Gestión de Recursos para la Información.

Software e Ingeniería del Software.

J. S. Sánchez

Características del Software.

- **El software se desarrolla, no se fabrica en un sentido clásico.**
- **El software no se "estropea".**
- **La mayoría del software se construye a medida, en vez de ensamblar componentes existentes.**

Crisis del Software.

- **La planificación y estimación de costes son frecuentemente muy imprecisas.**
- **La productividad de software no se corresponde con su demanda.**
- **No se tiene tiempo de recoger datos sobre el proceso de desarrollo del software ⇒ sindatos históricos como guía, la estimación no podrá ser buena.**

- **La insatisfacción del cliente con el software terminado se produce con demasiada frecuencia.**
- **La calidad del software es, en general, cuestionable.**
- **El software existente puede ser muy difícil de mantener ⇒ el mantenimiento se lleva la mayor parte de la inversión en software.**

Mitos del Software.

- **Mitos de Gestión.**

1.- "Nuestra gente dispone de las herramientas de desarrollo de software más avanzadas; después de todo, les compramos los ordenadores más nuevos."

2.- "Si fallamos en la planificación, podemos añadir más programadores y adelantar el tiempo perdido."

•Mitos del Cliente.

1.- "Una declaración general de los objetivos es suficiente para empezar a escribir los programas; podemos dar los detalles más adelante."

2.- "Los requisitos del proyecto cambian continuamente, pero los cambios pueden acomodarse fácilmente."

•Mitos de los Desarrolladores.

1.- "Una vez que escribimos el programa y hacemos que funcione, nuestro trabajo ha terminado."

2.- "Haste que no tenga el programa ejecutándose, realmente no tengo forma de comprobar su calidad."

3.- "Lo único que se entrega al terminar el proyecto es el programa funcionando."

Ingeniería del Software (IS).

- **Métodos: indican "cómo" construir técnicamente el software. Abarcan diversas tareas, tales como planificación y estimación de proyectos, análisis de los requisitos del sistema y del software, diseño de estructuras de datos, procedimientos algorítmicos, codificación, prueba y mantenimiento.**

- **Herramientas: suministran un soporte automático o semiautomático para los métodos. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser usada por otra, se establece un sistema para el soporte del desarrollo del software llamado CASE.**
- **Procedimientos: permiten conjuntar los métodos y las herramientas, facilitando un desarrollo racional del software.**

Paradigmas de la IS.

- **La IS está compuesta por una serie de pasos que abarcan los métodos, las herramientas y los procedimientos: paradigmas.**
- **Tres tipos básicos de paradigmas:**
  - ⇒ el ciclo de vida clásico o modelo en cascada.
  - ⇒ construcción de prototipos.
  - ⇒ el modelo en espiral.

El ciclo de vida clásico.

- **Ingeniería y análisis del sistema.**
- **Análisis de los requisitos del software.**
- **Diseño.**
- **Codificación.**
- **Prueba.**
- **Mantenimiento.**

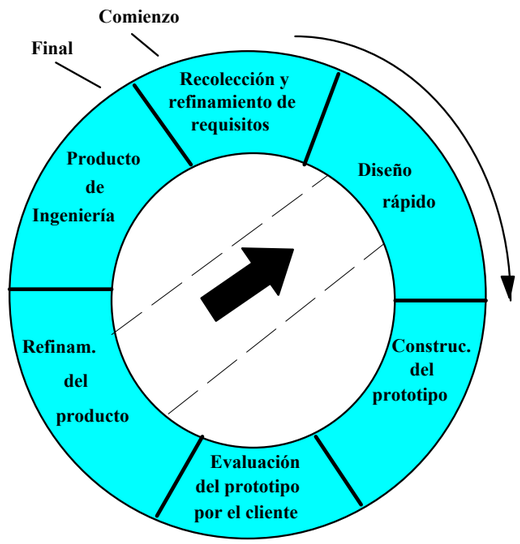
Inconvenientes:

⇒Los proyectos reales casi nunca siguen el flujo secuencial propuesto por este paradigma. Siempre hay iteraciones.

⇒Normalmente, es difícil para el cliente establecer al principio todos los requisitos, tal como lo requiere este paradigma.

⇒El cliente debe tener paciencia. Hasta llegar a las etapas finales del proyecto, no estará disponible una versión operativa del programa.

Construcción de prototipos.



Inconvenientes:

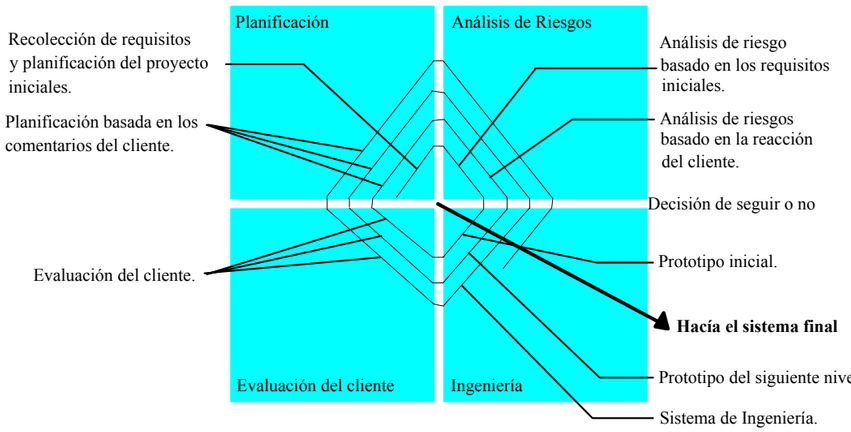
⇒El cliente ve funcionando lo que parece ser una primera versión del software solicitado, ignorando que, por las prisas en hacer que funcione, no se han contemplado aspectos de calidad o de mantenimiento del software a largo plazo. Cuando se le informa de que el producto debe ser reconstruido, el cliente se vuelve "loco" y solicita que se hagan las mejoras necesarias para hacer del prototipo un producto final que funcione, y el gestor de proyectos cede demasiado a menudo.

⇒El desarrollador impone ciertas elecciones de implementación (un sistema operativo o un lenguaje de programación inapropiado, simplemente porque ya está disponible y es conocido) con el fin de obtener un prototipo que funcione rápidamente. Después de algún tiempo, el técnico puede haberse familiarizado con estas elecciones y haber olvidado las razones por las que eran inapropiadas. La opción menos apropiada forma ahora parte integral del sistema.

Modelo en espiral.

Determinar objetivos, alternativas y restricciones.

Análisis de alternativas e identificación/resolución de riesgos.



Valoración de los resultados de la Ingeniería.

Desarrollo del producto de "siguiente nivel".

El modelo define 4 actividades principales (los 4 cuadrantes de la figura anterior):

- Planificación: determinar objetivos, alternativas y restricciones.
- Análisis de riesgo: análisis de alternativas e identificación/resolución de riesgos.
- Ingeniería: desarrollo del producto de "siguiente nivel".
- Evaluación del cliente: valoración de los resultados de la ingeniería.

Ventajas:

⇒Utiliza un enfoque evolutivo para la ingeniería del software, permitiendo reaccionar a los riesgos en cada nivel de la evolución del producto.

⇒Permite aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución del producto, utilizándolo como un mecanismo de reducción de riesgos.

⇒Mantiene el enfoque sistemático correspondiente a los pasos sugeridos por el ciclo de vida clásico, pero incorporándolo dentro de un marco de trabajo interactivo.

⇒Demanda una atención directa de riesgos técnicos en todas las etapas del proyecto y, si se aplica correctamente, debe reducir los riesgos antes de que se conviertan en problemáticos.

Inconvenientes:

⇒Puede resultar difícil convencer al cliente de que el enfoque evolutivo es el más apropiado.

⇒Requiere una considerable habilidad para la valoración del riesgo. Si no se descubre a tiempo un riesgo importante, pueden surgir graves problemas.

⇒El modelo en si mismo es bastante nuevo y no se ha utilizado tanto como los otros dos paradigmas, de modo que todavía falta tiempo para poder afirmar con total certeza su eficacia.