



**COLEGIO VOCACIONAL DE ARTES Y OFICIOS DE
CARTAGO NOCTURNO**

CURSO LECTIVO 2021

Primer Periodo

Enunciado Actividad Virtual ·1

**NIVEL: DECIMO
SECCIÓN: 10-05**

**Sub-Área: Diseño de Software
PROFESOR
Lic. Alexander Monge Vargas**

Alumno: Nelson Rodríguez Zúñiga

Abril, 2021

Contenido

Introducción.....	4
Sistemas de Información.....	4
¿Qué son sistemas de información?	4
Actividades.....	5
Ciclo de vida	5
Características de los sistemas de información:.....	6
Sistemas transaccionales	6
Sistemas de apoyo a las decisiones	6
Sistemas estratégicos.....	6
CRM(Costumer Relationship Management)	7
ERP (Enterprice Resource Plannig).....	8
Objetivos principales de ERP:.....	8
SCM (Supply Chain Management)	8
Modelos de la Ingeniería del Software	9
Modelo de Prototipo:	9
Etapas:.....	9
Ventajas:	9
Desventajas:	10
Modelo Espiral:	10
Ciclos o Iteraciones	10
Tareas.....	11
Ventajas	13
Desventajas	13
Modelo de Cascada.....	13
Fases:.....	13
Análisis de requisitos del software	13
Diseño del sistema	14
Diseño del programa.....	14
Codificación del ordeñado	14
Pruebas	14
Verificación	14
Mantenimiento	15

Ventajas	15
Desventajas	15
Modelo RUP	15
Características.....	16
Fases:.....	16
¿Cuándo usar RUP?	18
Ventajas del Modelo RUP	18
Desventajas del Modelo RUP	18
CONCLUSIONES	19

Introducción

A continuación, veremos datos importantes sobre los temas de Sistemas de información, así como también de Modelos de Ingeniería de software.

Se tomarán datos relevantes, información importante de cada tema, explicando sus características, partes importantes de ellas, además de ciclos, fases, ventajas y desventajas y también definiciones importantes.

Sistemas de Información

¿Qué son sistemas de información?

Es un conjunto de elementos orientados al tratamiento y la administración de datos e información, generados para cubrir una necesidad u objetivo, estos elementos formaran parte de las siguientes categorías:

- ✓ Personas.
- ✓ Actividades o técnicas de trabajo.
- ✓ Recursos materiales en general (recursos informáticos y de comunicación)

Los sistemas de información es un concepto muy genérico, dependiendo de para qué se vaya a usar así varia de su función.

Por ejemplo, en geografía y cartografía un sistema de información se utiliza para almacenar, editar, analizar, compartir y desplegar información georreferenciada.

En informática, es cualquier sistema de información que se utilice para obtener, almacenar, administrar, controlar, procesar, transmitir y recibir datos.

Actividades

Existen 4 actividades que producen información para el análisis y toma de decisiones:

- ✓ Recopilación de Datos: captura y recolecta datos en bruto tanto del interior como del exterior de la organización.
- ✓ Almacenamiento: Guardar de forma estructurada la información recopilada.
- ✓ Procesamiento: Convierte esa entrada de datos en una forma más significativa.
- ✓ Distribución: transfiere la información procesada a las personas o roles que la usarán.

Ciclo de vida

Existen pautas básicas para el desarrollo de un sistema de información para la organización:

- ✓ Codificación: con el algoritmo ya diseñado, se procede a su reescritura en un lenguaje de programación establecido (programación) en la etapa anterior, es decir, en códigos que la máquina pueda interpretar y ejecutar.
- ✓ Conocimiento de la organización: analizar y conocer todos los sistemas que forman parte de la organización, así como los futuros usuarios del sistema de información. En las empresas (fin de lucro presente), se analiza el proceso de negocio y los procesos transaccionales a los que dará soporte el SI.
- ✓ Determinar las necesidades: este proceso también se denomina elicitación de requerimientos. En el mismo, se procede identificar a través de algún método de recolección de información (el que más se ajuste a cada caso) la información relevante para el sistema de información que se propondrá.
- ✓ Diagnóstico: en este paso se elabora un informe resaltando los aspectos positivos y negativos de la organización. Este informe formará parte de la propuesta del sistema de información y, también, será tomado en cuenta a la hora del diseño.
- ✓ Diseño del sistema: una vez aprobado el proyecto, se comienza con la elaboración del diseño lógico del sistema de información; la misma incluye: el diseño del flujo de la información dentro del sistema, los procesos que se realizarán dentro del sistema, el diccionario de datos, los reportes de salida, etc. En este paso es importante para seleccionar la plataforma donde se apoyará el SI y el lenguaje de programación a utilizar.
- ✓ Identificación de problemas y oportunidades: el segundo paso es relevar las situaciones que tiene la organización y de las cuales se puede sacar una ventaja competitiva (Por ejemplo: una empresa con un personal capacitado en manejo informático reduce el costo de capacitación de los usuarios), así como las situaciones desventajosas o limitaciones que hay que sortear o que tomar en cuenta (Por ejemplo: el edificio de una empresa que cuenta con un espacio muy reducido y no permitirá instalar más de dos computadoras).

- ✓ Implementación: este paso consta de todas las actividades requeridas para la instalación de los equipos informáticos, redes y la instalación de la aplicación(programa) generada en la etapa de Codificación.
- ✓ Mantenimiento: proceso de retroalimentación, a través del cual se puede solicitar la corrección, el mejoramiento o la adaptación del sistema de información ya creado a otro entorno de trabajo o plataforma. Este paso incluye el soporte técnico acordado anteriormente.
- ✓ Propuesta: contando ya con toda la información necesaria acerca de la organización, es posible elaborar una propuesta formal dirigida hacia la organización donde se detalle: el presupuesto, la relación costo-beneficio y la presentación del proyecto de desarrollo del sistema de información.

Características de los sistemas de información:

Sistemas transaccionales. Su principal característica es:

- ❖ A través de estos suelen lograrse ahorros significativos de mano de obra, debido a que automatizan tareas operativas de la organización.

Sistemas de apoyo a las decisiones. La principal característica de estos es:

- ❖ Apoyan la toma de decisiones, por su misma naturaleza, son repetitivos y soportan decisiones no estructuradas que no suelen repetirse. Este tipo de sistemas puede incluir la programación de la producción, compra de materiales, flujo de fondos, proyecciones financieras, modelos de simulación de negocios, modelos de inventarios, etcétera.

Sistemas de apoyo de las decisiones. La principal característica de estos es que:

- ❖ Suelen introducirse después de la implantación de los sistemas transaccionales más relevantes de la empresa, porque estos últimos constituyen su plataforma de información.

Sistemas estratégicos. Su principal característica es:

- ❖ Su función primordial no es apoyar la automatización de procesos operativos ni proporcionar información para apoyar la toma de decisiones.

Ahora bien, los sistemas de gestión de información (SGI) permiten:

- ❖ Comprender la marcha de las organizaciones desde un enfoque analítico (donde queremos estar), evaluador (donde estamos) y creativo (donde podríamos estar).
- ❖ Develar oportunidades que merezcan ser explotadas y contrarrestar amenazas.
- ❖ Establecer los factores que resulten críticos y las necesidades asociadas con el SGI.
- ❖ Estudiar el impacto de los SGI en la posición del negocio y buscar nuevas oportunidades.

En el caso del diseño de un SGI precisa de:

- ❖ Un análisis previo de las necesidades de información de la organización.
- ❖ Un diagnóstico de la situación.
- ❖ Una auditoría de información que permita conocer los recursos de información disponibles y los que faltan, para qué y quiénes lo utilizan, qué valor se le añade en su uso, entre otros.

CRM(Costumer Relationship Management)

Es una estrategia de negocio orientada a la fidelización de clientes.

Permite a todos los empleados de una empresa disponer de información actualizada sobre los mismos, con el objetivo de optimizar la relación entre empresa y cliente. Además, ayuda a conocer todos los diferentes puntos de contacto con los cuales el cliente interactúa en la empresa.

CRM es un modelo de gestión basada en la satisfacción del cliente, el concepto más cercano es el de Marketing relacional.

A nivel de software para la administración de la relación con los clientes.

Sistemas informáticos de apoyo a la gestión de las relaciones con los clientes, a la venta y al marketing. 2 Dicho software puede comprender varias funcionalidades para gestionar las ventas y los clientes de la empresa: automatización y promoción de ventas, tecnologías data warehouse (almacén de datos) para agregar la información transaccional y proporcionar capa de reporting, dashboards e indicadores claves de negocio, funcionalidades para seguimiento de campañas de marketing y gestión de oportunidades de negocio, capacidades predictivas y de proyección de ventas.

La parte fundamental de toda estrategia CRM es la base de datos, con la cual se puede realizar un seguimiento constante a los clientes. Actualmente la manera innovadora que han encontrado

los almacenes de cadena para realizar seguimiento a sus clientes, es por medio de las tarjetas de puntos que otorgan regalos. De esta manera, por medio de la tarjeta, se puede saber en promedio cada cuánto el cliente visita el almacén y que productos suele adquirir. Uno de los mayores problemas para que las empresas exploten un producto CRM es el alto costo de estos productos comerciales, licencias adicionales como un sistema operativo y más aún el alto costo de la su implantación y puesta en marcha, afortunadamente existen también diversos proyectos de sistemas CRM con licencia

ERP (Enterprise Resource Plannig)

Sistema de planificación de recursos, su objetivo es la planificación de los recursos de la organización, es tener claramente identificado como llegar a los productos finales desde la materia prima, o sea, desde el inventario e insumos para poder determinar la cantidad que llegaremos a generar de productos finales, integran la información y los procesos de la organización en un solo sistema.

Son sistemas de gestión automatizan algunas prácticas de negocio asociadas con los aspectos operativos y productivos de la organización.

Objetivos principales de ERP:

- Optimización de procesos empresariales.
- Acceso a información de forma confiable, verás y a tiempo.
- La posibilidad de compartir información con las demás áreas de la organización buscando integridad y buen desarrollo de los procesos.
- Eliminación procesos incensarios dentro de la compañía.

SCM (Supply Chain Management)

O administración de cadena de suministros, se define como la unión de proveedores y consumidores por medio de una cadena. Se busca principalmente que los procesos que añaden más valor a la cadena, estén integrados para evitar disconformidades en uno u otro.

El modelo SCM, se toma cada vez con mayor interés por las empresas, debido a su alta efectividad. El objetivo de las empresas se deriva principalmente de dos factores: la globalización de mercados y producción y la evolución tecnológica.

La globalización de los mercados y la producción ha generado una voraz competencia mundial entre empresas, a la competencia se une el avance de muchas industrias por el intercambio

tecnológico, factor que hace más dura la competencia. Esto ha generado un ciclo de vida cada vez más corto para los productos.

La innovación tecnológica, de información, comunicación y de transporte ha permitido el desarrollo de métodos y herramientas cada vez eficaces para sincronizar la oferta y la demanda, por medio de la coordinación entre las actividades desarrolladas por proveedores, productores y distribuidores.

Modelos de la Ingeniería del Software

Modelo de Prototipo:

Pertenece a los modelos de desarrollo evolutivo. Se debe de construir en poco tiempo usando programa adecuados y optimizando recursos.

Este diseño se centra en una representación de la parte visual del software que serán vistos por el usuario final. Este será evaluado por el cliente para retroalimentación por este medio se refinan los requisitos del sistema que se desarrollará. La interacción ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

Etapas:

- ✓ Plan rápido.
- ✓ Modelo, diseño rápido.
- ✓ Construcción del prototipo.
- ✓ Desarrollo, entrega y retroalimentación
- ✓ Comunicación.
- ✓ Entrega del desarrollo final.

Ventajas:

- ☑ Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.
- ☑ También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina
- ☑ Se puede reutilizar el código.

La construcción de prototipos se puede utilizar como un modelo del proceso independiente, se emplea más comúnmente como una técnica susceptible de implementarse dentro del contexto de cualquiera de los modelos del proceso expuestos. Sin importar la forma en que éste se aplique, el paradigma de construcción de prototipos ayuda al desarrollo de software y al cliente a entender de mejor manera cuál será el resultado de la construcción cuando los requisitos estén satisfechos. De esta manera, este ciclo de vida en particular, involucra al cliente más profundamente para adquirir el producto.

Desventajas:

- ☒ El usuario tiende a crearse unas expectativas cuando ve el prototipo de cara al sistema final. A causa de la intención de crear un prototipo de forma rápida, se suelen desatender aspectos importantes, tales como la calidad y el mantenimiento a largo plazo, lo que obliga en la mayor parte de los casos a reconstruirlo una vez que el prototipo ha cumplido su función. Es frecuente que el usuario se muestre reacio a ello y pida que sobre ese prototipo se construya el sistema final, lo que lo convertiría en un prototipo evolutivo, pero partiendo de un estado poco recomendado.
- ☒ En aras de desarrollar rápidamente el prototipo, el desarrollador suele tomar algunas decisiones de implementación poco convenientes (por ejemplo, elegir un lenguaje de programación incorrecto porque proporcione un desarrollo más rápido). Con el paso del tiempo, el desarrollador puede olvidarse de la razón que le llevó a tomar tales decisiones, con lo que se corre el riesgo de que dichas elecciones pasen a formar parte del sistema final...

Modelo Espiral:

El desarrollo en espiral es un modelo de ciclo de vida del software definido por primera vez por Barry Boehm en 1986. Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. Las actividades no están fijadas a ninguna prioridad, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por el bucle interior.

Básicamente consiste en una serie de ciclos que se repiten en forma de espiral, comenzando desde el centro. Se suele interpretar como que dentro de cada ciclo de la espiral se sigue un Modelo Cascada, pero no necesariamente debe ser así. El Espiral puede verse como un modelo evolutivo que conjuga la naturaleza iterativa del modelo MCP con los aspectos controlados y sistemáticos del Modelo Cascada, con el agregado de gestión de riesgo.

Ciclos o Iteraciones

En cada vuelta o iteración hay que tener en cuenta:

- **Los Objetivos:** qué necesidad debe cubrir el producto.

- **Alternativas:** las diferentes formas de conseguir los objetivos de forma exitosa, desde diferentes puntos de vista como pueden ser:
 1. **Características:** experiencia del personal, requisitos a cumplir, etc.
 2. **Formas de gestión del sistema.**
 3. **Riesgo asumido con cada alternativa.**
- **Desarrollar y Verificar:** Programar y probar el software.

Si el resultado no es el adecuado o se necesita implementar mejoras o funcionalidades:

- Se planificará los siguientes pasos y se comienza un nuevo ciclo de la espiral. La espiral tiene una forma de [caracola](#) y se dice que mantiene dos dimensiones, la radial y la angular:
 1. **Angular:** Indica el avance del proyecto del software dentro de un ciclo.
 2. **Radial:** Indica el aumento del coste del proyecto, ya que con cada nueva iteración se pasa más tiempo desarrollando.

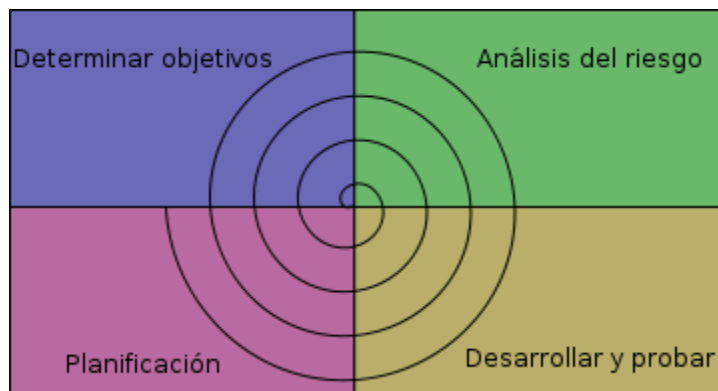
Este sistema es muy utilizado en proyectos grandes y complejos como puede ser, por ejemplo, la creación de un Sistema Operativo.

Al ser un modelo de Ciclo de Vida orientado a la gestión de riesgo se dice que uno de los aspectos fundamentales de su éxito radica en que el equipo que lo aplique tenga la necesaria experiencia y habilidad para detectar y catalogar correctamente los riesgos.

Tareas

Para cada ciclo habrá cuatro actividades:

1. **Determinar Objetivos.**
2. **Análisis del riesgo.**
3. **Desarrollar y probar.**
4. **'Planificación.'**



Determinar o fijar objetivos

Fijar también los productos definidos a obtener: requisitos, especificación, manual de usuario.
Fijar las restricciones.

Identificación de riesgos del proyecto y estrategias alternativas para evitarlos.

Hay una cosa que solo se hace una vez: planificación inicial.

Desarrollar, verificar y validar (probar)

Tareas de la actividad propia y de prueba.

Análisis de alternativas e identificación resolución de riesgos.

Dependiendo del resultado de la evaluación de los riesgos, se elige un modelo para el desarrollo, el que puede ser cualquiera de los otros existentes, como formal, evolutivo, cascada, etc. Así si 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 lo riesgos de protección son la principal consideración, un desarrollo basado en transformaciones formales podría ser el más apropiado.

Análisis del riesgo

Se lleva a cabo el estudio de las causas de las posibles amenazas y probables eventos no deseados y los daños y consecuencias que éstas puedan producir. Se evalúan alternativas. Se debe tener un prototipo antes de comenzar a desarrollar y probar.

En resumen, es para tener en cuenta los riesgos de cada uno de los ámbitos.

Mecanismos de control

La dimensión radial mide el coste.

La dimensión angular mide el grado de avance del proyecto.

Variaciones del Modelo En Espiral

Modelo en Espiral Típico de seis regiones

El modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora, a diferencia del modelo de proceso clásico que termina cuando se entrega el software.

Las 6 regiones que componen este modelo son las siguientes:

Comunicación con el cliente - Tareas necesarias para plantear la comunicación entre el desarrollador y el cliente.

Planificación - Tareas inherentes a la definición de recursos, el tiempo y otras informaciones relacionadas con el proyecto. Son todos los requerimientos.

Análisis de riesgos – Tareas para evaluar riesgos técnicos y otras informaciones relacionadas con el proyecto.

Ingeniería - Tareas para construir una o más representaciones de la aplicación.

Construcción y adaptación - Tareas requeridas para construir, probar, instalar y proporcionar soporte a los usuarios.

Evaluación del cliente - 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.³

Ventajas

El análisis del riesgo se hace de forma explícita y clara. Une los mejores elementos de los restantes modelos.

Reduce riesgos del proyecto

Incorpora objetivos de calidad

Integra el desarrollo con el mantenimiento, etc.

Además, es posible tener en cuenta mejoras y nuevos requerimientos sin romper con la metodología, ya que este ciclo de vida no es rígido ni estático.

Desventajas

Genera mucho tiempo en el desarrollo del sistema

Modelo costoso

Requiere experiencia en la identificación de riesgos

Modelo de Cascada

Es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase. Este modelo fue el primero en originarse y es la base de todos los demás modelos de ciclo de vida.

Fases:

Análisis de requisitos del software

En esta fase se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.

Es importante señalar que en esta etapa se debe consensuar todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del software de una manera.

Diseño del sistema

Descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

Es conveniente distinguir entre diseño de alto nivel o arquitectónico y diseño detallado. El primero de ellos tiene como objetivo definir la estructura de la solución (una vez que la fase de análisis ha descrito el problema) identificando grandes módulos (conjuntos de funciones que van a estar asociadas) y sus relaciones. Con ello se define la arquitectura de la solución elegida. El segundo define los algoritmos empleados y la organización del código para comenzar la implementación.

Diseño del programa

Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber qué herramientas usar en la etapa de Codificación.

Codificación del ordeñado

Es la fase en donde se implementa el código fuente, haciendo uso de prototipos así como de pruebas y ensayos para corregir errores.

Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.

Pruebas

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente y que cumple con los requisitos, antes de ser entregado al usuario final.

Verificación

Es la fase en donde el usuario final ejecuta el sistema, para ello el o los programadores ya realizaron exhaustivas pruebas para comprobar que el sistema no falle.

Mantenimiento

Una de las etapas más críticas, ya que se destina un 75 % de los recursos, es el mantenimiento del software ya que al utilizarlo como usuario final puede ser que no cumpla con todas nuestras expectativas.

Ventajas

Realiza un buen funcionamiento en equipos débiles y productos maduros, por lo que se requiere de menos capital y herramientas para hacerlo funcionar de manera óptima.

Es un modelo fácil de implementar y entender.

Está orientado a documentos.

Es un modelo conocido y utilizado con frecuencia.

Promueve una metodología de trabajo efectiva: Definir antes que diseñar, diseñar antes que codificar.

Desventajas

En la vida real, un proyecto rara vez sigue una secuencia lineal, esto crea una mala implementación del modelo, lo cual hace que lo lleve al fracaso.

El proceso de creación del software tarda mucho tiempo ya que debe pasar por el proceso de prueba y hasta que el software no esté completo no se opera. Esto es la base para que funcione bien.

Cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo.

Una etapa determinada del proyecto no se puede llevar a cabo a menos de que se haya culminado la etapa anterior.

Modelo RUP

Es un proceso de ingeniería de software, que hace una propuesta orientada por disciplinas para lograr las tareas y responsabilidades de una organización que desarrolla software.

Su meta principal es **asegurar la producción de software de alta calidad** que cumpla con las necesidades de los usuarios, con una planeación y presupuesto predecible.

Características

Dirigido por Casos de Uso:

Los casos de uso son los artefactos primarios para establecer el comportamiento deseado del sistema

Centrado en la Arquitectura:

La arquitectura es utilizada para conceptualizar, construir, administrar y evolucionar el sistema en desarrollo

Iterativo e Incremental:

Maneja una serie de entregas ejecutables.

Integra continuamente la arquitectura para producir nuevas versiones mejoradas.

Conceptualmente amplio y diverso

Enfoque orientado a objetos

En evolución continua

Adaptable

Repetible

Permite mediciones:

Estimación de costos y tiempo, nivel de avance, etc.

Fases:

Inicio (Inception)

El objetivo general de esta fase es establecer un acuerdo entre todos los interesados acerca de los objetivos del proyecto.

Es significativamente importante para el desarrollo de nuevo software, ya que se asegura de identificar los riesgos relacionados con el negocio y requerimientos.

Para proyectos de mejora de software existente, esta fase es más breve y se centra en asegurar la viabilidad de desarrollar el proyecto.

Elaboración

El objetivo en esta fase es establecer la arquitectura base del sistema para proveer bases estables para el esfuerzo de diseño e implementación en la siguiente fase.

La arquitectura debe abarcar todas las consideraciones de mayor importancia de los requerimientos y una evaluación del riesgo.

Construcción

El objetivo de la fase de construcción es clarificar los requerimientos faltantes y completar el desarrollo del sistema basados en la arquitectura base.

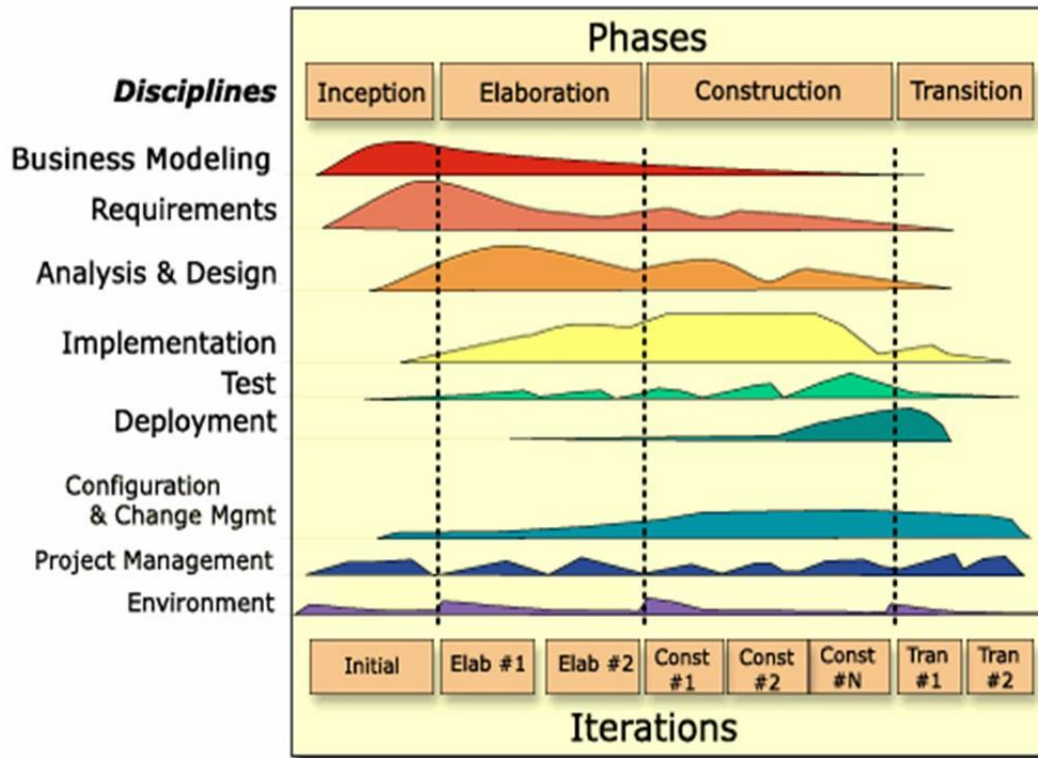
Vista de cierta forma esta fase es un proceso de manufactura, en el cual el énfasis se torna hacia la administración de recursos y control de las operaciones para optimizar costos, tiempo y calidad.

Transición

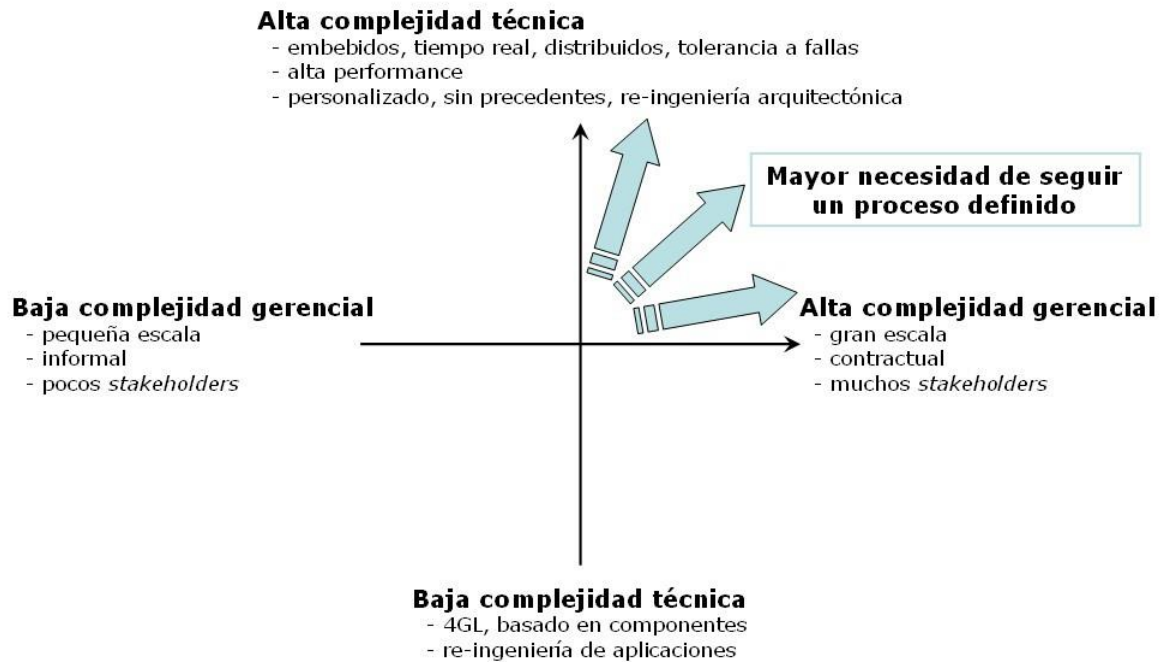
Esta fase se enfoca en asegurar que el software esté disponible para sus usuarios.

Se puede subdividir en varias iteraciones, además incluye pruebas del producto para poder hacer el entregable del mismo, así como realizar ajustes menores de acuerdo a ajuste menores propuestos por el usuario.

En este punto, la retroalimentación de los usuarios se centra en depurar el producto, configuraciones, instalación y aspectos sobre utilización.



¿Cuándo usar RUP?



Ventajas del Modelo RUP

- ☑ -Es el proceso de desarrollo más general de los existentes actualmente.
- ☑ -Es una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo).

Desventajas del Modelo RUP

- ☒ -Método pesado
- ☒ -Por el grado de complejidad puede ser no muy adecuado.
- ☒ -En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios.

CONCLUSIONES

Tomando en cuenta lo que se ha visto sobre estos temas, hay que tomar en cuenta la importancia de los sistemas de información para el manejo óptimo de la información y su uso para el éxito de una gestión correcta de una empresa, además del uso para optimización y el aprovechamiento de los sistemas informáticos para dicho éxito.

También la importancia de los diversos modelos que se pueden usar para el desarrollo, prueba, presentación, retroalimentación y finalización del proyecto, así como la satisfacción que se le debe de dar al cliente final.

También que con esto se debe de buscar la mejor opción que queramos usar a la hora de proceder con un futuro proyecto de algún cliente.

Tomar en cuenta cada una de sus ventajas y desventajas.