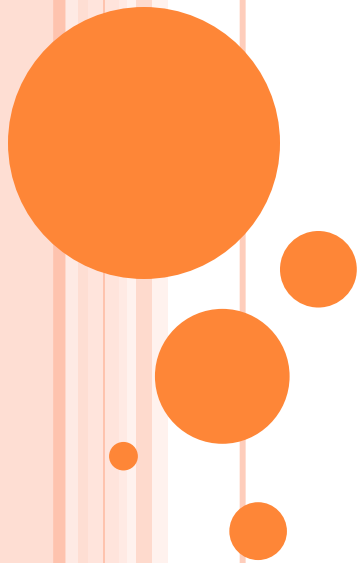



EL CICLO DE VIDA DEL DESARROLLO DE LOS SISTEMAS DE INFORMACIÓN.



PRINCIPIOS Y CRITERIOS PARA LA EVALUACIÓN DEL CICLO DE VIDA DE DESARROLLO DE SISTEMAS

- Se pueden enunciar algunos principios para desarrollar correctamente un sistema de información:
 - **1. Involucrar al usuario.**
 - El usuario es una parte imprescindible para el adecuado desarrollo de un sistema. Implicando al usuario se logrará mejor sus necesidades y reducir su potencial resistencia a los nuevos sistemas de información.
 - **2. Utilizar métodos de solución de problemas.**
 - Cualquier actividad compleja necesita aplicar lógicas contrastadas. El ciclo de vida es en sí un método de resolución de un problema específico.
 - **3. Abordar adecuadamente cada una de las fases.**
 - El ciclo de vida moderno incorpora una serie de fases: planificación, análisis, diseño, implantación y soporte de sistemas. En términos generales se puede decir que se desarrollan secuencialmente, y cada una de ellas incorpora mayor grado de detalle que la anterior. Las fases planificación y análisis han de abordarse correctamente, puesto que por muy inteligentes que sean las soluciones técnicas, sin un análisis correcto será muy difícil que el sistema sea todo lo útil que potencialmente podría ser.
- 

- **4. Normalizar y documentar.**

- Es fundamental que se fijen normas sobre las actividades, sobre las responsabilidades, requisitos documentales y controles de calidad para asegurar en el tiempo la supervivencia del sistema. Los analistas y programadores responsables de un sistema pueden dejar su puesto y si no existe la documentación apropiada, todo puede resultar caótico. La necesidad de documentar aumenta en la medida que el sistema que se desarrolle sea más complejo.

- **5. Justificar adecuadamente el sistema.**

- Desarrollar sistemas de información supone invertir en el futuro de la empresa. No se puede considerar un gasto, sino una inversión y como tal ha de plantearse.

- **6. Cancelar o revisar el proyecto si es necesario.**

- Si es necesario, durante el desarrollo se ha de ser lo suficientemente flexible como para cancelar un proyecto. Durante el ciclo de vida existen distintos momentos en los que se efectúa un control progresivo que es un control de la viabilidad del proyecto.

- **7. Descomponer y simplificar.**

- Un sistema complejo se ha de abordar dividiéndolo en subsistemas más simples. De esta manera disminuye la complejidad y es más abordable por el ser humano.

- **8. Diseñar sistemas flexibles.**

- Si los sistemas no se diseñan previendo futuras modificaciones, sólo servirán para momentos concretos en el tiempo. Si se hace necesario cambiar un sistema que no es flexible, consumirá muchos recursos y talento de las unidades involucradas en el soporte o mantenimiento del sistema.

CICLO DE VIDA DEL DESARROLLO DE SISTEMAS.

- Se puede decir que el ciclo de vida es una **herramienta de gestión de proyectos**- empleada para planificar, elaborar y controlar el proyecto de desarrollo de un sistema- y que involucra tanto a analistas como a ingenieros de software, programadores, propietarios y usuarios.
- Para el óptimo desarrollo de un sistema de información se deben seguir las siguientes **fases**:
 - 1. Planificación
 - 2. Análisis
 - 3. Diseño
 - 4. Implantación
 - 5. Mantenimiento



CICLO DE VIDA DEL DESARROLLO DE SISTEMAS.

1. Planificación de sistemas.

- La función de planificación pretende señalar y establecer prioridades sobre aquellas tecnologías y aplicaciones que producirán un **máximo beneficio** para la organización.
- El objetivo de esta fase consiste en decidir junto con el equipo humano de la empresa **donde** se va a implementar el sistema, los **objetivos** generales, específicos de la misma y **elaborar** los esquemas generales de la manera más clara y precisa.



CICLO DE VIDA DEL DESARROLLO DE SISTEMAS.

2. Análisis de los sistemas.

- Esta etapa ha sido clásicamente la **primera del ciclo** de vida del desarrollo de sistemas.
- Conlleva el estudio del sistema actual y la definición de las **necesidades reales** de los usuarios.
- Incluye las siguientes **fases**:
 - Análisis de la Viabilidad del Proyecto.
 - Análisis de los sistemas existentes.
 - Definir necesidades de usuarios y establecer prioridades.

CICLO DE VIDA DEL DESARROLLO DE SISTEMAS.

3. Diseño de sistemas.

- Durante esta etapa se diseña la solución informática para satisfacer las necesidades de la empresa, tal y como se especificaron.
- Esta etapa pierde su carácter de empresa para centrarse en cuestiones de **técnica informática**.
- El suceso que desencadena el inicio de esta fase es el documento de la **declaración de necesidades** de la empresa y las opiniones de los usuarios del sistema. La salida que esta etapa proporciona es la relación técnica de diseño que demuestra cómo se solucionan los problemas para los cuales se desarrolló el sistema.



CICLO DE VIDA DEL DESARROLLO DE SISTEMAS.

4. Implantación de sistemas.

- Comienza a partir de las especificaciones de diseño técnico proporcionadas en la fase anterior.
- Durante esta etapa se construye el sistema y se pone en funcionamiento.
- El output fundamental de esta etapa es el sistema de información en sí.

5. Mantenimiento de sistemas.

- Cualquier sistema implantado ha de mantenerse. Esta misión recae en un conjunto de expertos entre los que están los analistas y los programadores.
- El mantenimiento será necesario desde el momento en que se detecten algunas deficiencias en el sistema.
- Muchas veces serán los propios usuarios los que avisen de los problemas que el sistema ocasiona. Si los costes de mantenimiento fuesen demasiados altos, debería iniciarse el desarrollo de un nuevo sistema.

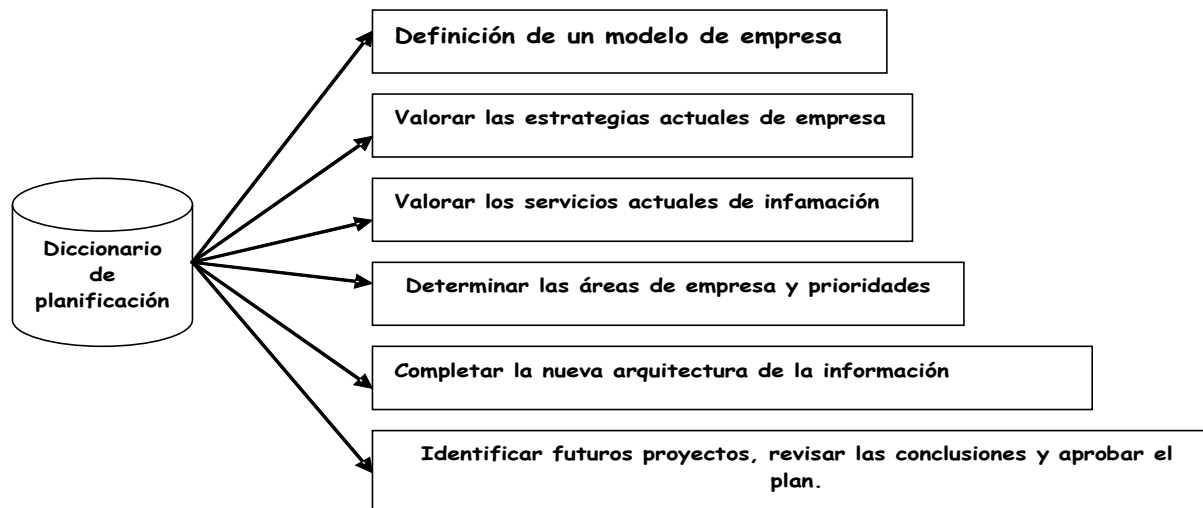
PLANIFICACIÓN DE SISTEMAS

- Como sabemos, la planificación de los sistemas de información es la **primera etapa** de un moderno ciclo de desarrollo y se puede considerar compuesta a su vez de tres **subetapas**:
 - Estudio de la misión y de los objetivos de la empresa.
 - Establecer una arquitectura de la información.
 - Analizar las áreas de empresa.
- **Estudio de la misión y de los objetivos de la empresa.**
- Para que los S.I sean verdaderamente útiles, han de contribuir a la misión de la empresa. Para aumentar el impacto positivo de las inversiones en sistemas de información, han de dirigirse a los objetivos, áreas y actividades que contribuyan en mayor medida al cumplimiento de la misión.
- Análisis de los factores fundamentales para el éxito.
- Análisis contextual. Especial referencia a la competencia.
- Análisis de las actividades sobre la base de la cadena de valor.
- Análisis del sistema de las actividades.



PLANIFICACIÓN DE SISTEMAS

- **Definición de una arquitectura de información.**
- La arquitectura de información se encarga del estudio, análisis, organización, disposición y estructuración de la información en la organización, y de la selección y presentación de los datos en los sistemas de información interactivos.
- Su principal objetivo es facilitar al máximo los procesos de comprensión y asimilación de la información, así como las tareas que ejecutan los usuarios en un espacio de información definido.
- Durante esta etapa de definición se han de realizar una serie de actividades que siguen una determinada secuencia, que se muestra a continuación junto con el diccionario de planificación en el que se archivan todos los documentos que se van generando.



PLANIFICACIÓN DE SISTEMAS

- Examen de las áreas de empresa.
- Es un examen general en un doble sentido, abarca todo un área de empresa y el nivel de detalle no es muy elevado.
- Las técnicas de estudio y análisis buscan conjuntamente el rediseño de los procesos para hacerlos más eficaces y eficientes. En los últimos años se ha hablado de la reingeniería de procesos o rediseño radical de los sistemas de información para mejorarlos y simplificarlos de forma intensa.
- Para desarrollar esta fase será necesario:
 - Constituir un equipo de análisis multifuncional.
 - Identificar las medidas de rendimiento del área de empresa.
 - Ampliar y desarrollar los modelos de áreas de empresa.
 - Valoración del rendimiento del área de empresa y de los sistemas.
 - Establecer proyectos y prioridades.
 - Planificar proyectos de desarrollo reales.
 - Revisar las conclusiones y aprobar el plan.



ANÁLISIS DE SISTEMAS

- El análisis de sistemas es el estudio de una aplicación del sistema de información y de empresa actual y la definición de las necesidades y las prioridades de usuario para conseguir una aplicación nueva o mejorada.
- Trata básicamente de determinar los objetivos y límites del sistema objeto de análisis, caracterizar su estructura y funcionamiento, marcar las directrices que permitan alcanzar los objetivos propuestos y evaluar sus consecuencia.
- Incluye las siguientes **fases**:
 - Análisis de la Viabilidad del Proyecto (o fase de inspección).
 - Análisis del sistema actual (o fase de estudio).
 - Definición y establecimiento de prioridades entre las necesidades de usuarios(o fase de definición).
- **Análisis de la Viabilidad del Proyecto.**
 - Objetivos de inspección.
 - Identificar los problemas, las oportunidades y las normas que dieron lugar a la solicitud del proyecto.



ANÁLISIS DE LA VIABILIDAD DEL PROYECTO.

- Determinar si resolver los problemas, aprovechar las oportunidades y cumplir las normas reportará beneficios a la empresa.
- ¿Qué **técnicas** se utilizarán en el EVS?
- Análisis coste/beneficio.
- Diagrama entidad/relación extendido.
- Sesiones de trabajo.
- Catalogación
- Impacto en la organización.
- Planificación
- Diagramas de actividades
- Matricial
- Presentación



ANÁLISIS DE LA VIABILIDAD DEL PROYECTO.

- ¿Dónde **interviene** cada uno de los siguientes participantes y cuáles su misión?
- **Comité de dirección:** participa al principio y al final del EVS. Su misión es indicar cuál debe ser el alcance del sistema y aprobar la solución final.
- **Usuarios expertos:** participan en el estudio de la situación actual, la definición de requisitos y el estudio de alternativas. Su objetivo es ayudar a conocer los sistemas existentes, exponer sus requisitos y dar su opinión en las alternativas de solución.
- **Especialistas en comunicaciones:** participan en el estudio de alternativas de solución. Su objetivo es definir los requisitos de comunicación de las distintas soluciones.



ANÁLISIS DEL SISTEMA ACTUAL

- Consiste en estudiar y analizar el sistema actual, siempre y cuando se cuente con un sistema actual, hago uso o no de la informática, dota al analista de una comprensión mas profunda del sistema.
- Los **objetivos**:
 - Conocer el entorno de empresa del sistema.
 - Conocer las causas y los efectos subyacentes del sistema.
 - Conocer las ventajas de aprovechar las oportunidades.
 - Conocer las implicaciones de no cumplir con las normas.




ESTABLECER LAS PRIORIDADES DE LOS USUARIOS

- Define a partir de la información qué necesita y que quiere el usuario que haga el sistema.
- **Objetivos:**
 - Definir las necesidades de la empresa sobre problemas.
 - Definir las necesidades de empresa sobre oportunidades
 - Definir las necesidades sobre normas.
- **Actividades:**
 - Identificar las necesidades.
 - Modelizar las necesidades de sistemas.
 - Elaborar prototipos de descubrimiento.
 - Definir prioridades entre las necesidades de empresa.
 - Modificar el ámbito y el plan de proyecto.
 - Revisar las especificaciones de las necesidades.



DISEÑO DE SISTEMAS

- El diseño de sistemas se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física.
 - La fase de diseño de sistemas encierra **cuatro etapas**:
 - **El diseño de los datos**: transforma el modelo de dominio de la información, creado durante el análisis, en las estructuras de datos necesarios para implementar el software.
 - **El diseño arquitectónico**: define la relación entre cada uno de los elementos estructurales del programa.
 - **El diseño de la interfaz**: describe como se comunica el software consigo mismo, con los sistemas que operan junto con el y con los operadores y usuarios que lo emplean.
 - **El diseño de procedimientos**: transforma elementos estructurales de la arquitectura del programa. La importancia del diseño del software se puede definir en una sola palabra **Calidad**, dentro del diseño es donde se fomenta la calidad del proyecto.
- 

EL DISEÑO DE PROCEDIMIENTOS

- El diseño es la única manera de materializar con precisión los requerimientos del cliente.
- El diseño del software es un proceso y un modelado a la vez. El proceso de diseño es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del sistema a construir.
- A lo largo del diseño se evalúa la calidad del desarrollo del proyecto con un conjunto de revisiones técnicas:
- El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente.
- Debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el software.
- El diseño debe proporcionar una completa idea de lo que es el software.



DISEÑO DE SISTEMAS

- **Diseño de la salida.**
- En este caso salida se refiere a los **resultados** e informaciones generadas por el sistema. Para la mayoría de los usuarios la salida es la única razón para el desarrollo de un sistema y la base de evaluación de su utilidad.
- Cuando se realiza un sistema, como analistas deben realizar lo siguiente:
- Determine que información presentar. Decidir si la información será presentada en forma visual, verbal o impresa y seleccionar el medio de salida.
- Disponga la presentación de la información en un formato aceptable.
- Decidir como distribuir la salida entre los posibles destinatarios.



DISEÑO DE SISTEMAS

- **Diseño de archivos.**
- Incluye decisiones con respecto a la naturaleza y contenido del propio archivo, como si fuera a emplear para guardar detalles de las transacciones, datos históricos, o información de referencia.
- Entre las **decisiones** que se toman durante el diseño de archivos, se encuentran las siguientes:
 - Los datos que deben incluirse en el formato de registros contenidos en el archivo.
 - La longitud de cada registro, con base en las características de los datos que contenga.
 - La secuencia a disposición de los registros dentro del archivo.
- No todos los sistemas requieren del diseño de todos los archivos, ya que la mayoría de ellos pueden utilizar los del viejo sistema y solo tenga que enlazarse el nuevo sistema al archivo maestro donde se encuentran los registros.



DISEÑO DE SISTEMAS

- **Diseño de interacciones con la base de datos.**
- La mayoría de los sistemas de información ya sean implantados en sistemas de cómputos grandes o pequeños, utilizan una base de datos que pueden abarcar varias aplicaciones, por esta razón estos sistemas utilizan un administrador de base de datos, en este caso el diseñador no construye la base de datos sino que consulta a su administrador para ponerse de acuerdo en el uso de la base de datos en el sistema.
- **Herramientas para el diseño de sistemas.**
- Apoyan el proceso de formular las características que el sistema debe tener para satisfacer los requerimientos detectados durante las actividades del análisis.
- **Herramientas para presentación.**
- Se utilizan para describir la posición de datos, mensajes y encabezados sobre las pantallas de las terminales, reportes y otros medios de entrada y salida.



DISEÑO DE SISTEMAS

- **Herramientas de especificación.**
- Apoyan el proceso de formular las características que debe tener una aplicación, tales como entradas, salidas, procesamiento y especificaciones de control. Muchas incluyen herramientas para crear especificaciones de datos.
- **Herramientas para el desarrollo de sistemas.**
- Estas herramientas nos ayudan como analistas a trasladar diseños en aplicaciones funcionales.
- **Herramientas para ingeniería de software.**
- Apoyan el proceso de formular diseños de software, incluyendo procedimientos y controles, así como la documentación correspondiente.
- **Generadores de códigos.**
- Producen el código fuente y las aplicaciones a partir de especificaciones funcionales bien articuladas.




DISEÑO DE SISTEMAS

- **Herramientas para pruebas.**
- Apoyan la fase de la evaluación de un sistema o de partes del mismo contra las especificaciones.
- Incluyen facilidades para examinar la correcta operación del sistema así como el grado de perfección alcanzado en comparación con las expectativas.
- La revolución del procesamiento de datos de manera computarizada, junto con las practicas de diseño sofisticadas están cambiando de forma dramática la manera en que se trasladan las especificaciones de diseño de sistemas de información funcionales.



FASE DE IMPLANTACIÓN

- Es la última fase del desarrollo de sistemas. Es el proceso de instalar equipos o software nuevos, como resultado de un análisis y diseño previo como resultado de la sustitución o mejoramiento de la forma de llevar a cabo un proceso automatizado.
 - Al implantar un sistema de información lo primero que debemos hacer es asegurarnos que el sistema sea operacional, es decir, que funcione de acuerdo con lo que requiere el análisis y permita que los usuarios puedan operar con él.
 - Existen varios **enfoques** de implementación:
 - Es darle responsabilidad a los grupos.
 - Uso de diferentes estrategias para el entrenamiento de los usuarios.
 - El analista de sistemas necesita ponderar la situación y proponer un plan de conversión que sea adecuado para la organización.
 - El analista necesita formular medidas de desempeño con las cuales evaluar a los usuarios.
 - Debe convertir físicamente el sistema de información antiguo en el nuevo modificado.
- 

FASE DE IMPLANTACIÓN

- **Capacitación de usuarios del sistema.**
- Es enseñar a los usuarios que se relacionan u operan en un proceso de implantación.
- La responsabilidad de esta capacitación de los usuarios primarios y secundarios es del **analista**, desde el personal de captura de datos hasta aquellos que toman las decisiones sin usar un ordenador.
- La empresa puede contratar los servicios de instructores externos pero el analista es la persona que puede ofrecer la mejor capacitación debido a que conoce al personal y el sistema mejor que cualquier otra persona.
- Si falta el analista la empresa puede contratar otros servicios de capacitación como son:
- **Vendedores:** son aquellos que proporcionan capacitación gratuita fuera de la empresa de uno o dos días.
- **Instructor pagado externamente:** son aquellos que pueden enseñar todo acerca de los ordenadores pero para algunos usuarios esta no es una capacitación necesaria.
- **Instructores en casa:** están familiarizados con el personal y pueden adecuar los materiales a sus necesidades, pero le faltaría experiencia en sistemas de información que es realmente la necesidad del usuario.
- El **objetivo de la capacitación** es lograr que los usuarios tengan el dominio necesario de las cosas básicas acerca de las maquinarias y procesos que se emplean para su operación de manera eficiente y segura.

LA EVALUACIÓN DEL SISTEMA

- Se lleva a cabo para identificar puntos débiles y fuertes del sistema implantado. La evaluación ocurre a lo largo de cualquiera de las siguientes cuatro dimensiones:
- **Evaluación operacional:** es el momento en que se evalúa la manera en que funciona el sistema, esto incluye su facilidad de uso, tiempo de respuesta ante una necesidad, como se adecuan los formatos en que se presenta la información, contabilidad global y su nivel de utilidad.
- **Impacto organizacional:** identifica y mide los b° operacionales para la empresa en áreas como las finanzas, eficiencia en el desempeño laboral e impacto competitivo.
- **Desempeño del desarrollo:** es la evaluación del proceso de desarrollo adecuado tomando en cuenta criterios como, tiempo y esfuerzo en el desarrollo concuerdan con presupuesto y estándares.
- **Prueba de sistemas:** dependiendo del tamaño de la empresa que usara el sistema y el riesgo asociado a su uso, puede hacerse la elección de comenzar la operación del sistema solo en un área de la empresa (como una prueba piloto), que puede llevarse a cabo en un departamento o con una o dos personas. Cuando se implanta un nuevo sistema lo aconsejable es que el viejo y el nuevo funcionen a la vez con la finalidad de comparar los R° que ambos ofrecen en su operación, además de dar tiempo al personal para adaptarse al nuevo sistema.

FASE DE UTILIZACIÓN Y MANTENIMIENTO

- La fase de mantenimiento de software aporta cambios al mismo para corregir defectos y dependencias encontradas durante su uso así como la adición de nuevas funciones para mejorar la usabilidad y aplicabilidad del software.
- Tipos de mantenimiento:
- **Perfectivo:** son las acciones llevadas a cabo para mejorar la calidad interna de los sistemas en cualquiera de sus aspectos; reestructuración del código, definición más clara del sistema y optimización del rendimiento y eficiencia.
- **Evolutivo:** son las incorporaciones, modificaciones y eliminaciones necesarias en un producto software para cubrir la expansión o cambio en las necesidades del usuario.
- **Adaptativo:** son las modificaciones que afectan a los entornos en los que el sistema opera.
- **Correctivo:** son aquellos cambios precisos para corregir errores del producto software.

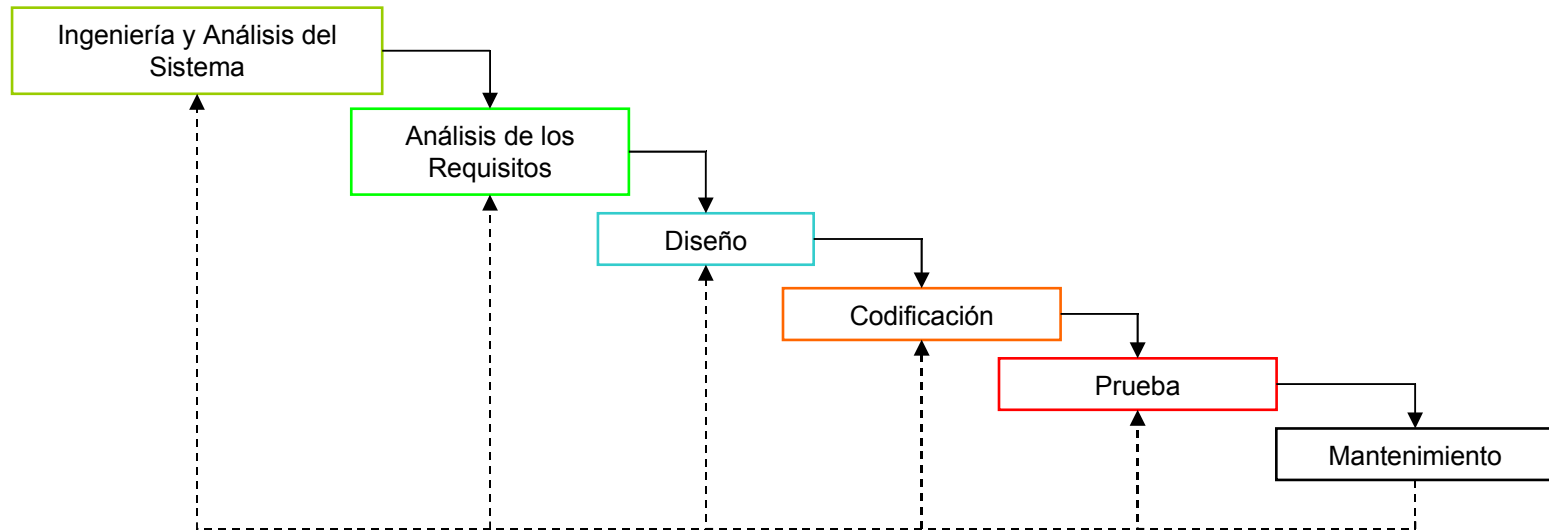


MODELOS DEL CICLO DE VIDA DE LOS SISTEMAS DE INFORMACIÓN

- Modelo clásico o en cascada.
- Modelo incremental.
- Modelo de desarrollo evolutivo.
- Modelo de prototipado de requerimientos.
- Modelo de espiral.
- Modelo de construcción de prototipos.
- Modelo de síntesis automática de software.



MODELO CLÁSICO O EN CASCADA.



Ingeniería y Análisis del Sistema: Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

Análisis de los requisitos del software: el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software (Analistas) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.




MODELO CLÁSICO O EN CASCADA.

- **Diseño:** el diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.
- **Codificación:** el diseño debe traducirse en una forma legible para la maquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.
- **Prueba:** una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.
- **Mantenimiento:** el software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debido a que hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.



MODELO INCREMENTAL.

- Los riesgos asociados con el desarrollo de sistemas largos y complejos son enormes. Una forma de reducir los riesgos es construir sólo una parte del sistema, reservando otros aspectos para niveles posteriores. El desarrollo incremental es el proceso de construcción siempre incrementando subconjuntos de requerimientos del sistema.
 - El modelo de desarrollo incremental provee algunos **beneficios** significativos para los proyectos:
 - Construir un sistema pequeño tiene siempre menos riesgo que construir un sistema grande.
 - Al ir desarrollando parte de las funcionalidades, es más fácil determinar si los requerimientos planeados para los niveles subsiguientes son correctos.
 - Si un error importante es realizado, sólo la última iteración necesita ser descartada.
 - Reduciendo el tiempo de desarrollo de un sistema decrecen las probabilidades que esos requerimientos de usuarios puedan cambiar durante el desarrollo.
 - Si un error importante es realizado, el incremento previo puede ser usado.
 - Los errores de desarrollo realizados en un incremento, pueden ser arreglados antes del comienzo del próximo incremento.
- 

MODELO DE DESARROLLO EVOLUTIVO.

- Construye una serie de grandes versiones sucesivas de un producto. El modelo evolutivo asume que los requerimientos no son completamente conocidos al inicio del proyecto.
- En el modelo evolutivo, los requerimientos son cuidadosamente examinados, y sólo esos que son bien comprendidos son seleccionados para el primer incremento. Los desarrolladores construyen una implementación parcial del sistema que recibe sólo estos requerimientos.
- El sistema es entonces desarrollado, los usuarios lo usan, y proveen retroalimentación a los desarrolladores. Basada en esta retroalimentación, la especificación de requerimientos es actualizada, y una segunda versión del producto es desarrollada y desplegada. El proceso se repite indefinidamente.
- Todo lo que uno tiene que hacer es construir un subconjunto de requerimientos conocidos y comprender al principio que muchos nuevos requerimientos es probable que aparezcan cuando el sistema sea desplegado o desarrollado.



MODELO DE PROTOTIPADO DE REQUERIMIENTOS.

- El prototipado de requerimientos es la creación de una implementación parcial de un sistema, para el propósito explícito de aprender sobre los requerimientos del sistema.
- Un prototipo es construido de una manera rápida tal como sea posible. Esto es dado a los usuarios, clientes o representantes de ellos, posibilitando que ellos experimenten con el prototipo. Estos individuos luego proveen la retroalimentación sobre lo que a ellos les gustó y no les gustó acerca del prototipo proporcionado, quienes capturan en la documentación actual de la especificación de requerimientos la información entregada por los usuarios para el desarrollo del sistema real.

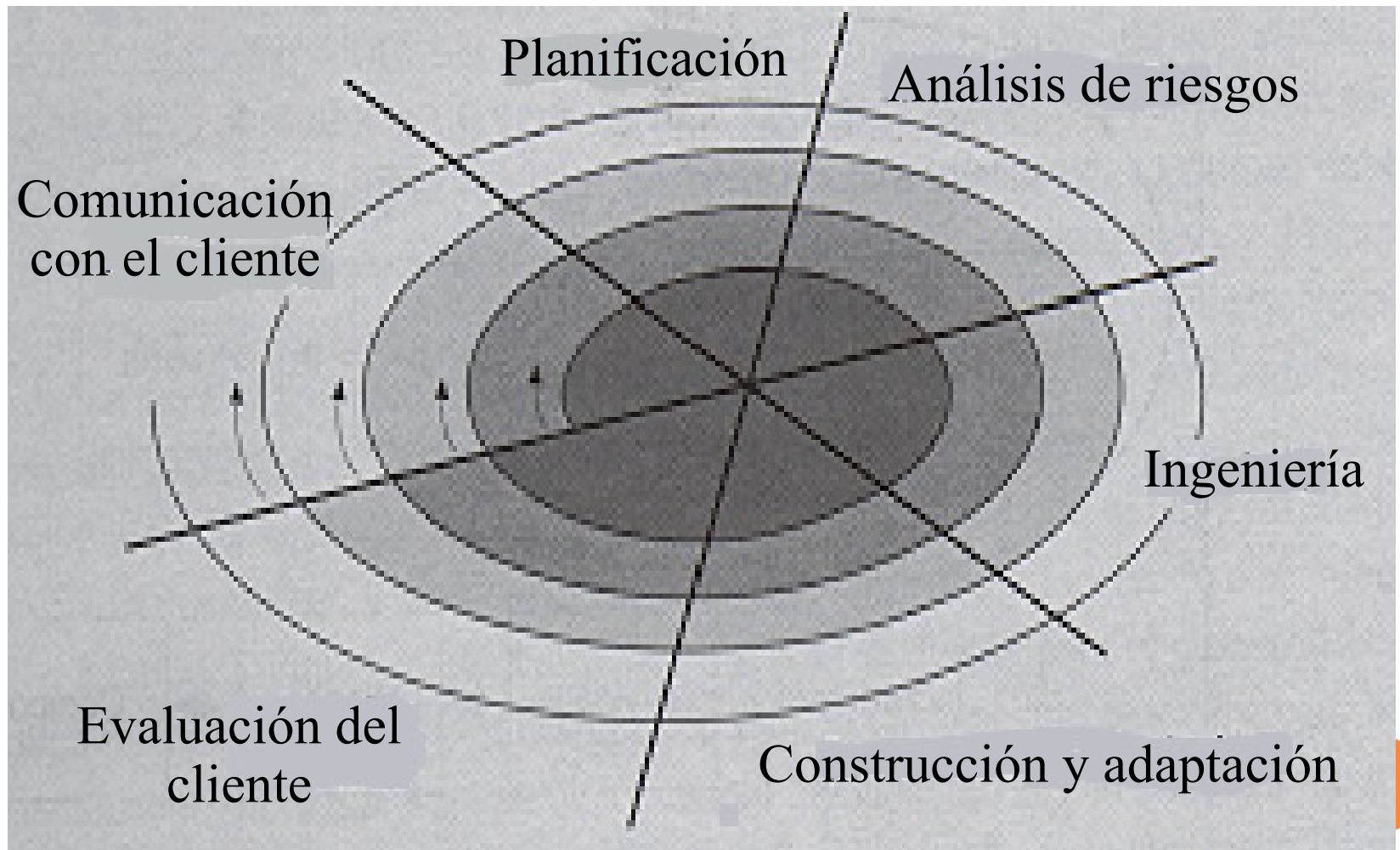


MODELO EN ESPIRAL

- Es un modelo evolutivo que combina el modelo clásico con el diseño de prototipos.
- Incluye la etapa de análisis de riesgos.
- Es ideal para crear productos con diferentes versiones mejoradas.
- Este es el enfoque más realista actualmente. El modelo en espiral se divide en un número de actividades estructurales, también llamadas regiones de tareas.
- Generalmente, existen entre tres y seis regiones de tareas:
- **Comunicación con el cliente:** las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- **Planificación:** las tareas requeridas para definir recursos, el tiempo y otras informaciones relacionadas con el proyecto. Son todos los requerimientos.
- **Análisis de riesgos:** las tareas requeridas para evaluar riesgos técnicos y otras informaciones relacionadas con el proyecto.
- **Ingeniería:** las tareas requeridas para construir una o más representaciones de la aplicación.
- **Construcción y adaptación:** las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.
- **Evaluación del cliente:** las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementación durante la etapa de instalación.



EL MODELO EN ESPIRAL



MODELO DE CONSTRUCCIÓN DE PROTOTIPOS



MODELO DE CONSTRUCCIÓN DE PROTOTIPOS

- Este modelo arranca con el establecimiento de los requerimientos del sistema, se definen los objetivos del sistema y los requisitos conocidos con base en las áreas de mayor prioridad e importancia para el sistema.
- Luego se hace un diseño preliminar , sobre el cual se construye un prototipo o modelo del sistema, compuesto a menudo de ventanas, tablas de la Base de Datos, formatos de entrada y de salida básicos.
- Un prototipo es una representación o modelo del producto de programación que incorpora componentes del producto real. Por lo regular, un prototipo tiene un funcionamiento limitado en cuanto a capacidades, confiabilidad o eficiencia.



MODELO DE SÍNTESIS AUTOMÁTICA DE SOFTWARE

- Se define el sistema utilizando un lenguaje formal.
- La implementación es automática, asistida por el Ordenador.
- La documentación se genera de forma automática.
- El mantenimiento se realiza "por sustitución" no mediante "parches".
- Dificultad en la participación del usuario.
- Diseños poco optimizados.

