

¿Que es un stakeholder?

El término stakeholder se utiliza para referirse a cualquier persona o grupo que se vera afectado por el sistema, directa o indirectamente.

Entre los stakeholders se encuentran:

- *Usuarios finales*
- *Ingenieros*
- *Gerentes*
- *Expertos del dominio*

Puntos de Vista

Se pueden utilizar como una forma de clasificar los stakeholders.

Existen tres tipos genéricos de puntos de vista:

- *Punto de vista de los interactuadores: representan a las personas u otros sistemas que interactúan directamente con el sistema. Pueden influir en los requerimientos del sistema de algún modo.*
- *Punto de vista indirecto: representan a los stakeholders que no utilizan el sistema ellos mismos pero que influyen en los requerimientos de algún modo.*
- *Punto de vista del dominio: representan las características y restricciones del dominio que influyen en los requerimientos del sistema.*
- *Su identificación puede ser difícil.*

Los mas específicos son:

1. *Los proveedores de servicios al sistema, los receptores de servicios del sistema*
2. *Los sistemas que deben interactuar*
3. *Las regulaciones y estándares a aplicar*
4. *Las fuentes de requerimientos*
5. *Los puntos de vista de las personas que lo van a desarrollar, administrar y mantener.*
6. *Puntos de vista del marketing y otros que generan requerimientos sobre las características del sistema*

Describe el proceso de elicitation. ¿Que es un requerimiento volatil y uno duradero?

Elicitación de Requisitos

- *Es el proceso de adquirir (“eliciting”) [sonsacar] todo el conocimiento relevante necesario para producir un modelo de los requerimientos de un dominio de problema.*

Objetivos:

- *Conocer el dominio del problema para poder comunicarse con clientes y usuarios y entender sus necesidades.*
- *Conocer el sistema actual (manual o informatizado).*
- *Identificar las necesidades, tanto explícitas como implícitas, de clientes y usuarios y sus expectativas sobre el sistema a desarrollar.*
- *La elicitación de requisitos es una actividad principalmente de carácter social, mucho más que tecnológico.*
- *Los problemas que se plantean son por tanto de naturaleza psicológica y social, más que técnicos.*
- *Nota: Requisitos = Requerimientos.*

Problemas de comunicación

- *Dificultad para expresar claramente las necesidades.*
- *No ser conscientes de sus propias necesidades.*
- *No entender cómo la tecnología puede ayudar.*
- *Miedo a parecer incompetentes por ignorancia tecnológica.*
- *No tomar decisiones por no poder prever las consecuencias, no entender las alternativas o no tener una visión global.*

¿quién tiene estas dificultades? Claramente el cliente papa.

- *Problemas de comunicación*
- *Cultura y vocabulario diferentes.*
- *Intereses distintos en el sistema a desarrollar.*
- *Medios de comunicación inadecuados (diagramas que no entienden los clientes y usuarios).*
- *Conflictos personales o políticos.*

¿quién tiene estas dificultades ? Supongo que ambos vieja.

•Limitaciones cognitivas (del desarrollador)

- *No conocer el dominio del problema.*
- *Hacer suposiciones sobre el dominio del problema.*
- *Hacer suposiciones sobre aspectos tecnológicos.*
- *Hacer simplificaciones excesivas.*

•Conducta humana

- *Conflictos y ambigüedades en los roles de los participantes.*
- *Pasividad de clientes, usuarios o ingenieros de requisitos.*
- *Temor a que el nuevo sistema lo deje sin trabajo.*

•Técnicos

- *Complejidad del dominio del problema.*
- *Complejidad de los requisitos.*
- *Múltiples fuentes de requisitos.*
- *Fuentes de información poco claras.*

Técnicas de elicitación

- *Muestreo de la documentación, las formas y los datos existentes.*
- *Investigación y visitas al lugar.*
- *Observación del ambiente de Trabajo.*
- *Cuestionarios.*
- *Entrevistas.*
- *Planeación conjunta de Requerimientos (JRP o JAD).*
- *Lluvia de Ideas – Brainstorming.*

Defina proceso y modelo de proceso. Características de un modelo de proceso.

¿Qué es un proceso de software?

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

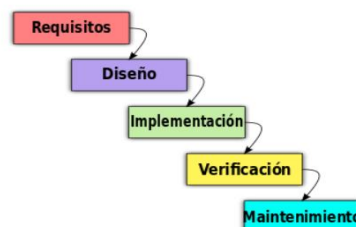
Además de las actividades, los procesos deben incluir:

- *Productos, que son los resultados de una actividad del proceso.*
- *Roles, que reflejan las responsabilidades de la gente que interviene en el proceso.*

- *Precondiciones y postcondiciones, que son declaraciones válidas antes y después de que se realice una actividad.*
- *No hay un proceso ideal, la mayoría de las organizaciones adoptan sus propios procesos de desarrollo de software.*
- *Para algunos sistemas críticos, se requiere un proceso estructurado, para otros uno menos formal y flexible.*

¿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.*
- *La mayoría de los modelos de proceso de software se basan en uno de los siguientes modelos generales o paradigmas*
- **Modelo en cascada:** *Representa las actividades anteriores y las representa como fases de proceso separadas.*



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



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

(MIRAR TEORIA DE MODELO DE PROCESOS)

¿Que es prueba de fiabilidad?

Wikipedia: En el campo de la psicología, la educación y la investigación social, la **fiabilidad** (también llamada técnicamente **confiabilidad**).

(Sommerville)

La fiabilidad de un sistema es la probabilidad de que el sistema funcione correctamente tal y como se ha especificado.

1. Fiabilidad. La probabilidad de que se tengan operaciones libres de caídas durante un tiempo definido en un entorno dado para un propósito específico.

2. Fiabilidad: Informalmente, la fiabilidad de un sistema es la probabilidad de que, durante un determinado periodo de tiempo, el sistema funcione correctamente tal y como espera el usuario

La fiabilidad se relaciona con la probabilidad de que se produzca un error en el momento de utilizar el sistema. Un programa puede contener defectos conocidos, pero aún puede considerarse como fiable por sus usuarios. Éstos pueden no usar nunca las características del sistema que están afectadas por esos defectos.

End (Sommerville).

(SlideShare- Confiabilidad)

Consiste en probar una aplicación para descubrir y eliminar errores antes de que se implemente el sistema.

Como existen infinitas combinaciones distintas a lo largo de una aplicación, no es muy probable que se encuentren todos los errores de una aplicación compleja.

Pero puede probar las situaciones más probables bajo condiciones normales de uso.

Con tiempo suficiente se pueden realizar pruebas más complicadas para detectar defectos menos evidentes.



¿Que es la calidad? Nombrar los modelos que conozca. De los modelos vistos en la materia cuales son los que se adaptan más fácilmente a los cambios.

- Calidad es un concepto manejado con bastante frecuencia en la actualidad, pero a su vez, su significado es percibido de distintas maneras.
- Al hablar de bienes y/o servicios de calidad, la gente se refiere normalmente a bienes de lujo o excelentes con precios elevados.
- Su significado sigue siendo ambiguo y muchas veces su uso depende de lo que cada uno entiende por calidad, por lo cual es importante comenzar a unificar su definición.

CROSBY: conformidad de los requerimientos

W. EDWARDS DEMING: Calidad en términos de la satisfacción del cliente

ARMAND V. FEIGENBAUM: Calidad como una relación directa entre los productos y servicios, y las necesidades del cliente.

KAORU ISHIKAWA: Establece que los requerimientos y necesidades de los clientes van cambiando lo que conlleva a una definición cambiante.

Conclusión...

- Se puede ver que coinciden en “conformar requerimientos del producto o servicio”, “lograr la satisfacción del cliente” y las relaciones entre estos conceptos.
- Pero la evaluación de los mismos, continúa dependiendo de la evaluación de sus características particulares, de manera subjetiva.
- En consecuencia lo más importante es definir claramente las características que nos interesa evaluar y su forma de evaluación.

Definiciones de Calidad

- * Capacidad de un producto o servicio para servir satisfactoriamente a los propósitos del usuario mediante su utilización.
- * Conformidad con los requisitos explícitos e implícitos de un cliente.
- * Ausencia de defectos e imperfecciones.

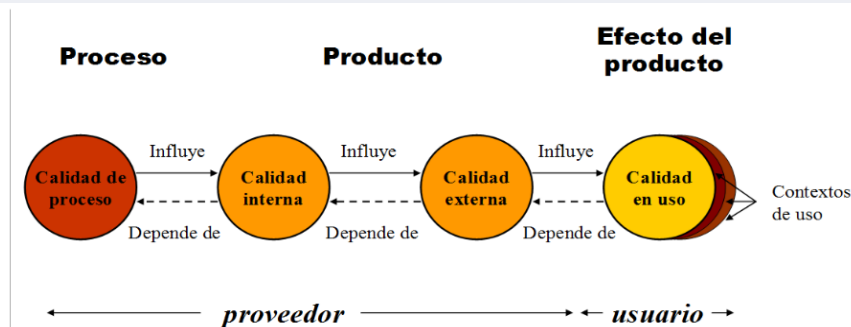
Modelos de calidad:

La calidad de un producto de software debe evaluarse usando un modelo de calidad que tiene en cuenta criterios para satisfacer las necesidades de los desarrolladores, mantenedores, adquisidores y usuarios finales.

- Los modelos de calidad pueden ser utilizados para construir mejores productos y asegurar su calidad.
- Se han desarrollado varios modelos de calidad para diferentes productos y procesos software.

Modelos de Calidad del Producto:

- ISO/IEC 9126: Tecnologías de la Información – Calidad de los productos software es un estándar internacional el cual clasifica la calidad del software en un conjunto estructurado de características de la siguiente manera.



*Mas recientemente ha aparecido la norma ISO 25000

Software Engineering -- Software product Quality Requirements and Evaluation (SQuaRE)

*SQuaRE cubre tres procesos de calidad complementarios: especificación de requisitos, medidas y evaluación.

*SQuaRE se creó para satisfacer necesidades que existían con la ISO 9126 y la ISO/IEC 14598, (primera generación de estándares de calidad de un producto de software). Por consiguiente, SQuaRE pertenece a la segunda generación de normas de calidad de un producto software

Modelos de Calidad de Proceso:

Se han desarrollado diferentes Modelos :

- CMM (Capability Maturity Model)/ CMMI
 - Desarrollado por SEI (Software Engineering Institute), organismo creado por el DoD de

USA

- Fuerte impacto en mejora del proceso
- Estipula un camino para la mejora
- Establece Áreas Clave que se deben atacar
- ISO 12207 – Modelos de Ciclos de Vida del Software
 - Actividades que debe incluir
- ISO 15504 – Modelo de Evaluación
- ISO 90003 Guía para la aplicación de la Norma ISO 9001 al Desarrollo, Suministro y Mantenimiento de Software.

De los modelos vistos en la materia cuales son los que se adaptan mas facilmente a los cambios.

Consultar: (Para mi)

Modelo en V.

El Modelo espiral (Boehm)

(Creo q esta seria la pregunta completa)

- **De los modelos de procesos vistos en la materia, ¿Cuáles son los que permiten mayor flexibilidad ante requerimientos cambiantes?. Justifique**

Rta:

Las metodologías ágiles, como por ejemplo RUP, permiten mayor flexibilidad ante requerimientos cambiantes ya que uno de los principios de este método es que el sistema se diseña para dar cabida a estos cambios, y para ello, los clientes son fuertemente implicados en todo el proceso de desarrollo, su papel es proporcional y priorizar nuevos requerimientos del sistema. Además, como el software se desarrolla en incrementos, el cliente especifica los requerimientos a incluir en cada incremento.

Hacer una tabla de decision (no me acuerdo el enunciado)

Final Juli:

1. Evolución de software. Que diferencias hay entre la ing tradicional y la de soft. Que es la ing de soft, explique sus conceptos.

Evolución del software

Durante los primeros años de la era de la computadora, el software se contemplaba como un añadido. El desarrollo del software se realizaba virtualmente sin ninguna planificación. El mismo se diseñaba a medida para cada aplicación y tenía una distribución relativamente pequeña.

La mayoría del software se desarrollaba y era utilizado por la misma persona u organización. La misma persona lo escribía, lo ejecutaba y, si fallaba, lo depuraba. El diseño era un proceso implícito y la documentación normalmente no existía.

En la segunda era en la evolución de los sistemas de computadora (1960-1970) la multiprogramación y los sistemas multiusuario introdujeron nuevos conceptos de interacción hombre - máquina. También se caracterizó por el establecimiento del software como producto.

En la tercera era en la evolución de los sistemas de computadora, que comenzó a mediados de los años setenta y continuó más allá de una década, el sistema distribuido, múltiples computadoras, cada una ejecutando funciones concurrentes y comunicándose con alguna otra, incrementó notablemente la complejidad de los sistemas informáticos. Las redes de área local y de área global, las comunicaciones digitales de alto ancho de banda y la creciente demanda de acceso "instantáneo" a los datos, supusieron una fuerte presión sobre los desarrolladores del software. La conclusión de la tercera era se caracterizó por la llegada y amplio uso de los microprocesadores.

La cuarta era de la evolución de los sistemas informáticos las potentes máquinas personales controladas por sistemas operativos sofisticados, en redes globales y locales, acompañadas por aplicaciones de software avanzadas se han convertido en la norma.

Las técnicas de la cuarta generación para el desarrollo del software están cambiando en la forma en que la comunidad del software construye programas informáticos.

Sin embargo, un conjunto de problemas relacionados con el software ha persistido a través de la evolución de los sistemas basados en computadora, y estos problemas continúan aumentando:

1. Los avances del software continúan dejando atrás nuestra habilidad de construir software para alcanzar el potencial del hardware.
2. Nuestra habilidad de construir nuevos programas no pueden ir al mismo ritmo de la demanda de nuevos programas, ni podemos construir programas lo suficientemente rápido como para cumplir las necesidades del mercado y de los negocios.
3. El uso extenso de computadoras ha hecho de la sociedad cada vez más dependiente de la operación fiable del software. Cuando el software falla, pueden ocurrir daños económicos enormes y ocasionar sufrimiento humano.
4. Continúan los esfuerzos para desarrollar software informático que tengan fiabilidad y alta calidad.

En respuesta a estos problemas, las prácticas de la Ingeniería del Software se están adoptando en toda la industria

Diferencias entre la Ingeniería tradicional de productos con la Ingeniería de Software

La diferencia existente entre la Ingeniería tradicional de productos y la Ingeniería de Software, es que a diferencia de la primera la Ingeniería de Software no fabrica un producto, sino que lo desarrolla al mismo tiempo que este no envejece sino que va cambiando hasta que cumple su tiempo de vida útil.

¿Qué es la ingeniería de software?

- Disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema incluyendo la evolución de éste, luego que se comienza a ejecutar.
 - Disciplina de la ingeniería
 - Hace que las cosas funcionen.
 - Se aplican teorías, métodos y herramientas.
 - Todos los aspectos de la producción de software
 - No sólo comprende los procesos técnicos del desarrollo de software, sino también se realizan actividades como la gestión de proyectos y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software
- La IEEE define a la Ingeniería de Software como:
 - (1) El uso de métodos sistemáticos, disciplinados y cuantificables para el desarrollo, operación y mantenimiento de software
 - El estudio de técnicas relacionadas con 1.

Richard Fairley propone:

- La Ingeniería de Software es la disciplina tecnológica y de administración que se ocupa de la producción y evolución sistemática de productos de software que son desarrollados y modificados dentro de los tiempos y costos estimados.
- Usa métodos sistemáticos "Cuantificables"
 - La cuantificación rigurosa de recursos, procesos y productos es una precondition para optimizar productividad y calidad. La "metrificación" y el control estadístico de procesos son claves en Ingeniería de Software.

- “Dentro de tiempos y costos estimados”
 - Un Ingeniero de Software debe cumplir contratos en tiempo y costos como es normal en obras de Ingeniería. Ello presupone la capacidad de medir, estimar, planificar y administrar proyectos.
- Para el “Desarrollo, operación y mantenimiento”
- La Ingeniería de Software se ocupa de todo el ciclo de vida de un producto, desde su etapa inicial de planificación y análisis de requerimientos hasta la estrategia para determinar cuándo y cómo debe ser retirado de servicio.
- La Ingeniería de Software surgió como reacción a las dificultades de desarrollar software sobre la base de habilidad, experiencia o intuición individuales.
- La aplicación de “métodos sistemáticos” y repetibles permite la producción y evolución de software por organizaciones distribuidas geográficamente y en el tiempo. La aplicación de estándares facilita la integración, reusabilidad mantenimiento de los productos.
- Es una “Disciplina tecnológica y de administración” .

En resumen:

La ingeniería de software trata de dar principios y métodos que permitan producir software confiable y eficiente, al menor costo posible.

- Para esto la ingeniería de software establece métodos, desarrolla herramientas automáticas o semiautomáticas y define procedimientos que establecen la relación de métodos y herramientas.

2. Que es el estudio de la factibilidad. Mencione las propiedades de un requerimiento. (Wikipedia)

Estudio de factibilidad también **Estudio de viabilidad** es el **análisis** financieros, **económicos** y **sociales** de una **inversión** (dada una opción **tecnológica** -estudio de pre-factibilidad). En la fase de **pre-inversión** la eventual etapa subsiguiente es el **diseño** final del **proyecto** (preparación del documento de proyecto), tomando en cuenta los **insumos** de un proceso productivo, que tradicionalmente son: tierra, **trabajo** y **capital** (que generan ingreso: renta, **salario** y ganancia).

Estudio de Viabilidad

- *Principalmente para sistemas nuevos*
- *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?
- *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.*

3. Define y diferencie a JAD y brainstorming.

JAD o JRP:

Planeación Conjunta de Requerimiento (JRP o JAD)

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)
- 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

Se basa en cuatro principios:

- Dinámica de grupos.
- Uso de ayudas visuales para mejorar la comunicación.
- Mantener un proceso organizado y racional.
- Durante la entrevista se trabajará sobre lo que se generará. (Interfaces, Reportes, etc.)
- 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) :

- 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.

• Los principios en que se basa esta técnica son:

- “Cuanto 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.

JAD vs BrainStorm: El brainstorming tiene la ventaja de que es muy fácil de aprender y requiere poca organización, pero puede no producir resultados con la misma calidad o nivel de detalle que otras técnicas.

Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del sistema, por lo que suele aplicarse en los primeros encuentros.

4. Que es un proceso. Que es un modelo de proceso.

¿Qué es un proceso de software?

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

Además de las actividades, los procesos deben incluir:

- Productos, que son los resultados de una actividad del proceso.
- Roles, que reflejan las responsabilidades de la gente que interviene en el proceso.
- Precondiciones y postcondiciones, que son declaraciones válidas antes y después de que se realice una actividad.
- No hay un proceso ideal, la mayoría de las organizaciones adoptan sus propios procesos de desarrollo de software.
- Para algunos sistemas críticos, se requiere un proceso estructurado, para otros uno menos formal y flexible.

¿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.

(MIRAR TEORIA DE MODELO DE PROCESO)

5. Que es una metodología ágil. Mencione las vistas en la materia.

Metodologías Ágiles:

“Es un enfoque **iterativo** e **incremental** (evolutivo) de desarrollo de software”

Iterativo: El desarrollo iterativo es una estrategia de reproceso en la que el tiempo se separa para revisar y mejorar partes del sistema.

Incremental: Es una estrategia programada y en etapas, en la que las diferentes partes del sistema se desarrollan en diferentes momentos o en diferentes velocidades, y se entregan a medida que se completan.

Objetivos:

- Producir software de alta calidad con un costo efectivo y en el tiempo apropiado.
- Esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto.
- Ofrecer una alternativa los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Reseña...

Una Metodología Ágil es aquella en la que “se da prioridad a las tareas que dan resultados directos y que reducen la burocracia tanto como sea posible” [Fowler], adaptándose además rápidamente al cambio de los proyectos. Ese enfoque ha sido utilizado desde hace más de dos décadas por un grupo de profesionales de software.

- ☐ **XP Extreme Programming** (www.extremeprogramming.org)
- ☐ **Scrum**
- ☐ **DSDM** (Dynamic Systems Development Method) www.dsdm.org
- ☐ **Crystal Methods** (Cockburn's Crystal Family Methodologies) www.crystallmethodologies.org
- ☐ **ASD Adaptive Software Development** (www.adaptivesd.com)
- ☐ **FDD Feature-Driven Development**

6. Defina y diferencia CMM y CMMI. Describa sus niveles.

CMM (Modelo de Capacidad y Madurez):
(Teoría 2013)

Posee una sola forma de aplicación (Escalonada) que permite un enfoque dirigido por niveles de madurez que indican cómo se desempeña una organización en base a la madurez en las áreas de proceso.

Presenta 5 (cinco) niveles de madurez.

Es un modelo de evaluación de los procesos de una organización. Fue desarrollado inicialmente para los procesos relativos al desarrollo e implementación de software.

Este modelo establece un conjunto de prácticas o procesos clave agrupados en Áreas Clave de Proceso. Para cada área de proceso define un conjunto de buenas prácticas.

A su vez estas Áreas de Proceso se agrupan en cinco "niveles de madurez":

1 – Inicial:

- Desempeño basado en la competencia del personal
 - Frecuentemente la organización vive “apagando incendios”
 - Aparecen héroes
 - Dificultad para encarar mejoras a largo plazo
 - La organización actúa esencialmente por reacción

2 – Repetible:

- La organización.
 - Estableció la gestión de los proyectos de software, y está documentado.
 - Usa políticas organizacionales.
 - Repite prácticas exitosas desarrolladas en proyectos previos.
 - Información (recursos, tiempo, esfuerzo) compartida por medios informales.

3 – Definido:

- El proceso para la gestión y las actividades de ingeniería está documentado e integrado en un proceso estándar para la organización.
- Todos los proyectos usan una versión documentada y aprobada del proceso estándar de la organización.
- La información del proceso se halla estandarizada y es compartida a través de la BDD de procesos de software.

4 – Gestionado:

- La organización aplica los principios de la gestión estadística de procesos para controlar el proceso del software
- La dirección tiene bases objetivas para tomar decisiones
- Puede predecir el desempeño en un entorno cuantificado realista
- Usa los datos como base para decisiones, objetivos y mejoras

5 – Optimizado:

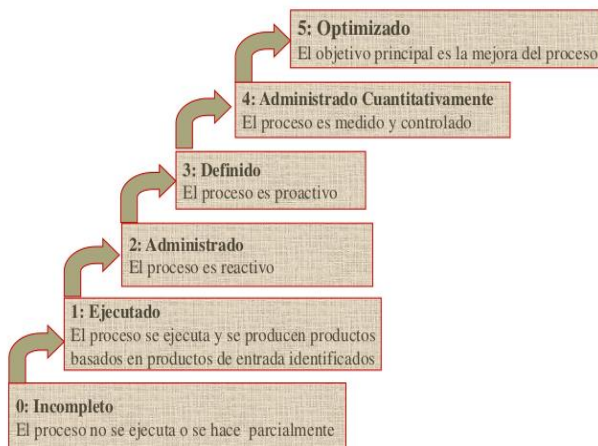
- La organización
 - identifica y elimina causas de desempeño pobre
 - mejora continua del proceso en base a gestión del cambio del proceso y de la tecnología

CMMI :

Posee dos vistas que permiten un enfoque diferente según las necesidades de quien vaya a implementarlo.

- Escalonado
 - centra su foco en la madurez de la organización. Igual que CMM.
- Continuo
 - Enfoca las actividades de mejora y evaluación en la capacidad de los diferentes procesos. Presenta 6 (seis) niveles de capacidad. Los niveles de capacidad indican qué tan bien se desempeña la organización en un área de proceso individual.

Representación continua:



7. Modelar con tablas como se viste juanita según si hace frío o calor, si está soleado o lluvioso, si se corta o no el pelo.

PREGUNTAS SUELTAS

1- Cómo nació la Ingeniería de Software y cuál ha sido su evolución?

- La Ingeniería de Software surgió como reacción a las dificultades de desarrollar software sobre la base de habilidad, experiencia o intuición individuales.
- La aplicación de “métodos sistemáticos” y repetibles permite la producción y evolución de software por organizaciones distribuidas geográficamente y en el tiempo. La aplicación de estándares facilita la integración, reusabilidad mantenimiento de los productos.
- Es una “Disciplina tecnológica y de administración”.

Como nació:

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; lo que dio origen a la “CRISIS DEL SOFTWARE”, es decir que los programas sin documentación, los cuales eran difíciles de mantener, y generar software nuevo o mantener los viejos (ingeniería inversa: a través de su código fuente se obtiene un relevamiento del sistema, con una buena documentación para las modificaciones futuras), tenía un costo muy alto.

El valor estratégico del software llevó a la Organización del Tratado del Atlántico Norte 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”.

Evolución:

La ingeniería de software evoluciona continuamente. Y en dicha evolución podemos encontrar cuatro etapas (eras) fundamentales:

- Era funcional – años 60: Se comienza a estudiar como 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: Se intensifica la necesidad de que el producto tenga atributos que satisfagan las necesidades explícitas e implícitas del usuario: aparecen conceptos tales como mantenibilidad, confiabilidad, eficiencia, usabilidad, etc.

1. Dé al menos dos definiciones de Ingeniería de Software y explique sus conceptos principales y diferencias.

- **De IEEE:** Es el uso de métodos sistemáticos, disciplinados y cuantificables para el desarrollo, operación y mantenimiento de software.

Concepto de “métodos sistemáticos”: La aplicación de estos métodos repetibles, permite la producción y evolución del Software. La aplicación de estándares facilita la integración, reusabilidad y mantenimiento de los productos.

Concepto de “cuantificables”: La cuantificación de los recursos, procesos y productos, garantizan la optimización de la productividad y la calidad. La metrificación y control estadístico también es clave para la Ingeniería de Software.

- **De Richard Fairley:** Es la disciplina tecnológica y de administración que se ocupa de la producción y evolución sistemática de productos de software que son desarrollados y modificados dentro de los tiempos y costos estimados.

Concepto de “disciplina tecnológica y de administración”: No solo se maneja el concepto de tecnología sino que también tenemos el concepto de administración debido a que se incluyen aspectos como el análisis de riesgo, el análisis de costo, la planificación de proyectos o el desarrollo de estándares.

Concepto de “dentro de tiempos y costos estimados”: Un ingeniero de Software debe cumplir contratos en tiempo y costos. Esto presupone la capacidad de medir, estimar, planificar y administrar proyectos.

Concepto de “desarrollo, operación y mantenimiento”: La Ingeniería de Software se ocupa de todo el ciclo de vida de un producto, desde su etapa inicial de planificación y análisis de requerimientos hasta la estrategia para determinar cuándo y cómo debe ser retirado de servicio.

Diferencias:

La IEEE se limita a hablar de métodos y técnicas para el desarrollo de Software, a diferencia de Richard Fairley que también introduce el concepto de “tiempo y costos”. Es decir que hace hincapié en que es necesario poseer una capacidad de medir, estimar, planificar y administrar los proyectos con el fin de conocer de que manera el mismo satisficará el objetivo planteado.

1. Defina Ingeniería de Requerimientos. Describa las técnicas de elicitación.

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 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”

Técnicas de elicitación:

- *Muestreo de la documentación, las formas y los datos existentes.*
- *Investigación y visitas al lugar.*
- *Observación del ambiente de Trabajo.*
- *Cuestionarios.*
- *Entrevistas.*
- *Planeación conjunta de Requerimientos (JRP o JAD).*
- *Lluvia de Ideas – Brainstorming.*

(1)Muestreo de la documentación, las formas y los datos existentes

- Recolección de hechos a partir de la documentación existente.
- ¿Qué tipo de documentos pueden enseñar algo acerca del sistema?
 - Organigrama (identificar el propietario, usuarios claves).
 - Memos, notas internas, minutas, registros contables.
 - Solicitudes de proyectos de sistemas de información anteriores.
- Permiten conocer el historial que origina el proyecto.
- Documentos que describen la funcionalidad del negocio que esta siendo analizada.
 - Declaración de la misión y plan estratégico de la organización .
 - Objetivos formales del departamento en cuestión.
 - Políticas, restricciones, procedimientos operativos.
 - Formularios de operaciones realizadas.
 - Bases de Datos.
 - Sistemas en funcionamiento.
- Documentación de sistemas anteriores.
 - Diagramas.
 - Diccionario o Repositorios de proyecto.
 - Documentos de diseño.
 - Manuales de operación y/o entrenamiento.
- Técnicas de muestreo de documentos y archivos.

(2)Investigación y visitas al sitio .

- Investigar el dominio.
- Patrones de soluciones (mismo problema en otra organización).
- Revistas especializadas.
- Buscar problemas similares en internet.
- Consultar otras organizaciones.

(3)Observación del ambiente de trabajo

- El analista se convierte en observador de las personas y actividades con el objeto de aprender acerca del sistema.
- Lineamientos de la observación:
 - Determinar quién y cuándo será observado.
 - Obtener el permiso de la persona y explicar el porqué será observado.
 - Mantener bajo perfil.
 - Tomar nota de lo observado.
 - Revisar las notas con la persona apropiada.
 - No interrumpir a la persona en su trabajo.

Ventajas

- Datos confiables
- El analista puede ver exactamente lo que se hace (tareas difíciles de explicar con palabras).
- Análisis de disposiciones físicas, tránsito, iluminación, ruido.
- Económica en comparación con otras técnicas.

Desventajas

- La gente se siente incómoda siendo observada.
- Algunas actividades del sistema pueden ser realizadas en horarios incómodos.
- Las tareas están sujetas a interrupciones.
- Tener en cuenta que la persona observada puede estar realizando las tareas de la forma “correcta” y no como lo hace habitualmente.

(4)Cuestionarios

- Documento que permite al analista recabar información y opiniones de los encuestados
- Recolectar hechos de un gran número de personas.
- Detectar un sentimiento generalizado.
- Detectar problemas entre usuarios.
- Cuantificar respuestas.

Ventajas

- Respuesta rápida
- Económicos
- Anónimos
- Estructurados de fácil análisis

Desventajas

- Número bajo de respuestas
- No responde a todas las preguntas
- Preguntas rígidas
- No se puede analizar el análisis corporal
- No se pueden aclarar respuestas incompletas
- Difíciles de preparar

Tipos de Cuestionario

- Formato libre (Abiertos)

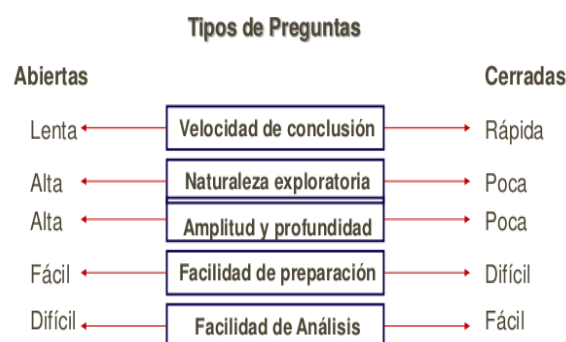
Diseñado para ofrecer al encuestado más flexibilidad en la respuesta

- ¿Qué reportes recibe actualmente?
- ¿Hay problemas con los reportes?

- Formato fijo (Cerrados)

Requieren la selección de una respuesta entre respuestas posibles predefinidas

- ¿Es útil el reporte que utiliza actualmente? SI NO



Tipo de información obtenida :

- Actitud

- Lo que las personas dicen que quieren
- Creencias
 - Lo que las personas creen que es verdad
- Comportamiento
 - Lo que realmente hacen
- Características
 - De las personas o cosas
- Cuándo usar Cuestionarios
 - Las personas están dispersas geográficamente
 - Diferentes oficinas o ciudades
 - Muchas personas involucradas
 - Clientes o usuarios
 - Queremos obtener opiniones generales
 - Queremos identificar problemas generales

(5)Entrevistas

- Técnica de exploración mediante la cual el analista de sistemas recolecta información de las personas a través de la interacción cara a cara.
- Es una conversación con un propósito específico, que se basa en un formato de preguntas y respuestas en general.
- Conocer opiniones y sentimientos del entrevistado.
- Tipo de información obtenida: opiniones, objetivos, procedimientos informales, sentimientos.

Ventajas

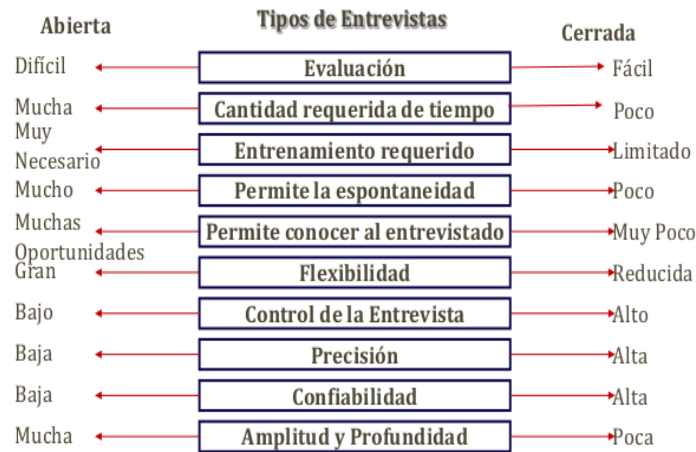
- El entrevistado se siente incluido en el proyecto
- Es posible obtener una retroalimentación del encuestado
- Es posible adaptar las preguntas de acuerdo al entrevistado
- Información no verbal
- Observando las acciones y expresiones del entrevistado

Desventaja

- Costosas
 - Tiempo y recursos humanos
- Las entrevistas dependen en gran parte de las habilidades del entrevistador
- No aplicable a distancia

Tipos de entrevistas :

- Estructuradas (Cerradas)
 - El encuestador tiene un conjunto específico de preguntas para hacérselas al entrevistado
 - Se dirige al usuario sobre un requerimiento puntual
 - No permite adquirir un amplio conocimiento del dominio
- No estructuradas (Abiertas)
 - El encuestador lleva a un tema en general
 - Sin preparación de preguntas específicas
 - Iniciar con preguntas que no dependen del contexto, para conocer el problema, la gente involucrada, etc.



Tipos de Preguntas

• Abiertas

- Permite al encuestado responder de cualquier manera
- ¿Qué opinión tiene del sistema actual?
- ¿Cómo describe su trabajo?

• Cerradas

- Las respuestas son directas, cortas o de selección específica
- ¿Quién recibe este informe?
- ¿Cuántas personas utilizan el sistema?

• Sondeo

- Permite obtener más detalle sobre un tema puntual
- ¿Podría dar detalles sobre...?
- ¿Podría dar un ejemplo de...?

Preguntas Abiertas

• Ventajas

- Revelan nueva línea de preguntas
- Hacen más interesante la entrevista
- Permiten espontaneidad

• Desventajas

- Pueden dar muchos detalles irrelevantes
- Se puede perder el control de la entrevista
- Parece que el entrevistador no tiene los objetivos claros

Preguntas cerradas

• Ventajas

- Ahorran tiempo
- Se mantiene más fácil el control de la entrevista
- Se consiguen datos relevantes

• Desventajas

- Pueden aburrir al encuestado
- No se obtienen detalles

• Organización de una entrevista

- Piramidal (Inductivo)
- Embudo (Deductivo)
- Diamante (Comb. de las anteriores)

(6)Planeación conjunta de Requerimientos (JRP o JAD).

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)*
- **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*

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

(7)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.*

Los principios en que se basa esta técnica son:

- *“Cuanto 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.*

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

1. Enuncie un par de definiciones de ingeniería de software que abarquen los aspectos salientes.

- La IEEE define a la Ingeniería de Software como:
 - (1) El uso de métodos sistemáticos, disciplinados y cuantificables para el desarrollo, operación y mantenimiento de software
 - El estudio de técnicas relacionadas con 1.

Richard Fairley propone:

- La Ingeniería de Software es la disciplina tecnológica y de administración que se ocupa de la producción y evolución sistemática de productos de software que son desarrollados y modificados dentro de los tiempos y costos estimados.

1) Qué es la Ingeniería de Software

- Disciplina de la ingeniería que comprende todos los aspectos de la producción de software

desde las etapas iniciales de la especificación del sistema incluyendo la evolución de éste, luego que se comienza a ejecutar.

- Disciplina de la ingeniería
 - Hace que las cosas funcionen.
 - Se aplican teorías, métodos y herramientas.
- Todos los aspectos de la producción de software
 - *No sólo comprende los procesos técnicos del desarrollo de software, sino también se realizan actividades como la gestión de proyectos y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software*

1. Describa la evolución del software? Que diferencias tiene la Ingeniería tradicional de productos con la Ingeniería de Software? Defina Ingeniería de Software y explique sus conceptos principales.

Evolución del software

Durante los primeros años de la era de la computadora, el software se contemplaba como un añadido. El desarrollo del software se realizaba virtualmente sin ninguna planificación. El mismo se diseñaba a medida para cada aplicación y tenía una distribución relativamente pequeña.

La mayoría del software se desarrollaba y era utilizado por la misma persona u organización. La misma persona lo escribía, lo ejecutaba y, si fallaba, lo depuraba. El diseño era un proceso implícito y la documentación normalmente no existía.

En la segunda era en la evolución de los sistemas de computadora (1960-1970) la multiprogramación y los sistemas multiusuario introdujeron nuevos conceptos de interacción hombre - máquina. También se caracterizó por el establecimiento del software como producto.

En la tercera era en la evolución de los sistemas de computadora, que comenzó a mediados de los años setenta y continuó más allá de una década, el sistema distribuido, múltiples computadoras, cada una ejecutando funciones concurrentes y comunicándose con alguna otra, incrementó notablemente la complejidad de los sistemas informáticos. Las redes de área local y de área global, las comunicaciones digitales de alto ancho de banda y la creciente demanda de acceso "instantáneo" a los datos, supusieron una fuerte presión sobre los desarrolladores del software. La conclusión de la tercera era se caracterizó por la llegada y amplio uso de los microprocesadores.

La cuarta era de la evolución de los sistemas informáticos las potentes máquinas personales controladas por sistemas operativos sofisticados, en redes globales y locales, acompañadas por aplicaciones de software avanzadas se han convertido en la norma.

Las técnicas de la cuarta generación para el desarrollo del software están cambiando en la forma en que la comunidad del software construye programas informáticos.

Sin embargo, un conjunto de problemas relacionados con el software ha persistido a través de la evolución de los sistemas basados en computadora, y estos problemas continúan aumentando:

1. Los avances del software continúan dejando atrás nuestra habilidad de construir software para alcanzar el potencial del hardware.
2. Nuestra habilidad de construir nuevos programas no pueden ir al mismo ritmo de la demanda de nuevos programas, ni podemos construir programas lo suficientemente rápido como para cumplir las necesidades del mercado y de los negocios.
3. El uso extenso de computadoras ha hecho de la sociedad cada vez más dependiente de la operación fiable del software. Cuando el software falla, pueden ocurrir daños económicos enormes y ocasionar sufrimiento humano.
4. Continúan los esfuerzos para desarrollar software informático que tengan fiabilidad y alta calidad.

En respuesta a estos problemas, las prácticas de la Ingeniería del Software se están adoptando en toda la industria

Diferencias entre la Ingeniería tradicional de productos con la Ingeniería de Software

La diferencia existente entre la Ingeniería tradicional de productos y la Ingeniería de Software, es que a diferencia de la primera la Ingeniería de Software no fabrica un producto, sino que lo desarrolla al mismo tiempo que este no envejece sino que va cambiando hasta que cumple su tiempo de vida útil.

Definición y conceptos:

- Disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema incluyendo la evolución de éste, luego que se comienza a ejecutar.
 - Disciplina de la ingeniería
 - Hace que las cosas funcionen.
 - Se aplican teorías, métodos y herramientas.
 - Todos los aspectos de la producción de software
 - *No sólo comprende los procesos técnicos del desarrollo de software, sino también se realizan actividades como la gestión de proyectos y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software*
- La IEEE define a la Ingeniería de Software como:
 - (1) El uso de métodos sistemáticos, disciplinados y cuantificables para el desarrollo, operación y mantenimiento de software
 - El estudio de técnicas relacionadas con 1.

Richard Fairley propone:

- La Ingeniería de Software es la disciplina tecnológica y de administración que se ocupa de la producción y evolución sistemática de productos de software que son desarrollados y modificados dentro de los tiempos y costos estimados.
- Usa métodos sistemáticos “Cuantificables”
 - La cuantificación rigurosa de recursos, procesos y productos es una precondition para optimizar productividad y calidad. La “metrificación” y el control estadístico de procesos son claves en Ingeniería de Software.
- “Dentro de tiempos y costos estimados”
 - Un Ingeniero de Software debe cumplir contratos en tiempo y costos como es normal en obras de Ingeniería. Ello presupone la capacidad de medir, estimar, planificar y administrar proyectos.
- Para el “Desarrollo, operación y mantenimiento”
 - La Ingeniería de Software se ocupa de todo el ciclo de vida de un producto, desde su etapa inicial de planificación y análisis de requerimientos hasta la estrategia para determinar cuándo y cómo debe ser retirado de servicio.

En resumen:

- La ingeniería de software trata de dar principios y métodos que permitan producir software confiable y eficiente, al menor costo posible.
- Para esto la ingeniería de software establece métodos, desarrolla herramientas automáticas o semiautomáticas y define procedimientos que establecen la relación de métodos y herramientas

1. ¿Qué es el Software? Describa su evolución.

- ¿Qué es Software?
- *Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación (IEEE)*

Evolución del software

Durante los primeros años de la era de la computadora, el software se contemplaba como un añadido. El desarrollo del software se realizaba virtualmente sin ninguna planificación. El mismo se diseñaba a medida para cada aplicación y tenía una distribución relativamente pequeña. La mayoría del software se desarrollaba y era utilizado por la misma persona u organización. La misma persona lo escribía, lo ejecutaba y, si fallaba, lo depuraba. El diseño era un proceso implícito y la documentación normalmente no existía.

En la segunda era en la evolución de los sistemas de computadora (1960-1970) la multiprogramación y los sistemas multiusuario introdujeron nuevos conceptos de interacción hombre - máquina. También se caracterizó por el establecimiento del software como producto.

En la tercera era en la evolución de los sistemas de computadora, que comenzó a mediados de los años setenta y continuó más allá de una década, el sistema distribuido, múltiples computadoras, cada una ejecutando funciones concurrentes y comunicándose con alguna otra, incrementó notablemente la complejidad de los sistemas informáticos. Las redes de área local y de área global, las comunicaciones digitales de alto ancho de banda y la creciente demanda de acceso "instantáneo" a los datos, supusieron una fuerte presión sobre los desarrolladores del software. La conclusión de la tercera era se caracterizó por la llegada y amplio uso de los microprocesadores.

La cuarta era de la evolución de los sistemas informáticos las potentes máquinas personales controladas por sistemas operativos sofisticados, en redes globales y locales, acompañadas por aplicaciones de software avanzadas se han convertido en la norma.

Las técnicas de la cuarta generación para el desarrollo del software están cambiando en la forma en que la comunidad del software construye programas informáticos.

Sin embargo, un conjunto de problemas relacionados con el software ha persistido a través de la evolución de los sistemas basados en computadora, y estos problemas continúan aumentando:

1. Los avances del software continúan dejando atrás nuestra habilidad de construir software para alcanzar el potencial del hardware.
2. Nuestra habilidad de construir nuevos programas no pueden ir al mismo ritmo de la demanda de nuevos programas, ni podemos construir programas lo suficientemente rápido como para cumplir las necesidades del mercado y de los negocios.
3. El uso extenso de computadoras ha hecho de la sociedad cada vez más dependiente de la operación fiable del software. Cuando el software falla, pueden ocurrir daños económicos enormes y ocasionar sufrimiento humano.
4. Continúan los esfuerzos para desarrollar software informático que tengan fiabilidad y alta calidad.

En respuesta a estos problemas, las prácticas de la Ingeniería del Software se están adoptando en toda la industria

1. ¿Qué es el software? ¿Cuál ha sido su evolución?

Idem a la anterior.

PROCESOS Y REQUERIMIENTOS!

2- Defina "proceso" y "modelo de proceso" y enumere las características de cualquier proceso.

Proceso: Hace referencia a la serie de pasos que involucran actividades, restricciones y recursos que producen un resultado esperado. Este comprende un conjunto de procedimientos organizados de tal manera de satisfacer las metas.

Modelo de proceso: Es una representación abstracta de un proceso del software que presenta una visión de ese proceso. Estos modelos pueden incluir actividades que son parte de los procesos y productos de software y el papel de las personas involucradas en la ingeniería de software.

CARACTERISTICAS DE CUALQUIER PROCESO:

- Establece todas las actividades
- Utiliza recursos, está sujeto a restricciones y genera productos intermedios y finales
- Puede estar compuesto por subprocesos
- Cada actividad tiene entradas y salidas definidas
- Las actividades se organizan en una secuencia
- Existen principios que orientan sobre las metas de cada actividad
- Las restricciones pueden aplicarse a una actividad, recurso o producto.

2. Sería posible combinar modelos de procesos? Si la respuesta es posible, da un ejemplo

Si es posible combinar modelos de procesos.

Ej: modelo de cascada con prototipado

2. Enumere los modelos genéricos del proceso de software. Explique los modelos de prototipos desechables y evolutivos.

• **Enumere los modelos genéricos del proceso de software:**

- Modelo en cascada
- Desarrollo evolutivo
- Desarrollo formal de sistemas
- Desarrollo basado en la reutilización
- Metodologías ágiles

Prototipo:

Un prototipo es un producto parcialmente desarrollado que permite que clientes y desarrolladores examinen algunos aspectos del sistema propuesto, y decidan si éste es adecuado o correcto para el producto terminado.

Esta es una alternativa de especificación para tratar mejor la incertidumbre, la ambigüedad y la volubilidad de los proyectos reales.

Ventajas: La especificación se puede desarrollar de forma creciente. Tan pronto como los usuarios desarrollen un mejor entendimiento de su problema, este se reflejará en el sistema de software.

Desventajas: El proceso no es visible. Es muy costoso producir documentos que reflejen cada versión. A menudo los sistemas tienen una estructura deficiente. Los cambios continuos tienden a corromper la estructura de software. Se requieren herramientas y técnicas especiales

Evolutivo: El objetivo es obtener el sistema a entregar. Permite que todo el sistema o alguna de sus partes se construyan rápidamente para comprender o aclarar aspectos y asegurar que el desarrollador, el usuario y el cliente tengan una comprensión unificada tanto de lo que se necesita como de lo que se propone como solución.

Desechable: El objetivo es obtener el conjunto de requerimientos. Luego se combina con otro paradigma para obtener el producto final. Este enfoque de prototipos supone que el modelo simulará alguna o todas las funciones que el usuario desea y que al concluirse el modelado los programas se descartarán y reemplazarán con programas reales. Es peligroso tentarse a adoptarlo como producto final

Si el prototipo se descarta y se reemplaza con el sistema real puede que no quede documentación de los requerimientos del usuario

Características de un proyecto candidato:

- Usuarios que no examinarán los modelos abstractos
- Usuarios que no determinarán sus requerimientos inicialmente
- Sistemas en línea y operados por pantalla
- Sistemas con énfasis en los formatos de E/S más que en los detalles algorítmicos
- Sistemas en los que haya que explorar aspectos técnicos
- Si el usuario tiene dificultad al tratar con los modelos gráficos para modelar los requerimientos y el comportamiento
- Si se considera que el análisis estructurado consume demasiado tiempo
- Si se enfatiza el aspecto de la interfaz humana

3. Describa el modelo de procesos incremental y diferéncielo del modelo iterativo.

Desarrollo por fases:

Se desarrolla el sistema de tal manera que puede ser entregado en piezas. Esto implica que existen dos sistemas funcionando en paralelo: el sistema operacional y el sistema en desarrollo. Se desarrollan versiones.

Existen dos maneras de desarrollar por fases:

Incremental: El sistema es particionado en subsistemas de acuerdo con su funcionalidad. Cada entrega agrega un subsistema.

En un proceso de desarrollo incremental, los clientes identifican, a grandes rasgos, los servicios que proporcionará el sistema. Identifican qué servicios son más importantes y cuales menos. Entonces, se definen varios incrementos en donde cada uno proporciona un subconjunto de la funcionalidad del sistema.

Una vez que los incrementos del sistema se han identificado, los requerimientos para los servicios que se van a entregar en el primer incremento se define en detalles, y éste se desarrolla.

Una vez que un incremento se completa y entrega, los clientes pueden ponerlo en servicio. Esto significa que tienen una entrega temprana de parte de la funcionalidad del sistema. Pueden experimentar con el sistema, lo cual les ayuda a clarificar sus requerimientos para los incrementos posteriores y para las últimas versiones del incremento actual. Tan pronto como se completan los nuevos incrementos, se integran en los existentes de tal forma que la funcionalidad del sistema mejora con cada incremento entregado.

Iterativo: El ciclo de vida iterativo se basa en la amplificación y refinamiento sucesivos de un sistema software mediante múltiples iteraciones, con retroalimentación cíclica. El sistema se incrementa a lo largo del tiempo, iteración tras iteración. La salida de una iteración no es un prototipo experimental o desechable, más bien es un subconjunto con calidad de producción del sistema final. Cada iteración conlleva la elección de un pequeño conjunto de requisitos que permite rápidamente diseñar, implementar y probar.

Beneficios del desarrollo iterativo

- Reducción temprana de riesgos.
- Progreso visible desde las primeras etapas.
- Una temprana retroalimentación, refinando el sistema a las necesidades reales de los usuarios.
- Gestión de la complejidad: el equipo no se ve abrumado por la «parálisis del análisis».
- El conocimiento adquirido en un ciclo se puede utilizar metódicamente para mejorar el propio proceso de desarrollo.

3. Elija dos modelos de proceso, explíquelos y diferéncielos.

4. Enumere Los Modelos de Procesos que conoce y descríbalos brevemente

(falta describirlos brevementes)

- Modelo en Cascada
- Modelo en V
- Modelo de Prototipos
- Desarrollo por fases
 - Incremental
 - Iterativo
- El modelo espiral

5- De los modelos de procesos vistos en la materia, ¿Cuáles son los que permiten mayor flexibilidad ante requerimientos cambiantes?. Justifique

(PREGUNTAR A GERO, NO ME CONVINCE ESTA Rta)

Las metodologías ágiles, como por ejemplo RUP, permiten mayor flexibilidad ante requerimientos cambiantes ya que uno de los principios de este método es que el sistema se diseña para dar cabida a estos cambios, y para ello, los clientes son fuertemente implicados en todo el proceso de desarrollo, su papel es proporcional y priorizar nuevos requerimientos del sistema. Además, como el software se desarrolla en incrementos, el cliente especifica los requerimientos a incluir en cada incremento.

2. Por qué cambian los requerimientos? Describa y enumere requerimientos volátiles y duraderos.

Los requerimientos cambian y esto persiste a lo largo de la vida del sistema. Los cambios ocurren por:

¿Por qué cambian los requerimientos?

- Porque al analizar el problema, no se hacen las preguntas correctas a las personas correctas (En sistemas grandes hay una comunidad diversa de usuarios)
- Porque los clientes y los usuarios son distintos
- Porque cambió el problema que se estaba resolviendo
- Porque los usuarios cambiaron su forma de pensar o sus percepciones
- Porque cambió el ambiente de negocios (mercado, etc.)

Requerimientos volátiles

- Cambian durante el desarrollo del sistema o después que se puso en operación. (Ej.: cambios gubernamentales)

Entre los requerimientos volátiles podemos encontrar:

- Requerimientos Cambiantes: cambian porque se modifica el ambiente (entorno).
- Requerimientos emergentes: surgen como ampliación (al incrementar la comprensión del cliente).
- Requerimientos consecuentes: surgen por la introducción del sistema. Pueden cambiar los procesos de la organización por desarrollar nuevas formas de trabajo.
- Requerimientos de compatibilidad: cambian porque interactúan con otros sistemas que cambian.

Requerimientos duraderos

- Relativamente estables, se derivan de la actividad principal de la organización.

2. Por qué cambian los requerimientos? Describa los requerimientos funcionales y no funcionales.

Idem anterior.

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.

Dentro de los requerimientos NO funcionales podemos encontrar:

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.

2- Explique las descripciones estáticas de requerimientos. Ejemplifique.

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.

2. Explique los requerimientos funcionales y no funcionales? Explique los objetivos de la especificación de requerimientos.

Idem anterior.

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.

Una especificación de requerimientos debe ser:

- Correcta
- Rastreado

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

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.

2) cuál es la principal fuente de errores en la definición y análisis de requerimientos. ¿Por qué?

LA COMUNICACION!!!!!!!!!!

- *La elicitación de requisitos es una actividad principalmente de carácter social, mucho más que tecnológico.*
- *Los problemas que se plantean son por tanto de naturaleza psicológica y social, más que técnicos.*

• *Nota: Requisitos = Requerimientos.*

Problemas de comunicación

- *Dificultad para expresar claramente las necesidades.*
 - *No ser conscientes de sus propias necesidades.*
 - *No entender cómo la tecnología puede ayudar.*
 - *Miedo a parecer incompetentes por ignorancia tecnológica.*
 - *No tomar decisiones por no poder prever las consecuencias, no entender las alternativas o no tener una visión global.*
- ¿quién tiene estas dificultades?*

- *Cultura y vocabulario diferentes.*
 - *Intereses distintos en el sistema a desarrollar.*
 - *Medios de comunicación inadecuados (diagramas que no entienden los clientes y usuarios).*
 - *Conflictos personales o políticos.*
- ¿quién tiene estas dificultades ?*

Elicitación de Requisitos

- *Limitaciones cognitivas (del desarrollador)*
- *No conocer el dominio del problema.*
- *Hacer suposiciones sobre el dominio del problema.*
- *Hacer suposiciones sobre aspectos tecnológicos.*
- *Hacer simplificaciones excesivas.*
- *Conducta humana*
- *Conflictos y ambigüedades en los roles de los participantes.*
- *Pasividad de clientes, usuarios o ingenieros de requisitos.*
- *Temor a que el nuevo sistema lo deje sin trabajo.*
- *Técnicos*
- *Complejidad del dominio del problema.*

- Complejidad de los requisitos.
- Múltiples fuentes de requisitos.
- Fuentes de información poco claras.

2. ¿Qué es un Requerimiento? Describa Requerimiento Funcional y No funcional.

- 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

Definición IEEE-Std-610

1. Condición o capacidad que necesita el usuario para resolver un problema o alcanzar un objetivo
2. Condición o capacidad que debe satisfacer o poseer un sistema o una componente de un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto.
3. Representación documentada de una condición o capacidad como en 1 o 2.

Idem anterior.

2) ¿Qué es estudio de factibilidad ? Describa

Estudio de factibilidad también **Estudio de viabilidad** es el análisis financieros, económicos y sociales de una inversión (dada una opción tecnológica -estudio de pre-factibilidad). En la fase de pre-inversión la eventual etapa subsiguiente es el diseño final del proyecto (preparación del documento de proyecto), tomando en cuenta los insumos de un proceso productivo, que tradicionalmente son: tierra, trabajo y capital (que generan ingreso: renta, salario y ganancia).

Estudio de Viabilidad

- Principalmente para sistemas nuevos
- 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?

- 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.

3. ¿Qué es un requerimiento? ¿Qué clasificaciones conoce?

Idem.

Tipos de requerimientos

- Requerimientos funcionales
- Requerimientos no funcionales

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 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.)*

4) Describa cuatro tipos de requerimientos no funcionales que pueden existir en un sistema. Ejemplifique

Numero de terminales:

Número de usuarios simultáneos:

Numero de transacciones a manejar dentro de ciertos periodos de tiempo.

La información en la base de datos se guardara encriptada. (respecto seguridad)

3) ¿qué es un requerimiento? Cuáles son los funcionales y cuáles los no funcionales? De que tipo son:

“los documentos a entregar deberán ajustarse a la reglamentación de la empresa”

“el sistema debe administrar las ventas por internet”

1)NF

2)NF

3. Defina Ingeniería de Requerimientos.

Idem a otra anterior.

4. Describa el proceso de Ingeniería de Requerimientos.

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, interfases, rendimiento y limitaciones.

Ingeniería de requerimientos 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.

La meta del proceso de ingeniería de requerimientos es crear y mantener un documento de requerimientos del sistema. El proceso general corresponde a cuatro subprocesos de alto nivel de la ingeniería de requerimientos, estos son:

- 1) Estudio de viabilidad
- 2) Obtención y análisis de requerimientos
- 3) Especificación de requerimientos
- 4) Validación de requerimientos
- 5) Gestion de requerimientos (PARA MI NO VA, Nico)

Estudio de viabilidad: para todos los sistemas nuevos, el proceso de ingeniería de requerimientos

debería empezar con un estudio de viabilidad. La entrada de este es un conjunto de requerimientos de negocio preliminares, una descripción resumida del sistema y de cómo este pretende contribuir a los procesos de negocio. Los resultados del estudio de viabilidad deberían ser un informe que recomiende si merece o no la pena seguir con la ingeniería de requerimientos y el proceso de desarrollo del sistema.

Un estudio de viabilidad es un estudio corto y debería resolver las siguientes preguntas:

- El sistema contribuye a los objetivos generales de la organización?
- 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?

Obtención y análisis de requerimientos: La extracción de los requerimientos es una parte especialmente crítica del proceso, debemos trabajar con los usuarios y los clientes (stakeholders), para comprender un problema cuando todavía no se ha encontrado una solución. Una de las maneras de realizar un análisis de problema consiste en identificar las personas, los procesos y los recursos involucrados, y después documentar las relaciones que existen entre ellos. Se averigua que elementos de datos pasan de un rol a otro, y cuáles procesos transforman los datos de una forma o estado a otra. Durante la extracción de los requerimientos se interroga a usuarios y clientes sobre los mismos aspectos. Por lo general resulta útil separar los requerimientos en tres categorías:

- 1) Requerimientos que deben ser absolutamente satisfechos.
- 2) Requerimientos que son muy deseables pero no indispensables.
- 3) Requerimientos que son posibles, pero que podrían eliminarse.

El proceso genérico de obtención y análisis de requerimientos consta de 4 pasos:

- *Descubrimiento de requerimientos:* aquí se utilizan técnicas como entrevistas, cuestionarios, JADs y brainstorm.
- *Clasificación y organización de requerimientos:* toma la recopilación no estructurada de requerimientos, y los organiza en grupos relacionados y coherentes.
- *Ordenación por prioridades y negociación de requerimientos:* inevitablemente, cuando existen muchos stakeholders, los requerimientos entraran en conflicto. Esta actividad se refiere a ordenar según las prioridades los requerimientos, y a encontrar y resolver los requerimientos en conflicto a través de la organización.
- *Documentación de requerimientos:* se documentan los requerimientos, se pueden producir documentos de requerimientos formales o informales.

Especificación de requerimientos: hay dos clases de documentos de requerimientos que se realizan en la extracción y el análisis de requerimientos. Por una parte, la extracción nos permite escribir un documento de **definición de requerimientos**, escrita en términos que el cliente puede entender, la definición de requerimientos es un listado completo de todas las cosas que el cliente espera que haga el sistema propuesto. Representa una comprensión entre el cliente y el desarrollador de lo que el cliente necesita o desea, y por lo general es escrito en forma conjunta por el cliente y el desarrollador. Por la otra parte, la **especificación de requerimientos** reitera la definición en los términos técnicos apropiados para el desarrollo del diseño de un sistema; es la contrapartida técnica al documento de definición de requerimientos y es escrito por analistas de requerimientos.

Objetivos de la especificación de requerimientos:

- Permiten que los desarrolladores expliquen como han entendido lo que el cliente pretende del sistema
- Indican a los diseñadores que funcionalidad y características va a tener el sistema resultante
- Indican al equipo de pruebas que demostraciones llevar a cabo para convencer al cliente de

que el sistema que se le entrega es lo que había ordenado.

Descripción estática de requerimientos: 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 y permiten razonar sobre comportamiento de sistemas que cambian poco (o nada) en el tiempo

Descripciones dinámicas: se considera un sistema en función de los cambios que ocurren a lo largo del tiempo. El sistema está en un estado particular hasta que un estímulo lo obliga a cambiar su estado.

Las técnicas para describir un sistema en términos de estados son:

- 1) Tablas de decisión
- 2) Redes de Petri
- 3) Diagrama de transición de estados
- 4) Tablas de transición de estados
- 5) ...

Validación de requerimientos: trata de mostrar que los requerimientos realmente definen el sistema que el cliente desea. La validación de requerimientos es importante debido a que los errores en los requerimientos pueden conducir a importantes costos si se descubren más tarde.

Durante el proceso de validación de requerimientos, se deben llevar a cabo verificaciones sobre requerimientos en el documento de requerimientos:

- *Verificaciones de validez*
- *Verificaciones de consistencia*
- *Verificaciones de completitud*
- *Verificaciones de realismo*
- *Verificabilidad*

Se pueden utilizar, en conjunto o de forma individual, varias técnicas de validación de requerimientos:

1. *Revisiones de requerimientos*
2. *Construcción de prototipos*
3. *Generación de casos de prueba*

Gestión de requerimientos: los requerimientos para sistemas software grande son siempre cambiantes. Pero ¿por qué cambian los requerimientos? :

- Porque al analizar el problema, no se hacen las preguntas correctas a las personas correctas (En sistemas grandes hay una comunidad diversa de usuarios)
- Porque los clientes y los usuarios son distintos
- Porque cambió el problema que se estaba resolviendo
- Porque los usuarios cambiaron su forma de pensar o sus percepciones
- Porque cambió el ambiente de negocios (mercado, etc.)

3¿Qué es Elicitación de requisitos?¿Cual es el impacto de los errores en la etapa de requisitos?

Elicitación de Requisitos

• *Es el proceso de adquirir ("eliciting") [sonsacar] todo el conocimiento relevante necesario para producir un modelo de los requerimientos de un dominio de problema.*

Objetivos:

- *Conocer el dominio del problema para poder comunicarse con clientes y usuarios y entender sus necesidades.*
- *Conocer el sistema actual (manual o informatizado).*

- Identificar las necesidades, tanto explícitas como implícitas, de clientes y usuarios y sus expectativas sobre el sistema a desarrollar.
- La elicitación de requisitos es una actividad principalmente de carácter social, mucho más que tecnológico.
- Los problemas que se plantean son por tanto de naturaleza psicológica y social, más que técnicos.
- Nota: Requisitos = 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

Problemas de comunicación

- Dificultad para expresar claramente las necesidades.
- No ser conscientes de sus propias necesidades.
- No entender cómo la tecnología puede ayudar.
- Miedo a parecer incompetentes por ignorancia tecnológica.
- No tomar decisiones por no poder prever las consecuencias, no entender las alternativas o no tener una visión global.

¿quién tiene estas dificultades?

- Cultura y vocabulario diferentes.
- Intereses distintos en el sistema a desarrollar.
- Medios de comunicación inadecuados (diagramas que no entienden los clientes y usuarios).
- Conflictos personales o políticos.

¿quién tiene estas dificultades ?

Elicitación de Requisitos

- Limitaciones cognitivas (del desarrollador)
- No conocer el dominio del problema.
- Hacer suposiciones sobre el dominio del problema.
- Hacer suposiciones sobre aspectos tecnológicos.
- Hacer simplificaciones excesivas.
- Conducta humana
- Conflictos y ambigüedades en los roles de los participantes.
- Pasividad de clientes, usuarios o ingenieros de requisitos.
- Temor a que el nuevo sistema lo deje sin trabajo.
- Técnicos
- Complejidad del dominio del problema.
- Complejidad de los requisitos.
- Múltiples fuentes de requisitos.
- Fuentes de información poco claras.

3) Defina proceso de licitación. Describa la técnica JAD.

Idem.

JAD:

4. Describir al menos 3 problemas que pueden surgir cuando los requisitos deben obtenerse de tres o mas clientes diferentes.

(Ian Sommerville)

Obtener y comprender los requerimientos de estos stakeholders, trae problemas como:

1. Debido a que los diferentes stakeholders tienen requerimientos distintos, que pueden expresar de varias formas. Los ingenieros de requerimientos tienen que considerar todas las fuentes potenciales de requerimientos y descubrir las concordancias y los conflictos que pueden surgir.
2. Inevitablemente, los stakeholders tienen opiniones diferentes sobre la importancia y prioridad de los requerimientos, y algunas veces estas opiniones están reñidas. Durante el proceso se deberían organizar frecuentes negociaciones con los stakeholders para que se pueda llegar a acuerdos.
3. Es imposible satisfacer completamente a todos los stakeholders. pero si algún stakeholder piensa que sus opiniones no se han considerado adecuadamente. Deliberadamente puede intentar socavar el proceso de ingeniería de requerimientos.

4- Como solucionaría el problema de tener los usuarios de su futuro proyecto en distintas ciudades y. los que se encuentran en la misma ciudad no se ponen de acuerdo con lo que quieren

En primer lugar, como se dificulta una reunión por cuestiones de distancia, lo que propondría es realizar un cuestionario para mandarlo vía a los distintos usuarios y así recibir las respuestas de estos. Otra opción sería coordinar una fecha y hora para la realización efectiva de una videoconferencia de manera tal de solventar los problemas y darles solución.

En segundo lugar, para solucionar el problema de que no se ponen de acuerdo, propondría utilizar una técnica de elicitación, como lo es la lluvia de ideas, ya que esta es útil para resolver la falta de consenso entre los usuarios.

4- Describa un Diagrama de Transición de Estados y modelice a través de esta técnica los estados de software de un lavarropas.

4. Describa las tablas y diagramas de transición de estados. Hagan ejemplo

Diagramas de transición de estados

Un diagrama de transición de estados describe al sistema como un conjunto de estados donde el sistema reacciona a ciertos eventos posibles (externos o internos).

$$f(S_i, C_j) = S_k$$

Al estar en el estado S_i , la ocurrencia de la condición C_j hace que el sistema cambie al estado S_k .

Componente:

- *Estados*: identifican un período de tiempo de un objeto/entidad (no instantáneo) en el cual está esperando alguna operación
- *Transición*: Paso de un estado a otro
 - Evento [Condición] / Acción
- *Estado Inicial/ Final*

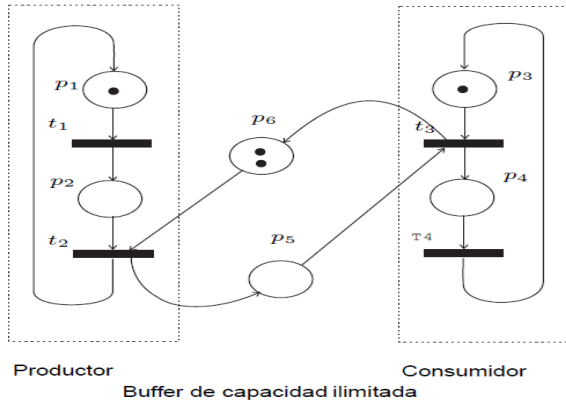
Tablas de transición de estados

Una tabla de transición de estados es una tabla que muestra que estado se moverá la máquina de estados, basándose en el estado actual y otras entradas. Una tabla de estados es esencialmente una tabla de verdad en la cual algunas de las entradas son el estado actual, y las salidas incluyen el siguiente estado, junto con otras salidas.

Componente:

- Estado actual.
- Estrada.
- Próximo Estado.

2. Describa las Redes de Petri. Cuál es su aplicación? Modelice el problema de Productor-Consumidor con buffer ilimitado.



4. Describa las componentes de una Red de Petri y los conceptos para su ejecución. Modelice el funcionamiento de una fábrica donde se produce un tipo de artículo que necesita una materia prima determinada. La fábrica cuenta con tres máquinas iguales en las que se puede realizar el procesamiento. Cada máquina puede procesar de a un artículo por vez.

• Definición formal

Una estructura de Red de Petri es una 4-upla

$$C = (P, T, I, O)$$

Lugares: (o sitios)

$P = \{P_1, P_2, \dots, P_m\}$

Transiciones (eventos)

$T = \{T_1, T_2, \dots, T_n\}$

Función de
entrada

$I: T \rightarrow P^\infty$

Función de
salida

$O: T \rightarrow P^\infty$

Multigrafo (de un nodo puede partir más de un arco), bipartito, dirigido

4) El Diagrama de Transición de Estados y las Redes de Petris describen diferentes aspectos. ¿Cuáles son? Dé un ejemplo sencillo de cada notación.

4. Describa el modelo de Casos de Uso.

Resaltado en fotocopia.

3- Definición de calidad y calidad total. Mejoramiento de la Calidad. Enumere las Normas de Calidad que conoce.

- Calidad es un concepto manejado con bastante frecuencia en la actualidad, pero a su vez, su significado es percibido de distintas maneras.
- Al hablar de bienes y/o servicios de calidad, la gente se refiere normalmente a bienes de lujo o excelentes con precios elevados.
- Su significado sigue siendo ambiguo y muchas veces su uso depende de lo que cada uno

entiende por calidad, por lo cual es importante comenzar a unificar su definición.

CROSBY: conformidad de los requerimientos

W. EDWARDS DEMING: Calidad en términos de la satisfacción del cliente

ARMAND V. FEIGENBAUM: Calidad como una relación directa entre los productos y servicios, y las necesidades del cliente.

KAORU ISHIKAWA: Establece que los requerimientos y necesidades de los clientes van cambiando lo que conlleva a una definición cambiante.

Conclusión...

- Se puede ver que coinciden en “conformar requerimientos del producto o servicio”, “lograr la satisfacción del cliente” y las relaciones entre estos conceptos.
- Pero la evaluación de los mismos, continúa dependiendo de la evaluación de sus características particulares, de manera subjetiva.
- En consecuencia lo más importante es definir claramente las características que nos interesa evaluar y su forma de evaluación.

Definiciones de Calidad

* Capacidad de un producto o servicio para servir satisfactoriamente a los propósitos del usuario mediante su utilización.

* Conformidad con los requisitos explícitos e implícitos de un cliente.

* Ausencia de defectos e imperfecciones.

Calidad Total: *Es un proceso con evolución continua, que requiere de un sistema consistente, que a través de su buen manejo continuo debe agregar valor en cada etapa del proceso, siendo sus fines satisfacer al cliente y obtener lucros permanentes, lo cual exige medición y luego mejoras continuas sin fin.*

• Mejoramiento de la Calidad

En el proceso del mejoramiento de la calidad se debe evitar escatimar costos iniciales a costa de merma en la calidad, ya que eso traerá como consecuencia costos totales altos. Además debe tomarse en cuenta que al entregar un sistemas en tiempo sin confiabilidad es entregar un producto defectuoso en tiempo, por eso debe realizarse una planificación no especulativa para la fecha de entrega del producto final y por último, no debe pensarse que al remover defectos individuales del software no es mejorar la calidad.

Normas de calidad

Calidad del producto:

- 1) ISO/IEC 9126 mas reciente ha aparecido la norma ISO 25000.

Calidad del proceso:

- 1) CMM
- 2) CMMI

4. Defina calidad y calidad total. Describa las normas de calidad que conozca.

Hecho y ademas resaltado.

5- Describa las ampliaciones del análisis estructurado para modelar sistemas de tiempo real.

Sistemas tiempo real: STR

Características:

- Responden al mundo real

- En un tiempo prefijado
- Deben ser fiables, reinicializables y recuperables a fallas

Ejemplos: Control de procesos, investigación médico.

Ampliaciones del análisis estructurado:

- Muchas aplicaciones de software son dependientes del tiempo y procesan más información orientada al control que a los datos, por ej: control de naves, procesos de fabricación, etc...
- Las primeras ampliaciones que se hacen a este método están efectuadas por Ward y Mellor, y posteriormente lo hacen Hatley y Pirbhai y GoldSmith.
- Estas ampliaciones permiten reflejar el flujo de control y el procesamiento de control, así como el procesamiento y el flujo de datos.

Notación: (Ward y Mellop)

- Los procesos de control representan con burbujas punteadas las funciones que coordinan o sincronizan.
- Los flujos de control representan con líneas punteadas señales o interrupciones.
- Los almacenamientos de control representan mediante líneas dobles punteadas los datos de control permanentes del sistema.
- La descripción de las burbujas pueden completarse con diagramas de transición de estados.

Notación: (Hatley y Pirbhai)

- Genera un diagrama de flujo de control (DFC) que contiene los mismos procesos que el DFD y muestra los flujos de control.
- Los flujos de control se representan con líneas punteadas.
- Usa una barra entre el flujo de control para referenciar una especificación de control.

Técnicas SADT: Análisis y diseño estructurado

La parte del análisis representa al sistema como un conjunto ordenado de diagramas (actigramas y datagramas).

Cada diagrama representa una transformación, y a lo sumo se utilizan seis diagramas para describir una función (si no la función debe ser redefinida).

Falta las de GoldSmith.

5. Defina la evolución del concepto de calidad al concepto de calidad total.

Para alcanzar la calidad total, es necesario incluir en los planes estratégicos de la organización, el concepto de gestión de calidad, el cual consta de los siguientes procesos:

- Planeamiento de la calidad: Lo que se busca es precisar los clientes internos y externos, determinar las necesidades y elaborar productos y servicios que la satisfagan, con el objetivo de planear la calidad.
- Control de la calidad: Monitorear, medir, comparar y ajustar productos y servicios de acuerdo a lo planificado para identificar imperfecciones de la calidad.
- Mejoramiento de la calidad: Eliminación de procesos fracasados y del despilfarro de los errores con el objetivo de corregir las imperfecciones de la calidad.

5. Describa el concepto de calidad total. ¿Quién fue Edward Deming?. Explique el "circulo de Deming"

Calidad Total: Es un proceso con evolución continua, que requiere de un sistema consistente, que a

través de su buen manejo continuo debe agregar valor en cada etapa del proceso, siendo sus fines satisfacer al cliente y obtener lucros permanentes, lo cual exige medición y luego mejoras continuas sin fin.

Edward Deming: Un pionero y profeta de la Calidad Total. Un estadista, profesor y fundador de la Calidad Total. Ignorado en EEUU, Deming fue a Japón en 1950 y enseñó a los administradores, ingenieros y científicos Japoneses como producir calidad.

Treinta años después, luego de ver un documental en televisión, titulado, "Si Japón puede, porque nosotros no" corporaciones como Ford, General Motors y Dow Chemical, entre otras, buscaron la asesoría de Deming. A partir de esos momentos comenzó a ser ampliamente solicitado.

Deming compartió sus ahora famosos "14 puntos" y "7 pecados mortales" con algunas de las corporaciones más grandes de América.

Sus estándares de calidad se convirtieron en sitios comunes en los libros de administración.

Círculo de Deming:

- **Disminuyen los costos:** Hay menos desperdicios, menor reproceso, menos errores, menos retrasos, se aprovecha mejor el tiempo de máquinas, sólo se emplean los insumos necesarios.
- **Mejora la productividad:** Las horas hombre y las horas máquina no se malgastan, se aprovechan mejor. Los recursos renovables no se deterioran.
- **Conquista el mercado:** Se planifica y se produce en relación con las preferencias y requerimientos del cliente, y se mejora continuamente en ese sentido.
- **Se permanece en el negocio:** El cliente satisfecho volverá a comprar y recomendará hacer los mismos a otros clientes.
- **Se incrementa el trabajo:** El mercado ha sido conquistado y está en aumento. El operario está orgulloso de su trabajo.

5. Describa los modelos del Análisis Estructurado. Describa las extensiones para Sistemas de Tiempo Real.

Modelos del análisis estructurado:

1. Modelado de funciones del sistema.

Diagrama de flujo de datos: Es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por "conductos" y almacenamientos de datos.

Representa la transformación de entradas a salidas y es también llamado diagrama de burbujas o modelo de proceso.

Es una herramienta comúnmente utilizada por sistemas operacionales en los cuales las funciones del sistema son de gran importancia y son más complejas que los datos que este maneja.

2. Modelado de datos del sistema.

Diagrama Entidad-Relación: Es una herramienta para visualizar la información que maneja el sistema y su vinculación.

Es un módulo de red que describe con un alto nivel de abstracción la distribución de datos almacenados en un sistema.

Es efectiva para la comunicación con los usuarios ejecutivos de mayor nivel de una organización o el grupo de administración de base de datos.

Entidades: Simbolizan objetos (concretos o conceptuales) de la aplicación.

Relaciones: Simbolizan las asociaciones entre entidades.

Atributos: Describen propiedades de las entidades y relaciones y pueden ser identificadores (claves) o descriptores.

3. Modelado del comportamiento del sistema

Diagrama de transición de Estados: Es la herramienta que permite representar el comportamiento temporal del sistema, indica cómo se mueve el sistema de un estado a otro. Estados: Representa un período de tiempo en el que el sistema sigue un comportamiento observable.

Cambios de estado: Representan transiciones entre un estado u otro.

Condiciones previas: Asociadas a cada cambio de estado.

Condiciones resultantes: Asociadas a cada cambio de estado.

Sistemas tiempo real: STR

Características:

- Responden al mundo real
- En un tiempo prefijado
- Deben ser fiables, reinicializables y recuperables a fallas

Ejemplos: Control de procesos, investigación médico.

Ampliaciones del análisis estructurado:

- Se utilizan para modelizar aplicaciones conducidos por sucesos que producen información de control y/o que producen información con limitaciones de tiempo.

Notación: (Ward y Mellop)

- Los procesos de control representan con burbujas punteadas las funciones que coordinan o sincronizan.
- Los flujos de control representan con líneas punteadas señales o interrupciones.
- Los almacenamientos de control representan mediante líneas dobles punteadas los datos de control permanentes del sistema.
- La descripción de las burbujas pueden completarse con diagramas de transición de estados.

Notación: (Hatley y Pirbhai)

- Genera un diagrama de flujo de control (DFC) que contiene los mismos procesos que el DFD y muestra los flujos de control.
- Los flujos de control se representan con líneas punteadas.
- Usa una barra entre el flujo de control para referenciar una especificación de control.

Técnicas SADT: Análisis y diseño estructurado

La parte del análisis representa al sistema como un conjunto ordenado de diagramas (actigramas y datagramas).

Cada diagrama representa una transformación, y a lo sumo se utilizan seis diagramas para describir una función (si no la función debe ser redefinida).

5. Describa las técnicas de comunicación: cuestionarios y lluvia de ideas.

5. Defina calidad y calidad total. Cómo pueden superarse los costos de la no calidad? Qué es mejoramiento de la calidad?.

Cómo pueden superarse los costos de la no calidad?

Los costos se pueden superar realizando mediciones adecuadas de la calidad y de los costos desde la perspectiva del cliente, ya que si no se mide algo, no se puede mejorar.

Mejoramiento de la Calidad

En el proceso del mejoramiento de la calidad se debe evitar escatimar costos iniciales a costa de merma en la calidad, ya que eso traerá como consecuencia costos totales altos. Además debe tomarse en cuenta que al entregar un sistema en tiempo sin confiabilidad es entregar un producto defectuoso en tiempo, por eso debe realizarse una planificación no especulativa para la fecha de entrega del producto final y por último, no debe pensarse que al remover defectos individuales del software no es mejorar la calidad.

5) *Describe el Modelo en espiral.*

Modelo en espiral

• El modelo espiral (Boehm)

- Combina las actividades de desarrollo con la gestión del riesgo
- Trata de mejorar los ciclos de vida clásicos y prototipos.
- Incorpora objetivos de calidad y gestión de riesgos
- Elimina errores y alternativas no atractivas al comienzo
- Permite iteraciones, vuelta atrás y finalizaciones rápidas
- Cada ciclo empieza identificando:
 - Los objetivos de la porción correspondiente
 - Las alternativas
 - Restricciones
- Cada ciclo se completa con una revisión que incluye todo el ciclo anterior y el plan para el siguiente

5. *¿Cuáles son las actividades genéricas a cualquier Modelo de Proceso?*

CREO QUE SON LAS DE LA TEORIA 2, LA OTRA ES QUE SEAN LAS DE LA TEORIA 7.

5. *¿Cuáles son las actividades genéricas en cualquier proceso?*

CREO QUE SON LAS DE LA TEORIA 2, LA OTRA ES QUE SEAN LAS DE LA TEORIA 7.

6- *Definición de calidad. Componentes de la calidad. Definición de calidad total.*

Componentes

- Calidad de la Infraestructura
- incluye, por ejemplo, la calidad de las redes, y sistemas de software.
- Calidad de Software
- de las aplicaciones de software construidas, o mantenidas, o con el apoyo de IS.
- Calidad de Datos
- que ingresan en el sistema de información.
- Calidad de Información
- está relacionada con la calidad de los datos.
- Calidad de gestión
- incluye el presupuesto, planificación y programación.
- Calidad de servicio
- incluye los procesos de atención al cliente

6. *Definición de calidad. Definición de calidad total. ¿Qué es CMM? Explique.*

CMM:

6. *Defina Calidad del Software. ¿Cómo se divide? Describa.*

Calidad de Software

Se divide en :

- Calidad del producto obtenido
 - Calidad del proceso de desarrollo
- Son dependientes.

- **Producto (Hatton, 1995)**

- Un producto es de buena calidad si le sirve a quien lo adquiere y si este lo usa para realizar las tareas para lo que fue concebido.
- Aunque el software tiene aspectos intangibles, un producto software es sin embargo un bien en sí mismo e incluye sus documentos asociados.

- **Proceso**

- Un proceso malo, mal concebido e implementado generará productos de mala calidad.
- Un proceso bueno, bien concebido e implementado generará la mayor cantidad de veces productos de buena calidad.

* La calidad de un producto de software debe evaluarse usando un modelo de calidad que tiene en cuenta criterios para satisfacer las necesidades de los desarrolladores, mantenedores, adquirentes y usuarios finales.

* Los modelos de calidad pueden ser utilizados para construir mejores productos y asegurar su calidad.

* Se han desarrollado varios modelos de calidad para diferentes productos y procesos software.

6. Describa la técnica de Lluvia de Ideas.

Marcado en apunte.

6- ¿Qué es un prototipo?. Describa los tipos de prototipos que conozca. ¿Cuáles son los proyectos candidatos a prototipar?

1)

- Un prototipo es un producto parcialmente desarrollado que permite que clientes y desarrolladores examinen algunos aspectos del sistema propuesto, y decidan si éste es adecuado o correcto para el producto terminado.
- Esta es una alternativa de especificación para tratar mejor la incertidumbre, la ambigüedad y la volubilidad de los proyectos reales.

2)

Tipos

- **Evolutivos**

- El objetivo es obtener el sistema a entregar.
- Permite que todo el sistema o alguna de sus partes se construyan rápidamente para comprender o aclarar aspectos y asegurar que el desarrollador, el usuario y el cliente tengan una comprensión unificada tanto de lo que se necesita como de lo que se propone como solución

- **Descartables**

- No tiene funcionalidad
- Se utilizan herramientas de modelado

3)

Proyectos candidatos

- Usuarios que no examinarán los modelos abstractos
- Usuarios que no determinarán sus requerimientos inicialmente
- Sistemas con énfasis en los formatos de E/S más que en los detalles algorítmicos
- Sistemas en los que haya que explorar aspectos técnicos

- Si el usuario tiene dificultad al tratar con los modelos gráficos para modelar los requerimientos y el comportamiento
- Si se enfatiza el aspecto de la interfaz humana

Para asegurar el éxito

- Debe ser un sistema con el que se pueda experimentar
- Debe ser comparativamente barato (< 10%)
- Debe desarrollarse rápidamente
- Énfasis en la interfaz de usuario
- Equipo de desarrollo reducido
- Herramientas y lenguajes adecuados

6) ¿ Qué son las metodología ágiles ? Describa sus características

“Es un enfoque **iterativo** e **incremental** (evolutivo) de desarrollo de software”

Iterativo: El desarrollo iterativo es una estrategia de reproceso en la que el tiempo se separa para revisar y mejorar partes del sistema.

Incremental: Es una estrategia programada y en etapas, en la que las diferentes partes del sistema se desarrollan en diferentes momentos o en diferentes velocidades, y se entregan a medida que e completan.

Objetivos:

- Producir software de alta calidad con un costo efectivo y en el tiempo apropiado.
- Esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto.
- Ofrecer una alternativa los procesos de desarrollo de software tradicionales, caracterizados por ser rigidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Reseña...

Una Metodología Ágil es aquella en la que “se da prioridad a las tareas que dan resultados directos y que reducen la burocracia tanto como sea posible” [Fowler], adaptándose además rápidamente al cambio de los proyectos. Ese enfoque ha sido utilizado desde hace más de dos décadas por un grupo de profesionales de software.

7- Modelizar a través de una tabla de decisión la recepción de presentaciones en un congreso.

Las presentaciones se envían para un Workshop en particular o al Congreso en General. Las presentaciones pueden ser resúmenes o artículos completos.

En el caso que sean resúmenes y siempre que la evaluación sea satisfactoria, se le solicita a los autores envíen el artículo completo. Si la evaluación NO es satisfactoria, se rechaza el artículo o se reenvía el resumen para su adaptación a los autores. En otro caso se acepta la presentación.

7. Modelizar usando una tabla de decisión:

Se trata de la obtención del carnet de conducir. Puede ser la primera vez o una renovación.

Siendo la primera vez se cobra un arancel de 75\$. En otro caso se cobra 30\$.

Si se paga al contado se le hace un 10% de descuento.

En el caso de ser una persona > de 70 años, la duración del carnet es de 1 año, en otro caso es de 5 años.

De tratarse de la primera vez se toma un examen teórico, otro práctico y otro de aptitud. En caso de ser renovación se toma el examen de aptitud.

7. Modelizar usando una tabla de decisión:

Se trata de la obtención del carnet de conducir. Puede ser la primera vez o una renovación.

Siendo la primera vez se cobra un arancel de 75\$. En otro caso se cobra 30\$. Si se paga al

contado se le hace un 10% de descuento. En el caso de ser una persona > de 70 años, la duración del carnet es de 1 año, en otro caso es de 5 años. De tratarse de la primera vez se toma un examen teórico, otro práctico y otro de aptitud. En caso de ser renovación se toma el examen de aptitud.

7. Especifique el funcionamiento de un contestador telefónico. Modele utilizando DTE.

7- Modelizar usando una tabla de decisión: *Se trata de la suscripción a revistas de publicación periódica.*

La suscripción puede ser a revistas nacionales o internacionales. En el caso de ser revistas nacionales se envía una nota por correo y en el caso de revistas internacionales se envía un mail para suscribirse. Si la periodicidad de la publicación es anual se salda el monto completo de la suscripción. En el caso que sea una revista internacional y la forma de pago sea por depósito deben solicitarse los datos de la cuenta. En cualquier caso que la forma de pago sea a través de un giro deben solicitarse los datos del individuo.

7. *Modelizar usando una tabla de decisión la organización de un torneo de handball. Los equipos son locales o del exterior y pueden pertenecer a la categoría menores o cadetes. Hay equipos femeninos o masculinos. Todos los equipos femeninos juegan por la mañana. Los locales pueden planificarse durante los días de la semana. La categoría menores masculina juega los sábados. Todos los equipos deben inscribirse.*

7. *Defina calidad y calidad total. Describa las normas ISO y CMM.*

7. Modelizar usando una tabla de decisión el comportamiento de un brazo robótico:

Si el objeto está ubicado a más de 1 metro, se provoca un desplazamiento.

Si el objeto está por encima del alcance, se gira 40° hacia arriba.

Si el objeto es redondo, se abre la pinza y en ese caso si estaba a más de un metro se envía una señal.

Luego de los acomodamientos necesarios, en cualquier caso se realiza la captura del objeto.

7) *Represente a través de tablas de decisión. Si Adriana está en el padrón, cumple condición y no está doblemente empadronada o lo está pero tiene el comprobante de elección de lugar de voto, puede votar a Centro y a Claustro. Si está doblemente empadronada y no tiene comprobante sólo puede votar a Centro, lo mismo ocurre de no cumplir condición.*

7. *Tabla de Decisión:*

Los pacientes pueden concurrir en forma particular o con Obra Social, en ese caso puede ser IOMA u OSDE. El arancel particular es de \$50, por IOMA se entrega un Bono, por OSDE se hace un cupón. Los remedios por IOMA se recetan en un formulario tipo A, por OSDE en un formulario tipo B y particular en un formulario tipo C. Los pacientes con obra social se registran en un listado.

(Nota: en el final dudamos si siempre un paciente requiere medicamentos, la profesora se había ido, estaba otro profesor, que no pudo hacer más que decirnos que aclaremos qué considerábamos y en función de eso plantear la solución.)

7. *Se desea modelizar mediante una tabla de decisión el siguiente problema. Ignacio ha decidido a comprarse un vehículo y hay que ayudarlo a decidir que tipo de vehículo comprar. Si logra llegar a un acuerdo en su trabajo entonces comprará una furgoneta para hacer viajes de reparto. Caso contrario debe evaluar la plata con la que cuenta. Si tiene ahorrado mas de 100 mil pesos y su hermano le presta dinero extra entonces comprará un minibus para realizar traslados de pasajeros ambulatorios, pero si su hermano no tiene dinero extra para prestarle entonces comprará una camioneta 4X4 para trabajar en el campo de un amigo. Si sus ahorros no superan los 100 mil pesos, sin importar que su hermano le preste, entonces comprará un utilitario para envío de encomiendas.*

Realizar la reducción en el caso de que sea posible.

7) Realizar la tabla de decisión de manera completa y si fuera necesario realizar la reducción correspondiente.

Una empresa quiere determinar los descuentos a aplicar al sueldo de sus empleados según estos criterios: La empresa registra presentismo, cuándo un empleado no lo cumple obtendrá un 15% de descuento. Los básicos mayores a 5000\$ obtendrán 5% de descuento en concepto de ganancias. Para empleados asociados al sindicato existe un descuento del 5%. Los descuentos son acumulables.

6. Modelizar a través de una tabla de decisión el problema de rendir asignaturas por los alumnos. Los alumnos pueden rendir "libre" o "con cursada aprobada". El examen puede ser escrito, oral o ambos. Los alumnos que dan libre pasan en primer turno. Si el examen es sólo escrito la nota está a la semana siguiente. Siempre que el examen tenga una parte oral se califica en el momento.

Final Gero, Diciembre (1ra o 2da, no me acuerdo)

- 1) Que conocimientos debe tener un ingeniero de software?*
- 2) Que es un requerimiento? ¿Que tipos de requerimientos conoce? Describa*
- 3) definir que es una especificacion de requisitos de software (SRS). Indicar algunos 5 criterios a cumplir para una buena SRS.*
- 4) Indicar la afirmacion falsa respecto de un caso de uso*
 - a) Su relacion se expresa mediante una interzaf.*
 - b) Sirven para modelar el contexto del problema.*
- 5) Describa el modelo en espiral.*
- 6) Describa los modelos de calidad presentados en la materia.*