

Metodologías de desarrollo de software

1.7 Nivel de madurez del proceso software

Método diseñado por el Instituto de Ingeniería del Software (SEI) en la universidad Carnegie Mellon (EE.UU.), evalúa el nivel de madurez del proceso de desarrollo de software.

Hay cinco niveles de madurez:

1. **Inicial.** La empresa no dispone de procesos y controles definidos. En este caso no hay control efectivo de la gestión de proyectos de software.
2. **Repetible.** La empresa dispone de un control básico de gestión de proyectos. El problema es que cualquier modificación supone riesgo elevado de fracaso.

01/11/02

IAGP

1

Metodologías de desarrollo de software

3. **Definido.** Se monitorizan y mejoran los procesos. Las empresas en este nivel disponen de:
 - Un grupo de proceso para mejorar el proceso software
 - Una metodología que describe las actividades
4. **Gestionado.** La empresa posee controles avanzados, métricas y realimentación.
5. **Optimizado.** Se emplean métricas con el objetivo de optimización, sobre todo de los procesos más débiles.

Este modelo es útil para mostrar el estado y las áreas de problemas clave en empresas.

01/11/02

IAGP

2

Metodologías de desarrollo de software

1.8 Situación de la industria de tecnologías de la información

- § Las economías de los países desarrollados dependen en gran parte del software.
- § Más y más sistemas son actualmente controlados por software.
- § La Ingeniería de Software concierne a teorías, métodos y herramientas para el desarrollo profesional de software.
- § El gasto en la Ingeniería de Software, representa un alto porcentaje del PIB de los países desarrollados.
- § Los costos del software superan el costo del sistema. El costo del software en una PC es a menudo más caro que la PC.
- § Cuesta más mantener el software que desarrollarlo. Para sistemas con una larga vida, este costo se multiplica.
- § La Ingeniería de Software concierne a un desarrollo efectivo en cuanto a costes del software.

01/11/02

IAGP

3

Metodologías de desarrollo de software

La información tiene su origen en:

- Ministerio de Ciencia y Tecnología
- Asociaciones de empresas del sector (por ej, Sedisi)
- Empresas de estudios de mercado

La informática española está orientada hacia los servicios y el software.

- Crece el empleo y las inversiones.
- Muy por debajo del resto de Europa
- No hay datos fiables de la industria del software

01/11/02

IAGP

4

Metodologías de desarrollo de software

Tipos de productos de software:

- Productos genéricos.

Productos que son producidos por una organización para ser vendidos al mercado.

- Productos hechos a medida.

Sistemas que son desarrollados bajo pedido a un desarrollador específico.

La mayor parte del gasto del software es en productos genéricos, pero hay más esfuerzo en el desarrollo de los sistemas hechos a medida.

01/11/02

IAGP

5

Metodologías de desarrollo de software

Novedad de IBM

IBM va a informar esta semana a sus 500 mayores clientes sobre sus planes destinados a desarrollar sistemas informáticos con capacidad para "autoarreglarse" cuando se estropeen.

Autonomic Computing, que forma parte de la división de software de la compañía, también trabajará para estandarizar aquellos esfuerzos enfocados en el desarrollo de programas de 'software' para otras compañías.

01/11/02

IAGP

6

Metodologías de desarrollo de software

1.9 Importancia de la metodología

Muchos factores repercuten en la persona que trabaja en el desarrollo de software:

- Cambios en sistema operativo
- Cambios en lenguaje de programación
- Organización del proyecto
- Estándares considerados

La productividad es la cantidad de trabajo que puede ser realizada por una persona. Hay parámetros que pueden hacerla variar mucho.

La productividad no considera la calidad del producto.

01/11/02

IAGP

7

Metodologías de desarrollo de software

El objetivo de cualquier empresa de desarrollo de software es disponer de personas muy cualificadas.

El ingeniero de software trabaja en equipo, aunque algunos no son partidarios de seguir una metodología.

Ejemplo en la industria del automóvil:

- Volvo tiene organizada la producción en equipo, compartiendo responsabilidades.
- BMW no está organizada la fabricación en equipo, hay falta de comunicación entre los trabajadores.

Ambas empresas producen vehículos análogos a los mismos precio, pero con diferente metodología.

01/11/02

IAGP

8

Metodologías de desarrollo de software

1.10 Modelo de métodos formales

Este modelo acompaña a un conjunto de actividades que conducen a la especificación matemática del software.

Los métodos formales permiten que se desarrollen y verifiquen los sistemas mediante una rigurosa notación matemática.

Se eliminan muchos de los problemas habituales de las tecnologías convencionales, como ambigüedad, inconsistencia e incompletitud.

Se logra un producto libre de defectos, aunque requiere más tiempo y es difícil de comprender por personas sin muchos conocimientos.

01/11/02

IAGP

9

Metodologías de desarrollo de software

- Estos métodos es poco probable que sean de amplio uso en el futuro.
- Se convierten en una aproximación para el desarrollo de sistemas críticos de seguridad y estándar.
- Estos cambios expanden el perfil hacia el proceso de software.

01/11/02

IAGP

10

Metodologías usadas en ingeniería del software

1.11 Criterios para evaluar una metodología

- Debe de **ajustarse a los objetivos**. Un desarrollo siempre es para lograr unos objetivos, luego ha de adaptarse a ellos.
- Debe **cubrir el ciclo entero de desarrollo**. Ha de seguir tres etapas: investigación, análisis de requisitos y diseño.
- Debe **integrar las distintas fases del ciclo de desarrollo**. Es importante la rastreabilidad en ambos sentidos y la fácil interacción entre etapas del desarrollo.
- Debe de **incluir la realización de validaciones**. Es necesaria una validación de cada fase antes de pasar a la siguiente.
- Debe soportar la **determinación de la exactitud del sistema**. Implica diversas fases, como por ejemplo la correspondencia entre el sistema y sus especificaciones y que cumple con las necesidades del usuario.

01/11/02

IAGP

11

Metodologías usadas en ingeniería del software

- Debe ser la **base de una comunicación efectiva**. Ha de existir durante todo el proceso comunicación entre analistas, programadores, usuarios y gestores.
- Debe de **funcionar en un entorno dinámico orientado al usuario**. Se ha de producir a lo largo del ciclo de vida, una transferencia de conocimientos hacia el usuario. Se han de usar técnicas sencillas, como por ejemplo diagramas.
- Debe de **especificar claramente los responsables** de los resultados. Cada tarea ha de estar indicada por el responsable de ella.
- Debe de poder **emplearse en un entorno amplio** de proyectos software. Se ha de adoptar una metodología útil para un gran número de sistemas a elaborar, tanto en tamaño, vida, complejidad y entornos.

01/11/02

IAGP

12

Metodologías usadas en ingeniería del software

- Se debe de **poder enseñar**. Incluso en una organización sencilla serán muchos los usuarios, cada persona debe de comprender el sistema que está usando.
- Debe de estar **soportada por herramientas CASE**. Permiten mejorar la productividad, tanto a nivel individual como del proyecto entero.
- Debe **soportar la eventual evolución del sistema**. Muchos sistemas duran incluso 10 años, tiempo en el que será necesaria una evolución.
- Debe incluir **actividades conducentes a mejorar el proceso de desarrollo**. Se ha de disponer de mediciones del proceso para identificar la calidad y coste de cada etapa. Se suelen usar herramientas CASE.

01/11/02

IAGP

13

Metodologías usadas en ingeniería del software

¿Qué criterios son los más importantes?

Hay pocos estudios en este aspecto.

La tendencia, para los aspectos más importantes es:

- Capacidad de la metodología para desarrollar sistemas con la calidad requerida.
- Satisfacción del usuario
- Capacidad de desarrollar gran variedad de sistemas
- Ha de soportar todas las etapas del desarrollo.

01/11/02

IAGP

14

Metodologías usadas en ingeniería del software

1.12 Paradigmas de programación

Un paradigma de programación es un modelo básico de diseño y desarrollo de programas, que permite generarlos con unas directrices específicas, tales como: estructura modular, gran cohesión, alta rentabilidad, etc.

Puede parecer sorprendente la existencia de varios paradigmas de programación, aunque la mayoría de los programadores sólo está familiarizado con uno, el de la programación procedimental.

Hay multitud, son muy conocidos:

- Basado en reglas de producción
- Funcional
- Heurístico
- Orientado al objeto

01/11/02

IAGP

15

Metodologías usadas en ingeniería del software

Un paradigma de programación se define como: una colección de modelos conceptuales que juntos modelan el proceso de diseño y determinan, al final, la estructura de un programa.

Para que el proceso sea efectivo, las características del lenguaje deben reflejar de forma adecuada los modelos conceptuales del paradigma bajo consideración.

Entonces se dice que el lenguaje soporta el paradigma, a veces es difícil distinguirlo del propio paradigma, se identifica con él.

Floyd distingue tres categorías de paradigmas:

- Los que soportan técnicas de programación de bajo nivel
- Los que soportan métodos de diseño de algoritmos
- Los que soportan soluciones de programación de alto nivel

01/11/02

IAGP

16

Metodologías usadas en ingeniería del software

Los que soportan la programación de alto nivel se agrupan en:

- Solución procedimental u operacional. Describe etapa a etapa cómo obtener la solución
- Solución demostrativa. Especifica la solución describiendo ejemplos (es una variante de la anterior)
- Solución declarativa. Indica las características que debe tener la solución, sin describir cómo procesarla

Paradigmas procedimentales u operacionales

Su característica es la secuencia computacional realizada etapa a etapa para resolver el problema.

La mayor dificultad es saber si el valor computado es correcto.

01/11/02

IAGP

17

Metodologías usadas en ingeniería del software

Son de dos tipos:

- Con efecto de lado. Modifican continuamente la representación de sus datos. Hay dos tipos:
 - Imperativo
 - Orientado a objetos
- Sin efecto de lado. Actúan creando nuevos datos continuamente.

Los paradigmas procedimentales definen explícitamente la secuencia, que se puede procesar en serie o paralelo. A su vez, este último puede ser síncrono (procesos simples aplicados a muchos objetos simultáneamente) o asíncrono (cooperación de procesos paralelos).

01/11/02

IAGP

18

Metodologías usadas en ingeniería del software

1.13 Paradigmas declarativos

El programa se **construye** indicando hechos, reglas, restricciones, ecuaciones, transformaciones y otras propiedades derivadas del conjunto de valores que configuran la solución.

No hay una descripción de las etapas a seguir para lograr una solución.

Eliminan la necesidad de probar que el valor calculado es la solución.

Estos paradigmas y los lenguajes que los soportan suelen ser pseudodeclarativos.

Ejemplos son: funcional, lógico y de transformación.

01/11/02

IAGP

19

Metodologías usadas en ingeniería del software

1.14 Paradigmas demostrativos

El programador **no especifica** los procedimientos para construir una solución. Se presentan soluciones de problemas similares, que posibilitan al sistema que generalice una solución.

Otra solución es la programación asistida, el sistema observa acciones que el programador ejecuta y si son semejantes a acciones pasadas, intentará inferir cuál es la próxima acción.

Las principales objeciones son:

- Si no se verifican exhaustivamente pueden producir programas erróneos
- La capacidad de inferencia es muy limitada
- Es difícil verificar cuando un programa es correcto

01/11/02

IAGP

20

Metodologías usadas en ingeniería del software

1.15 Paradigma imperativo

Se caracteriza por un **modelo** abstracto de ordenador, que consiste en un **gran almacenamiento** de memoria.

Este paradigma viene bien representado mediante la arquitectura Von Neumann, utiliza su modelo de máquina para conceptualizar las soluciones.

Existe un programa en memoria que se va ejecutando secuencialmente y que toma unos datos de la memoria, efectúa cálculos y actualiza la memoria.

La programación mediante este paradigma consiste en:

determinar qué datos son requeridos para el cálculo, asociarlos a una dirección de memoria y efectuar paso a paso una secuencia de transformaciones.

01/11/02

IAGP

21

Metodologías usadas en ingeniería del software

Las características fundamentales son:

- Celda de memoria para almacenar valores
- Operaciones de asignación
- Repetición

Es el conocido como paradigma algorítmico.

Algoritmo es un conjunto finito de reglas diseñadas para crear una secuencia de operaciones para resolver un tipo específico de problemas.

Según Niclus Wirth, un programa viene definido por:

Algoritmos + Estructura de datos = Programa

01/11/02

IAGP

22

Metodologías usadas en ingeniería del software

Tipos de lenguajes imperativos:

- Orientados a expresiones. Predominan las expresiones. Como C, FORTRAN, Algol y Pascal.
- Orientados a sentencias. Predominan las sentencias. Como COBOL y PL1.

Biografía de Von Neumann



01/11/02

IAGP

23

Metodologías usadas en ingeniería del software

Matemático estadounidense nacido en 1903 en Budapest (Hungría), estudió en Zurich y en las universidades de Berlín y Budapest. Desarrolló la rama de las matemáticas conocida como teoría de juegos.

- Ø Viajó a Estados Unidos en 1930 para unirse al claustro de la Universidad de Princeton.
- Ø A partir de 1933 se incorporó al Instituto de Estudios Avanzados de Princeton (Nueva Jersey).
- Ø Adquirió la nacionalidad estadounidense en 1937 y durante la II Guerra Mundial ejerció como asesor en el proyecto de la bomba atómica de Los Álamos.
- Ø En marzo de 1955 fue nombrado miembro de la Comisión de Energía Atómica de los Estados Unidos.

01/11/02

IAGP

24

Metodologías usadas en ingeniería del software

Von Neumann fue un gran matemático. Destacó por sus aportaciones fundamentales a la **teoría cuántica**, especialmente el concepto de anillos de operadores (actualmente conocido como álgebra de Neumann) y también por su trabajo de iniciación de las matemáticas aplicadas, principalmente la **estadística** y el análisis numérico.

También es conocido por el diseño de computadoras electrónicas de gran velocidad

En 1952 planificó la primera computadora que utilizaba un programa archivado flexible, el MANIAC I.

En 1956, la Comisión de Energía Atómica le concedió el premio Enrico Fermi por sus aportaciones a la teoría y diseño de los ordenadores electrónicos.

01/11/02

IAGP

25

Metodologías usadas en ingeniería del software

1. 15 Paradigma funcional

Se basa en el modelo **matemático** de composición funcional. El resultado de un cálculo es la entrada del siguiente ya sí sucesivamente.

Las funciones se pueden tratar como datos y es posible su uso como parámetros, realizar cálculos y ser devueltas como valores normales.

Un lenguaje que soporta este paradigma es el COMMONLISP. Es un lenguaje funcional procedimental.

El lenguaje LISP (LIST Processing) fue diseñado en 1959 por John McCarthy, en el Instituto Tecnológico de Massachusetts.

01/11/02

IAGP

26

Metodologías usadas en ingeniería del software

John McCarthy, profesor Senior de Hoover Institution, nació en Boston in 1927. Estudió matemáticas en el California Instituto de Tecnología y en 1951 en la universidad Princeton.



Fue profesor de Ciencias de la Computación en la universidad de Stanford desde 1962 y director del laboratorio de Inteligencia Artificial desde 1965 a 1980. Es también profesor Charles M. Pigott en la escuela de ingeniería.

Fue uno de los fundadores en 1948 de la inteligencia artificial y a partir de 1958 trabajó en problemas epistemológicos.

Empleó la lógica matemática para probar la corrección de los programas