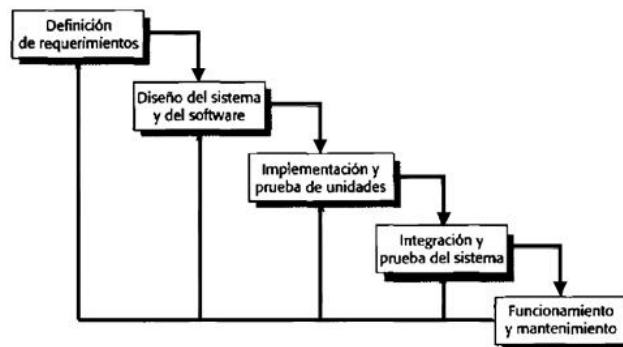


1. MODELOS DEL PROCESO SOFTWARE

El modelo de proceso de desarrollo de software es quizás la pieza más importante de este engranaje conocido como ingeniería de software. Existen varios modelos para el proceso de desarrollo software. Los modelos están conformados por etapas que son generales a todos los enfoques. Las diferencias están básicamente en los tiempos en los cuales se realizan dichas etapas, la simultaneidad, la prioridad que se le da a cada etapa, entre otros elementos. Otras características se pueden observar en detalle en la definición de cada uno de los modelos.

Figura 1. Ciclo de vida del Software



Fuente: Ian Sommerville. Ingeniería del Software, Séptima Edición

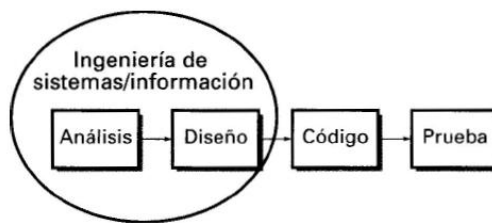
Algo importante en el análisis del proceso software es que los procesos no son excluyentes y un proceso bien puede ser la unión de varios de ellos.

1.1 **MODELO LINEAL SECUENCIAL O EN CASCADA**¹

El modelo en lineal secuencial es también conocido como el ciclo de vida del software, está conformado por las etapas de Análisis de requerimientos, Diseño, Codificación y pruebas. Cada etapa tiene dependencias de finalización y orden la una de la otra. En ese orden de ideas no se podrá iniciar la etapa de codificación sino no se ha pasado ya por la etapa de análisis y diseño previamente y respectivamente.

Debido a la forma en la que este proceso aborda la solución de proyectos software tiene algunas ventajas y falencias.

Figura 2. Modelo Lineal Secuencial



Fuente: Roger S. Pressman. Ingeniería del Software, Un enfoque práctico. Quinta edición. McGraw Hill

Entre otras las ventajas y desventajas de este modelo son:

- Los proyectos reales raras veces siguen el modelo secuencial que propone el modelo. Aunque el modelo lineal puede acoplar

¹ Roger S. Pressman. Ingeniería del Software, Un enfoque práctico. Quinta edición. McGraw Hill

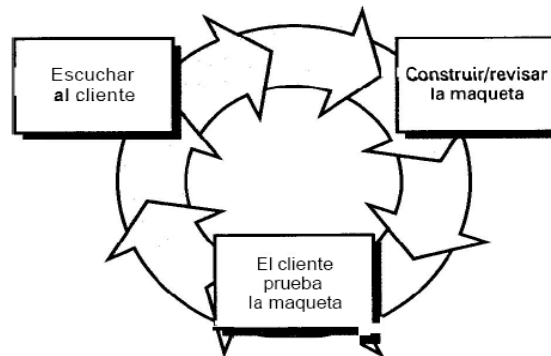
interacción, lo hace indirectamente. Como resultado, los cambios pueden causar confusión cuando el equipo del proyecto comienza.

- Este modelo es ampliamente usado en la industria del software y pese a que tiene algunas dificultades como las mencionadas arriba es mejor que no aplicar ningún modelo de proceso para el desarrollo de software.
- A menudo es difícil que el cliente exponga explícitamente todos los requisitos. El modelo lineal secuencial lo requiere y tiene dificultades a la hora de acomodar la incertidumbre natural al comienzo de muchos proyectos.
- El cliente debe tener paciencia. Una versión de trabajo del (los) programa(s) no estará disponible hasta que el proyecto esté muy avanzado. Un grave error puede ser desastroso si no se detecta hasta que se revisa el programa.
- Es difícil afrontar cambios propuestos por el cliente en fases diferentes de la fase de análisis y definición de requisitos y entre más adelantado esté el proyecto más costosos son los cambios o modificaciones.

1.2 PROCESO DE SOFTWARE POR PROTOTIPOS²

El paradigma de desarrollo por prototipos permite refinar sistemas complejos con base en un sistema mínimo definido al principio del proceso de la especificación del sistema y de los cuales el cliente no tiene la definición completa de requisitos.

Figura 3. El paradigma de construcción por prototipos



Fuente: Roger S. Pressman. Ingeniería del Software, Un enfoque práctico. Quinta edición. McGraw Hill

El proceso comienza con la definición de requisitos por parte del cliente y así se llega a los objetivos globales del sistema. Con base en los requisitos conocidos se diseña un primer acercamiento a la solución el cual se implementa como una versión funcional del software. Dicho prototipo es evaluado por el cliente y así definir nuevos alcances y nuevos

² Roger S. Pressman. Ingeniería del Software, Un enfoque práctico. Quinta edición. McGraw Hill

requisitos. Por medio de dichas iteraciones, de un prototipo a otro se logra el sistema completo.

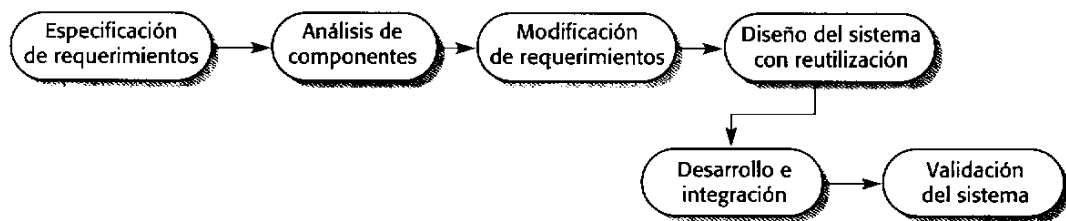
Este enfoque tiene ventajas y desventajas entre otras están:

- El cliente ve lo que parece una versión del software y no tiene en cuenta que para realizarse un prototipo rápido, se rebaja la calidad del producto software de esa manera el cliente puede no entender que se debe realizar de nuevo el sistema y querrá tener como versión final de su sistema el prototipo ya desarrollado.
- De forma demasiado frecuente la gestión de desarrollo del software es muy lenta.
- Documentar el desarrollo suele ser tedioso y puede perderse control por parte de los jefes de proyectos debido a la cantidad de versiones desarrolladas.
- Por el otro lado el paradigma por prototipos también cuenta con una aceptación amplia debido a que el cliente le gusta ver versiones inmediatas de su sistema. También suele ser muy efectivo para el manejo y la identificación de errores en etapas tempranas del proceso de desarrollo.

1.3 PROCESO SOFTWARE BASADO POR COMPONENTES³

Es una práctica habitual por parte de los desarrolladores la inclusión de códigos desarrollados previamente por el equipo de desarrollo. Esta práctica informal no depende del proceso software que se esté siguiendo en el desarrollo de un sistema de esta manera dicha práctica haya una definición formal en el procesos software basada por componentes. Aunque en la definición de requisitos y la etapa de validación es comparable con los demás modelos en las etapas intermedias los cambios son significativos.

Figura 4. El proceso software basado por componentes



Fuente: Ian Sommerville. Ingeniería del Software, Séptima Edición

Este consta de 4 etapas de desarrollo:

- **Análisis de componentes:** Con base en los requerimientos del cliente se analizan los componentes factibles de ser implementados para dicha necesidad.

³ Ian Sommerville, (2005). Ingeniería del Software. Addison Wesley. Séptima Edición

- **Modificación de componentes:** Con base en los requerimientos del cliente y los componentes encontrados en la etapa de análisis de componentes se hacen modificaciones a los componentes para hallar versiones ajustadas a las necesidades del cliente.
- **Diseño del sistema con reutilización:** Se realiza un diseño con base en los componentes que se reutilizarán. Si ciertos componentes no están disponibles de acuerdo a lo requerido por el diseño se debe diseñar nuevo software, nuevos componentes.
- **Desarrollo e Integración:** Los componentes que no existen y fueron diseñados en la etapa anterior se deben desarrollar, y luego comenzar con el proceso de integración de los componentes tanto los existentes como los nuevos componentes.

Ventajas:

- Entregas más rápidas debido a la reducción en los tiempos de desarrollo.
- Las aplicaciones son modulares permitiendo el desarrollo independiente de componentes los cuales alimentan el almacén de componentes para ser incluidos en otros proyectos.

Desventajas:

- Si la organización no tiene control sobre cada componente del almacén de componentes pierde sentido la orientación hacia este modelo.

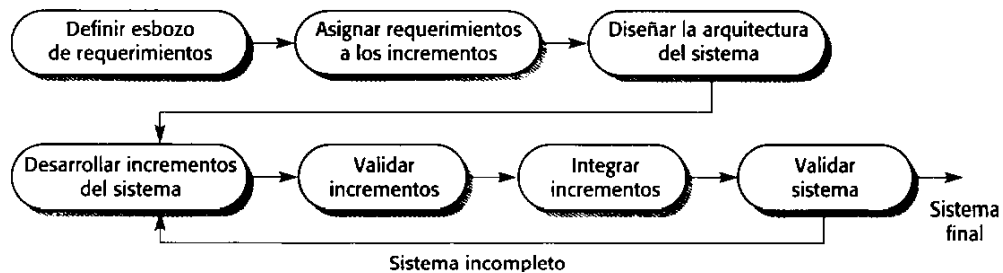
- El cliente exige tiempos basado en el hecho que ya muchas cosas están desarrolladas y no siempre los componentes existentes cumplen con las necesidades de los clientes.

1.4 PROCESO DEL SOFTWARE POR ENTREGAS INCREMENTALES⁴

Este modelo del proceso software es un proceso intermedio entre las mejores características del modelo en cascada y el modelo evolutivo. El analista comienza por realizar una definición de los requerimientos del sistema completo y luego diseña una estructura por incrementos o entregables de la aplicación. Cada incremento se desarrolla siguiendo los pasos que propone el modelo en cascada o el ciclo de vida del software. Cada incremento define o entrega un subconjunto de funcionalidad del software completo y son definidos, desarrollados y entregados con base en la prioridad del cliente. Una vez que los incrementos del sistema se han identificado, los requerimientos para los servicios que se van a entregar en el primer incremento se definen en detalle, y éste se desarrolla. Durante el desarrollo, se puede llevar a cabo un análisis adicional de requerimientos para los requerimientos posteriores, pero no se aceptan cambios en los requerimientos para el incremento actual.

⁴ Ian Sommerville, (2005). Ingeniería del Software. Addison Wesley. Séptima Edición

Figura 5. Proceso del software por entregas incrementales



Fuente: Ian Sommerville. Ingeniería del Software, Séptima Edición

Ventajas:

- Los clientes no tienen que esperar hasta que el sistema completo se entregue para sacar provecho de él. El primer incremento satisface los requerimientos más críticos de tal forma que pueden utilizar el software inmediatamente.
- Los clientes pueden utilizar los incrementos iniciales como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema.
- Existe un bajo riesgo de un fallo total del proyecto. Aunque se pueden encontrar problemas en algunos incrementos, lo normal es que el sistema se entregue de forma satisfactoria al cliente.
- Puesto que los servicios de más alta prioridad se entregan primero, y los incrementos posteriores se integran en ellos, es inevitable que los servicios más importantes del sistema sean a los que se les hagan más pruebas. Esto significa que es menos probable que los clientes

encuentren fallos de funcionamiento del software en las partes más importantes del sistema.

Desventajas:

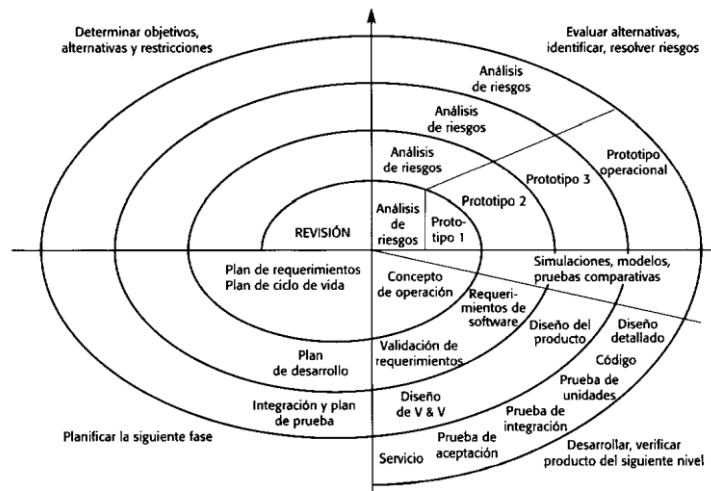
- Puede ser difícil adaptar las necesidades del cliente a incrementos de tamaño apropiado.
- Pueden existir componentes comunes a todos los incrementos y debido a que los requerimientos no se definen en detalle hasta que se implemente el incremento puede ser difícil identificar los recursos comunes a todos los incrementos.

1.5 PROCESO SOFTWARE EN ESPIRAL⁵

El modelo en espiral es una mirada diferente al proceso software. Más que representar el proceso del software como una secuencia de actividades con retrospectiva de una actividad a otra, se representa como una espiral. Cada ciclo en la espiral representa una fase del proceso del software. Así el ciclo más interno podría referirse a la viabilidad del sistema. el siguiente ciclo a la definición de requerimientos el siguiente ciclo al diseño del sistema, y así sucesivamente. Cada ciclo del espiral se divide en 4 sectores:

⁵ Ian Sommerville, (2005). Ingeniería del Software. Addison Wesley. Séptima Edición

Figura 6. Proceso del software en espiral



Fuente: Ian Sommerville. Ingeniería del Software, Séptima Edición

Definición de objetivos: Para esta fase del proyecto se definen los objetivos específicos. Se identifican las restricciones del proceso y el producto, y se traza un plan detallado de gestión. Se identifican los riesgos del proyecto. Dependiendo de estos riesgos, se planean estrategias alternativas.

Evaluación y reducción de riesgos: Se lleva a cabo un análisis detallado para cada uno de los riesgos del proyecto identificados. Se definen los pasos para reducir dichos riesgos. Por ejemplo, si existe el riesgo de tener requerimientos inapropiados se puede desarrollar un prototipo del sistema.

Desarrollo y Validación: Después de la evaluación de riesgos se elige un modelo para el desarrollo del sistema. Por ejemplo si los riesgos en la interfaz de usuario son dominantes un modelo de desarrollo apropiado podría ser la construcción de prototipos evolutivos. Si los riesgos de

seguridad son la principal consideración, un desarrollo basado en transformaciones formales podría ser el más apropiado, y así sucesivamente. El modelo en cascada puede ser el más apropiado para el desarrollo si el mayor riesgo identificado es la integración de los subsistemas.

Planificación: El proyecto se revisa y se toma la decisión de si se debe continuar con un ciclo posterior de la espiral. Si se decide continuar. Se desarrollan los planes para la siguiente fase del proyecto.

La diferencia principal entre el modelo en espiral y los otros modelos del proceso del software es la consideración explícita del riesgo en el modelo en espiral. Informalmente, el riesgo significa sencillamente algo que puede ir mal.

Desventajas:

- Se requiere una considerable habilidad para evaluar riesgos, si un riesgo importante no es considerado podría traer consecuencias graves para el buen desarrollo del proyecto.

Ventajas:

- Permite un alto control de las actividades que serán desarrolladas a la vuelta de cada ciclo del espiral.
- Administración y control de riesgos en medio del proceso de desarrollo. Es explícitamente definido para cada etapa.