

INT

Introducción

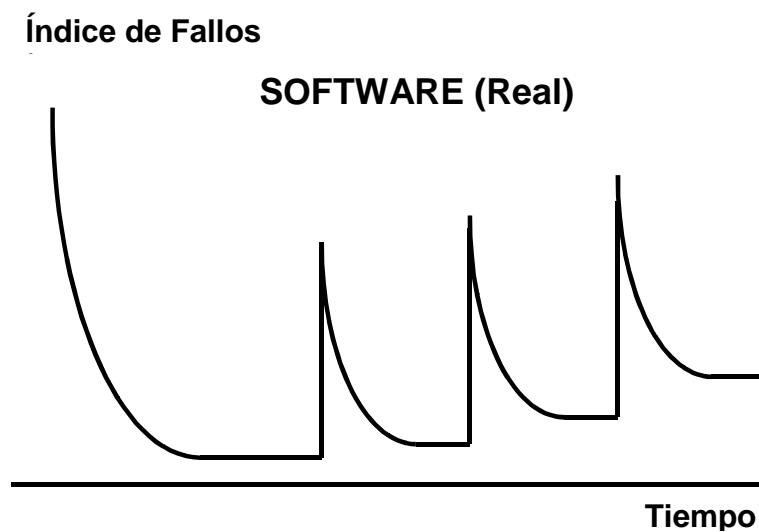
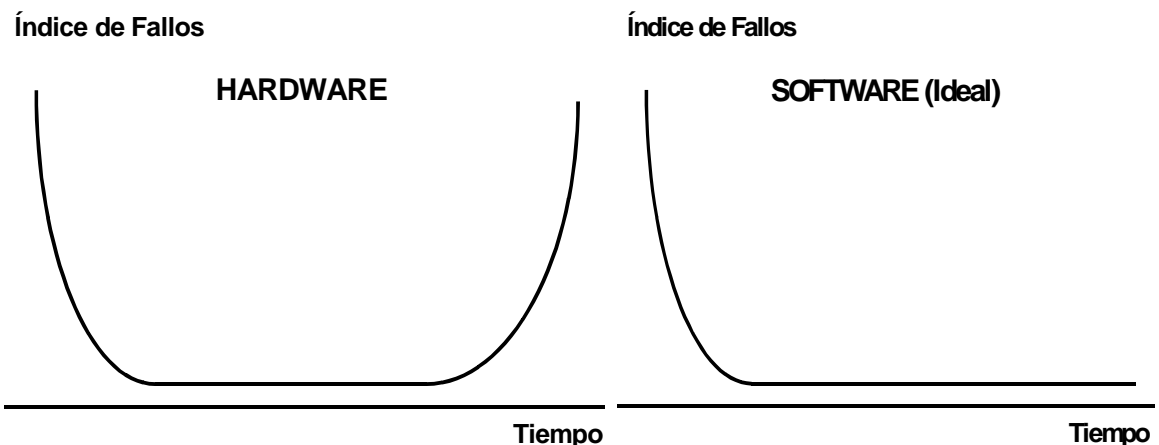
Software	2
Metodologías y Herramientas.....	5
Procesos de Software	8
Modelos de Proceso Software	9
Visión Genérica de la IS.....	15
Métrica Versión 3.....	17
Estructura Principal	20
Interfaces.....	22
Técnicas y Prácticas	23
Participantes	26
Métrica 3 vs. Métrica 2	28
Bibliografía.....	29

Software

Instrucciones + Estructuras de Datos + Documentación

Características

- El software se desarrolla, no se fabrica. Los costes se concentran en la ingeniería, no en la producción.
- El software no se estropea, pero se “deteriora”.



- **La mayoría del software se construye a medida. No existen catálogos de componentes software.**

Reutilización:

- **Bibliotecas**
- **Técnicas de programación estructurada y modular**
- **Técnicas orientadas a objetos**

Problemas

“Crisis del Software”

- **La planificación y estimación de costes son muy imprecisos**
 - **No existen datos sobre proyectos anteriores**
 - **Gestores de proyectos no especializados en el desarrollo del software**
- **Productos no entregados a tiempo**
- **Falta de comunicación con los usuarios**
- **Baja productividad**
 - **Desarrollo sobre especificaciones ambiguas**
 - **Modificar especificaciones sobre “la marcha”**
 - **Falta total de documentación del desarrollo y modificaciones**

- **Mala calidad**
 - **Pocas o nulas pruebas exhaustivas ⇒ Software con errores**
- **El producto no responde a las necesidades de usuarios**
 - **Poco interés y tiempo en análisis de requisitos**
 - **Errores en el producto en su entrega**
 - **Los productos se vuelven a diseñar y/o desarrollar**
- **Gran diversidad (productos, métodos, documentación)**

Necesidades

- **Asegurar calidad en el software**
- **Conseguir plazos y costes razonables**
- **Marco homogéneo de referencia**

Solución (parcial?): Ingeniería del Software

“Aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software” (IEEE – 93)

- **Metodología de Planificación y Desarrollo de Sistemas de Información MÉTRICA Versión 3**
- **Plan General de Calidad Aplicable al Desarrollo de Equipos Lógicos**

Metodologías y Herramientas

Método

Un determinado enfoque (“approach”) para solucionar un determinado problema. Indican como construir técnicamente el software.

Consta de:

- Notación (gráfica y/o textual)
- Técnicas que han de ser utilizadas para solucionar el problema

Metodología

- Estudio de los métodos
- Conjunto de métodos integrados para alcanzar una meta común

Definen la secuencia en la que se aplican los métodos, productos o documentos requeridos, controles de calidad,...

(Definiciones parcialmente extraídas del SEI: Software Engineering Institute, Carnegie Mellon University)

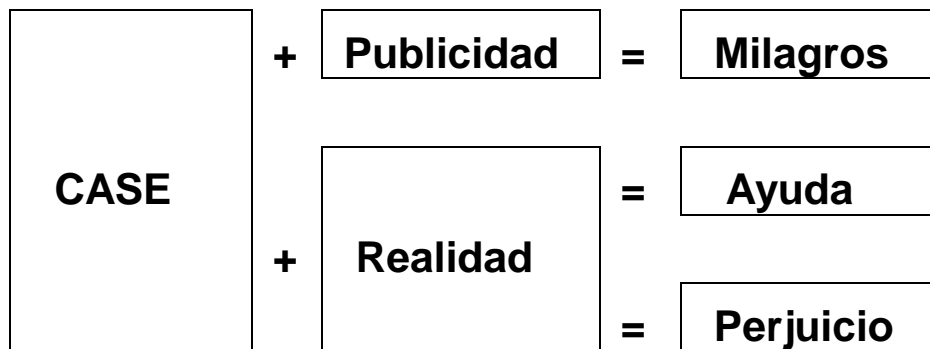
Herramientas

Productos para automatizar un método, parte o varios

- **CAD/CAM/CAE (Computer Aided Design, Manufacturing and Engineering)**
- **CASE (Computer Aided Software/System Engineering)**

- **UpperCASE (Principio del ciclo de vida. ej. análisis)**
- **LowerCASE (Final del ciclo de vida, ej. generador de código)**
- **Toolset (Varias Herramientas integradas para soportar una metodología)**
- **IPSE (Integrated Programming Support Environment) Soporte de todo el ciclo de vida**
- **I-CASE (Integrated CASE). Para diferenciar de herramienta no integradas**
- **MetaCASE (Generador de herramientas CASE)**

Es la tecnología CASE la solución?



- **Por sí mismas, las herramientas NO SOLUCIONAN NADA**
- **Hay que CONOCER a fondo los métodos de desarrollo, gestión y control**
- **CASE posibilita y fomenta el uso de métodos. Es una solución parcial**

Ayuda a rebajar costes?

- **PRODUCCIÓN (análisis, diseño, implementación):**
 - No decrecen sensiblemente**
 - Incluso crecen algo**
- **MANTENIMIENTO:**
 - Decrece drásticamente**

Procesos de Software

Conjunto de actividades y productos obtenidos durante el desarrollo de un sistema software, independientemente de su tamaño o complejidad

Actividades de un proceso genérico

- **Especificación y Análisis**

Se definen las funcionalidades y restricciones del software a construir

- **Desarrollo**

Producción propiamente dicha de forma que cumpla con las especificaciones

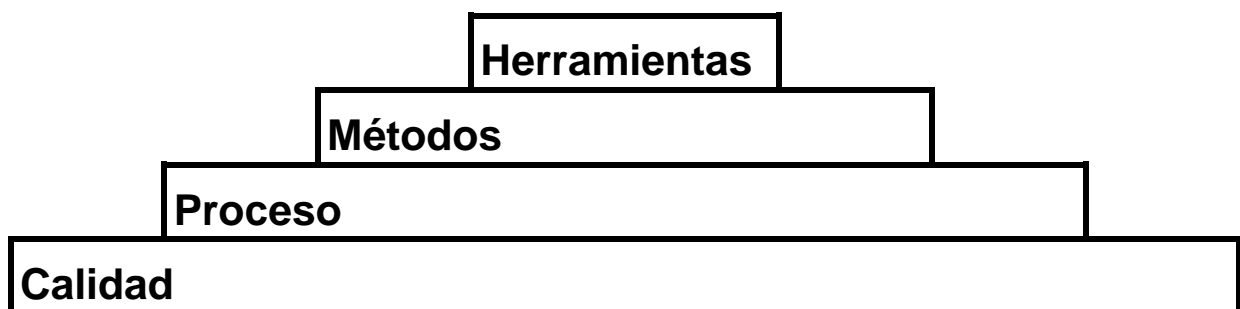
- **Validación**

Asegurar que el software es lo que el cliente requiere

- **Evolución**

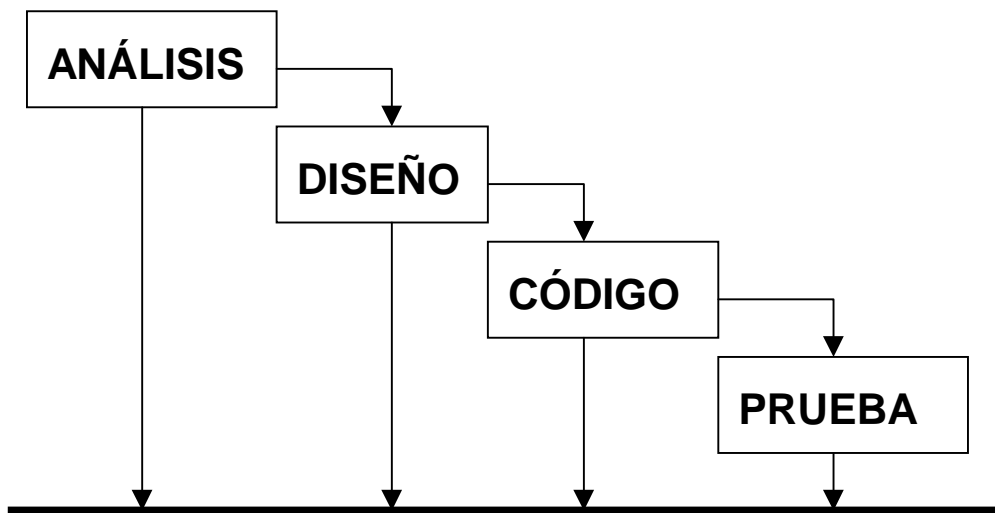
Adaptación a posibles cambios

Capas en la Ingeniería del Software



Modelos de Proceso Software

Lineal Secuencial o “en Cascada”

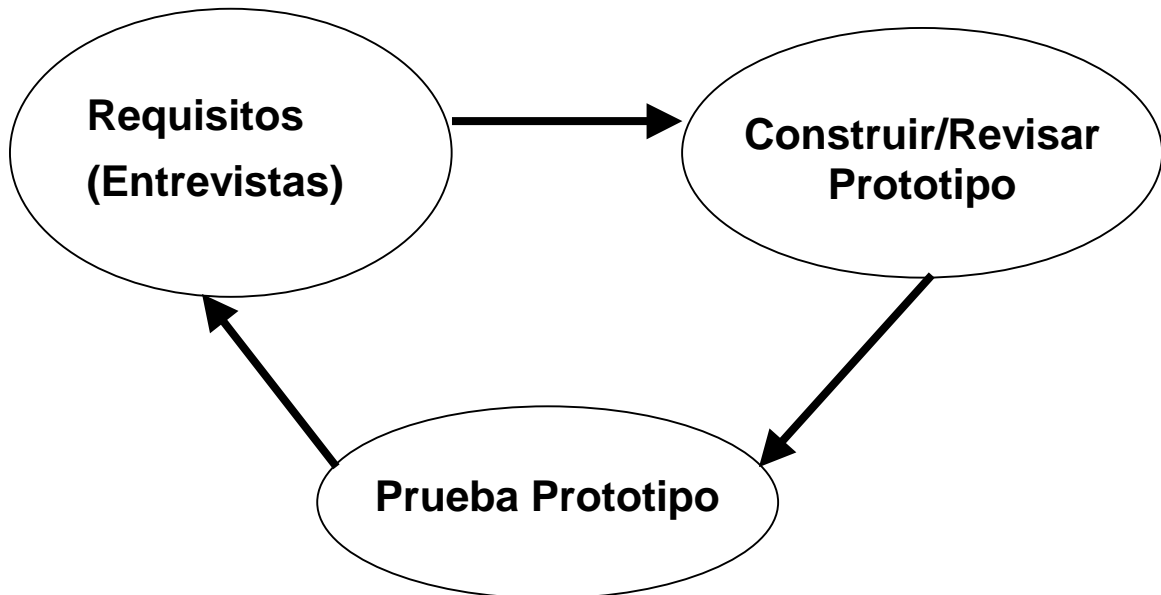


Problemas:

- Difícil ajustar a proyectos reales
- Difícil que el usuario exponga los requisitos al principio
- No está disponible versión operativa hasta las últimas etapas

Orientado a Prototipos

Mecanismo para la identificación de requisitos del software



Problemas:

- La gestión del desarrollo de software es muy lenta
- Compromisos implementación/eficiencia (Lenguaje de programación, sistema operativo, etc.)

Modelo Incremental

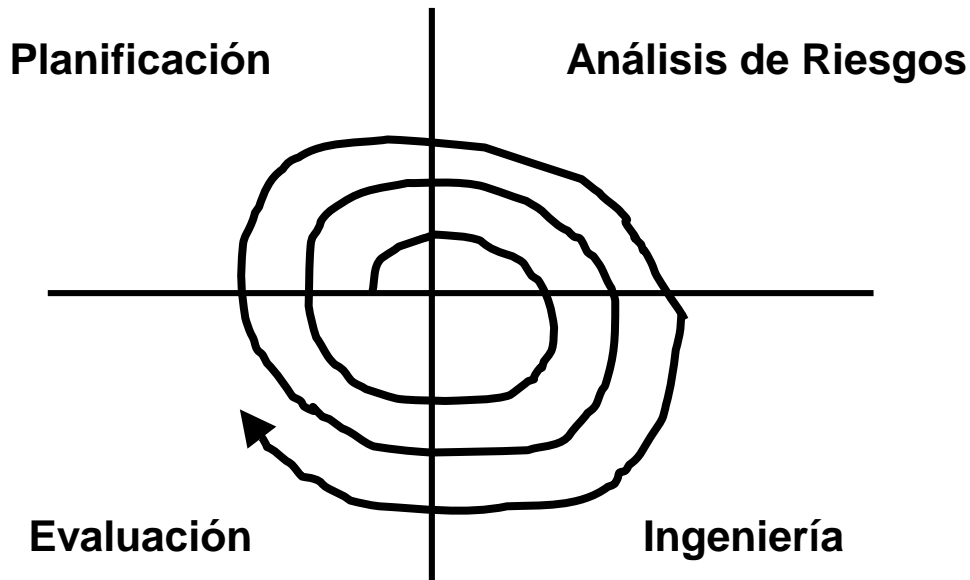
Basado en el ciclo de vida lineal y en la construcción de prototipos

- **En primer lugar, se crea un primer “incremento” (*producto esencial*) utilizando el modelo en cascada que contempla los requisitos básicos.**
- **Se elaboran nuevos incrementos, incorporando en cada uno nuevos requisitos no detectados hasta el momento, o no incluidos en anteriores incrementos**
- **Así, los primeros incrementos son versiones “desmontadas” del producto final**
- **Cada incremento genera un producto operacional (o versión “incompleta”)**

¿Incremental vs. Prototipos?

Espiral (Boehm, 1988)

Combina las principales ventajas del ciclo de vida en cascada y el orientado a prototipos.



- El primer ciclo de la espiral especifica el producto
- Cada uno de los siguientes un nuevo prototipo (refinado)

Problemas:

- Poca experimentación en proyectos

Basado en Componentes

- Reutilización del software
- Adecuado en tecnologías orientadas a objetos
- Especialización del modelo en espiral, donde las aplicaciones se desarrollan a partir de componentes previamente creados
- Ingeniería:
 - Identificación de componentes candidatos
 - Acceso a los componentes de la *librería*
 - Construcción, en su caso, de componentes y almacenamiento en la librería
- El proceso anterior forma la primera iteración en la espiral
- En futuras iteraciones, se deben ensamblar los componentes

Problemas

- Excesiva dependencia de la calidad y robustez de la librería de componentes

Ciclos de Vida Orientados a Objeto

- En sus orígenes la Orientación a Objeto (OO) enfatizaba sobre todo en los lenguajes de programación (Lenguajes de Programación Orientados a Objeto)
- A medida que el interés aumentaba en la industria, el estudio se traslada al Diseño, y posteriormente al Análisis
- Básicamente, la mayoría de metodologías especializan los paradigmas iterativos (construcción de prototipos) y evolutivos (incremental y espiral).
- En general, en estas metodologías la tupla (Análisis-Diseño-Construcción) es más compacta que en las metodologías estructuradas, ya que todo el proceso se basa en clases y objetos

Ejemplo: Rational Unified Process (RUP)

(<http://www.rational.com/products/whitepapers>)

- Proceso Software basado en el uso efectivo de la notación UML (Unified Modeling Language)
- Utilización a lo largo de todo el ciclo de vida de patrones (“templates”) y herramientas
- Cuatro fases secuenciales:
 1. Incepción: Definición inicial del proyecto, plan de proyecto, costes, personal, riesgos, etc.
 2. Elaboración: Descripción de arquitectura, riesgos revisados, plan revisado, manual de usuario inicial, etc.
 3. Construcción: Desarrollo, integración y prueba de los componentes
 4. Transición: Pruebas beta por los usuarios, conversión de datos, plan de formación de usuarios, etc.

Visión Genérica de la IS

Independientemente del paradigma, se puede ver el proceso de desarrollo:

Definición (QUÉ)

- Estudio del Sistema
- Planificación
- Análisis de Requisitos

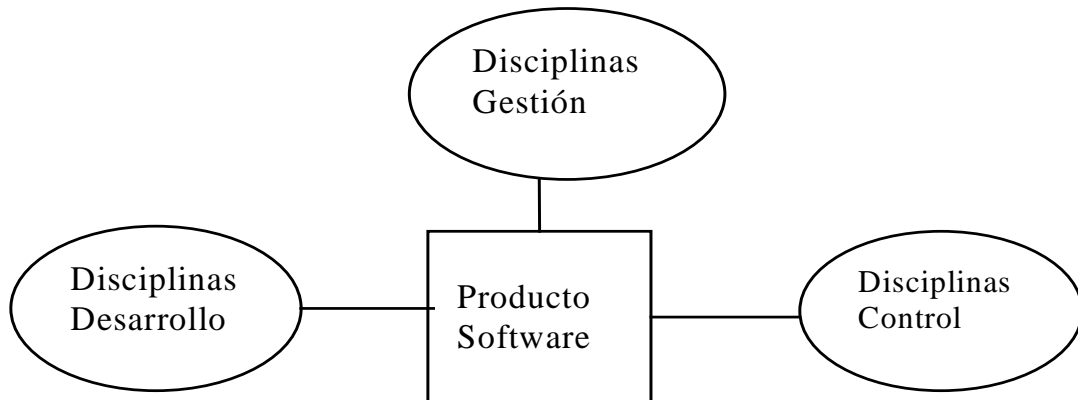
Desarrollo (CÓMO)

- Diseño (arquitectura, estructuras de datos, ...)
- Codificación
- Prueba

Mantenimiento (CAMBIO)

- Corrección
- Adaptación
- Mejora

Desarrollo de un producto software



Normativa del MAP

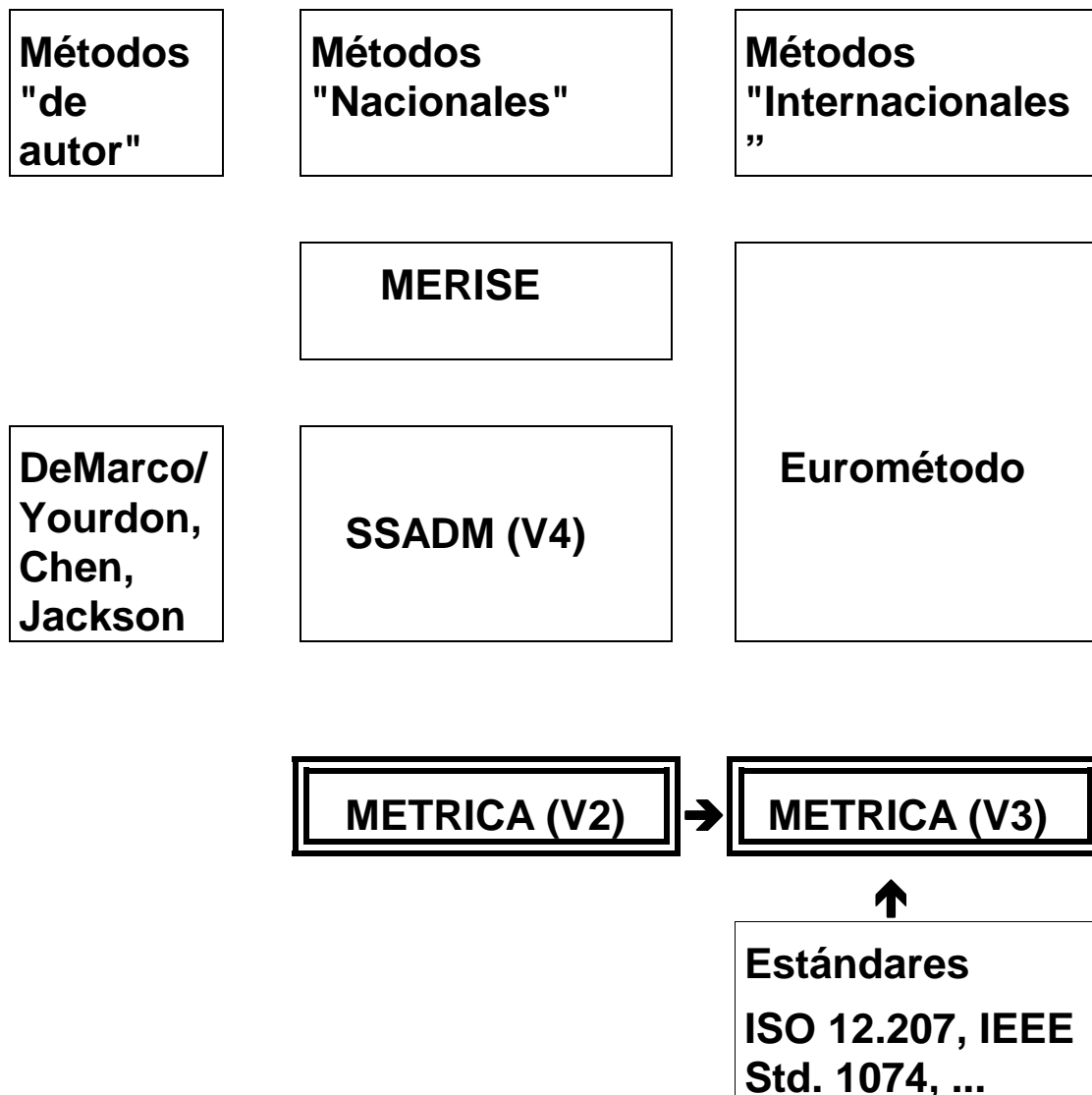
Disciplinas de desarrollo	Metodologías de desarrollo	Métrica Versión 3
Disciplinas de Control	Aseguramiento de la calidad (SQA: Software Quality Assurance) Gestión de Configuraciones (SCM: Software Configuration Management)	Plan General de Garantía de Calidad aplicable al desarrollo de equipos lógicos Interfaz de Calidad de Métrica 3 Plan de Gestión de Configuraciones (en proyecto) Interfaz de Gestión de la Configuración de Métrica 3

Métrica Versión 3

Génesis de Métrica

- Ministerio para las Administraciones Públicas
- Otros Ministerios
- Empresa Consultora (Coopers & Lybrand)

Origen de Métrica



Objetivos

- **Abierta:** apta para Administración y empresas privadas
- **Formal:** establece pasos necesarios
- **Orientada a obtener productos de calidad:** énfasis en revisiones formales
- **Flexible:** adaptable a distintos proyectos, entornos, complejidades y ciclos de vida
- **Sencilla:** práctica y de fácil uso

Organización de la metodología

(<http://www.map.es/csi>)

- Estructura Principal
- Interfaces
- Técnicas
- Participantes

- Herramientas de Apoyo
 - **AUTOFORMACIÓN**
 - **SELECTOR (Selección de Herramientas: Selección de productos software del mercado)**
 - **GESMET (Gestor Metodológico: Adaptación de Métrica 3 a proyectos concretos)**

Estructura Principal

Objetivo

Definir la estructura del proyecto: Ciclo de vida

- **Procesos**
- **Actividades**
- **Tareas**

Basada en un enfoque orientado a proceso

Distingue entre Procesos Principales (Planificación, Desarrollo y Mantenimiento) y Procesos de Interfaz (Calidad, Seguridad, Gestión y Configuración)

Basada principalmente en el ciclo de vida secuencial o de cascada (“waterfall”)

Como consecuencia de cada fase, se indican los PRODUCTOS a conseguir:

- **Documentos**
- **Código**

Procesos Principales

- **Planificación de Sistemas de Información (PSI)**
- **Desarrollo de Sistemas de Información**
 - **Estudio de Viabilidad del Sistema (EVS)**
 - **Análisis del Sistema de Información (ASI)**
 - **Diseño del Sistema de Información (DSI)**
 - **Construcción del Sistema de Información (CSI)**
 - **Implantación y Aceptación del Sistema (IAS)**
- **Mantenimiento de Sistemas de Información (MSI)**

Planificación	Desarrollo			Mantenimiento
	Análisis	Diseño	Construcción e Implantación	

	EVS		CSI	
	Estudio de Viabilidad del Sistema de Información		Construcción del Sistema de Información	
PSI		DSI		MSI
Planificación de Sistemas de Información		Diseño del Sistema de Información		Mantenimiento del Sistema de Información
	ASI		IAS	
	Análisis del Sistema de Información		Implantación y Aceptación del Sistema de Información	

Interfaces

Definen un conjunto de actividades de tipo organizativo o de soporte al proceso de desarrollo y/o productos

- **Gestión de Proyectos (GP): planificación, seguimiento y control de actividades y recursos humanos y materiales**
- **Seguridad (SG): análisis de riesgos lógicos**
- **Gestión de la Configuración (GC): definir y controlar los cambios en la configuración del sistema, modificaciones y versiones.**
- **Aseguramiento de la Calidad (CAL): marco de referencia para la definición y puesta en marcha de planes de aseguramiento de la calidad.**

Técnicas y Prácticas

Técnicas

Heurísticas y procedimientos con notaciones específicas (sintaxis y semántica), apoyados en estándares (p.e., modelo entidad/relación)

Prácticas

Procedimientos y notaciones con menor rigidez (p.e., realización de entrevistas)

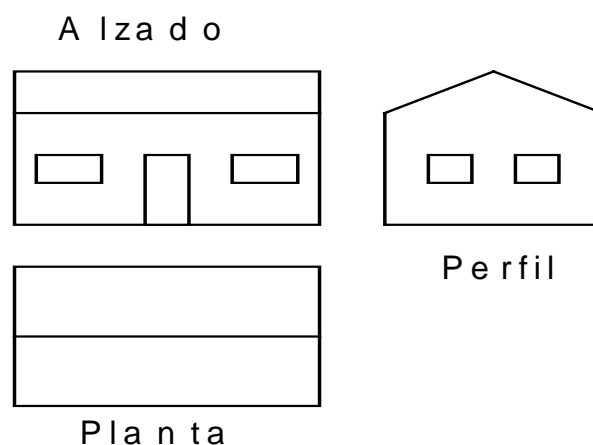
Diferentes técnicas, la mayor parte de ellas con el objetivo de crear un modelo (parcial) del sistema

Qué es un modelo?

Una forma de representar un sistema (norma IEEE)

Ejemplo en Arquitectura

Un mismo objeto puede representarse de acuerdo con tres vistas diferentes



Para Sistemas Informáticos

- Típicamente tres modelos (tres vistas diferentes del mismo sistema)
- Al menos dos son necesarios

Modelo	Objetivo	Técnicas
Modelo Funcional (dinámico)	Lo que hace	Diagramas de Flujo de datos (DFD) Casos de Uso (CU)
Modelo de datos Retenidos (estático)	Lo que recuerda	Diagramas Entidad-Relación (ERD)
Modelo de transición de estados (evolutivo)	Los diferentes estados que caracterizan al sistema	Diagramas de transición de estados (STD)

Técnicas más relevantes utilizadas en Métrica 3 (Estructurado)

Objetivo	Técnica			
Modelo funcional	Diagramas de Flujo de Datos	DFD	Data Flow Diagram	DFD
	Diagrama de Casos de Uso	DCU		
Modelo de datos	Diagrama de Estructura lógica de Datos	DED	Logical Data Structure	LDS
Modelo de transición de estados	Diagramas de Transición de Estados	DTE	State Transition Diagrams	STD
Diseño	Diagramas de estructura de cuadros	DEC	Structure Diagrams	SD
Pruebas			Testing	

Participantes

Definición de perfiles de los participantes en los procesos principales de la metodología

Coinciden con los definidos en la herramienta AUTOFORMACIÓN

Perfiles establecidos:

- **Directivo**
- **Jefe de Proyecto**
- **Consultor**
- **Analista**
- **Programador**

Organización de la documentación		EP	T
Introducción	INT		
Diagramas de Flujo de Datos	DFD		X
Diagramas de Casos de Uso	DCU		X
Diagramas de Estructura Lógica de Datos	DED		X
Diagramas de Transición de Estados	DTE		X
Historia de la Vida de las Entidades	HVE		X
Estudio de Viabilidad del Sistema	EVS	X	
Análisis del Sistema de Información	ASI	X	
Diseño	DIS		X
Diseño del Sistema de Información	DSI	X	
Pruebas	PRU		X

- **EP: Métrica-3: Estructura Principal**
- **T: Métrica-3: Técnicas**

Métrica 3 vs. Métrica 2

	MÉTRICA 3	MÉTRICA 2
Estructura Metodología	Procesos	Fases
	Actividades Tareas	Módulos Actividades Tareas
Planificación	PSI	PSI
Desarrollo	EVS	ARS
	ASI	EFS
	DSI	DTS
	CSI	DCS
	IAS	DPU PIA
Mantenimiento	MSI	

Aportaciones Métrica 3

- **Compatibilidad con los últimos estándares en Calidad e Ingeniería del Software**
- **Interfaz con Procesos de Gestión y Control**
- **Cubre desarrollos estructurados y orientados a objeto**
- **Ampliación para tecnologías Cliente/Servidor e Interfaces de Usuario Gráficas**
- **Técnicas soportadas por varias herramientas CASE**

Bibliografía

- **"Metodología de Planificación, Desarrollo y Mantenimiento de Sistemas de Información, MÉTRICA 3", Ministerio para las Administraciones Públicas (MAP), 2001. (Accesible desde <http://www.map.es/csi/metrica3/index.html>)**
- **"Plan General de Calidad Aplicable al Desarrollo de Equipos Lógicos", Ministerio para las Administraciones Públicas (MAP), 1992**
- **"Ingeniería del Software: Un enfoque Práctico, Quinta Edición", R.S. Pressman. McGraw-Hill, 2002**
- **"Ingeniería del Software, Sexta Edición", I. Sommerville. Addison-Wesley, 2002**
- **"Análisis y diseño detallado de aplicaciones informáticas de gestión", M.G. Piattini y otros, Ra-Ma 1996**
- **"Ingeniería del Software de Gestión: Análisis y Diseño de Aplicaciones", A. de Amescua y otros. Paraninfo, 1995**
- **"Utilización de UML en Ingeniería del Software con Objetos y Componentes", P. Stevens, R. Pooley. Addison-Wesley, 2002.**
- **"El Lenguaje Unificado de Modelado", G. Booch, J. Rumbaugh, I. Jacobson, Addison-Wesley, 2002.**
- **"Structured Analysis and System Specification", T. DeMarco. Yourdon Press, 1979**
- **"Modern Structured Analysis", E. Yourdon. Prentice-Hall, 1989**
- **"IEEE Software Engineering Standards Collection", Institute of Electrical and Electronic Engineers. IEEE Computer Society, 1994**

- **"IEEE Transactions on Software Engineering". Revista de *Institute of Electrical and Electronic Engineers*, IEEE Computer Society.**
- **"ACM Transactions on Software Engineering and Methodology". Revista de *Association for Computing Machinery*.**
- **"Novática". Revista de la *Asociación de Técnicos de Informática* (ATI)**
- **"The NATO Software Engineering Conferences" 1968, 1969. (Accesible desde: <http://www.cs.ncl.ac.uk/old/people/brian.randell/home.for mal/NATO/>)**