

Ciclos, Procesos y Metodologías de Desarrollo de Software

Análisis y Diseño de Sistemas de Información

UNIDAD 2

Desarrollo de un Sistema de Información

Desarrollo de un Sistema de Información

- Desarrollo de software para un sistema de información
- Para desarrollar de manera adecuada cualquier tipo de software es necesario realizar una gran cantidad de actividades,
- De éstas, la programación es sólo una más y en muchas ocasiones no la más importante
- A este conjunto de etapas y procesos necesarios para un buen desarrollo de software se le denomina ciclo de desarrollo de software

Definiciones

- La norma IEEE 1074 define el ciclo de vida del desarrollo de sistemas como *“Una aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software”*
- La norma ISO 12207-1 tiene la siguiente definición *“Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso”*

Procesos de Ciclo de Vida

Procesos del Ciclo de Vida del Desarrollo de Software

- Procesos de Desarrollo
- Procesos Principales
- Procesos de Soporte
- Procesos de la Organización

Procesos Principales

- Son aquellos que resultan útiles a las personas que inician o realizan el desarrollo, la operación o el mantenimiento del software durante su ciclo de vida

Procesos Principales

- Procesos de adquisición. Actividades y tareas que el comprador, el cliente o el usuario realiza para adquirir un sistema o un producto de software
- Proceso de suministro. Se inicia con la decisión de preparar una propuesta para responder a la petición de un usuario
- Proceso de desarrollo. Implica el análisis de requisitos, diseño, codificación, integración, pruebas, instalación y aceptación

Procesos Principales

- Proceso de explotación. Incluye la explotación del software y el soporte operativo a los usuarios
- Proceso de mantenimiento. Aparece cuando el software necesita modificaciones, ya sea en el código o en la documentación asociada, debido a un error, una deficiencia, un problema o la necesidad de mejora o adaptación

Procesos de Soporte

- Se utilizan para llevar a cabo funciones tales como la gestión, la formación del personal o la mejora del proceso

Procesos de Soporte

- Proceso de documentación. Registra la información producida por un proceso o actividad del ciclo de vida
- Proceso de gestión de la configuración. Este proceso aplica ciertos procedimientos administrativos y técnicos durante todo el ciclo de vida del sistema
- Proceso de aseguramiento de la calidad. Aporta una confianza en la que los procesos y los productos de software del ciclo de vida cumplen con los requisitos especificados y se ajustan a los planes establecidos

Procesos de Soporte

- Proceso de verificación. Determina si los requisitos de un sistema o del software están completos y son correctos
- Proceso de validación. Determina si el sistema o software final cumplen con los requisitos previstos para su uso
- Proceso de revisión conjunta. Sirve para evaluar el estado del software y sus productos en una actividad del ciclo de vida o una fase de un proyecto

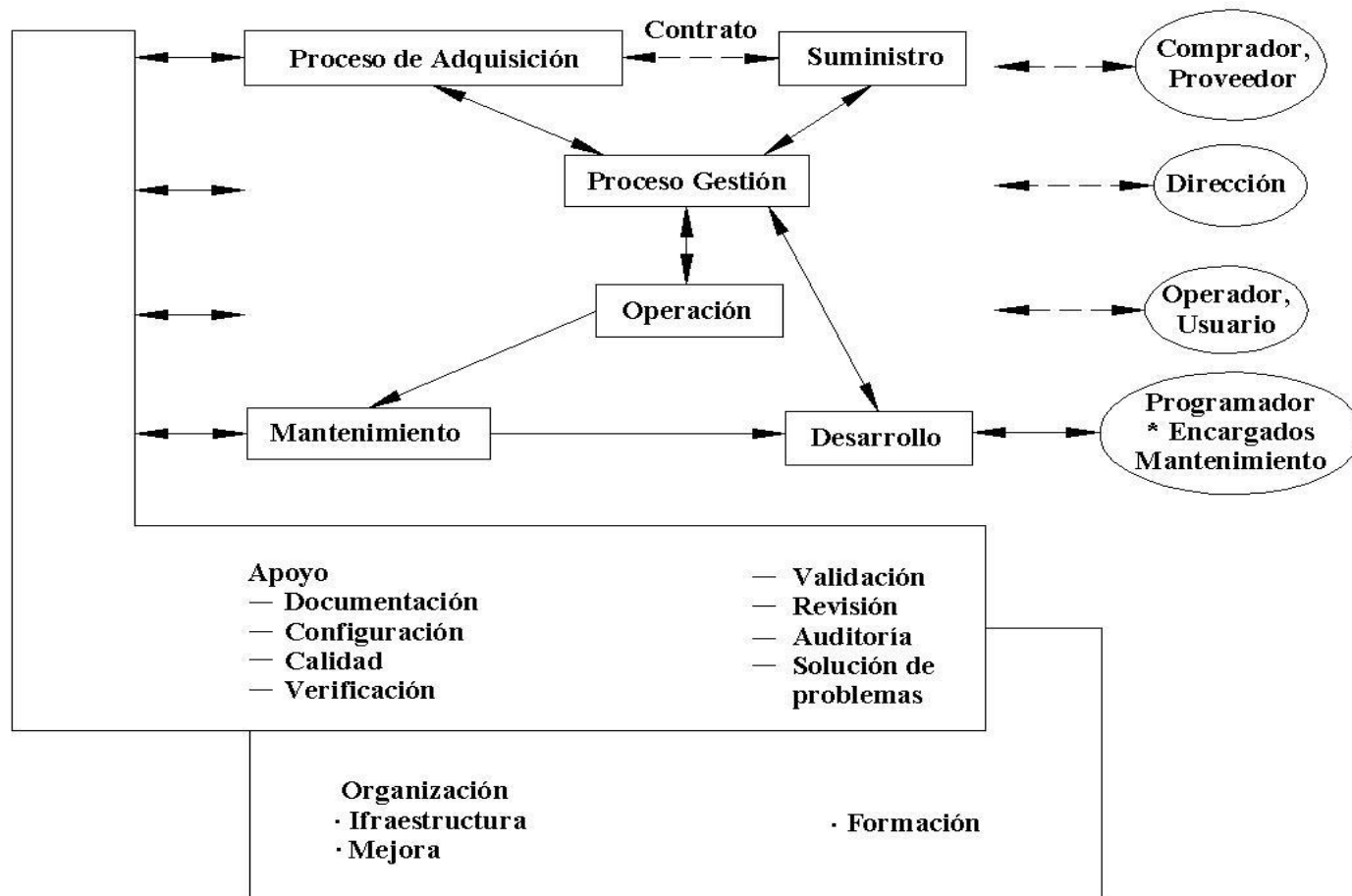
Procesos de Soporte

- Proceso de auditoría. Permite determinar si se han cumplido los requisitos, los planes y el contrato
- Proceso de resolución de problemas. Permite analizar y eliminar los problemas (por ejemplo inconformidades) descubiertos durante el desarrollo

Procesos de la Organización

- Se utilizan para llevar a cabo funciones tales como la gestión, la formación del personal o la mejora del proceso. Se llevan a cabo normalmente a nivel organizativo, fuera del ámbito de proyectos y contratos específicos.
 - Proceso de gestión
 - Proceso de infraestructura
 - Proceso de mejora
 - Proceso de formación

Interacción entre los Procesos



Procesos de Desarrollo

- Análisis de Requisitos del Sistema
- Diseño de la Arquitectura del Sistema
- Análisis de Requisitos de Software
- Diseño de la Arquitectura del Software
- Diseño Detallado de Software
- Codificación y Prueba de Software
- Integración del Software
- Pruebas del Software
- Integración del Sistema
- Prueba del Sistema

Elementos del Proceso

- Análisis de Requerimientos del Sistema. Se especifican los requisitos del sistema, incluyendo las funciones y las capacidades que debe incluir, requisitos de seguridad, de interfaces de operación y mantenimiento
- Diseño de la Arquitectura del Sistema. Se establece una arquitectura de alto nivel, la cuál identifica los principales componentes de hardware y de software

Elementos del Proceso

- Análisis de Requisitos de Software. Se documentan los requisitos del sistema especificados en el Análisis, incluyendo las especificaciones de las características de calidad que debe cumplir el sistema
- Diseño de la Arquitectura del Software. Se deben transformar los requisitos del software en una arquitectura o estructura de alto nivel que identifica sus componentes principales. También se elabora una versión preliminar de los manuales de usuario y de los requisitos que las pruebas deben cumplir

Elementos del Proceso

- Diseño Detallado de Software. Se realiza un diseño detallado para cada componente software y las bases de datos. Se actualizan los manuales de usuario y los requisitos que las pruebas deben cumplir
- Codificación y Prueba de Software. Se desarrollan y se documentan los distintos componentes de software y de las bases de datos. Se realizan pruebas de los distintos componentes y se actualizan de nueva cuenta los manuales de usuario

Elementos del Proceso

- Integración del Software. Se integran los componentes del software y se prueban según sea necesario, se actualizan nuevamente los manuales de usuario
- Pruebas del Software. El desarrollador lleva a cabo la prueba de cualificación en función de los requisitos especificados

Elementos del Proceso

- Integración del Sistema. Se integran los elementos software y hardware junto con las operaciones manuales
- Prueba del Sistema. Análoga a la del software, pero se lleva a cabo de acuerdo con los requisitos especificados para el sistema

Introducción a las Metodologías de Desarrollo

Metodologías de Desarrollo

- Definición Formal. Conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información

Definición

- Se define una metodología como el conjunto de componentes que especifican:
 - Cómo se debe dividir un proyecto en etapas
 - Qué tareas se llevan a cabo en cada etapa
 - Qué salidas se producen y cuándo se deben producir
 - Qué restricciones se aplican
 - Qué herramientas se van a utilizar

Procedimientos

- La descomposición del proceso se realiza hasta el nivel de tareas o actividades elementales
- Para cada tarea se identifica un procedimiento que define la forma de ejecutarla
- El resultado de aplicar un procedimiento, se obtienen uno o más productos
- El sistema deseado estará formado por un conjunto de productos finales

Propiedades de una Metodología

- Para aplicar un procedimiento se pueden utilizar una o más técnicas, las cuáles suelen ser, con mucha frecuencia, gráficas con apoyos textuales y formales
- Para la realización de una técnica, se utilizan como apoyo las herramientas
- Una metodología puede seguir uno o varios modelos de ciclo de vida
- El ciclo de vida indica qué es lo que hay obtener a lo largo del desarrollo del proyecto, pero no cómo

Características

- Existencia de Reglas Predefinidas. La metodología debe indicar las reglas que definan sus fases, las tareas, productos intermedios, técnicas, herramientas, ayudas al desarrollo y formatos de documentación
- Cobertura del Ciclo de Desarrollo. Debe contener los pasos a realizar desde el planteamiento de un sistemas hasta su mantenimiento, proporcionando mecanismos para integrar los resultados de una fase a la siguiente

Características

- Verificaciones Intermedias. Se debe contemplar la realización de verificaciones sobre los productos generados en cada fase para comprobar su corrección. Se realizan por medio de revisiones de software para detectar inconsistencias, inexactitudes o cualquier tipo de defecto generado durante el proceso de desarrollo

Características

- Planificación y Control. Debe proporcionar una forma de desarrollar software de manera planificada, para que no se disparen los costos ni se amplíen los tiempos de entrega
- Comunicación Efectiva. Debe proporcionar un medio de comunicación efectiva entre los desarrolladores para facilitar el trabajo en grupo y con los usuarios

Características

- Utilización sobre varios proyectos. La metodología debe ser flexible para que pueda emplearse sobre varios proyectos que pueden variar en tamaño o entorno
- Una organización no debe utilizar una metodología diferente para cada proyecto, sino que se debe amoldar a un proyecto en concreto

Características

- Fácil Formación. Debe ser sencillo para la organización capacitar al personal en todos los procedimientos definidos por la metodología
- Herramientas CASE. La metodología debe estar soportada por herramientas automatizadas que mejoren la productividad del equipo de desarrollo y la calidad de los productos resultantes

Características

- Mejoras y Calidad. La metodología debe contener actividades que mejore el proceso de desarrollo. Debe contener indicadores para evaluar la calidad y el costo asociado a cada etapa del proceso
- Soporte al Mantenimiento. La metodología debe facilitar, en mayor grado posible, las modificaciones sobre los sistemas existentes

Características

- Reutilizable. Soporte de la reutilización de software. Se deben incluir procedimientos para la creación, mantenimiento y recuperación de componentes reutilizables

Modelos de Ciclo de Vida

Definición

- Modelo que permite especificar la forma en que se llevarán a cabo los procesos, actividades y tareas

Modelo en cascada

- El número de fases o etapas que se proponen en este ciclo suele variar, aunque normalmente incluyen:
 - Análisis de requisitos del sistema
 - Análisis de requisitos del software
 - Diseño preliminar
 - Diseño detallado
 - Codificación
 - Pruebas
 - Mantenimiento

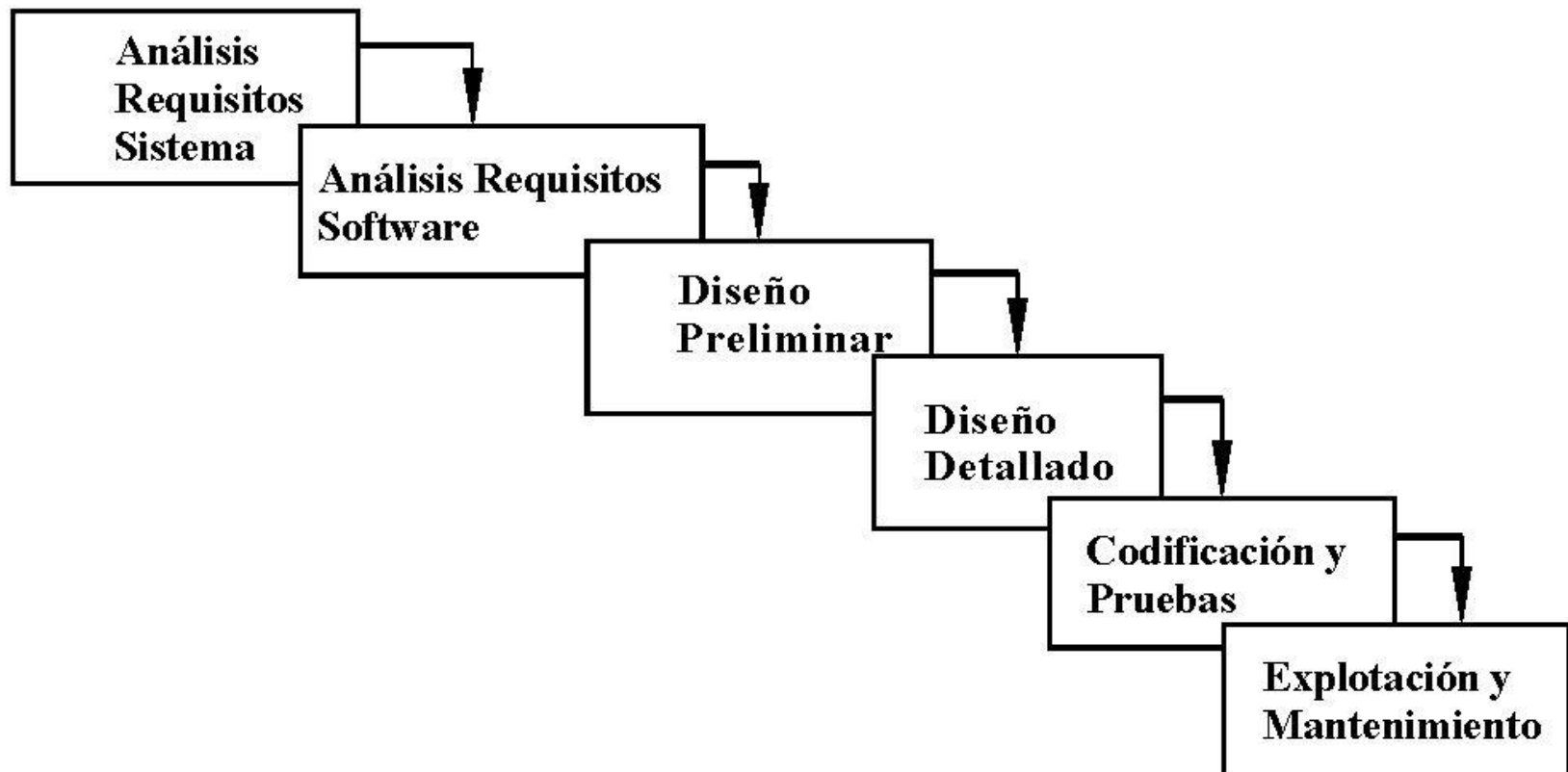
Características

- Cada fase empieza cuando se ha terminado la fase anterior
- Para pasar de una fase a otra es necesario conseguir todos los objetivos de la etapa previa
- Ayuda a prevenir que se sobrepasen las fechas de entrega y los costos esperados
- Al final de cada fase el personal técnico y los usuarios tienen la oportunidad de revisar el progreso del proyecto

Desventajas

- No refleja el proceso real de desarrollo de software. Los procesos reales rara vez siguen este flujo secuencial, puesto que siempre hay iteraciones
- Se tarda mucho tiempo en pasar por todo el ciclo, dado que hasta que no se finalice una fase no se pasa a la siguiente
- Acentúa el fracaso de la industria del software con el usuario final, ya que el sistema en funcionamiento no estará disponible hasta las fases finales del proyecto

Modelo en Cascada



Modelo Incremental

- En el modelo incremental se va creando el sistema de software añadiendo componentes funcionales al sistema (llamados incrementos)

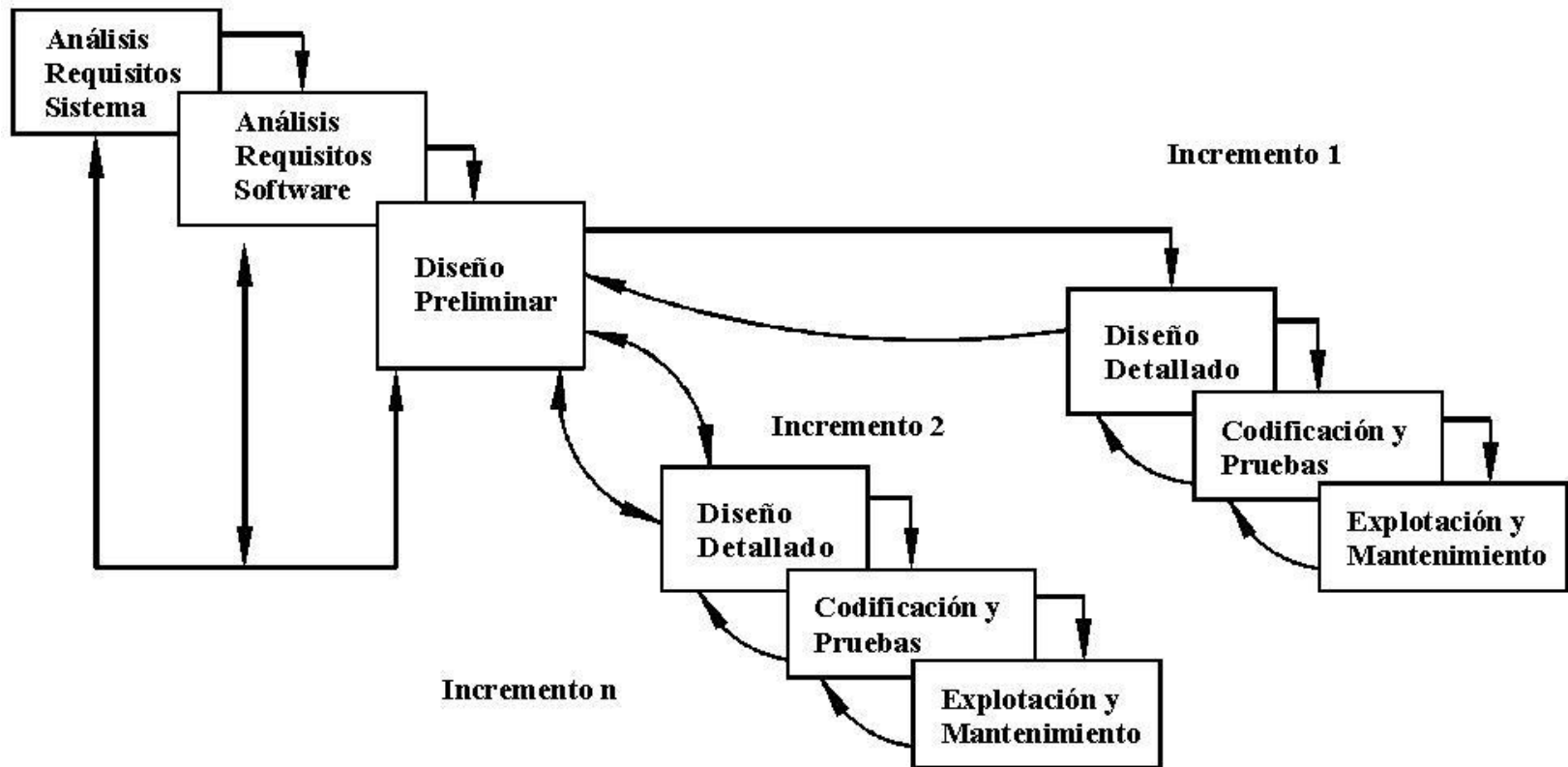
Características

- En cada paso sucesivo, se actualiza el sistema con nuevas funcionalidades o requisitos, es decir, cada versión o refinamiento parte de una versión previa y le añade nuevas funciones
- Este modelo se ajusta a entornos de alta incertidumbre, por no tener la necesidad de poseer un conjunto exhaustivo de requisitos, especificaciones, diseños, etc al comienzo del desarrollo del sistema

Desventajas

- A pesar que permite el cambio continuo de requisitos, aún existe el problema de determinar si los requisitos propuestos son válidos
- Los errores en los requisitos se detectan tarde y su corrección resulta tan costosa como en el modelo en cascada

Modelo Incremental



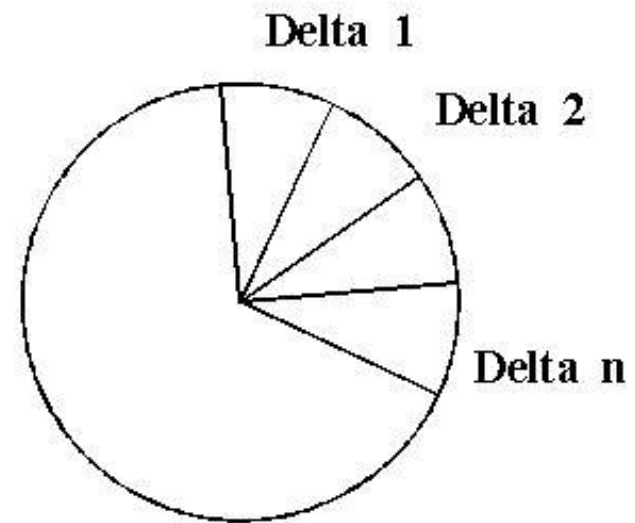
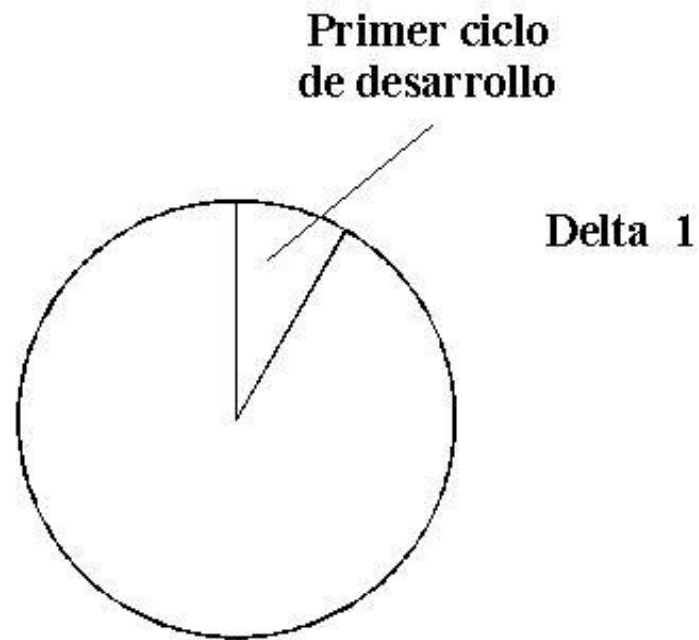
Modelo evolutivo

- Es una extensión del modelo incremental, en éste, los incrementos se realizan de manera secuencial en lugar de en paralelo
- Desde el punto de vista del cliente el sistema evoluciona según se van entregando los incrementos
- Desde el punto de vista del desarrollador, los requerimientos que son claros desde el principio dictan el incremento inicial
- Los incrementos para cada uno de los siguiente ciclos de desarrollo estarán basados en la experiencia de los incrementos anteriores

Características

- Se entrega temprano una parte del sistema, aunque no estén completos los requisitos
- Se permite entregar parte del sistema como herramienta para la generación de requerimientos faltantes
- Se obtienen beneficios para el sistema mediante entregas iniciales, mientras las entregas posteriores estén en desarrollo

Modelo Evolutivo



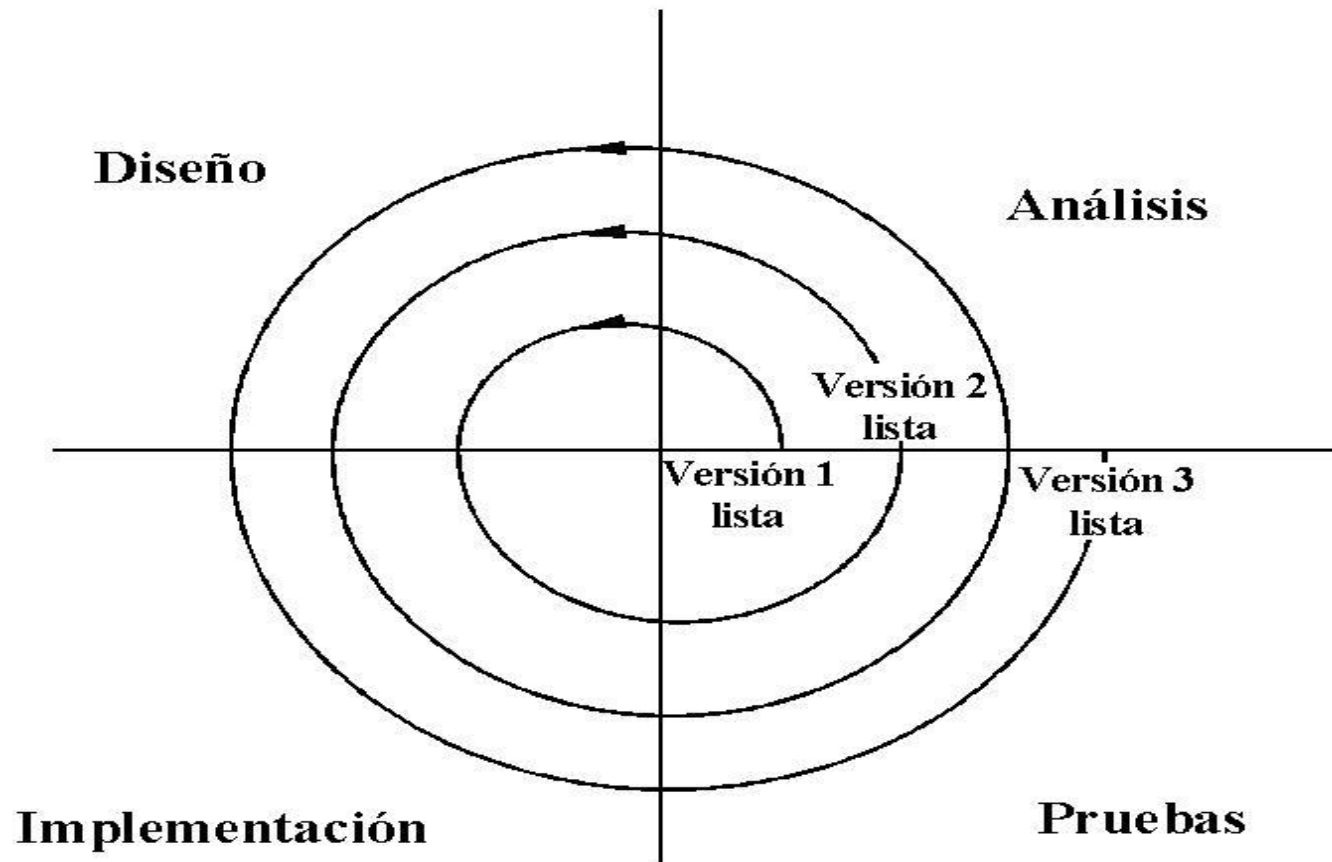
Modelo en Espiral

- El modelo de espiral se basa en una estrategia para reducir el riesgo del proyecto en áreas de incertidumbre
- Enfatiza ciclos de trabajo, cada uno de los cuales estudia el riesgo antes de proceder al siguiente ciclo
- Cada ciclo comienza con la identificación de los objetivos, soluciones alternativas, restricciones asociadas con cada alternativa y finalmente se procede a su evaluación

Propiedades

- Una actividad comienza cuando se entienden los objetivos y riesgos involucrados
- Basado en la evaluación de soluciones alternas, se usan las herramientas que mejor reduzcan los riesgos
- El desarrollo se incrementa en cada etapa, permitiendo prototipos sucesivos del producto
- Con algunas variantes, éste es el modelo de proceso más importante en la actualidad

Modelo en Espiral



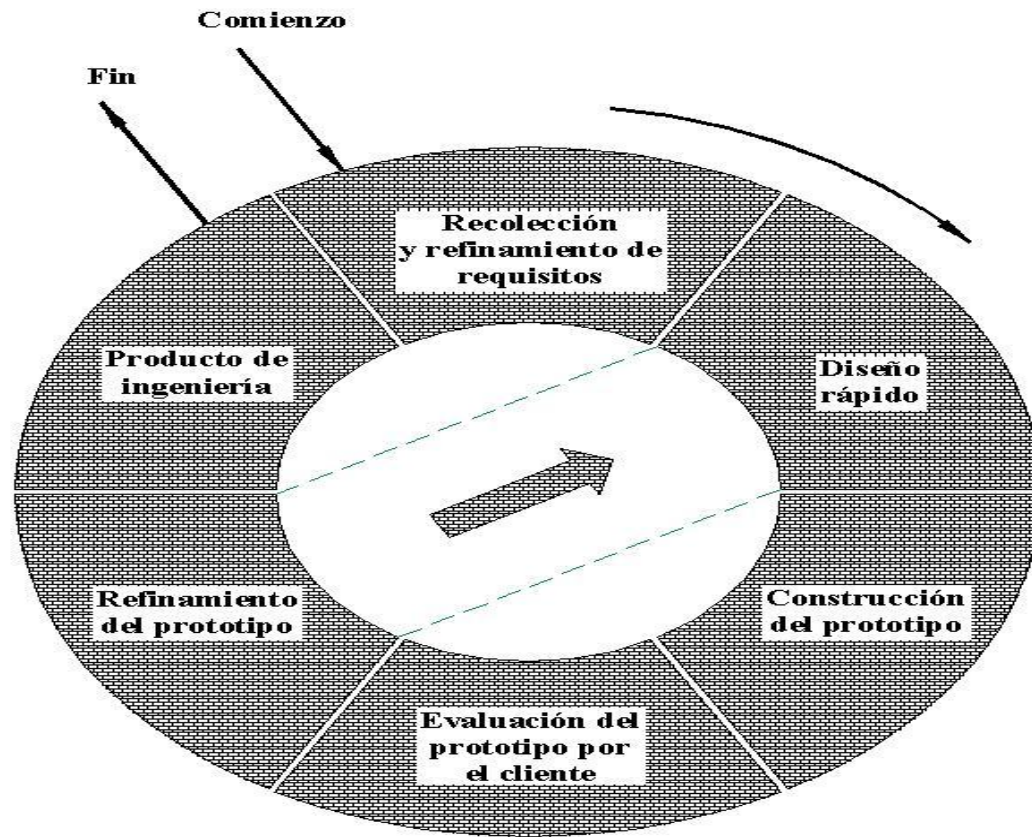
Modelo de Prototipo

- Se basa en un diseño rápido y la creación de un prototipo en las primeras fases del desarrollo del proyecto. Sus propiedades son:
- No modifica el ciclo de vida
- Reduce el riesgo de generar productos que no satisfagan los requisitos
- Reduce costos y aumenta las probabilidades de éxito
- Una vez identificados todos los requisitos mediante el prototipo, se construye el producto de ingeniería
- Deben desarrollarse versiones funcionales rápidamente
- Tiene gran importancia la interfaz de usuario

Desventajas

- Las principales desventajas de este prototipo son:
 - El cliente, al ver el prototipo en operación puede creer que el sistema esta completo
 - El desarrollador puede construir el sistema final solamente ampliando el prototipo lo que disminuye la calidad del producto final

Modelo de Prototipo



El Desarrollo Iterativo e Incremental

Introducción

- El desarrollo iterativo es un enfoque para el desarrollo de software que requiere un entrenamiento y poseer ciertos conocimientos
- El ciclo de vida iterativo se basa en la ampliación y refinamiento sucesivos del sistema mediante múltiples iteraciones, con retroalimentación cíclica y adaptación como elementos principales que convergen hacia un sistema adecuado

Productos de una Iteración

- Un producto de software listo para su distribución
- Código de los componentes del sistema
- Manuales para el uso del sistema
- Documentación técnica asociada

Fases del Desarrollo

- Las fases del desarrollo que se encuentran en el modelo iterativo e incremental son:
 - Inicio
 - Elaboración
 - Construcción
 - Transición

Etapa de Inicio

- Define los objetivos, alcances y limitaciones del proyecto
- Estima los recursos y viabilidad
- La fase de inicio debe tener una duración corta
- Bocetos de interfaz de usuario

Etapa de Elaboración

- Definir la mayoría de los casos de uso
- Implementación y prueba de elementos básicos de la Arquitectura
- Dos y cuatro iteraciones de entre dos y seis semanas
- Empezar pronto con la programación
- Realizar pruebas realistas
- Detallar la mayoría de los casos de uso

Etapa de Construcción

- Produce un sistema ejecutable en el ambiente del usuario
- Se detallan los casos de uso que no lo estén
- Se obtiene un manual de usuario preliminar
- Se obtiene el software ejecutable
- Se obtienen los diagramas y documentación del sistema

Etapa de Transición

- Debe Satisfacer los requerimientos planteados a satisfacción de los usuarios
- Correcciones a la versión anterior
- Obtener un producto operacional en el ambiente del usuario
- Capacitar a los usuarios en el uso del sistema
- Afinar los manuales de usuario

El Proceso Unificado

El Proceso Unificado (UP)

- El Proceso Unificado es una metodología cuyo fin es entregar un producto de software, es decir un conjunto de actividades necesarias para transformar un conjunto de requerimientos de usuario en un sistema de software

El Proceso Unificado

- Es un proceso iterativo, en donde el desarrollo se realiza a través de mini-proyectos de duración corta (semanas) llamadas iteraciones

Elementos del UP

- Desarrollo Iterativo de Software
- Permite comprender los requerimientos que hacen crecer el sistema
- Sigue un modelo que busca las tareas más riesgosas, reduciendo así los riesgos del proyecto

Elementos del UP

- Administración de Requerimientos
- Describe como se obtienen, organizan, documentan los requerimientos
- Captar y comunicar los requerimientos de la organización
- Documentar las decisiones

Elementos del UP

- Uso de Arquitecturas basadas en componentes
- Se basa en diseñar una arquitectura que sea flexible, fácil de modificar, comprensible y que se fundamenta en la reutilización de sus componentes
- Modelado visual del software
- Modela visualmente la organización
- Permite analizar la consistencia entre los componentes, el diseño y su implementación

Propiedades

- Las propiedades del UP son:
 - Fomenta el desarrollo iterativo e incremental
 - Dirigido por casos de uso
 - Centrado en la arquitectura

Dirigido por Casos de Uso

- Énfasis en los requerimientos del usuario
- El conjunto de todos los casos de uso constituyen el modelo de casos de uso

Centrado en la Arquitectura

- Conjunto de decisiones significativas respecto a la organización
- Selección de los componentes estructurales e interfaces que constituyen un sistema
- Casos de uso relacionados con la Función
- Arquitectura relacionada con la Forma
- Los casos de uso y la arquitectura se desarrollan y evolucionan en paralelo

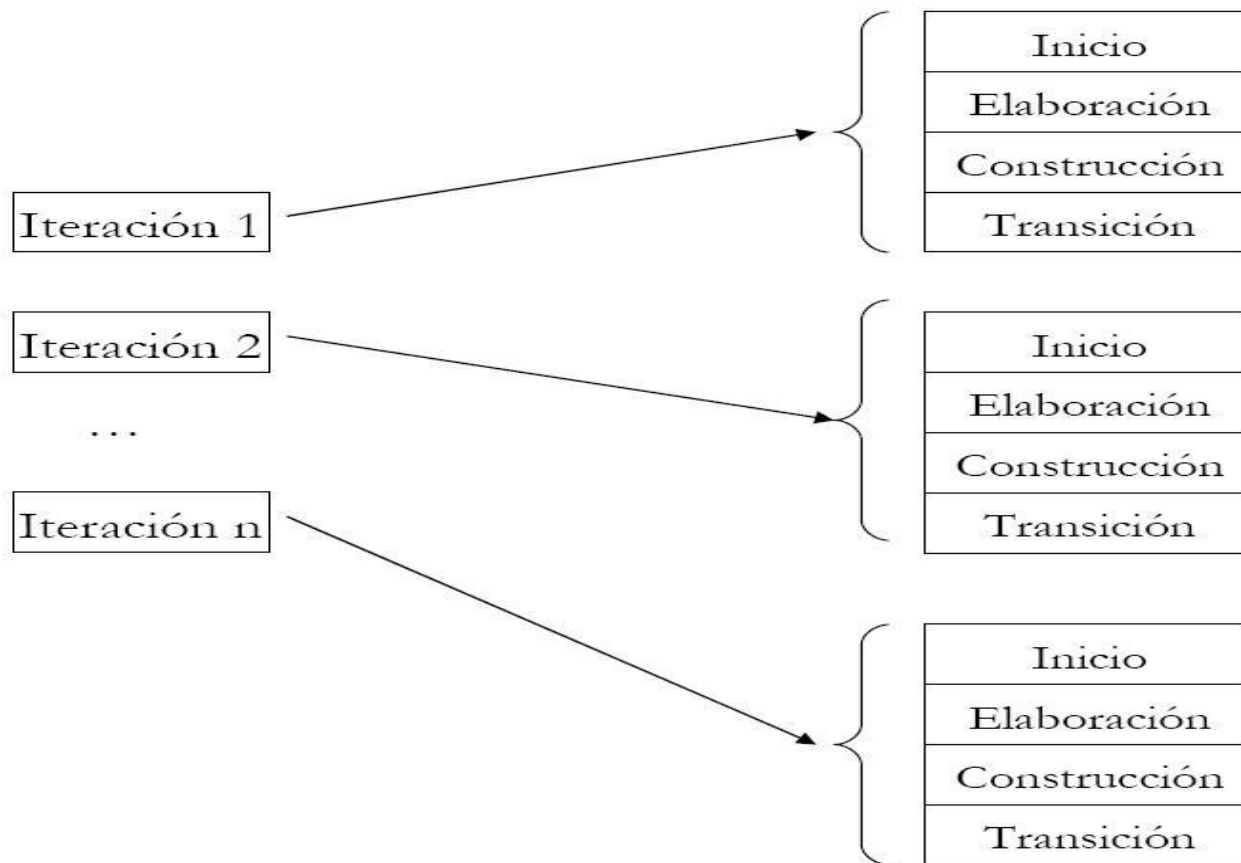
Etapas del UP

- Está formado por las siguientes etapas:
 - Inicio. Visión aproximada, análisis del negocio, alcance
 - Elaboración. Visión refinada, implementación de la base de la arquitectura, más requisitos
 - Construcción. Implementación iterativa de requisitos
 - Transición. Pruebas e instalación

Consideraciones Especiales

- En la fase de Inicio, NO se determinan los requisitos, se realiza un estudio de viabilidad
- En la fase de Elaboración, se implementa la arquitectura que servirá como base, NO se refiere al desarrollo del sistema

Esquema del UP

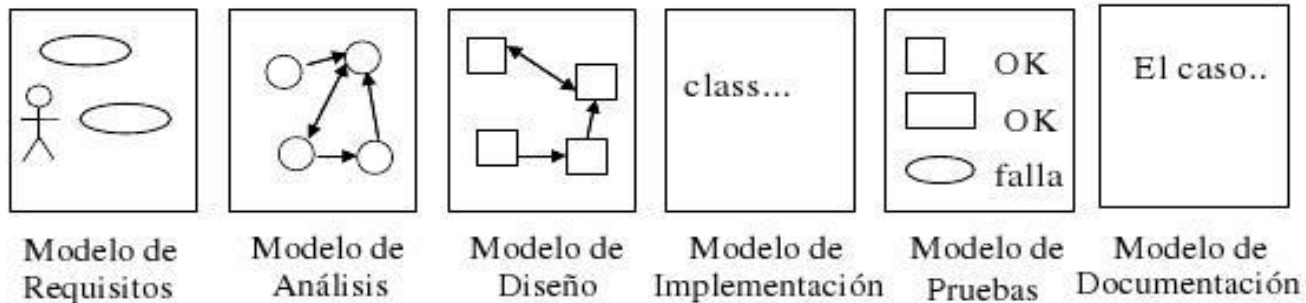


Disciplinas

- Es un conjunto de actividades que se realizan en un área o una etapa. Estas disciplinas generarán varios productos de trabajo o artefactos.

Flujos de Trabajo o Disciplinas

- El UP está formado por diferentes flujos de trabajo que contienen en diferentes niveles a las fases del desarrollo incremental. Estos flujos de trabajo son:



Artefactos

- Los artefactos son productos de trabajo que se generan en cada una de las disciplinas y que van evolucionando conforme se desarrolla el proyecto.
 - Modelo del Negocio: Modelo del Dominio
 - Requisitos: Modelo de Casos de Uso, Glosario, Especificaciones Complementarias
 - Diseño: Modelo de Diseño, Modelo de Arquitectura
 - Implementación: Modelo de Implementación

Reutilización

Fundamentos de la Reutilización

- Pueden existir similitudes entre distintos sistemas
- El software se representa como una combinación de módulos
- Diseñar aplicaciones es la suma de especificaciones de módulos e interrelaciones

Ventajas y Desventajas

- Ventajas
 - La principal ventaja es que puede reducir los tiempos y costos del desarrollo
- Desventajas
 - Puede ser difícil reconocer los componentes reutilizables
 - Dificultad para catalogar y recuperar esos componentes

Reutilización de Software

