

Ingeniería de Software I

Conceptos de ingeniería de software

2016

Un poco de historia sobre la IS

- En los años 60, con la sofisticación creciente de los sistemas de software crecieron también las dificultades para desarrollarlos o adaptarlos. Las dificultades desbordaban los recursos técnicos de una heterogénea clase profesional.

“CRISIS DEL SOFTWARE”

- El valor estratégico del software llevó a la Organización del Tratado del Atlántico Norte (OTAN) a organizar un par de conferencias que tuvieron carácter fundacional para la Ingeniería de Software (Garmish 1968 y Roma 1969).
- El propósito de estas conferencias fue identificar la raíz de los problemas que enfrentaba la incipiente industria del software y sentar las bases de procesos sistemáticos, repetibles y confiables. En la reunión de Roma se comenzó a utilizar la expresión “Ingeniería de Software”.

Un poco de historia sobre la IS



- Era funcional – años 60
 - Se estudia cómo explotar la tecnología para hacer frente a las necesidades funcionales de las organizaciones
- Era de control – años 70
 - Aparece la necesidad de desarrollar software en tiempo, planeado y controlado. Se introduce el modelo de ciclo de vida en fases.

Era de costos – años 80

- La importancia de la productividad en el desarrollo de software se incrementa sustancialmente. Se ponen en práctica varios modelos de costos.

Era de calidad – años 90 a la actualidad

- Se intensifica la necesidad de que el producto tenga atributos que satisfagan las necesidades explícitas e implícitas del usuario: mantenibilidad, confiabilidad, eficiencia, usabilidad



¿Qué conocimientos debe tener un IS?

- El Ingeniero de Software debe tener una combinación de conocimientos científicos, metodológicos, tecnológicos y administrativos.
- El Ingeniero debe estar familiarizado con la aplicación de métodos formales: lógica, estadística, simulación y con el uso de notaciones de modelización, especificación, diseño, programación
- El Ingeniero debe poder aplicar metodologías de documentación, análisis, especificación, diseño, implementación y prueba. Debe conocer las ventajas y limitaciones de cada notación y cada técnica. Debe saber cómo y cuándo aplicarlas.

¿Qué conocimientos debe tener un IS?

- El Ingeniero debe conocer las tecnologías y productos: sistemas operativos, lenguajes, herramientas CASE, bases de datos, sistemas generadores de interfaces, bibliotecas de código.
- El Ingeniero debe conocer técnicas de administración de proyectos: planificación, análisis de riesgos, control de calidad, seguimiento de proyectos, control de subcontratistas, etc.
- En los últimos años se observa una especialización de los ingenieros de software por dominio de aplicación o por actividad

Responsabilidad profesional y ética

» La Ingeniería de Software se desarrolla en un marco económico, social y legal.

- Los IS deben aceptar responsabilidades más amplias que las responsabilidades técnicas

» No debe utilizar sus capacidad y habilidades de forma deshonesto, o de forma que deshonor la profesión.

Responsabilidad profesional y ética

»Confidencialidad

- Respetar la confidencialidad de sus empleados y clientes

»Competencia

- No falsificar el nivel de competencia y aceptar responsabilidades fuera de su capacidad

»Derechos de la propiedad intelectual

- Conocer la leyes vigentes sobre las patentes y copyright

»Uso inapropiado de las computadoras

- No debe utilizar sus habilidades técnicas para utilizar de forma inapropiada otras computadoras

»Existen diferentes organización cono ACM o IEEE que sugieren diferentes códigos de ética a respetar

Técnicas de elicitación

Técnicas de elicitación

1. Muestreo de la documentación, los formularios y los datos existentes.
2. Investigación y visitas al lugar.
3. Observación del ambiente de Trabajo.
4. Cuestionarios.
5. Entrevistas.
6. Planeación conjunta de Requerimientos (JRP o JAD).
7. Lluvia de Ideas - Brainstorming .

Planeación Conjunta de Requerimientos (JRP)

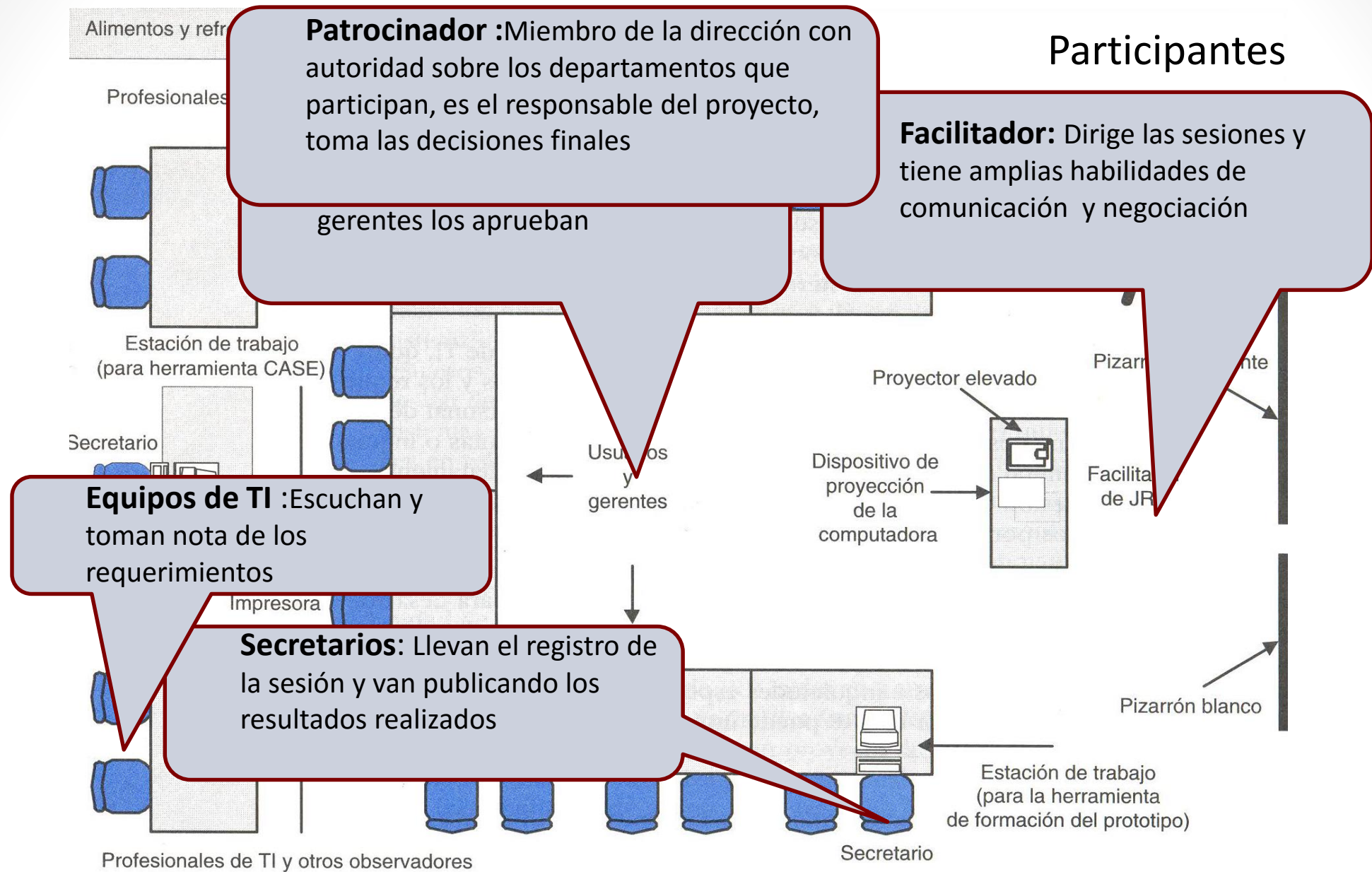
Proceso mediante el cual se conducen reuniones de grupo altamente estructurados con el propósito de analizar problemas y definir requerimientos

- Requiere de extenso entrenamiento
- Reduce el tiempo de exploración de requisitos
- Amplia participación de los integrantes
- Se trabaja sobre lo que se va generando
- Alguna bibliografía la menciona como JAD
- (Joint Application Design)



Planeación Conjunta de Requerimientos (JRP)

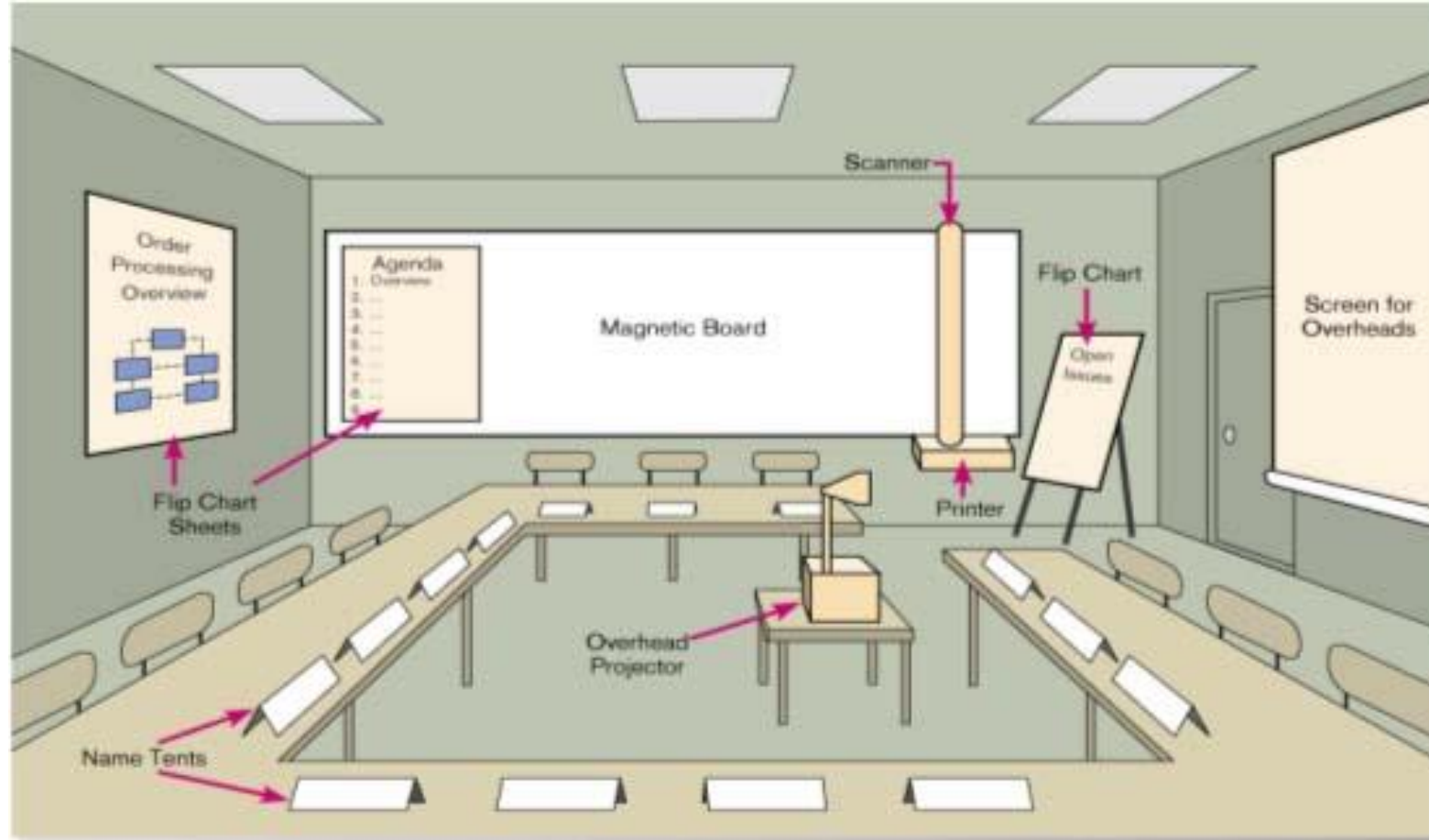
- Ventajas
 - Ahorro de tiempo
 - Usuarios involucrados
 - Desarrollos creativos
- Desventajas
 - Es difícil organizar los horarios de los involucrados
 - Es complejo encontrar un grupo de participantes integrados y organizados



Planeación Conjunta de Requerimientos (JRP)

- Cómo planear las sesiones de JRP
 - Selección de una ubicación para las sesiones de JRP
 - Selección de los participantes
 - Preparar la agenda

Espacio para el JRP



Planeación Conjunta de Requerimientos (JRP)

Beneficios del JRP

- JRP involucra activamente a los usuarios y la gerencia en el proyecto de desarrollo
- JRP reduce el tiempo de la etapa de requerimientos
- Si se incorporan prototipos, los mismos ya confirman el diseño del sistema

Lluvia De Ideas (Brainstorming)



Lluvia De Ideas (Brainstorming)

- Técnica para generar ideas al alentar a los participantes para que ofrezcan tantas ideas como sea posible en un corto tiempo sin ningún análisis hasta que se hayan agotado las ideas.
- Se promueve el desarrollo de ideas creativas para obtener soluciones.
- Se realizan reuniones del equipo involucrado en la resolución del problema, conducidas por un director.

Lluvia De Ideas (Brainstorming)

- Los principios en que se basa esta técnica son:
 - “Cuantas más ideas se sugieren, mejores resultados se conseguirán”.
 - La producción de ideas en grupos puede ser más efectiva que la individual.
 - Las ideas de una persona pueden hacer que aparezcan otras por “contagio”.
 - A veces las mejores ideas aparecen tarde.
 - Es mejor elegir sobre una variedad de soluciones.

Lluvia De Ideas (Brainstorming)

- Incluye una serie de fases de aplicación:
 - Descubrir hechos, Producir ideas, Descubrir soluciones
- Clave para resolver la falta de consenso entre usuarios
- Es útil combinarlo con la toma de decisiones
- Ayuda a entender el dominio del problema
- Encara la dificultad del usuario para transmitir
- Ayuda a entender: al usuario y al analista



Bibliografía

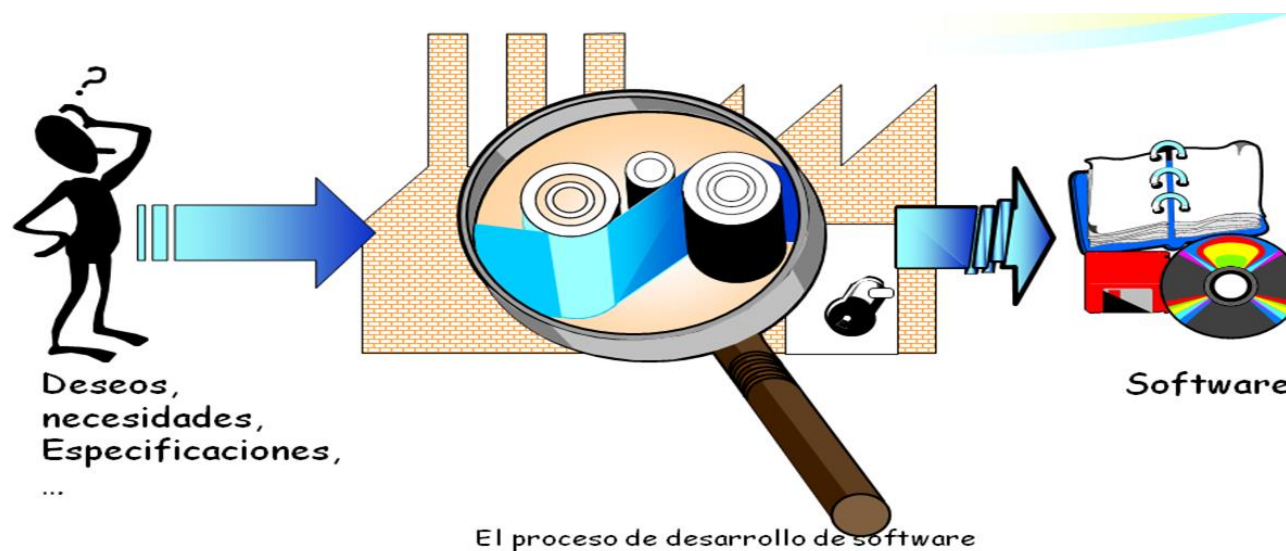
- Libros consultados para conceptos de IS y técnicas de comunicación
 - Whitten-Bentley, Análisis de Sistemas Diseño y Métodos, Capítulo 5, Mc Graw Hill 2008
 - Kendall y Kendall, Análisis y diseño de Sistemas, Capítulo 4, Pearson Prentice Hall 2005
 - Pfleeger, Capítulo 4 , Ingeniería de Software, Pearson Prentice Hall 2002

Ingeniería de Software I

Requerimientos I -

¿Qué es un proceso de software?

- Es un conjunto de actividades y resultados asociados que producen un producto de software.



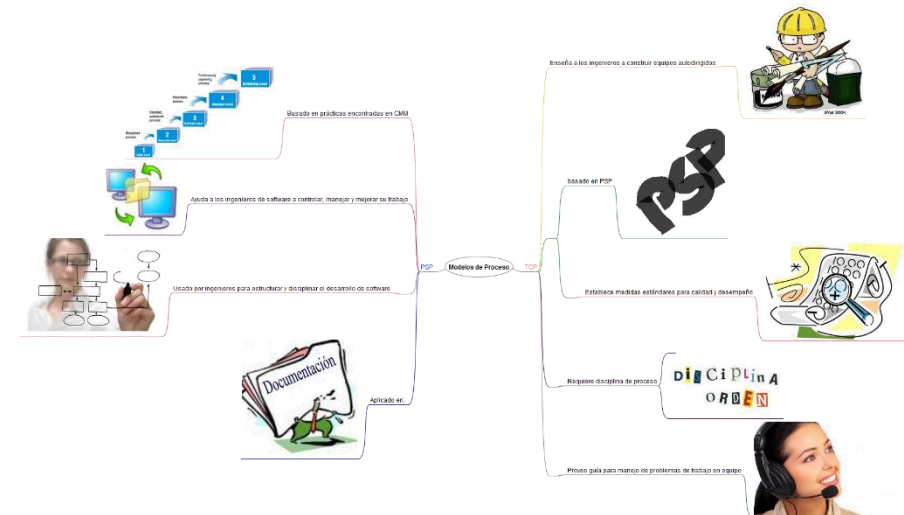
¿Qué es un proceso de software?

- Actividades fundamentales de los procesos:
 - Especificación del software
 - Desarrollo del software
 - Validación del software
 - Evolución del software
- Los IS son los responsables de realizar estas actividades.



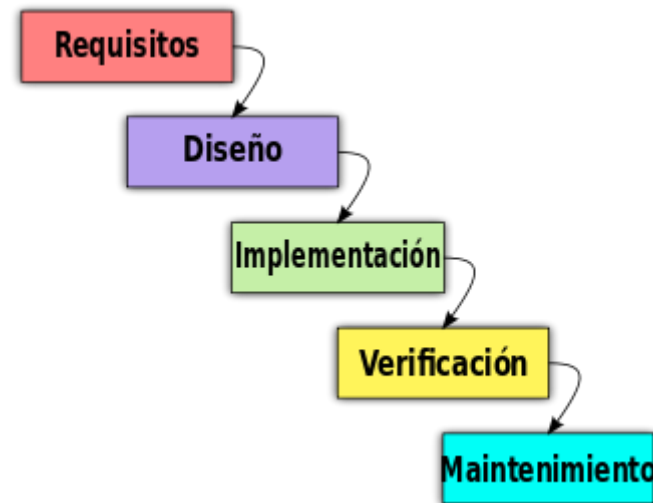
¿Qué es un modelo de proceso de software?

- Es una representación simplificada de un proceso de software que presenta una visión de ese proceso.
- Estos modelos pueden incluir actividades que son partes de los procesos y productos de software, y el papel de las personas involucradas.



¿Qué es un modelo de proceso de software?

- La mayoría de los modelos de proceso de software se basan en uno de los siguientes modelos generales o paradigmas
 - Modelo en cascada: Las etapas se representan cayendo en cascada. Cada etapa de desarrollo se debe completar antes que comience la siguiente



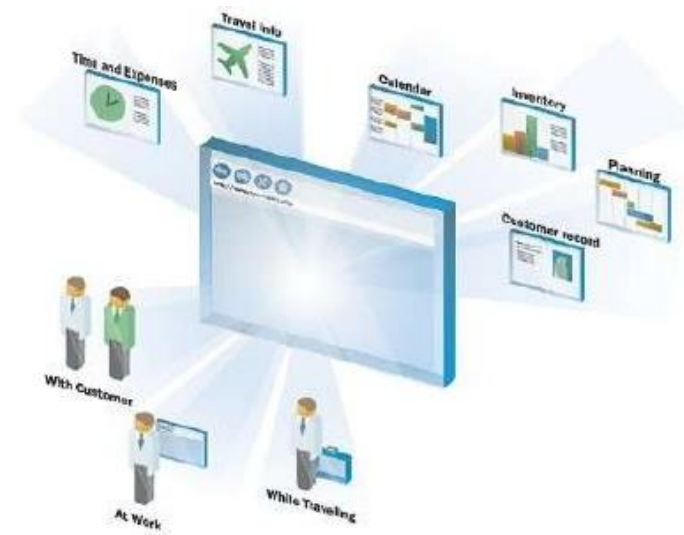
¿Qué es un modelo de proceso de software?

- Desarrollo iterativo : Un sistema inicial se desarrolla rápidamente a partir de una especificación abstracta. Éste se refina basándose en las peticiones del cliente.



¿Qué es un modelo de proceso de software?

- IS basada en componentes: Esta técnica supone que las partes ya existen. El proceso se enfoca en la integración de las partes.



- Más adelante se verán los modelos de procesos más detalladamente.

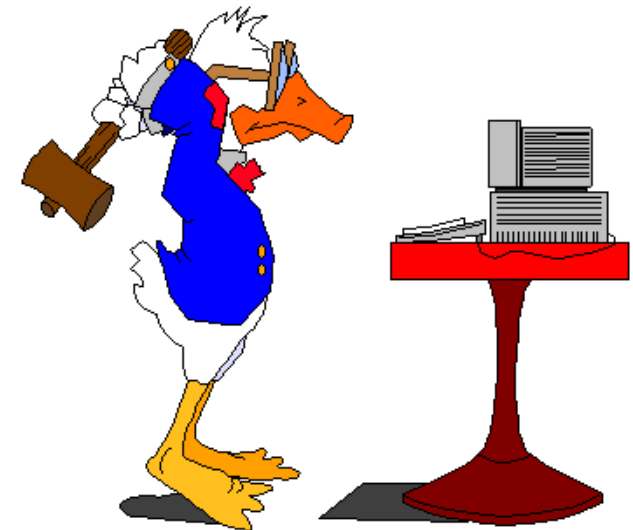
Requerimientos

- Un requerimiento (o requisito) es una característica del sistema o una descripción de algo que el sistema es capaz de hacer con el objeto de satisfacer el propósito del sistema



Requerimientos

- Impacto de los errores en la etapa de requerimientos
 - El software resultante puede no satisfacer a los usuarios
 - Las interpretaciones múltiples de los requerimientos pueden causar desacuerdos entre clientes y desarrolladores
 - Puede gastarse tiempo y dinero construyendo el sistema erróneo



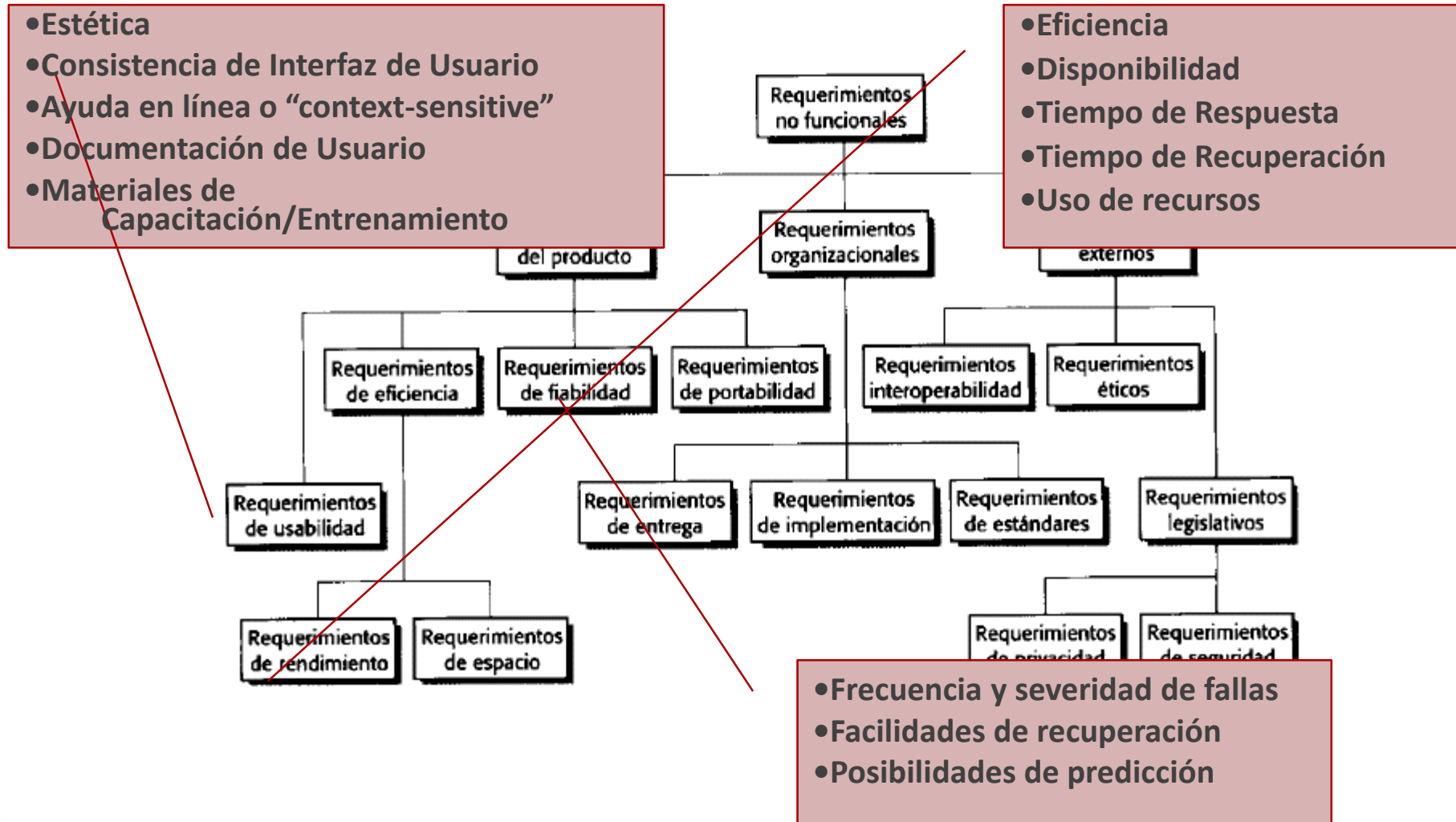
Requerimientos

- Tipos de requerimientos
 - Requerimientos funcionales
 - Describen una interacción entre el sistema y su ambiente. Cómo debe comportarse el sistema ante determinado estímulo.
 - Describen lo que el sistema debe hacer, o incluso cómo NO debe comportarse.
 - Describen con detalle la funcionalidad del mismo.
 - Son independientes de la implementación de la solución.
 - Se pueden expresar de distintas formas
 - Requerimientos no funcionales
 - Describen una restricción sobre el sistema que limita nuestras elecciones en la construcción de una solución al problema.

Requerimientos

- Tipos de requerimientos
 - Requerimientos no funcionales
 - Requerimientos del producto
 - Especifican el comportamiento del producto (usabilidad, eficiencia, rendimiento, espacio, fiabilidad, portabilidad).
 - Requerimientos organizacionales
 - Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador (entrega, implementación, estándares).
 - Requerimientos externos
 - Interoperabilidad, legales, privacidad, seguridad, éticos.

Requerimientos No Funcionales



Requerimientos- Tipos

- Otras Clasificaciones
 - Requerimientos del dominio
 - Reflejan las características y restricciones del dominio de la aplicación del sistema. Pueden ser funcionales o no funcionales y pueden restringir a los anteriores. Como se especializan en el dominio son complicados de interpretar.
 - Requerimientos por Prioridad
 - Que deben ser absolutamente satisfechos
 - Que son deseables pero no indispensables
 - Que son posibles, pero que podrían eliminarse

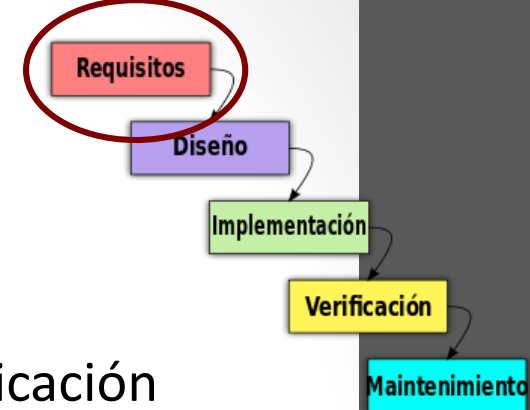
Requerimientos - Tipos

- Otras Clasificaciones
 - Requerimientos del Usuario
 - Son declaraciones en lenguaje natural y en diagramas de los servicios que se espera que el sistema provea y de las restricciones bajo las cuales debe operar.
 - Pueden surgir problemas por falta de claridad, confusión de requerimientos, conjunción de requerimientos.
 - Requerimientos del Sistema
 - Establecen con detalle los servicios y restricciones del sistema.
 - Es difícil excluir toda la información de diseño (arquitectura inicial, interoperabilidad con sistemas existentes, etc.)

Ingeniería de Requerimientos

- La ingeniería de requerimientos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en donde se describen las funciones que realizará el sistema.

Ingeniería de requerimientos es el proceso por el cual se transforman los requerimientos declarados por los clientes, ya sean hablados o escritos, a especificaciones precisas, no ambiguas, consistentes y completas del comportamiento del sistema, incluyendo funciones, interfaces, rendimiento y limitaciones”



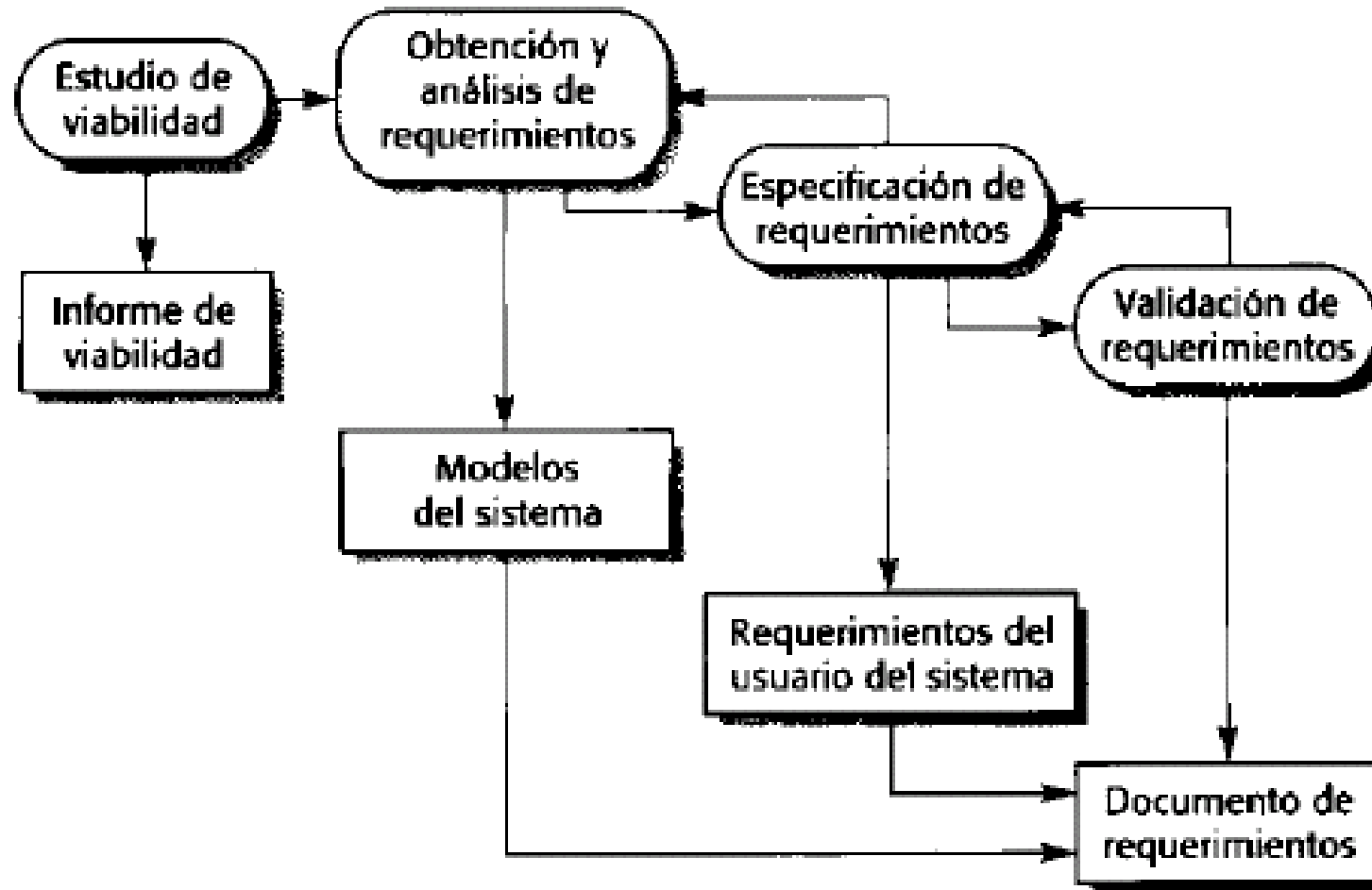
Ingeniería de Requerimientos

- También es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos.”
- “Ingeniería de requerimientos” es un enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto”

Ingeniería de Requerimientos

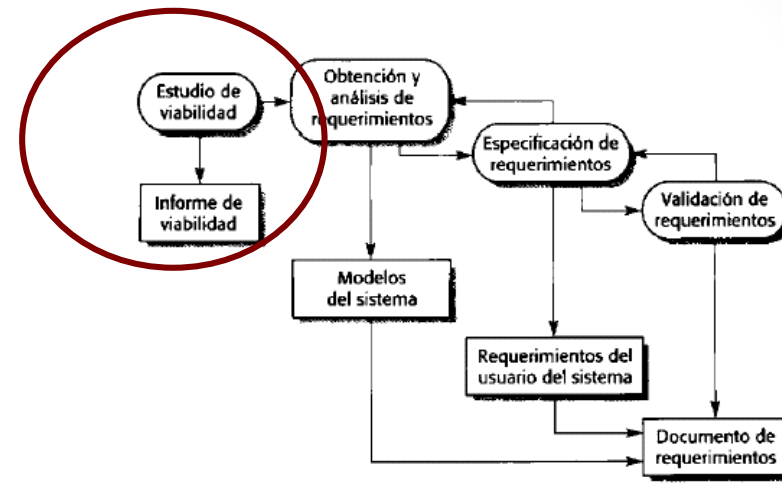
- Importancia
 - Permite gestionar las necesidades del proyecto en forma estructurada
 - Mejora la capacidad de predecir cronogramas de proyectos
 - Disminuye los costos y retrasos del proyecto
 - Mejora la calidad del software
 - Mejora la comunicación entre equipos
 - Evita rechazos de usuarios finales.

Ingeniería de Requerimientos



Ingeniería de Requerimientos

Estudio de Viabilidad



Ingeniería de Requerimientos

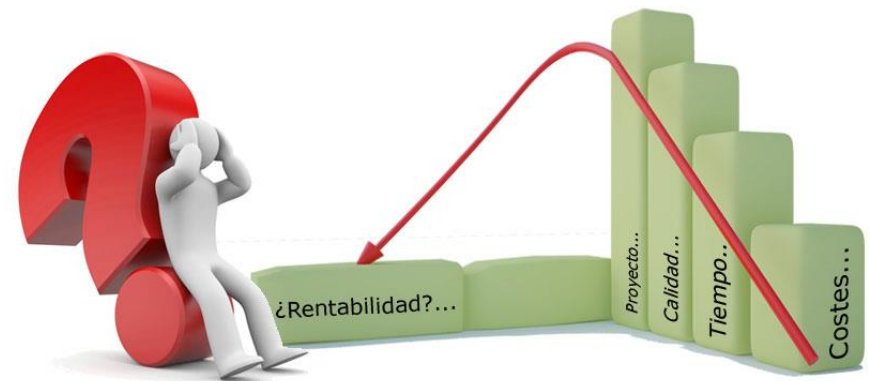
Estudio de Viabilidad

- A partir de una descripción resumida del sistema se elabora un informe que recomienda la conveniencia o no de realizar el proceso de desarrollo
- Responde a las siguientes preguntas:
 - ¿El sistema contribuye a los objetivos generales de la organización?(Si no contribuye, entonces no tiene un valor real en el negocio)
 - ¿El sistema se puede implementar con la tecnología actual?
 - ¿El sistema se puede implementar con las restricciones de costo y tiempo?
 - ¿El sistema puede integrarse a otros que existen en la organización?

Ingeniería de Requerimientos

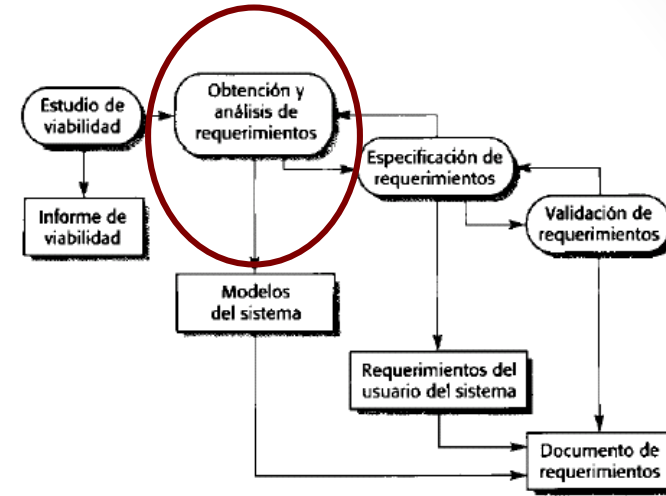
Estudio de Viabilidad

- Una vez que se ha recopilado toda la información necesaria para contestar las preguntas anteriores se debería hablar con las fuentes de información para responder nuevas preguntas y luego se redacta el informe, donde debería hacerse una recomendación sobre si debe continuar o no el desarrollo.



Ingeniería de Requerimientos

Obtención y análisis de requerimientos o
Elicitación de requerimientos



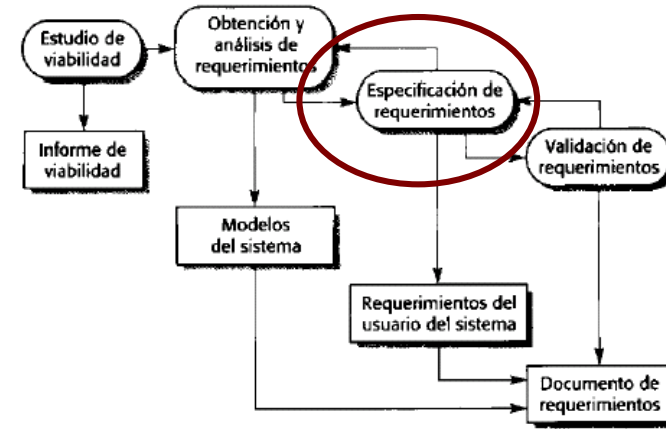
Ingeniería de Requerimientos

Obtención y análisis de requerimientos

- Propiedades de los Requerimientos
 - Necesario: Su omisión provoca una deficiencia.
 - Conciso: Fácil de leer y entender
 - Completo: No necesita ampliarse
 - Consistente: No contradictorio con otro
 - No ambiguo: Tiene una sola implementación
 - Verificable: Puede testearse a través de inspecciones, pruebas, etc.

Ingeniería de Requerimientos

Especificación de requerimientos



Ingeniería de Requerimientos

Especificación de requerimientos

- Objetivos
 - Permitir que los desarrolladores expliquen cómo han entendido lo que el cliente pretende del sistema
 - Indicar a los diseñadores qué funcionalidad y características va a tener el sistema resultante
 - Indicar al equipo de pruebas qué demostraciones llevar a cabo para convencer al cliente de que el sistema que se le entrega es lo que había pedido.

Ingeniería de Requerimientos

Especificación de requerimientos

- Correcta
- No ambigua
- Completa
- Verificable
- Consistente
- Comprensible por los consumidores
- Modificable
- Rastreable
- Independiente del diseño
- Anotada
- Concisa
- Organizada
- Utilizable en operación y mantenimiento

Ingeniería de Requerimientos

Especificación de requerimientos

- Documento de definición de requerimientos
 - Listado completo de todas las cosas que el cliente espera que haga el sistema propuesto
- Documento de especificación de requerimientos
 - Definición en términos técnicos
- Documento de especificación de requerimientos de Software IEEE Std. 830-1998 (SRS)
 - Objetivo:
 - Brindar una colección de buenas prácticas para escribir especificaciones de requerimientos de software (SRS).
 - Se describen los contenidos y las cualidades de una buena especificación de requerimientos.

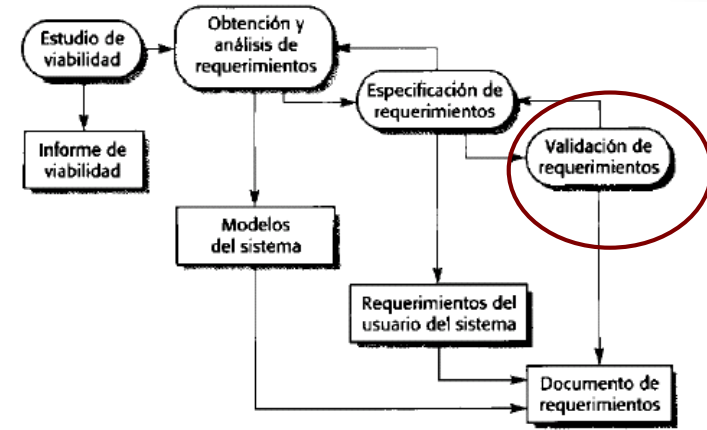
Ingeniería de Requerimientos

Especificación de requerimientos

- Aspectos básicos de una especificación de requerimientos
 - Funcionalidad
 - ¿Qué debe hacer el software?
 - Interfaces Externas
 - ¿Cómo interactuará el software con el medio externo (gente, hardware, otro software)?
 - Rendimiento
 - Velocidad, disponibilidad, tiempo de respuesta, etc.
 - Atributos
 - Portabilidad, seguridad, mantenibilidad, eficiencia
 - Restricciones de Diseño
 - Estándares requeridos, lenguaje, límite de recursos, etc.

Ingeniería de Requerimientos

Validación de requerimientos



Ingeniería de Requerimientos

Validación de requerimientos

- Es el proceso de certificar la corrección del modelo de requerimientos contra las intenciones del usuario.
- Trata de mostrar que los requerimientos definidos son los que estipula el sistema. Se describe el ambiente en el que debe operar el sistema.
- Es importante, porque los errores en los requerimientos pueden conducir a grandes costos si se descubren más tarde

Ingeniería de Requerimientos

Validación de requerimientos

- Validación
 - Al final del desarrollo evaluar el software para asegurar que el software cumple los requerimientos
- Verificación
 - Determinar si un producto de software de una fase cumple los requerimientos de la fase anterior
- Sobre estas definiciones:
 - la validación sólo se puede hacer con la activa participación del usuario
 - validación: hacer el software correcto
 - verificación: hacer el software correctamente

Ingeniería de Requerimientos

Validación de requerimientos

- ¿Es suficiente validar después del desarrollo del software?
 - La evidencia estadística dice que NO
 - Cuanto más tarde se detecta, más cuesta corregir (Boehm)
 - Bola de nieve de defectos
 - Validar en la fase de especificación de requerimientos puede ayudar a evitar costosas correcciones después del desarrollo
- ¿Contra qué se verifican los requerimientos?
 - No existen “los requerimientos de los requerimientos”
 - No puede probarse formalmente que un Modelo de Requerimientos es correcto. Puede alcanzarse una convicción de que la solución especificada en el modelo de requerimientos es el correcto para el usuario.

Ingeniería de Requerimientos

Validación de requerimientos

- Comprenden
 - Verificaciones de validez (para todos los usuarios)
 - Verificaciones de consistencia (sin contradicciones)
 - Verificaciones de completitud (todos los requerimientos)
 - Verificaciones de realismo (se pueden implementar)
 - Verificabilidad (se puede diseñar conjunto de pruebas)

Ingeniería de Requerimientos

Validación de requerimientos

- Técnicas de validación

Pueden ser manuales o automatizadas

- Revisiones de requerimientos (formales o informales)
 - Informales
 - Los desarrolladores deben tratar los requerimientos con tantos stakeholders como sea posible.
 - Formal
 - El equipo de desarrollo debe conducir al cliente, explicándole las implicaciones de cada requerimiento
- Antes de una revisión formal, es conveniente realizar una revisión informal.
- Construcción de prototipos
- Generación de casos de prueba

Técnicas de Especificación de Requerimientos

Técnicas de Especificación de Requerimientos

- Estáticas
 - Se describe el sistema a través de las entidades u objetos, sus atributos y sus relaciones con otros. No describe como las relaciones cambian con el tiempo.
 - Cuando el tiempo no es un factor mayor en la operación del sistema, es una descripción útil y adecuada.
 - Ejemplos: Referencia indirecta, Relaciones de recurrencia, Definición axiomática, Expresiones regulares, Abstracciones de datos, entre otras.

Técnicas de Especificación de Requerimientos

- Dinámicas
 - Se considera un sistema en función de los cambios que ocurren a lo largo del tiempo.
 - Se considera que el sistema está en un estado particular hasta que un estímulo lo obliga a cambiar su estado.
 - Ejemplos: Tablas de decisión, Diagramas de transición de estados, Tablas de transición de estados, Diagramas de persianas, Diagramas de transición extendidos, Redes de Petri, entre otras.

Técnicas de Especificación de Requerimientos Estáticas

- Descripción del sistema con una referencia indirecta al problema y su solución.
- Se define "QUÉ" se hace, no "CÓMO".
- Ejemplo: sistema que resuelva k ecuaciones con n incógnitas => NO se declara el método de resolución, puede NO existir la solución.
- Relaciones de recurrencia
 - Descripción del sistema mediante una función que define su valor en función de términos anteriores.
 - Ejemplo: Expresar la serie de Fibonacci
 - $F(0) = 1$ $F(1) = 1$ $F(n+1) = F(n) + F(n-1)$

- Referencia indirecta (ecuaciones implícitas)

Técnicas de Especificación de Requerimientos Estáticas

- Definición axiomática
 - Se definen las propiedades básicas de un sistema a través de operadores y axiomas (debe ser un conjunto completo y consistente)
 - Se generan teoremas a través del comportamiento del sistema y se demuestran
 - Ejemplos: Sistemas expertos, Definición de TADs, etc.

Técnicas de Especificación de Requerimientos Estáticas

- Expresiones regulares
 - Se define un alfabeto y las combinaciones permitidas. Cuando un sistema procesa un conjunto de cadenas de datos, permite definir las cadenas de datos aceptables
 - Alfabeto
 - ÁTOMOS: (símbolos básicos) a, b, c .
 - ALTERNACIÓN: $(a | b) = \{a, b\}$
 - COMPOSICIÓN: $(ab) = \{ab\}$
 - ITERACIÓN: $(a)^* = \{e, a, aa, \dots\}$ $(a)^+ = \{a, aa, \dots\}$
 - Se definen las combinaciones válidas
 - $(a(b | c)) = \{ab, ac\}$
 - $(a(b | c))^+ = \{ab, ac, abac, acab, \dots\}$

Técnicas de Especificación de Requerimientos Estáticas

- Abstracciones de datos
 - Para aquellos sistemas en los que los datos determinan las clases de acciones que se realizan (importa para qué son).
 - Se categorizan los datos y se agrupan los semejantes.
 - El diccionario contiene los TIPOS DE DATOS (clases) y los DATOS (objetos).
 - Se organizan de tal manera de aprovechar las características compartidas.

Técnicas de Especificación de Requerimientos Dinámicas

- Tablas de Decisión
 - Es una herramienta que permite presentar de forma concisa las reglas lógicas que hay que utilizar para decidir acciones a ejecutar en función de las condiciones y la lógica de decisión de un problema específico.
- Describe el sistema como un conjunto de:
 - Posibles CONDICIONES satisfechas por el sistema en un momento dado
 - REGLAS para reaccionar ante los estímulos que ocurren cuando se reúnen determinados conjuntos de condiciones y
 - ACCIONES a ser tomadas como un resultado.

Técnicas de Especificación de Requerimientos Dinámicas

- Tablas de Decisión
 - Construiremos las tablas con:
 - condiciones simples y acciones simples
 - Las condiciones toman sólo valores Verdadero o Falso
 - Hay 2^N Reglas donde N es el nro. de condiciones

	REGLA1	REGLA2
COND1			
COND2			
.....			
ACCION1			
ACCION2			
.....			

Técnicas de Especificación de Requerimientos Dinámicas- **Tablas de Decisión**

- Como se llena la tabla?

A partir de un enunciado se debe:

1. Identificar las condiciones y cuales las acciones.
2. Completar la tabla teniendo en cuenta:
 - a) Si hay condiciones que son opuestas, debe colocarse una de ellas porque por la negativa se obtendrá la otra.
 - b) Las condiciones deben ser atómicas.
3. Se construye las reglas

Técnicas de Especificación de Requerimientos Dinámicas- **Tablas de Decisión**

- **Tablas de Decisión**
 - Modelizar el problema de remisión de mercadería con las siguientes consideraciones:
 - Si el comprador no es cliente se imprime un mensaje de aviso y no se remite.
 - Si no hay stock y el comprador es cliente no se remite.
 - Si hay stock y el comprador es cliente se remite

Técnicas de Especificación de Requerimientos Dinámicas- **Tablas de Decisión**

- **Tablas de Decisión**
 - Modelizar el problema de remisión de mercadería con las siguientes consideraciones:
 - Si el comprador no es cliente se imprime un mensaje de aviso y no se remite.
 - Si no hay stock y el comprador es cliente no se remite.
 - Si hay stock y el comprador es cliente se remite

Técnicas de Especificación de Requerimientos Dinámicas

- Si el comprador no es cliente se imprime un mensaje de aviso y no se remite.
- Si no hay stock y el comprador es cliente no se remite.
- Si hay stock y el comprador es cliente se remite

Condiciones	{		Reglas			
		Es cliente	V	V	F	F
		Hay stock	V	F	V	F
Acciones	{	Imprime mensaje de aviso			X	X
		Se remite	X			
		No se remite		X	X	X

Técnicas de Especificación de Requerimientos Dinámicas

- Tablas de Decisión
 - Especificaciones completas
 - Aquellas que determinan acciones (una o varias) para todas las reglas posibles.
 - Especificaciones redundantes
 - Aquellas que marcan para reglas que determinan las mismas condiciones acciones iguales.
 - Especificaciones contradictorias
 - Aquellas que especifican para reglas que determinan las mismas condiciones acciones distintas.

Técnicas de Especificación de Requerimientos Dinámicos

- Tablas de Decisión
 - Redundancia y Contradicción

	Reglas						
C1	V	V	F	F
C2	V	V	.	.	.	V	V
C3	V	F	F	F
A1			X	X
A2	X				.		
A3		X	.	.		X	X

Redundante

	Reglas						
C1	V	V	F	F
C2	V	V	.	.	.	V	V
C3	V	F	F	F
A1				X
A2	X				.	X	
A3		X	.	.		X	

Contradictoria

Técnicas de Especificación de Requerimientos Dinámicas

- Tablas de Decisión
 - Reducción de Complejidad (Redundancia)
 - Combine las reglas en donde sea evidente que una alternativa no representa una diferencia en el resultado.
 - El guión [—] significa que la condición 2 puede ser S o N, y que aún así se realizará la acción.

Condición 1:	S	S
Condición 2	S	N
Acción 1	X	X

Condición 1:	S
Condición 2	—
Acción 1	X

Técnicas de Especificación de Requerimientos Dinámicos

- Tablas de Decisión
 - Reducción de Complejidad (Redundancia)
 - Algebra de bool

	Reglas			
Es cliente	V	V	F	F
Hay stock	V	F	V	F
Imprime mensaje de aviso			X	X
Se remite	X			
No se remite		X	X	X

Reglas		
V	V	F
V	F	—
		X
X		
	X	X

Bibliografía

- Libros consultados para técnicas de especificación de requerimientos
 - Pfleeger, Capítulo 4 , Ingeniería de Software, Pearson Prentice Hall 2002
 - Sommerville Ian, Capítulos 6 y 7, Ingeniería de software, Addison Wesley 2005