÓN

Las Metodologías de Desarrollo de Sistemas MÉTRICA II (España) y MERISE (Francia), son de las más conocidas y se han implantado con el fin de solucionar la problemática que resulta de la escasa documentación de los sistemas y de la falta de comunicación con los usuarios durante el proceso de desarrollo, lo que genera productos que no responden totalmente a las necesidades de los usuarios. En este sentido, se definen un conjunto de métodos, procedimientos, técnicas y herramientas que facilitan la construcción de los Sistemas de Información.

La forma de trabajar que es utilizada en estas metodologías se estructura en fases.

MERISE.- Es una metodología diseñada para la confección de Planes Informáticos que se subdivide en fases (Estudio Previo, Detallado, Escenarios, Desarrollo del Plan y Seguimiento). Utilizado por la Administración Pública francesa, se ha ido imponiendo en la empresa privada desde hace más de 30 años.

MÉTRICA II.- Ha sido diseñada por un grupo de trabajo constituido al efecto por personal procedente de distintos Ministerios y Organismos de la Administración Española con la asistencia externa de la empresa Coopers & Lybrand.

Esta metodología es una guía formal, para la Planificación, Análisis, Diseño y Construcción e Implantación de Sistemas de Información

empleando conceptos y técnicas de Ingeniería de Sistemas información, y que es una herramienta de dominio público.

El principal objetivo de las metodologías MÉTRICA II y MERISE, es el poder crear un entorno que permita a un equipo de trabajo construir Sistemas, que sean capaces de:

- ✦ Dar solución a los objetivos considerados prioritarios en la Administración.
- ✦ Se desarrollen cuando el usuario los necesite y de acuerdo con los presupuestos y duración estimados.
- ✦ De calidad que se mantengan fácilmente para soportar los cambios futuros de la organización.

Todo ello utilizando un vocabulario común y un conjunto completo de tareas y productos finales que ayuden a construir con éxito Sistemas de Información.

La principal característica de estas metodologías es su flexibilidad ya que se adaptan a gran variedad de sistemas y ciclos de vida. Su carácter público y abierto ha permitido su utilización en departamentos informáticos de las Administraciones Públicas y de empresas privadas.

Entendiendo por esto que metodología es: la forma en que deben hacerse las cosas.

CAPÍTULO 1

INGENIERÍA DE SOFTWARE

CAPÍTULO 1.- INGENIERÍA DE SOFTWARE

Ingeniería del Software es la INGENIERÍA que trata de construir software de ALTA CALIDAD a BAJO COSTO

1.1.- Concepto de ciclo de vida

El ciclo de vida indica que es lo que hay que obtener a lo largo del desarrollo de un proyecto pero no cómo hacerlo.

Una aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software.

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 de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso.

Pero traducido a palabras un poco más comprensibles podríamos decir que el ciclo de vida es, un conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace hasta que se retira o se reemplaza.

1.2.- Ciclos de vida Estructurados

1.2.1.- Análisis estructurado

Son técnicas para la construcción de modelos, estos se crean reflejando el flujo y contenido de la información. Se estudia el sistema en partes y se ve en forma, de que este funcione así sabremos lo que se debe crear o construir.

Los ciclos de vida estructurados podríamos decir que son aquellos que se realizan por partes.

Entre los ciclos de vida estructurados más conocidos y utilizados tenemos los siguientes:

- ✦ Ciclo de vida en Cascada.
- ✦ Ciclo de vida en Espiral.
- ✦ Ciclo de vida Incremental.

1.2.2.- Ciclo de vida en Cascada

También llamado ciclo de vida clásico, fue el Primer ciclo empleado en 1970.

Trabaja con ejecución secuencial de una serie de fases como se muestra en la figura 1.1. Cada fase genera documentación para la siguiente fase.

Una característica importante de este ciclo es, que, para el buen funcionamiento de los sistemas se deben realizar todas las fases y por lo mismo es muy tardado desarrollarlo.

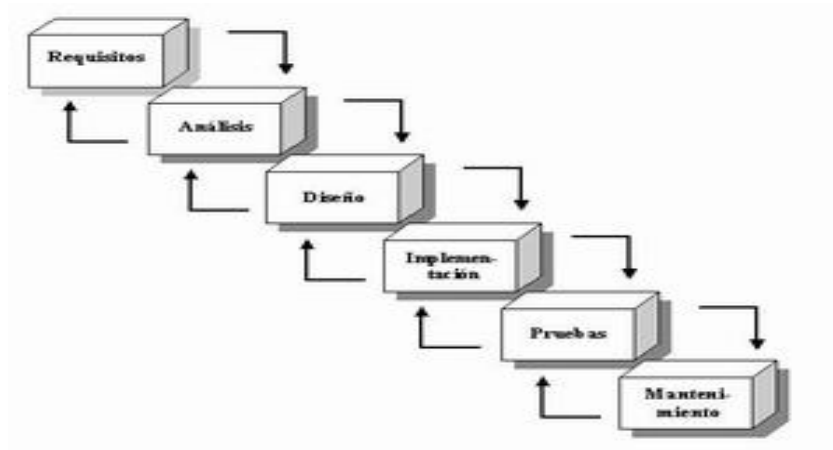


Fig. 1.1 El ciclo de vida en cascada

1.2.3 Ciclo de vida en Espiral.

El ciclo de vida en espiral fue definido por Boehm en 1986 trabaja en base a cuatro actividades que son la planificación, análisis de riesgo, ingeniería y evaluación del cliente. Permite eliminar errores o alternativas no deseadas.

Lo que hace diferente este ciclo de otros es:

- ✦ Que existe un reconocimiento explícito de las diferentes alternativas para alcanzar los objetivos de un proyecto.

- ✦ La identificación de riesgos asociados con cada una de las alternativas.
- ✦ La división de los proyectos en ciclos.
- ✦ El modelo se adapta a cualquier tipo de actividad.
- ✦ Se evalúan las alternativas respecto a los objetivos y las restricciones.
- ✦ Cada ciclo se completa con una revisión que incluye todo el ciclo anterior y el plan para el siguiente.

En la figura 1.2 se muestra el funcionamiento del ciclo en espiral.

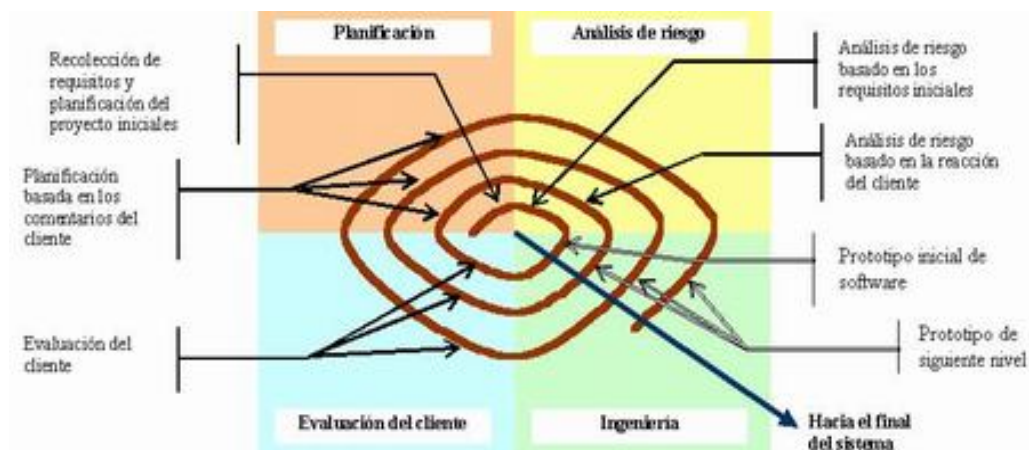


Fig.1.2 El ciclo de vida en Espiral

1.2.4.- Ciclo de vida Incremental.

Con este ciclo de vida se pretende dividir el trabajo, como se muestra en la figura 1.3, es muy recomendable para los proyectos grandes, aparte que en este, los usuarios se involucran más con su sistema.

Para los sistemas tradicionales que funcionan como si fueran una misma cosa no es recomendable este ciclo, aparte de que es costoso.

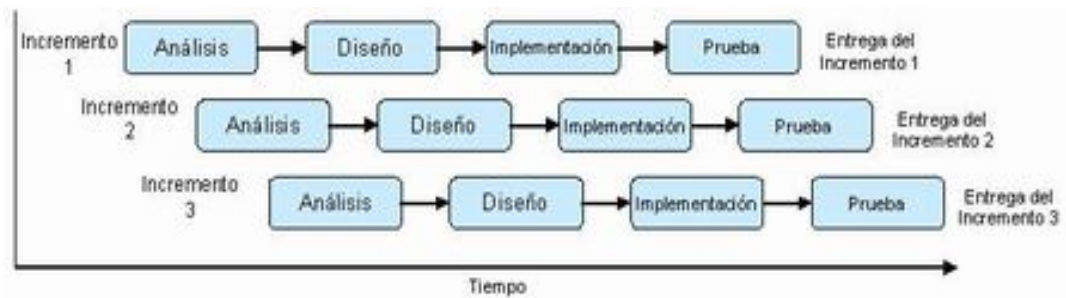


Fig.1.3 El ciclo de vida incremental

1.3.- Ciclos de vida Orientados a Objetos

1.3.1.- Análisis Orientado a Objetos (AOO)

El análisis orientado a objetos surge por la necesidad y dificultad que presentaban los ciclos de vida estructurados en cuanto a lo largo de cada uno, la dificultad que había para poder hacer correcciones en diseño y la falta de portabilidad del código.

Por ello es que nace este tipo de análisis, porque puede atacar estos problemas en cualquier momento del ciclo de vida del sistema, con la ayuda de varios métodos y entre los más conocidos encontramos los siguientes:

- ✦ Rational Unified Process (RUP)
- ✦ Dynamic Systems Development Method (DSDM)
- ✦ BOOCH

1.3.2 Rational Unified Process (RUP)

El proceso de ciclo de vida de RUP se divide en cuatro fases bien conocidas llamadas Incepción, Elaboración, Construcción y Transición. Esas fases se dividen en iteraciones, como se muestra en la figura 1.4, cada una de las cuales produce una pieza de software demostrable. La duración de cada iteración puede extenderse desde dos semanas hasta seis meses. Las fases son:

1. **Inicio.** Se especifican los objetivos del ciclo de vida del proyecto y las necesidades de cada participante.
2. **Elaboración.** Se analiza el dominio del problema y se define el plan del proyecto. RUP presupone que la fase de elaboración brinda una arquitectura suficientemente sólida junto con requerimientos y planes bastante estables.

3. **Construcción.** Se desarrollan, integran y verifican todos los componentes y rasgos de la aplicación. RUP considera que esta fase es un proceso de manufactura.
4. **Transición.** Comienza cuando el producto está suficientemente maduro para ser entregado.

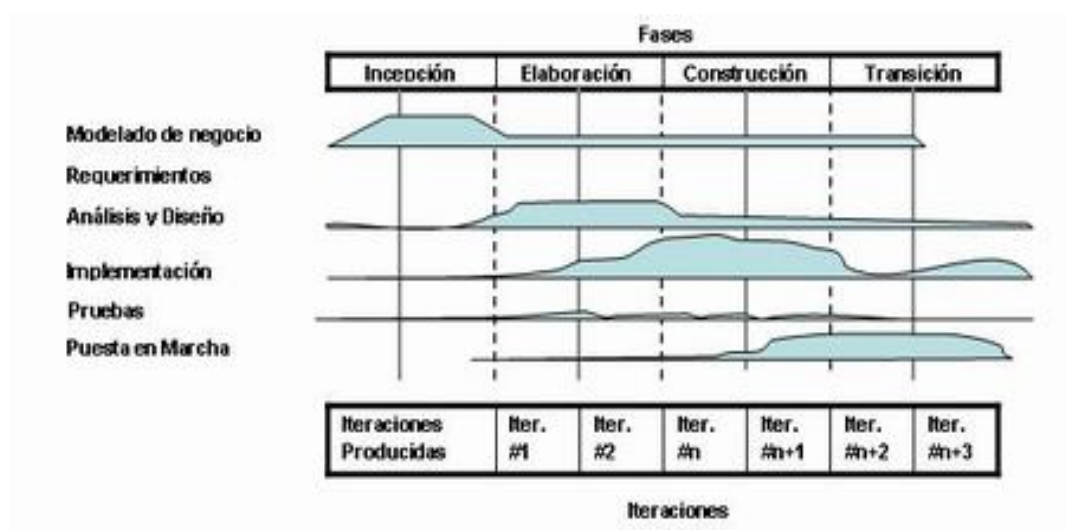


Fig. 1.4 Ciclo de vida RUP

RUP define algunas prácticas comunes:

1. **Desarrollo iterativo de software.** Las iteraciones deben ser breves y proceder por incrementos pequeños. Esto permite identificar riesgos y problemas tempranamente y reaccionar frente a ellos en consecuencia.

2. **Administración de requerimientos.** Identifica requerimientos cambiantes y postula una estrategia disciplinada para administrarlos.
3. **Uso de arquitecturas basadas en componentes.** La reutilización de componentes permite asimismo ahorros sustanciales en tiempo, recursos y esfuerzo.
4. **Modelado visual del software.** Se deben construir modelos visuales, porque los sistemas complejos no podrían comprenderse de otra manera. Utilizando una herramienta como UML, la arquitectura y el diseño se pueden especificar sin ambigüedad y comunicar a todas las partes involucradas.
5. **Prueba de calidad del software.** RUP pone bastante énfasis en la calidad del producto entregado.
6. **Control de cambios.** La madurez del software se puede medir por la frecuencia y tipos de cambios realizados.

1.3.3.- Dynamic Systems Development Method (DSDM)

DSDM consiste en cinco fases:

1. Estudio de viabilidad.
2. Estudio del negocio.
3. Iteración del modelo funcional.

4. Iteración de diseño y versión.
5. Implementación.

Las últimas tres fases son iterativas e incrementales. De acuerdo con la iniciativa de mantener el tiempo constante, las iteraciones de DSDM son cajas de tiempo. La iteración acaba cuando el tiempo se consume. Se supone que al cabo de la iteración los resultados están garantizados. Una caja de tiempo puede durar de unos pocos días a unas pocas semanas.

1. **Estudio de factibilidad.** Se evalúa el uso de DSDM o de otra metodología conforme al tipo de proyecto, variables organizacionales y de personal. Si se opta por DSDM, se analizan las posibilidades técnicas y los riesgos.
2. **Estudio del negocio.** Se analizan las características del negocio y la tecnología.
3. **Iteración del modelo funcional.** En cada iteración se planea el contenido y la estrategia, se realiza la iteración y se analizan los resultados.
4. **Iteración de diseño y construcción.** Aquí es donde se construye la mayor parte del sistema.

5. **Despliegue.** El sistema se transfiere del ambiente de desarrollo al de producción. Se entrena a los usuarios, que ponen las manos en el sistema.

La configuración del ciclo de vida de DSDM se representa con un diagrama característico, figura 1.5:

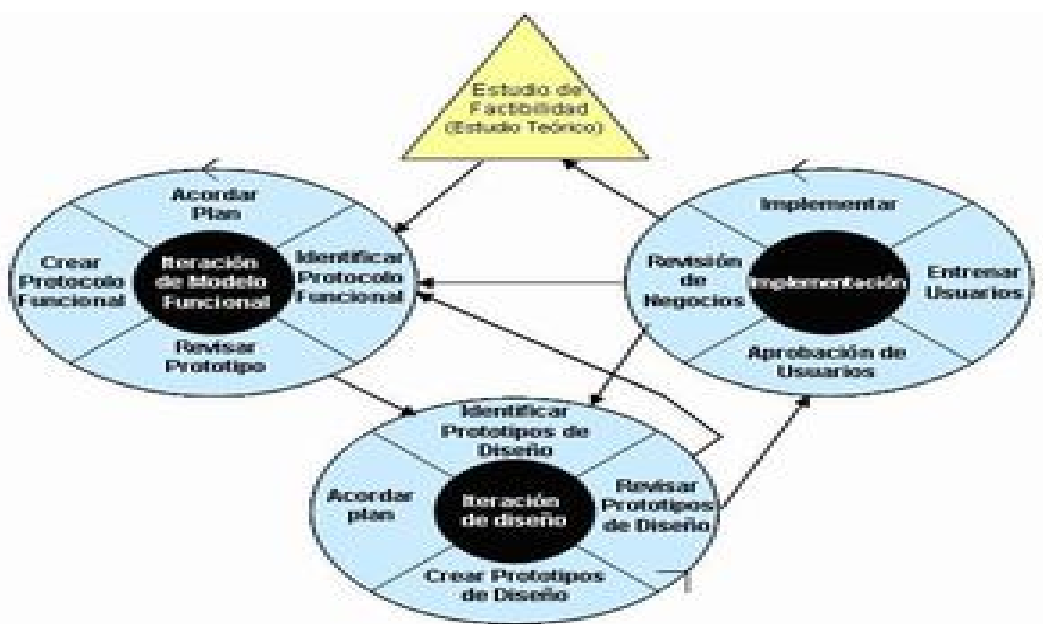


Fig. 1.5 El ciclo DSDM

1.3.4.- Desarrollo de BOOCH

BOOCH estableció un ciclo de vida que se preocupa más por el diseño detallado y para describir las actividades a realizarse en la creación orientada a objetos, BOOCH utiliza una serie de diagramas para hacerlo.

Las etapas de este ciclo son:

1.- Análisis de requerimiento.- En esta etapa se identifican las necesidades de los usuarios, para poder identificar las principales funciones del sistema.

2.- Análisis de dominio.- Aquí definimos el modelado Orientado a Objetos del sistema con una serie de tareas como lo son:

- ✦ Identificar operaciones
- ✦ Encontrar atributos
- ✦ Definir clases
- ✦ Definir relaciones.
- ✦ Establecer Herencias
- ✦ Validar el diseño

3.- Diseño.- Es el proceso donde se elige una forma de implementación, de modo que sea eficiente para realizar las actividades antes mencionadas.

1.3.5.- XP (Programación Extrema)

1.3.5.1.- Qué es XP

La XP (*Programación Extrema*) es un movimiento relativamente nuevo en el mundo del desarrollo: sus ideas centrales son bastante revolucionarias. + Esta diseñada para entornos dinámicos

- + Pensada para equipos pequeños (hasta 10 programadores)
- + Orientada fuertemente hacia la codificación
- + Énfasis en la comunicación informal, verbal

1.3.5.2.- Antecedentes

Las raíces de la XP yacen en la comunidad de Smalltalk, y en particular la colaboración cercana de Kent Beck y Ward Cunningham a finales de los 80s. Ambos refinaron sus prácticas en numerosos proyectos a principios de los 90s, extendiendo sus ideas de un desarrollo de software adaptable y orientado a la gente.

El paso crucial de la práctica informal a una metodología ocurrió en la primavera de 1996. A Kent se le pidió revisar el progreso del proyecto de nómina C3 para Chrysler. El proyecto estaba siendo llevado en Smalltalk por una compañía contratista, y estaba en problemas. Debido a la baja calidad de la base del código, Kent recomendó tirar la base del código en su totalidad y empezar desde el principio. El proyecto entonces reinició bajo su dirección y subsecuentemente se

volvió el buque insignia temprano y el campo de entrenamiento de la XP.

La primera fase del C3 fue muy exitosa y comenzó a principios de 1997. El proyecto continuó desde entonces y después se encontró con dificultades, lo que resultó en la cancelación del desarrollo en 1999. (Lo cual prueba, si ninguna otra cosa, que la XP no es garantía de éxito.)

La XP empieza con cuatro valores: Comunicación, Retroalimentación, Simplicidad y Coraje. Construye sobre ellos una docena de prácticas que los proyectos XP deben seguir. Muchas de estas prácticas son técnicas antiguas, tratadas y probadas, aunque a menudo olvidadas por muchos, incluyendo la mayoría de los procesos planeados. Además de resucitar estas técnicas, la XP las teje en un todo sinérgico donde cada una refuerza a las demás.

Una de las más llamativas, así como inicialmente atractiva para mí, es su fuerte énfasis en las pruebas. Mientras todos los procesos mencionan la comprobación, la mayoría lo hace con muy poco énfasis. Sin embargo la XP pone la comprobación como el fundamento del desarrollo, con cada programador escribiendo pruebas cuando escriben su código de producción. Las pruebas se integran en el proceso de integración continua y construcción lo que rinde una plataforma altamente estable para el desarrollo futuro.

En esta plataforma XP construye un proceso de diseño evolutivo que se basa en refactorar un sistema simple en cada iteración. Todo el

diseño se centra en la iteración actual y no se hace nada anticipadamente para necesidades futuras. El resultado es un proceso de diseño disciplinado, lo que es más, combina la disciplina con la adaptabilidad de una manera que indiscutiblemente la hace la más desarrollada entre todas las metodologías adaptables.

CAPÍTULO 2

METODOLOGÍA MÉTRICA II

CAPÍTULO 2.- METODOLOGÍA MÉTRICA II

2.1.- Antecedentes

MÉTRICA II ha sido diseñada por un grupo de trabajo constituido al efecto por personal procedente de distintos Ministerios y Organismos de la Administración Española con la asistencia externa de la empresa Coopers & Lybrand.

Esta metodología es una guía formal, aunque flexible en su utilización, para la Planificación, Análisis, Diseño y Construcción e Implantación de Sistemas de Información empleando conceptos y técnicas de Ingeniería de Sistemas de Información y Tecnología de la Información.

El ámbito original de aplicación de MÉTRICA II ha sido la Administración General del Estado pero no hay ninguna razón que impida su utilización por las Administraciones Locales y Autónoma e incluso por compañías privadas de cualquier tipo.

2.1.1.- Objetivos

A medida que crece el volumen de información a manejar en la Administración Española, aumenta la necesidad de disponer de una Tecnología de la Información que soporte dinámicamente el funcionamiento normal de los distintos departamentos que la constituyen.

Dicho soporte ha de ser dinámico en el sentido de que debe adaptarse con facilidad a las condiciones, externas e internas, cambiantes de la Organización. Por otra parte, ha de ser eficaz y atenerse estrictamente a las necesidades del usuario. Para ello la comunicación entre las Unidades usuarias y los desarrolladores es un factor vital y determinante.

La problemática de los desarrolladores que no utilizan ninguna metodología de desarrollo, se puede resumir así:

Escasa o nula documentación de los sistemas, lo que dificulta las tareas de desarrollo, implantación y especialmente la de mantenimiento. Falta de comunicación con los usuarios, lo que genera productos no entregados a tiempo y que, además, no responden totalmente a las necesidades de los usuarios.

Se justifica, por tanto, la implantación de una Metodología de Desarrollo de Sistemas en la Administración, en la que se defina un conjunto de métodos, procedimientos, técnicas y herramientas que faciliten la construcción de Sistemas de Información, con el fin de:

- ✦ Satisfacer todas las necesidades de los departamentos usuarios implicados.
- ✦ Generar la documentación asociada, para facilitar su mantenimiento posterior.
- ✦ Mejorar la productividad de las organizaciones.

El principal objetivo de la metodología MÉTRICA II es crear un entorno que permita al equipo de trabajo construir sistemas, que:

- ✦ Den solución a los objetivos considerados prioritarios en la organización.
- ✦ Se desarrollen cuando el usuario los necesite y de acuerdo con los presupuestos y duración estimados.
- ✦ De calidad que se mantengan fácilmente para soportar los cambios futuros de la organización.
- ✦ Todo ello utilizando un vocabulario común y un conjunto completo de tareas y productos finales que ayuden a construir con éxito Sistemas de Información.

2.2.- Estructura de la metodología MÉTRICA II

MÉTRICA II ofrece un marco de trabajo en el que se define:

- ✦ Una estructura de proyecto que sirva de guía al equipo de trabajo e involucre a los usuarios en su desarrollo y en sus puntos decisivos.
- ✦ Un conjunto de productos finales a desarrollar.
- ✦ Un conjunto de técnicas para obtener los productos finales.

- ✦ Las diferentes responsabilidades y funciones de los miembros del equipo de proyecto y de los usuarios.

Con este fin, se describe en detalle la sucesión de pasos, estructurados en Fases, Módulos, Actividades y Tareas, que se han de seguir en el desarrollo de sistemas informáticos, así como los productos que se obtienen en cada uno de dichos pasos. Estos productos pueden ser, productos finales o bien productos intermedios que servirán para la realización de algún paso posterior. Por último se describe la estructura final de la documentación obtenida.

Las razones que han llevado a definir esta estructura de Fases y Módulos son las siguientes:

El término Fase conlleva la idea de secuencia, y presenta las características que a continuación se indican:

Establece un conjunto formal de Productos que deben ser entregados por el equipo de trabajo antes de que se inicie la siguiente Fase. De esta forma, se pueden dividir los proyectos en una serie de tareas preestablecidas, que facilitarán las labores de Planificación y Control de Proyectos.

El final de cada Fase requiere una aceptación formal de las conclusiones a las que se ha llegado al término de la misma.

El producto final obtenido en cada Fase es un documento que se utiliza para el inicio de la siguiente fase.

La división en Módulos obedece a razones de homogeneidad: Un módulo es un grupo de actividades y tareas que se realizan para producir un conjunto específico de productos finales.

MÉTRICA II está dividida en cinco Fases que se descomponen en siete Módulos. Los Módulos, a su vez, se descomponen en Actividades y éstas en Tareas como se muestra en la figura 2.1.

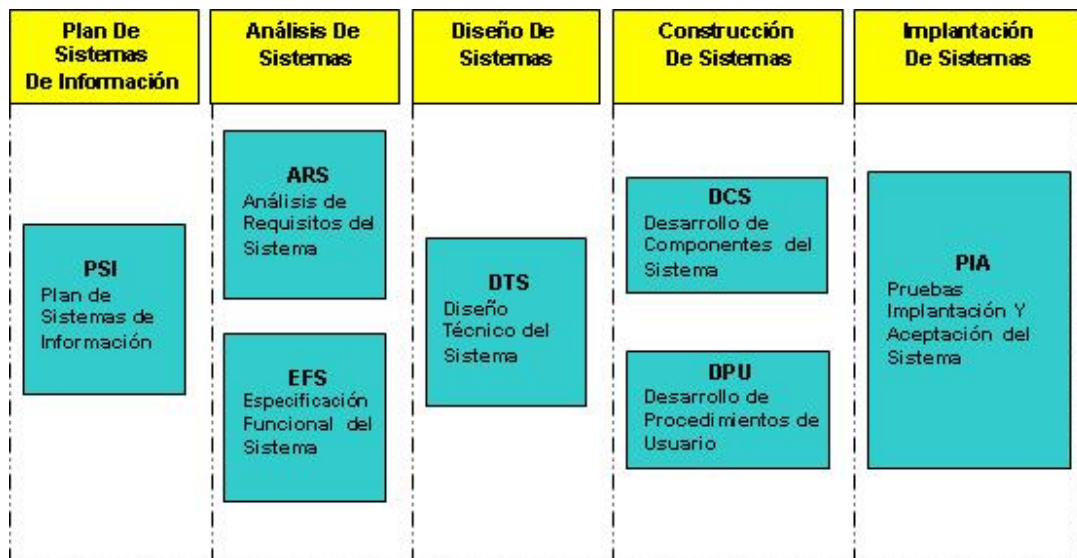


Fig. 2.1 Visión General de MÉTRICA II

Las fases en las que se divide MÉTRICA II son:

- ✦ **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**

La estructura de MÉTRICA II no está asociada al modelo de desarrollo de ciclo de vida en cascada. Ya que MÉTRICA II prescribe gran cantidad de retornos a nivel de actividades, módulos e incluso de fases como se verá más adelante.

MÉTRICA II está apoyada en una serie de técnicas que dan el soporte práctico necesario para el desarrollo óptimo de las Actividades definidas en ella, y permite el empleo de herramientas tecnológicas avanzadas (CASE, Lenguajes 4 Generación, etc.) que facilitan dicho desarrollo.

Es importante destacar que aún contemplando aspectos de Gestión de Proyectos, Gestión de Calidad y Gestión de Configuración, MÉTRICA II no pretende soportar todas las actividades relacionadas con estos conceptos de Ingeniería de Sistemas.

Sin embargo, aporta un nexo de unión con dichos conceptos, identificando el lugar donde conectan la metodología de desarrollo de sistemas y el resto de los aspectos asociados al desarrollo de cualquier Sistema de Información.

En cualquier caso permite poner los cimientos de lo que sería una construcción de sistemas con un enfoque de ingeniería.

A continuación se describen con más detalle cada una de las fases de la Metodología MÉTRICA II.

2.3.- FASE 0: Plan de Sistemas de Información

La realización de un Plan de Sistemas de Información dentro de cualquier Organización, tiene como finalidad asegurar la adecuación entre los objetivos estratégicos de la misma y la información necesaria para soportar los grandes objetivos. Esto hace que una metodología de planificación de sistemas abarque a toda la organización y exige tener en cuenta una serie de conceptos, en cuanto a planificación de estrategias que desbordan el marco específico de una Metodología de Desarrollo de sistemas.

En la figura 2.2 se muestra la representación del Plan de Sistemas de Información.

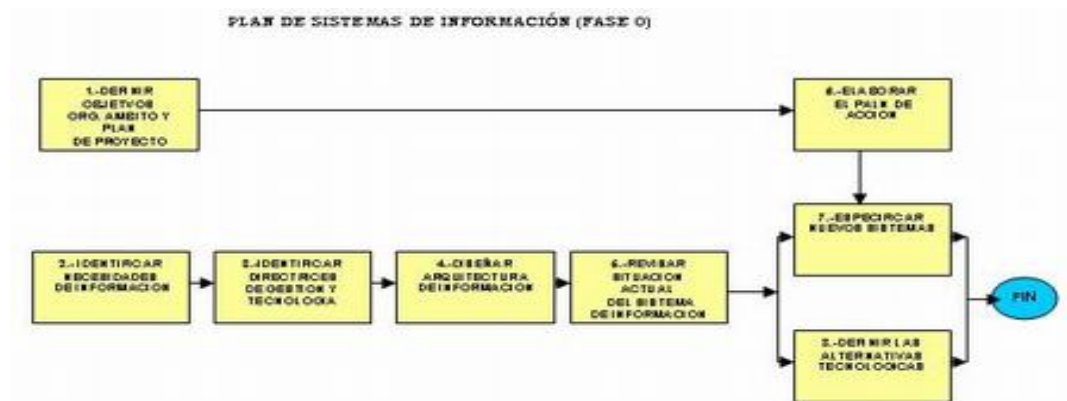


Fig. 2.2 Plan de Sistemas de Información

Conscientes de esta diferencia en cuanto al ámbito que se pretende cubrir con una Metodología de Planificación de Sistemas, es necesario, sin embargo, establecer una relación directa entre ambas metodologías, con el fin de que la información obtenida con una concepción estratégica sirva de entrada y punto de partida para la especificación de los sistemas concretos a desarrollar.

Para ello, se ha definido la Fase 0 de la Metodología MÉTRICA II, con los siguientes objetivos:

1.- Definir la información necesaria que se debe obtener con la realización de una Metodología de Planificación, en cuanto a objetivos estratégicos de la Organización y factores críticos de éxito para satisfacer estos objetivos.

2.- Definir la Arquitectura de la Información (procesos y datos) que satisfará los objetivos estratégicos de la Organización.

3.- Definir los nuevos sistemas a desarrollar que permitan implantar dicha Arquitectura. La información obtenida servirá de punto de partida para el desarrollo de cada uno de estos sistemas con MÉTRICA II.

Para el desarrollo de un sistema aislado, cuya necesidad no se derive de un Plan de Sistemas, no se utilizará la Fase 0 de la Metodología.

2.4.- FASE 1: Análisis de Sistemas

El propósito de esta Fase es:

En primer lugar, describir el alcance, los objetivos y los requisitos del Sistema. Basándose en todo esto, el equipo del proyecto puede examinar distintas alternativas que podrían solucionar el problema y recomendar una de ellas. Con la finalización del primer módulo de esta Fase, Análisis de requisitos del Sistema (*figura 2.3*), se obtendrá, como producto final, un documento donde se establecerá:

- ✦ El alcance del Proyecto.
- ✦ El Modelo Lógico Actual de Procesos y el Esquema Lógico Actual de Datos.
- ✦ Los requisitos de usuario.
- ✦ El análisis de alternativas, y la solución propuesta.



Fig. 2.3 Análisis de Requisitos del Sistema

En segundo lugar el objetivo de esta Fase es elaborar un conjunto de especificaciones formales que describan la funcionalidad del Sistema para su aprobación por parte del usuario. Esta descripción se documentará en el módulo siguiente de esta Fase, Especificación Funcional del Sistema (*figura 2.4*), que deberá incluir:

- ✦ Definición de los Subsistemas.
- ✦ Definición de los datos del Sistema.
- ✦ Interfases de usuario y prototipos.
- ✦ Especificación de la entrega.

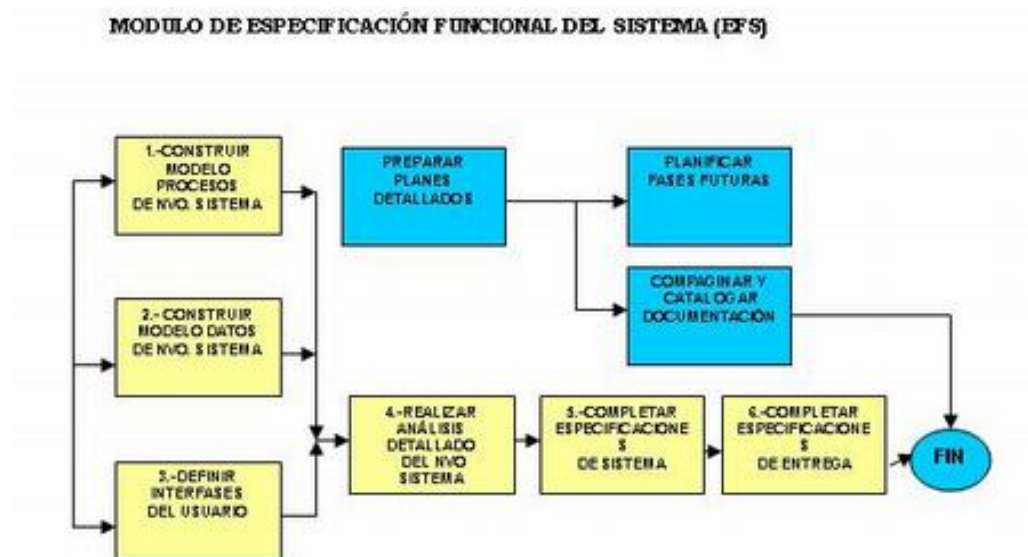


Fig. 2.4 Especificación Funcional del Sistema

2.5.- FASE 2: Diseño de Sistemas

El propósito de esta Fase de Diseño de Sistemas será obtener un conjunto de especificaciones físicas que constituirán el punto de partida para la construcción del Sistema (*figura 2.5*).



Fig. 2.5 Diseño Técnico del Sistema

Durante el desarrollo de las actividades definidas en esta Fase, se deberá tener en cuenta el entorno tecnológico donde se implantará el sistema.

Este aspecto específico hace necesaria una adaptación muy especial de esta Fase de MÉTRICA II al entorno físico que posea el Departamento o Unidad de la Administración que comience a utilizar en sus proyectos los estándares aquí representados.

2.6.- FASE 3: Construcción del Sistemas

El propósito de esta Fase será construir el sistema partiendo del conjunto de especificaciones físicas del mismo, obtenidas durante la Fase anterior (*figura 2.6*).

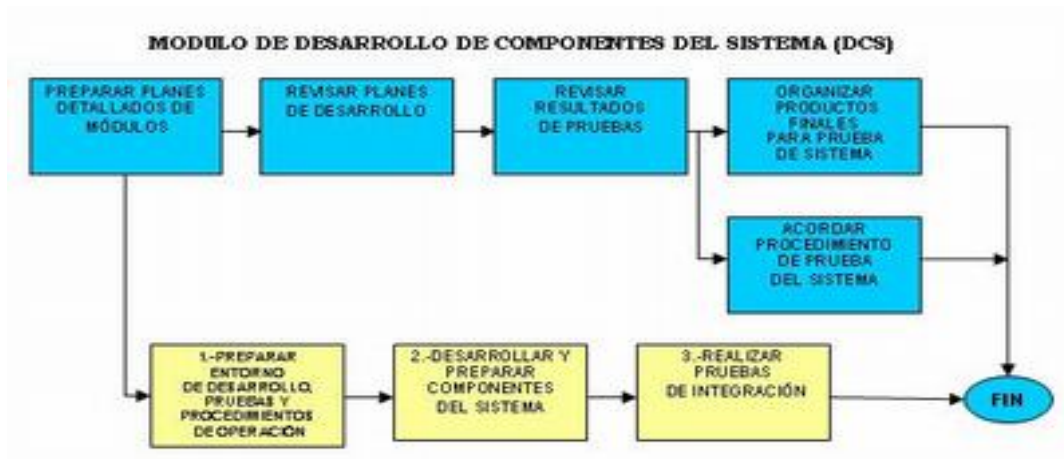


Fig. 2.6 Desarrollo de Componentes del Sistema

Asimismo, se contemplará la realización de las pruebas unitarias necesarias para asegurar el perfecto funcionamiento de los programas desarrollados.

Durante esta Fase se establecerá la estrategia para desarrollar los procedimientos de usuario y el plan de formación a usuario, identificando los recursos para su realización (*figura 2.7*).

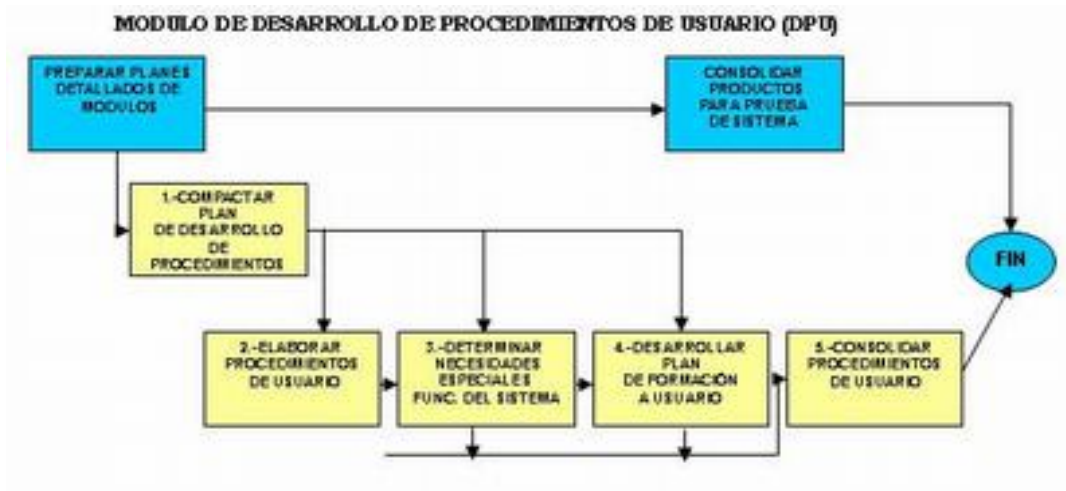


Fig. 2.7 Desarrollo de Componentes del Sistema

2.7.- FASE 4: Implantación de Sistemas

El propósito de la Fase de Pruebas e Implantación es probar el equipo lógico, los procedimientos de usuario y la efectividad de la formación para que, una vez aceptado el sistema, se implante y pase a funcionar en un entorno real de producción (figura 8).

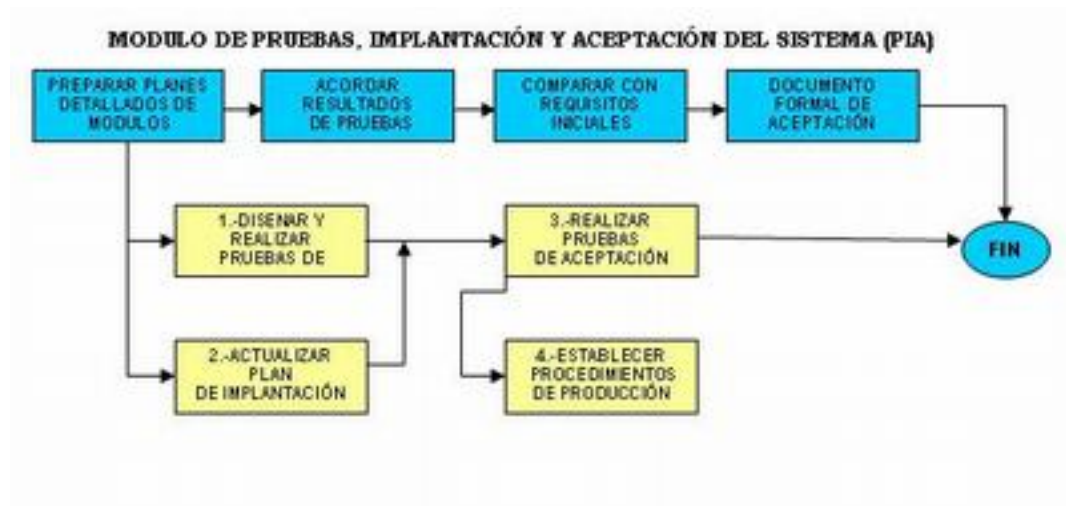


Fig. 2.8 Pruebas, Implantación y Aceptación del Sistema

El objetivo fundamental es conseguir la aceptación final del sistema por parte de los usuarios del mismo, para ello:

- 1.-** Se combinarán por primera vez todo el equipo lógico y los procedimientos para un trabajo del sistema real.
- 2.-** Se realizarán las pruebas de aceptación, las cuales constituyen un procedimiento formal ejecutado por los usuarios que permite verificar que el sistema producido es totalmente funcional y satisface los requisitos iniciales, como un paso previo a su implantación.
- 3.-** Se realizarán los procedimientos necesarios para la implantación y puesta en producción del sistema.

2.8.- Organización de la Metodología MÉTRICA II

La Metodología MÉTRICA II está organizada en tres guías.

La GUÍA DE REFERENCIA es el documento en que se describe el cuerpo completo de la Metodología. En él se presenta:

- a)** Una definición de las Fases, Módulos, Actividades y Tareas a desarrollar, incluyendo una serie de gráficos que muestran las Actividades incluidas en cada una de las Fases y la relación de dichas actividades con otras a desarrollar anterior o posteriormente.
- b)** Una descripción detallada de cada uno de los Módulos, incluyendo las Actividades y Tareas correspondientes indicando la(s) Técnica(s) a

emplear para realizarlas. Para cada actividad se especifican las funciones de los distintos actores implicados en el desarrollo de sistemas: Jefe del Proyecto, Equipo del Proyecto, Comité de Dirección, Director del Proyecto, Grupo de Calidad, Grupo de Usuarios, Especialistas en sistemas y Responsables Técnicos.

c) Una descripción de los productos finales que deberán desarrollarse al final de cada Módulo con ejemplos en algunos casos.

Esta guía permitirá consultar en detalle aspectos del desarrollo de proyecto, por ello al ocurrir esto en un número limitado de ocasiones, no será necesario que cada persona posea una copia de la misma, siendo suficiente que exista una copia por proyecto o grupo de desarrollo.

La GUÍA DE TÉCNICAS es el documento en el que se describen en detalle las técnicas que soportan las distintas fases definidas en MÉTRICA II.

Las técnicas de MÉTRICA II son:

- ✦ Diagramas de flujo de datos
- ✦ Modelado de datos
- ✦ Historia de la vida de las entidades
- ✦ Entrevistas

- ✦ Diseño estructurado
- ✦ Análisis Coste-Beneficio
- ✦ Pruebas
- ✦ Factores críticos de éxito
- ✦ Técnicas matriciales

Para cada técnica se indican sus objetivos, descripción y utilización de la técnica en MÉTRICA y en algunos casos ejemplos.

De la misma forma que en la guía anterior, será suficiente una guía por proyecto o grupo de desarrollo.

La GUÍA DE USUARIO, es un resumen del Manual de Referencia que puede ser empleado para consultas rápidas.

Esta guía será la que ayudará a las personas involucradas en un proyecto de desarrollo a seguir los estándares definidos en MÉTRICA II, por lo tanto, sería interesante que todos los participantes en proyectos de este tipo tengan su propia guía de usuario.

2.9.- Gestión de Proyectos

MÉTRICA II es una metodología flexible, pensada para permitir que el Jefe del Proyecto pueda seleccionar aquellos Módulos y Actividades que cubran las necesidades específicas del mismo, sin tener que

desarrollarlos todos o hacerlo con una estructura inadecuada y evitando, de esta manera, la realización de tareas innecesarias.

Teniendo en cuenta esto, se han identificado diferentes tipos de proyectos, según su duración, complejidad, tipo de ciclo de vida, alcance, etc...

Clasificando los proyectos según sus características se aportan seis Mapas de Actividades:

1. Proyectos Grandes (PG).
2. Proyectos Pequeños (PP).
3. Desarrollo Modular (DM).
4. Prototipo (PT).
5. Mantenimiento de Sistemas (MS).
6. Basada en Paquete (BP).

Estos Mapas de Actividades no tienen el propósito de ofrecer un catálogo al que hay que adaptarse de forma obligatoria, sino que constituyen simplemente una muestra de la adaptación de MÉTRICA II a determinados ciclos de vida considerados como estándares.

Una vez decidido el comienzo de un proyecto, se deberán emplear los conceptos aportados en los Mapas de Actividades para construir el Mapa de Actividades específico del proyecto.

Esta será la herramienta que permitirá al Jefe de Proyecto decidir, sin salirse del esquema de MÉTRICA II, qué actividades se ejecutarán y cuales no, qué productos finales se obtendrán y cuándo, y qué revisiones formales e informales se realizarán y quién tendrá la responsabilidad de ejecutarlas.

Este mecanismo permitirá flexibilizar la utilización de MÉTRICA II, facilitando su implantación en todo tipo de proyectos.

2.10.- Implantación de la Metodología MÉTRICA II

En primer lugar, hay que resaltar la importancia que tiene el proceso de implantación, en el proyecto de adopción de nuevos estándares, técnicas y herramientas en una Organización.

Este proceso determinará el éxito o el fracaso de la utilización adecuada del marco metodológico aportado por MÉTRICA II.

Las Guías que componen esta metodología no tienen el propósito de ofrecer en sí mismas un manual metodológico completo a toda la Administración. Esto es porque en la Administración conviven diferentes entornos, con máquinas, sistemas operativos, lenguajes de programación y hasta organizaciones diferentes.

Con esta perspectiva, MÉTRICA II define un marco de referencia común para el Desarrollo de Sistemas de Información en cada uno de los diferentes entornos. Por ello, en su construcción, se han concretado muchos de los aspectos asociados a la ejecución de proyectos informáticos, pero dejando una puerta abierta a la adaptación de otros aspectos asociados al entorno tecnológico.

Por esta razón, teniendo en cuenta que la mejor Metodología es la que se adapta a las necesidades de cada organización, se deberán considerar los siguientes objetivos en el proceso de implantación de MÉTRICA II:

(a) Completar los aspectos no cubiertos por las guías que componen MÉTRICA II, considerando lenguajes de programación, sistemas operativos, bases de datos, tipos de ordenador, estándares o procedimientos anteriormente establecidos, estándares sobre programación, nomenclatura, etc...

(b) Integrar en MÉTRICA II la utilización de las herramientas de ayuda al desarrollo que se consideren necesarias.

(c) Considerar el cambio cultural y organizativo que la implantación de MÉTRICA II tiene asociado, tratando de minimizar el posible impacto negativo que para los usuarios, personal de desarrollo, responsables de Unidades, pueda representar el comenzar a trabajar con nuevos estándares de desarrollo.

Teniendo en cuenta la consecución de estos objetivos, los pasos a seguir para la correcta implantación del entorno definido serían los siguientes:

PASO 1: Planificación de la implantación y selección de herramientas CASE

PASO 2: Adaptación de MÉTRICA II e integración de Herramientas CASE

PASO 3: Formación

PASO 4: Desarrollo de Proyecto(s)

Para la realización de estos pasos será de utilidad el soporte externo de un grupo especializado en Metodologías de desarrollo y técnicas de modelado, conocedores de MÉTRICA II, por personal externo o por un grupo mixto.

CAPÍTULO 3

METODOLOGÍA MERISE

CAPÍTULO 3.- METODOLOGÍA MERISE

3.1.- Antecedentes

Las bases de MERISE comenzaron en 1.972 por un equipo universitario de ingenieros de Aix-en-Provence. La primera versión salió a finales de 1.976.

El proyecto partió del Centre Technique Informatique del Ministerio de Industria Francés en Septiembre de 1.977, para cubrir las necesidades tanto de la administración como de las empresas. El proyecto finalizó en mayo de 1.978 dando lugar a MERISE como metodología de Análisis y Diseño de Sistemas de Información.

3.1.1.- Objetivos

La realización de un sistema de información no es evidente mientras haga falta reflexionar en donde debemos implementarlo. La frase de concepción necesita de métodos que permitan colocar un modelo sobre el cual apoyarse.

MERISE es un método de concepción, de desarrollo y de realización de proyectos informáticos. La meta de este método es llegar a realizar un sistema de información. El método MERISE está basado en la separación de los datos y de los procedimientos a efectuarse en más modelos conceptuales y físicos.

La separación de los datos y de los procedimientos asegura una vida más larga del modelo.

3.2.- Estructura de la Metodología MERISE

Las fases de la metodología MERISE son:

1. Estudio preliminar.
2. Estudio detallado.
3. Implementación.
4. Realización y puesta en marcha.

3.3.- Estudio Preliminar

Esta fase comprende lo que es la planificación del sistema en desarrollo.

La primera tarea del estudio preliminar es proporcionar un marco de trabajo, que permita organizar los recursos, los costos y el tiempo.

La segunda tarea es la estimación de los recursos requeridos, para el más óptimo desarrollo y desempeño del sistema.

Con lo anterior dicho, se puede simular una pirámide donde las herramientas (hardware y software), son la base proporciona la infraestructura de soporte al esfuerzo de desarrollo, en segundo nivel de la pirámide se encuentran los componentes reutilizables. Y en la parte mas alta de la pirámide se encuentra el recurso primario, las personas (el recurso humano).



Fig. 3.1 Recursos Para el Desarrollo del Sistema

Recurso Humano.- La Cantidad de personas requeridas para el desarrollo de un proyecto de software solo puede ser determinada después de hacer una estimación del esfuerzo de desarrollo, y seleccionar la posición dentro de la organización y la especialidad que desempeñara cada profesional.

Componentes Reutilizables.- La finalidad de utilizar componentes reutilizables es simplemente para evitar gastos innecesarios, que bien pueden ser aprovechados en algún otro recurso que sea indispensable.

Herramientas (Hardware Y Software).- Como ya se había dicho estas dependen de la estimación de los recursos.

3.4.- Estudio Detallado

El estudio detallado abarca lo que es el análisis y el diseño del sistema

3.4.1.- Análisis

El análisis se lleva a cabo teniendo en cuenta los siguientes objetivos:

- ✦ Identificar las necesidades del cliente.
- ✦ Evaluar que conceptos tiene el cliente sobre el sistema para establecer su viabilidad.
- ✦ Realizar un estudio técnico y económico.
- ✦ Asignar funciones a los recursos (hardware, software, personas, bases de datos, etc.).
- ✦ Establecer restricciones de presupuesto y planificación temporal

Para lograr estos objetivos se requiere tener un gran conocimiento y dominio del hardware y software así como la administración de personal y/o la administración de base de datos.

3.4.1.1.- Identificar Necesidades

Aquí se identifican las metas globales, se analizan las perspectivas del cliente, sus necesidades y requerimientos, sobre la planificación temporal y presupuestal, líneas de mercadeo y otros puntos que puedan ayudar a la identificación y desarrollo del proyecto.

3.4.1.2.- Estudio de Viabilidad

Muchas veces cuando se emprende el desarrollo de un proyecto de Sistemas los recursos y el tiempo no son realistas para su materialización sin tener pérdidas económicas y frustración profesional. La viabilidad y el análisis de riesgos están relacionados de muchas maneras, si el riesgo del proyecto es alto, la viabilidad de producir software de calidad se reduce, sin embargo se deben tomar en cuenta cuatro áreas principales de interés:

a) Viabilidad económica.

Una evaluación de los costos de desarrollo, comparados con los ingresos netos o beneficios obtenidos del producto o Sistema desarrollado.

b) Viabilidad Técnica.

Un estudio de funciones, rendimiento y restricciones que puedan afectar la realización de un sistema aceptable.

c) Viabilidad Legal.

Es determinar cualquier posibilidad de infracción, violación o responsabilidad legal en que se podría incurrir al desarrollar el Sistema.

El estudio de la viabilidad puede documentarse como un informe

aparte para la alta gerencia.

3.4.1.3.- Estudio Económico y Técnico.

El estudio económico incluye lo que llamamos, el análisis de costos – beneficios, significa una valoración de la inversión económica comparado con los beneficios que se obtendrán en la comercialización y utilidad del producto o sistema.

Muchas veces en el desarrollo de Sistemas estos son intangibles y resulta un poco difícil evaluarlo, esto varía de acuerdo a las características del sistema. El análisis de costos – beneficios es una fase muy importante de ella depende la posibilidad de desarrollo del Proyecto.

En el Estudio Técnico, el desarrollador evalúa los principios técnicos del Sistema y al mismo tiempo recoge información adicional sobre el rendimiento, fiabilidad, características de mantenimiento y productividad.

Los resultados obtenidos del estudio técnico son la base para determinar sobre si continuar o abandonar el proyecto, si hay riesgos de que no funcione, no tenga el rendimiento deseado, o si las piezas no encajan perfectamente unas con otras.

3.4.1.4.- Modelado de la Arquitectura del Sistema.

Cuando queremos dar a entender mejor lo que vamos a construir, en el caso de edificios, herramientas, aviones, maquinas, se crea un modelo a escala de menor tamaño.

Sin embargo cuando aquello que construiremos es un software, nuestro modelo debe tomar una forma diferente, deben representar todas las funciones y subfunciones, clases y subclases de un sistema. Los modelos se concentran en lo que debe hacer el sistema no en como lo hace, estos modelos pueden incluir notación gráfica, información y comportamiento del Sistema.

Todos los sistemas basados en computadoras pueden modelarse como transformación de la información empleando una arquitectura del tipo entrada y salida.

3.4.1.5.- Especificaciones del Sistema.

Aquí se documenta lo que sirve como fundamento para la ingeniería hardware, software, base de datos, e ingeniería humana. Describe la función y rendimiento del sistema y las dificultades que estarán presentes durante su desarrollo. Y se producen las especificaciones de los requisitos de software.

3.4.2.- Diseño

En el Diseño del Sistema se define el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un Sistema, con suficientes detalles como para permitir su interpretación y realización física.

El proceso de Diseño es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del Sistema a construir. A lo largo del diseño se evalúa la calidad del desarrollo del proyecto con un conjunto de revisiones técnicas:

- ⊕ El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente.
- ⊕ Debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el Software.
- ⊕ El diseño debe proporcionar una completa idea de lo que es el Software, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la Implementación.

Para evaluar la calidad de una presentación del diseño, se deben establecer criterios técnicos para hacer un buen diseño como son:

- ⊕ El diseño debe presentar una organización jerárquica que haga un uso inteligente del control entre los componentes del software.

- ✦ El diseño debe ser modular, es decir, se debe hacer una partición lógica del Software en elementos que realicen funciones y subfunciones específicas.
- ✦ El diseño debe contener abstracciones de datos y procedimientos.
- ✦ Se deben producir módulos que presenten características de funcionamiento independiente.
- ✦ El diseño debe conducir a interfaces que reduzcan la complejidad de las conexiones entre los módulos y el entorno exterior.

Estos criterios no se consiguen por casualidad. El proceso de diseño exige buena calidad a través de la aplicación de principios fundamentales de Diseño y una revisión exhaustiva.

Cuando se va a diseñar un sistema se debe tener presente que el proceso de un diseño incluye, concebir y planear algo en la mente, así como hacer un dibujo o modelo o croquis.

3.4.2.1.- Diseño de la Salida.

En este caso la salida se refiere a los resultados e informaciones generadas por el sistema. Para la mayoría de los usuarios la salida es la única razón para el desarrollo de un Sistema y la base de evaluación de su utilidad. Sin embargo cuando se realiza un sistema,

hay que considerar lo siguiente:

- ✦ Determinar que información presentar. Decidir si la información será presentada en forma visual, verbal o impresa y seleccionar el medio de salida.
- ✦ Hacer que la presentación de la información sea en un formato aceptable.
- ✦ Decidir como distribuir la salida entre los posibles destinatarios.

3.4.2.2.- Diseño de Archivos.

Incluye decisiones con respecto a la naturaleza y contenido del propio archivo, como si se fuera a emplear para guardar detalles de las transacciones, datos históricos, o información de referencia.

No todos los sistemas requieren del diseño de todos los archivos, ya que la mayoría de ellos pueden utilizar los del viejo Sistema y solo tenga que enlazarse el nuevo Sistema al Archivo maestro donde se encuentran los registros.

3.4.2.3.- Diseño de interacciones con la Base de Datos.

La mayoría de los sistemas de información ya sean implantados en sistemas de cómputo grandes o pequeños, utilizan una base de datos que pueden abarcar varias aplicaciones, por esta razón estos sistemas

utilizan un administrador de base de datos, en este caso el diseñador no construye la base de datos sino que consulta a su administrador para ponerse de acuerdo en el uso de esta en el sistema.

3.5.- Implementación

Su objetivo es producir una solución eficiente en un lenguaje ejecutable que implemente las decisiones adoptadas en la fase de diseño. Suele incluir la codificación y la prueba del sistema hasta obtener un paquete ejecutable sobre la plataforma (hardware y S.O.) requerida por el usuario.

Es interesante mencionar que todas las fases anteriores son conceptualmente independientes del lenguaje de programación seleccionado.

Es ahora en la fase de implementación cuándo se selecciona y utiliza un lenguaje de programación determinado; lo que sí es evidente es que el conocimiento del lenguaje de implementación puede orientar la fase de diseño como ocurre en el caso de los lenguajes de programación orientados a objetos, relacionando de forma más directa los objetos o módulos identificados con las construcciones del lenguaje.

Se divide el trabajo entre diversos componentes del equipo de trabajo, éstos han trabajado concurrentemente en el diseño detallado y en la subsiguiente implementación de diversos módulos.

El problema es que ahora es necesario integrar los diversos módulos y construir el sistema de software completo.

Se denomina **integración** al proceso de construir un sistema de software combinando componentes individuales en una unidad ejecutable. Este proceso de integración debe hacerse de forma ordenada para que se integren los módulos en función del uso que unos hacen de otros. La gestión del proyecto deberá asegurar que la integración se realiza adecuadamente.

Una vez obtenida la implementación del sistema es necesario probar que satisface los requisitos definidos inicialmente. Posiblemente, cada uno de los diseñadores que ha estado construyendo cada uno de los módulos ha probado que su implementación está de acuerdo con las decisiones tomadas en el diseño pero no puede asegurar que al integrarlo con otros no existan problemas de incompatibilidades o aspectos no considerados individualmente en cada módulo. Es necesario, por tanto, realizar pruebas a diferentes niveles hasta que el sistema en su conjunto sea aceptado por el usuario.

Al final de la fase, se genera el **Manual de Usuario** junto con el **código fuente** del sistema y las **pruebas** asociadas.

Aunque con la fase de implementación se dispone del «producto», no acaba con ella la actividad del equipo de desarrollo ni la metodología del software construido.

3.6.- Puesta en Marcha

3.6.1.- Transferencia del producto

La fase de **transferencia** del producto tiene como objetivo instalar el sistema de software desarrollado en el entorno del cliente y realizar las pruebas de aceptación necesarias. En muchas ocasiones el proceso de transferencia implica un período largo en el que se incluye la formación del usuario en el producto y la realización de las pruebas de aceptación junto con el usuario.

Debemos tener presente que el usuario deberá aceptar el sistema que se le entrega en función de los requisitos de usuario que dieron origen a todo el proceso. Por ello, es importante que durante el desarrollo sea posible conocer las decisiones asociadas con los requisitos de usuario.

Para muchos productos de consumo general, la fase de transferencia continúa las actividades de prueba iniciadas durante la implementación con la colaboración del cliente.

La entrega de productos es un reconocimiento implícito de que pueden existir problemas tanto de errores ocultos como de adecuación del producto al usuario que saldrán a la luz mediante la interacción con usuarios reales. No olvidemos que la prueba de un sistema puede demostrar la presencia de errores pero nunca su ausencia.

Se suele generar también en esta fase el documento de Historia del Proyecto que resume las lecciones aprendidas y de cuyo análisis se pueden extraer conclusiones para la mejora de los procesos de desarrollo en futuros proyectos.

3.6.2.- Evolución

Una vez que el producto de software ha entrado en operación regular por el usuario no es de ningún modo un sistema inmutable. Todo producto, software complejo debe adaptarse a un entorno que va cambiando (a las nuevas necesidades del cliente, a la evolución de la plataforma de ejecución, hardware o software, etc. Un producto de software que no evoluciona va haciéndose cada vez menos útil en ese entorno.

La evolución del sistema de software suele incluirse dentro de una fase denominada de mantenimiento aunque su implicación es mucho más amplia de lo que el término significa en otras metodologías o ciclos de vida. A nadie se le ocurre llevar su coche a un taller para que le incorporen un nuevo cilindro; sin embargo, parece que modificar líneas de código se puede hacer sin alterar la sustancia del producto, lo cual no es cierto.

Se suele hablar de tres tipos diferentes de mantenimiento:

1) Mantenimiento correctivo. Pretende eliminar problemas surgidos durante la fase de operación del sistema y que no han sido detectados anteriormente.

2) Mantenimiento perfectivo. Pretende mejorar la funcionalidad del sistema ya sea en relación con la eficiencia en ejecución del mismo (menor tiempo de respuesta, optimización del uso de la memoria, etc.), facilitar su uso, etc.

3) Mantenimiento evolutivo. Pretende modificar la funcionalidad del sistema para adaptarla a las nuevas necesidades del usuario o con el objetivo de adaptarlo a nuevas interfaces hardware o software.

CONCLUSIONES

CONCLUSIONES

Las metodologías facilitan la planificación, el control y seguimiento de los proyectos, además de mejorar la gestión de los recursos, tanto materiales como humanos.

Las características de las metodologías son muy amplias a comparación de los ciclos de vida.

Son pocas las metodologías que cuentan con una documentación concreta que informen al desarrollador sobre sus distintas fases y para su aplicación se necesitan expertos, puesto que no cualquier persona tiene el conocimiento de su aplicación.

La metodología MÉTRICA II ofrece una forma de trabajo que permite dividir los problemas que pudieran presentarse en el desarrollo de un proyecto, en elementos más simples para su solución, por lo que se convierte en una herramienta de mayor precisión y confiabilidad.

La metodología MERISE, es una metodología que simplifica la elaboración de los sistemas y basa su funcionamiento en cuatro fases cuando MÉTRICA II se trabaja con cinco.

La diferencia radica en que MÉTRICA II también divide sus fases en módulos y estos a su vez pueden ser divididos en actividades.

En MERISE no todas sus fases se dividen en módulos.

La diferencia mas importante entre estas dos metodologías esta en las necesidades reales que cada organización tiene para manejar su información.

Si se requiere tratar grandes cantidades de información y hacer un sistema complejo y se cuenta con los recursos humanos materiales y de tiempo necesario, ambas metodologías cubren las expectativas.

Generalmente los sistemas de información se hacen con un mínimo de recursos, y el diseño del sistema se puede formar con muchas decisiones a corto plazo. Esto realmente funciona muy bien si el sistema es pequeño, pero conforme el sistema crece llega a ser cada vez más difícil agregar nuevos aspectos al mismo. Además los bugs (malos funcionamientos del sistema) llegan a ser cada vez más frecuentes y más difíciles de corregir.

Por ello existen metodologías como las ya mencionadas, que permiten la elaboración de sistemas de las cuales hay una que se adapta mejor a las necesidades de cada desarrollador de software, o ya sea a la información que se trata de manejar en un sistema.

REFERENCIAS

Referencias Bibliográficas:

- [1] REZA BECERRIL, F. **“Ciencia Metodología e Investigación”**, 1ra. Ed. Pearson, México 1997

- [2] IEEE 1074

- [3] ISO 12207-1

- [4] PRESSMAN, REGER S. **“Ingeniería De Software, Un Enfoque Practico”**, 3a. Ed, Mc Graw Hill, Madrid 1993.

- [5] YOURDON, E., **“Análisis estructurado moderno”**. 1993: Prentice-Hall Hispanoamericana. El proceso de análisis.

- [6] MOLINA GÓMEZ, GRACIELA. Monografía **“Tópicos de ingeniería de software”** UAEH, Pachuca Hgo, 2000

- [7] **"METODOLOGÍA DE Planificación Y DESARROLLO DE SISTEMAS DE INFORMACIÓN. MÉTRICA VERSIÓN II"**. M.A.P. Editado por Ministerio para las Administraciones Publicas.

- [8] KLASTORIN, TED **“ADMINISTRACIÓN DE PROYECTOS”**, 1RA. ED. ALFAOMEGA GRUPO EDITOR S.A. DE C.V., SEPTIEMBRE 2005.

Referencias Electrónicas:

URL: <http://www.ati.es/novatica/1995/nov-dic/nv118.html>

URL: http://www.mailxmail.com/curso/empresa/desarrollo_proyectos

URL: <http://www.csi.map.es/csi/pg5m41.htm#9>

URL: <http://www.ati.es/metodologias/1998/nv118.html>

URL: www.microsoft.com/spanish/msdn/arquitectura/arquitectura_soft.asp

URL: <http://www.dsdm.org/>

URL: <http://www.csi.map.es/csi/pg5m41.htm>

URL: <http://dmi.uib.es/~bbuades/evs/sld003.htm>

URL: <http://www.ati.es/novatica/1995/nov-dic/nv118.html>

URL: <http://www.uned.es/413057/Introduccion.pdf>

URL: <http://www.inst-informatica.pt/v20/ibst/metodos/metrica.htm>

URL: <http://sqlpro.developpez.com/cours/modelisation/merise/?page=base>