**Modelo de proceso**

Es una estrategia de desarrollo que se complementa de las herramientas y de los métodos usados.

**¿Cómo se selecciona?**

Para seleccionar un proceso adecuado al sistema que se va a desarrollar, debemos de tomar en cuenta los siguientes puntos principales:

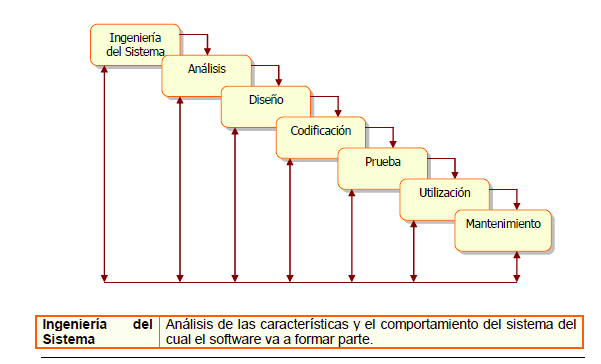
**La naturaleza del producto:**

* Aplicación que se necesite
* Métodos y herramientas
* Controles y entregas

**Modelo Lineal**

**Etapas:**

* Plan Operativo (definir el problema)
* Especificación de requerimientos (punto de vista de los desarrolladores y usuarios)
* Especificación Funcional (información del software que se va a trabajar)
* Diseño (satisfacer los requerimientos)
* Implementación (Codificación)
* Integración
* Validación y verificación
* Manutención



**Modelo Cascada**

La diferencia está en que el modelo cascada posee retroalimentación entre etapas para corregir errores.

**Etapas:**

* Ingenería y análisis del sistema
* Análisis de requisitos
* Diseño
* Codificación
* Pruebas
* Mantenimiento

**Ventajas:**

* Demanda un marco de trabajo claro que define las actividades involucradas en el software.
* Métodos mas utilizados en software y han sido exitosos.

**Desventajas:**

* Asumen que el diseñador puede distinguir entre lo que el sistema debe de hacer y cómo el sistema lo hará.
* Exceso de documentos.

**¿Por qué no utilizar el modelo?**

No recomiendan utilizar este tipo de paquetes en las empresas que necesiten un paquete de software compuesto de muchas piezas porque no es precisamente el mejor modelo para desarrollar interactividad y crear estructuras completas de software.

**Modelo Prototipo**

* Investigación preliminar
* Análisis y especificación
* Diseño y construcción
* Evaluación
* Modificación
* Diseño Técnico
* Programación y prueba
* Operación y mantención

**¿Dónde se utiliza?**

Se puede utilizar con un cliente obsesionado con la presentación de su software y con una calidad visual que lo haga mucho mas fácil de utilizar.

**Ventajas:**

* Reduce la incertidumbre del riesgo y los costos.
* Incrementa la aceptación del nuevo sistema.
* Mejora la suministración del proyecto.
* La comunicación entre los desarrolles y clientes.

**Desventajas:**

* La dependecia y herramientas de software
* No es posible aplicarla en cualquier proyecto
* Modelo Evolutivo: Todos los requerimientos estan sujetos a cambios a traves del tiempo, es por eso que busca la satisfaccion del cliente lo mas rapido posible.

**Modelo Evolutivo**

**Etapas:**

* Especificacion inicial (base a los requerimientos definidos, además de la definición del problema a solucionar).
* Desarrollo del producto (énfasis en tiempo de liberación del software).
* Implementación, uso y evaluación (monitoreo de nuevos requerimientos).
* Re-especificación (esta ligada en forma de ciclo en el desarrollo del producto).

**Ventajas:**

Puede ser utilizado en sistemas interactivos pequeños, medianos o de vida corta, así como segmentos de grandes sistemas.

**Desventajas:**

* Poca visibilidad que existe en el proceso de construcción.
* Pobre especificación de requerimientos.
* Habilidades de programación muy elevadas (tiene enfoque de liberación rápida).

**Modelo Incremental**: Concepto parecido al anterior, se comienza el desarrollo del sistema para satisfacer un subconjunto de requesitos específico.

**Etapas:**

* Requerimientos
* Diseño
* Codificación y test unitario
* Integración del sistema
* Operación y mantención

**¿Dónde se utiliza?**

**Ejemplo:** Estos modelos nos permiten entregar piezas del software que sean necesarias para cubrir una actividad relativamente estable mientras se continúa perfeccionando el resto de interaciones.

**Modelo Espiral:** Modelo no secuencial y con nivel de complejidad, sin embargo, incluye un elemento muy útil e importante en el desarrollo del software, el análisis de riesgos.

**Etapas:**

* Comunicación con el cliente
* Planificación
* Análisis de riesgos
* Ingeniería
* Evaluación del cliente (aquí es el cliente que decide si está conforme con los resultados para continuar con el siguiente paso, de lo contrario, regresa a la etapa de planificación).
* Construcción y entrega

**Ventajas:**

* Prototipos para la idea de final e imagen detallada en los errores durante su funcionamiento
* Permite el cliente evaluar los factores de riesgos y con la misma información tomar decisiones.

**Desventajas:**

* Un análisis de riesgos mal hecho puede subestimar los errores del software que orilla a regresar a las etapas sin fín.

**¿Cuándo utilizarlo?**

* Un paquete de control de asignación de tarjetas de creditos a titulares de cuentas bancarias
* Protección de códigos de seguridad, etc.

**Cuadro comparativo se contrastan los diferentes criterios a la hora de escoger los métodos más adecuados para el desarrollo del sistema:**



**Modelo Scrum**

Según los expertos se considera el mejor modelo para desarrollar que los anteriores, metodología ágil de gestión de trabajo.

**Etapas:**

* Product Backlog: Es la toma de requerimientos (no es necesario tener todos los requisitos al principio, para empezar solo hace falta tener las ideas principales)
* Sprint Backlog: Se toman los requisitos que se van a desarrollar, se subdividen en tareas,
* Sprint: A diario se hace una reuion de 15 minutos donde particpa cada miembro del eqipo y cada uno se tiene que responder las siguientes preguntas: **¿qué hiciste ayer? ¿qué vas hacer hoy? ¿tienes algún impedimiento para seguir avanzando tu area?** Al final se trata de resolver todas las problemas que se presentan.
* Working increment of the software: Se le enseña al cliente todos los requisitos propuestos, el objetivo es ir construyento versions perfectamente funcionales en las que poco a poco se vayan añadiendo caracteristicas pero es fundamental que estas versiones sean funcionales para poder recibir un buen feedback del cliente.

**Ventajas:**

* Entrega de un producto funcional al finalizar cada sprint.
* Visualización del proyecto día a día.
* Equipos integrados y comprometidos con el proyecto.

**Desventajas:**

* No genera toda la evidencia de documentación como otras metodologías.
* A veces es mejor complementarlo con otros métodos.

**Conclusión:**

Se puede concluir que el método es el **Modelo Scrum**, permite una gran organización al dividir las tareas en segmentos, además, la reunión diaria fortalece el trabajo y la resolución en los problemas que pudiese tener un integrante de la unidad; un detalle no menos importante, no mide cuánto tiempo se lleva trabajando, sino en cuánto falta para terminar, que es lo que realmente importa.

Google utiliza este método.