¿Qué es la Ingeniería de Software?

> *"El establecimiento y uso de los principios de ingeniería robustos, orientados a obtener software económico que sea fiable y que funcione de manera eficiente sobre máquinas reales".*
>
> *Fritz Bauer*

Todos aquellos que pertenecemos a este mundo informatizado, intercomunicado, computarizado, y programable, que ya tenemos algunas décadas encima, podemos hacer memoria y recordar cómo se desarrollaba el software en sus comienzos, codificando, sin documentar, y es de allí, de este ensayo y error de donde nació la ingeniería de software.

La ingeniería de software es la aplicación de métodos y técnicas para resolver problemas con la ayuda de la informática, la cual proporciona las herramientas necesarias para definir adecuadamente el producto software. Debemos considerar el producto software no solo al programa como tal, ya que el producto se compone de:

- Software: - Fuentes.
  - Ejecutable.
- Documentación: - Manual Técnico.
  - Manual de Usuario.

El objetivo de la ingeniería de software no es necesariamente alcanzar la calidad perfecta, más bien busca la suficiente y necesaria para entregar un producto que satisfaga cada contexto de uso por parte de los usuarios.

William Mendoza Rodríguez
(57) 310 877 81 03 - (57) 316 557 65 30
Tel: (57) (1) 4079723 Bogotá, D.C.
willymen@porcomputador.net

Dentro de los objetivos de la ingeniería de software encontramos:

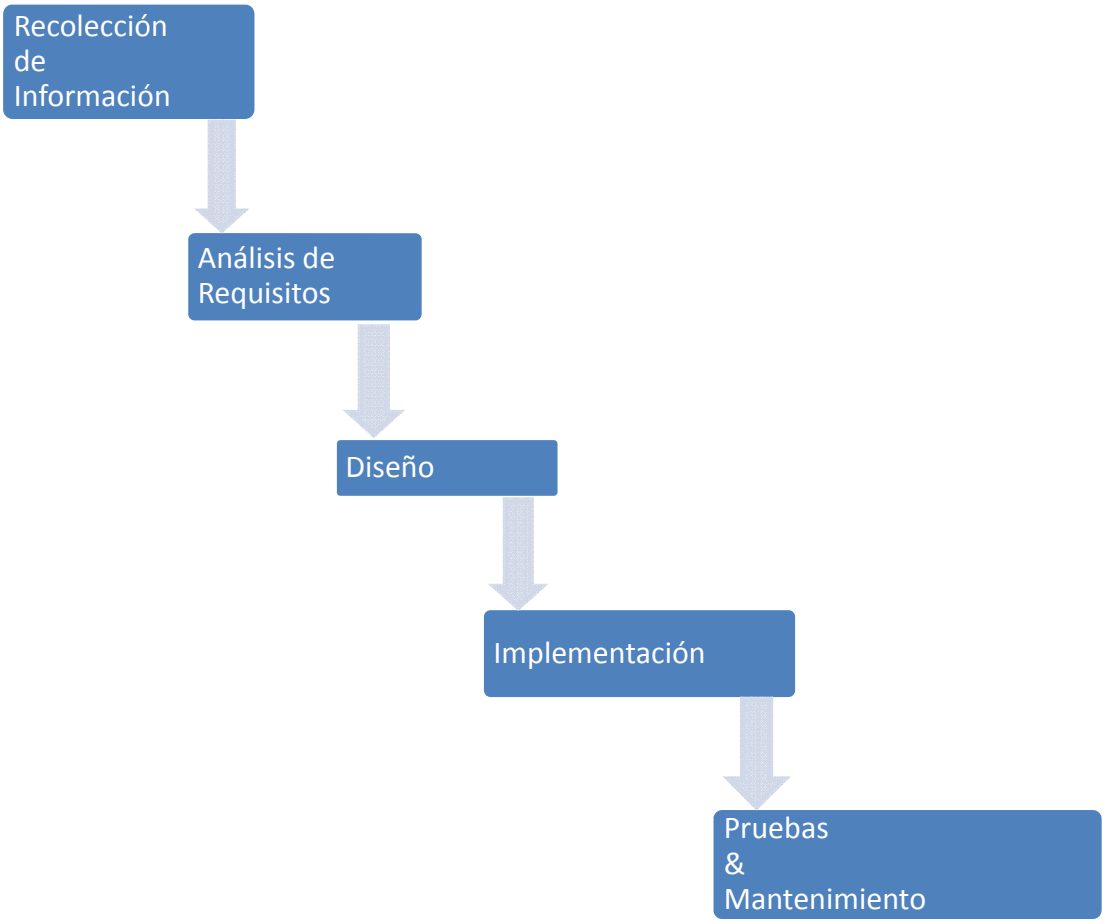
- Facilitar el control del proceso de desarrollo del producto software.
- Aumentar la productividad y trabajo de los ingenieros de software (Desarrolladores).
- Suministrar a los desarrolladores las bases necesarias para construir productos de alta calidad en una forma eficiente.
- Mejorar la calidad de los productos software.
- Definir una disciplina que garantice la producción y el mantenimiento de los productos.

En sus comienzos el código era desarrollado por unos pocos y a la medida, para sistemas específicos, poco se pensaba en la reutilización del código, y cuando era necesario hacer una modificación a este, muchas veces el coste de esta modificación era mayor a tener que realizarlo todo nuevamente, ya sea económicamente o en cuestión de esfuerzo y tiempo de dedicación. Si comparáramos el desarrollo de software con la fabricación de hardware, podremos encontrar muchas veces, desde hace ya unas décadas que es más fácil conseguir en el mercado microchips que desarrollan una tarea específica, pero aún es muy difícil conseguir un componente software de esta manera, no tenemos un catalogo, o centro comercial de software reutilizable .

Hoy en día se busca que los costes del software no sean altos, y que realmente satisfagan con las necesidades esperadas. Pero como podremos saber que el software que realicemos realmente cumple con los requerimientos esperados para suplir las necesidades existentes, en el momento indicado.

Analicemos un poco el ciclo de vida tradicional, para los proyectos de Software y de esta forma mirar que podría estar mal para corregir estos posibles errores, y mejorar la calidad del software que se fabrica en la actualidad:

William Mendoza Rodríguez
(57) 310 877 81 03 - (57) 316 557 65 30
Tel: (57) (1) 4079723 Bogotá, D.C.
willymen@porcomputador.net



Este es el ciclo de vida clásico (Cascada), pero en la práctica se han generado muchos inconvenientes, por lo cual se ha seguido últimamente el ciclo de vida iterativo incremental, donde se divide el problema en varios sub problemas, así como se realiza en programación (Programas, en subprogramas), y a cada uno de estos "pequeños problemas", aplicamos el ciclo de vida clásico, para así ir verificando cada una de las pequeñas partes que componen nuestro gran sistema, implementando poco a poco, dependiendo de los diferentes ciclos de desarrollo.

Existen otros modelos como, el evolutivo, el de espiral, entre otros, que no abordaremos precisamente en este documento.

William Mendoza Rodríguez
(57) 310 877 81 03 - (57) 316 557 65 30
Tel: (57) (1) 4079723 Bogotá, D.C.
willymen@porcomputador.net

Recolección de Información

(¿Qué?, Qué es lo que quiere el usuario.)

Considero que es una de las etapas más importantes del proyecto, por no decir que es la más importante, ya que en esta fase es donde se encuentran, definen, y acuerdan con todos los actores involucrados en el desarrollo del sistema todos los requerimientos o necesidades que suplirá el nuevo software. Abordaremos más profundamente este tema, más adelante, pues es el objetivo del curso.

Análisis de Requisitos del Software

(¿Qué? Qué es lo que yo le entendí al usuario)

Basados en una buena recolección de la información (Levantamiento de Requerimientos) necesaria para desarrollar el software, el analista entrara a interpretar hasta llegar a comprender el ámbito del sistema interconectando las necesidades, prioridades, funcionalidades, actores, requeridas para el desarrollo.

Diseño

(¿El Cómo? Como desarrollare el software)

"*El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación*", aquí se define la arquitectura del software, a nivel de persistencia, aplicación, presentación (GUI *Graphics User Interface*) la abstracción de los datos (paradigmas de programación, estructuras, objetos, etc.)

William Mendoza Rodríguez
(57) 310 877 81 03 - (57) 316 557 65 30
Tel: (57) (1) 4079723 Bogotá, D.C.
willymen@porcomputador.net

Implementación

(Lo Hagó)

Depende mucho de un buen diseño, cuando este se ha realizado efectiva y detalladamente, la implementación prácticamente se podrá realizar de manera mecánica, es la codificación por medio de un lenguaje de programación adecuado.

Pruebas & …

(Verificación de la calidad del software desarrollado).

Una vez que tenemos nuestro producto software debemos realizar pruebas para comprobar que cumple con los mínimos estándares de calidad del software de hoy como son:

Functionality	Funcionalidad
Usability	Usabilidad
Reliability	Fiabilidad
Performance	Desempeño
Supportability	Soporte
+ (Portability)	Portabilidad

Claro que estos patrones se deben contemplar desde el análisis y diseño, optimizando así los tiempos y costes del proyecto

…Mantenimiento.

(Servicio postventa)

Es normal que un software sufra cambios, después de haberse entregado al cliente final, realizando las correcciones (*Mantenimiento correctivo*) o adaptaciones (*Mantenimiento adaptativo*) necesarias en su momento. Se podrían

William Mendoza Rodríguez
(57) 310 877 81 03 - (57) 316 557 65 30
Tel: (57) (1) 4079723 Bogotá, D.C.
willymen@porcomputador.net

realizar diferentes técnicas para abordar este mantenimiento, para que el coste dentro del ciclo de vida sea menor. Estas soluciones podrían ser de tres tipos:

1. Ingeniería inversa.

   Análisis de un sistema para identificar sus componentes y las relaciones entre ellos, así como para crear representaciones del sistema en otra forma o en un nivel de abstracción más elevado. Utilizado para sistemas desarrollados por otras personas o compañías, en lenguajes de programación de generaciones anteriores, para desarrollar nuevos módulos o sistemas para acoplar en el ya existente.

2. Reingeniería:

   Modificación de un producto software, o de ciertos componentes, usando para el análisis del sistema existente técnicas de ingeniería inversa y, para la etapa de reconstrucción, herramientas de ingeniería directa, de tal manera que se oriente este cambio hacia mayores niveles de facilidad en cuanto a mantenimiento, reutilización, comprensión o evolución. Desarrollo de nuevas funcionalidades a un sistema ya existente bajo un lenguaje de programación que permita ingresar los nuevos módulos.

3. Reestructuración del software:

   Cambio de representación de un producto software, pero dentro del mismo nivel de abstracción.

Con estas técnicas se busca proporcionar los métodos necesarios para reconstruir el software, ya sea RE-programándolo, RE-documentándolo, (o en el peor de los casos documentándolo), RE-diseñándolo, o RE-haciendo alguna(s) de sus funcionalidad (es).

William Mendoza Rodríguez
(57) 310 877 81 03 - (57) 316 557 65 30
Tel: (57) (1) 4079723 Bogotá, D.C.
willymen@porcomputador.net

Esta curso como tal busca el estudio de la ingeniería de requerimientos, ya que se ha encontrado que del 40 al 60 % de los proyectos de software fracasan, por un estudio inadecuado de los requerimientos de los productos software.

Bibliografía:

- Roger S. Pressman. “Ingeniería de Software: Un Enfoque Práctico” Tercera Edicion. McGraw Hill
- Karl E. Wiegers “Software Reequirements” Primera Edición. Microsoft Press
- Larman “UML y Patrones” Primera Edición. Pearson
- http://www.monografias.com/trabajos5/desof/desof.shtml
- http://www.monografias.com/trabajos6/resof/resof.shtml
- http://www.monografias.com/trabajos5/inso/inso.shtml
- http://cnx.org/content/m17431/latest/

William Mendoza Rodríguez
(57) 310 877 81 03 - (57) 316 557 65 30
Tel: (57) (1) 4079723 Bogotá, D.C.
willymen@porcomputador.net