***Ingenieria de Software***

***Resumen teorico***

**¿Que es software?**

Conjunto de programas, procedimientos, datos y documentación que operan sobre un sistema de computo

**Tipos:**

**-** Genéricos: Softwares de propósito general, no están optimizados ni especificados para un cliente en particular.

- Personalizados: Softwares generados en base a las especificaciones del cliente.

**¿Que es un Stakeholder?**

Se refiere a cualquier persona o grupo que se vera afectado por el sistema, directa o indirectamente.

(“Proceso” no es lo mismo que “Proceso de Software”)

**Características de cualquier proceso:**

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

**¿Que es un modelo de proceso?**

Es una representación simplificada de un proceso de software, estos modelos pueden incluir actividades que son parte de los productos y procesos, y el papel de las personas que están involucradas.

**Estos se basan en los siguientes modelos:**

* En cascada: Se define el sistema en un conjunto de fases.
* Iterativo: El sistema se desarrolla rápidamente a partir de las especificaciones abstractas, luego se refina con las especificaciones del cliente.
* IS basada en componentes: Se basa en que el sistema ya esta hecho pero separado en partes, solo falta la unificación de estas.

**Tipos de modelos:**

* Modelo en cascada.
* Modelo en V.
* Modelo de prototipos.
* Desarrollado por fases:
  + - Incremental
    - Iterativo.
* Modelo en espiral.

**Modelo en cascada:**

**Pros:**

* Las etapas se representan cayendo en cascadas.
* Cada etapa se tiene que completar antes de que empiece la otra.
* Fácil explicación hacia los clientes.

**Contras:**

* No existen resultados hasta que esté todo terminado.
* Las fallas más importantes se encuentran al final.
* Resolver las fallas es muy difícil en las etapas finales.
* La necesidad de pruebas aumenta en las etapas finales.
* El “Congelar” una etapa es poco realista.
* Si existen cambios de parecer y/o cambio de ambiente y/o cambio de etc. habria que empezar a resolver los cambios desde donde se inicio.

**Modelo en V:**

* Muestra cómo se relacionan las actividades de prueba con las de análisis y diseño.
* Sugiere que la prueba unitaria y de integración también se aplique al diseño.
* La vinculación entre los lados “derechos” e “izquierdos” hace que, si se encuentran errores durante la validación y la verificación, el lado izquierdo de la V se puede volver a ejecutar.

**Modelo de prototipos:**

* Un prototipo es un producto parcialmente desarrollado que permite a clientes y desarrolladores examinar algunos aspectos finales y decidan si este es el correcto para seguir con el producto.
* Es una alternativa para tratar la incertidumbre.

**Tipos de prototipos:**

Evolutivos:

* El objetivo es tener el sistema a entregar.
* Permite que todo el sistema, o parte de él, se desarrolle rápidamente para ayudar a comprender los aspectos al desarrollador, cliente y al usuario de manera de unificar la solución.

Descartables:

* No tienen funcionalidad.
* Se utilizan herramientas de modelado.

Proyectos candidatos:

* Usuarios que no examinaron los modelos abstractos.
* Usuarios que no definieron sus requerimientos iniciales.
* Sistemas que prestan atención a la E/S más que a los detalles.
* Sistemas donde haya que explorar aspectos técnicos.
* Usuarios con problemas con los modelos gráficos para modelar los requerimientos.
* Si se enfatiza el aspecto de la interfaz humana.

Desarrollo por fases:

* Incrementales: El sistema es particionado en subsistemas y en cada entrega se agrega un nuevo subsistema.
* Iterativo: Se agrega todo el sistema desde un inicio y luego se sacan versiones nuevas con mejoras y cambios.

**Modelo en espiral:**

* Combina actividades de desarrollo con la gestión del riesgo.
* Trata de mejorar los ciclos de vida clásicos y de prototipos.
* Incorpora objetivos de calidad y gestión de riesgo.
* Elimina errores al inicio.
* Permite iteraciones, vuelta atrás y finalizaciones rápidas.
* Cada ciclo comienza identificando: Objetivos, alternativas y restricciones.
* Un ciclo se finaliza con la revisión del ciclo anterior y un plan para el siguiente.

**Metodologias Agiles:**

Es un enfoque iterativo e incremental para el desarrollo de software.

* Iterativo: El tiempo se particiona para revisar las partes del sistema.
* Incremental: El sistema se va desarrollando en tiempos distintos o en velocidades distintas, luego se entrega todo.

**Objetivos:**

* Hacer el software de calidad con tiempo y costo apropiados.
* Afianzar los valores que permiten a los equipos desarrollar software rapidamente y adaptandose a los cambios que puedan surgir durante el proyecto.
* Es una alternativa a los **procesos de desarrollo de software** tradicionales.

Una **metodología ágil** es aquella a la que se priorizan actividades que producen resultados directos y que reducen la burocracia al máximo, adaptándose a los cambios.

**Requerimientos:**

* Condiciones o requisitos que debe cumplir el sistema con el propósito satisfacer las necesidades especificadas.

**IEEE:** Condición que necesita el usuario/sistema/subconto del sistema para resolver un problema.

**Licitacion de requisitos/requerimientos:**

La licitación es el proceso por el cual se trata de sacar todo aquel dato relevante al cliente que sirva a la hora de emplear un modelo de solución para la resolución de un problema.

Objetivos:

* Conocer el dominio del problema para poder comunicarse con el usuario o cliente, y entender sus necesidades.
* Ver con que sistema se están manejando actualmente, ya sea uno manual o informatizado.
* Entender las necesidades implícitas y explícitas de clientes o usuarios, y ver qué esperan ellos del sistema.

Problemas relacionados con la comunicación:

* Dificultad para expresar claramente las necesidades.
* No ser consciente de las necesidades.
* No entender como la tecnología ayudaría con los problemas
* No tomar decisiones por no poder prever las consecuencias.
* Medios de comunicación inadecuados (diagramas que no se entiendan)
* Conflictos personales o políticos.

Impacto de los errores en la etapa de requerimientos:

* El software resultante puede no cumplir las necesidades de los usuarios.
* Las interpretaciones múltiples de requerimientos pueden causar problemas entre clientes y desarrolladores.
* Puede gastarse tiempo y dinero construyendo un sistema erróneo.

**Tecnicas de licitacion:**

* Muestreo de la información, las formas y datos existentes.
* Visitas al lugar.
* Observación del ambiente de trabajo.
* Cuestionarios
* Entrevistas.
* Planeación conjunta de requerimientos **JRP** o **JAD**.
* Brainstorming.

Planeacion conjunnta de requerimientos JRP o JAD:

* Proceso mediante el cual se realizan reuniones altamente estructuradas con el objetivo de entender el problema en cuestión.
* Necesitan un entrenamiento extenso.
* Amplia participación de los usuarios.
* Se trabaja sobre lo que se va generando.
* Reduce el tiempo de exploración de requisitos.

Ventajas:

* Usuarios y clientes involucrados en el proyecto.
* Reduce el tiempo de obtención de requerimientos.
* Si se incorporan prototipos, los mismos ya confirman el diseño del sistema.

Desventajas:

* Difícil coincidir los horarios de todos los involucrados.
* Difícil llevar a cabo una reunión con tanta gente involucrada.

Participantes del JRP:

* **Patrocinador**: Es quien toma las decisiones finales y esta a cargo de todos los departamentos involucrados.
* **Facilitador**: Posee amplios conocimientos de comunicación y negociación.
* **Usuarios y Gerentes**: Los usuarios transmiten los requerimientos a los gerentes y estos los aprueban o no.
* **Secretarios**: Llevan nota de la sesión y publican los resultados con herramientas CASE.
* **Equipo TI**; Toman nota de los requerimientos.

**Ingenieria de requerimientos:**

La **ingeniería de requerimientos** es una disciplina para desarrollar una especificación completa y concisa (**COMCON)** y no ambigua. Que se utiliza como punto de consenso entre todas las partes involucradas y donde se describen las funcionalidades del sistema.

Es un proceso mediante el cual se intercambian diferentes puntos de vista para modelar lo que se desea que el sistema vaya a realizar. Este proceso combina métodos, herramientas y actores donde el resultado es un modelo del cual se genera un documento de requerimientos.

Se enfoca en recolectar, organizar y documentar los requerimientos del sistema, también para mantener acuerdos sobre los cambios de requerimientos entre clientes y los equipos de desarrollo.

**Importancia:**

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

**La ingenieria de requerimientos incluye:**

Estudio de viabilidad: Por lo general se aplica en sistemas nuevos y consiste en a partir de una descripción resumida del sistema se elabora un informe breve donde se recomienda si realizar el proyecto o no.

Obtencion y analisis de requerimientos:

Propiedades de los requerimientos:

* Necesario: Que su omisión trae problemas.
* Conciso: Facil de entender.
* Completo: Que no necesita ampliarse.
* Consistente: Que no es contradictorio con ningún otro.
* No ambiguo: Que tiene una sola implementación.
* Verificable: Que pueden testearse.

Tipos de requerimientos:

* Funcionales:
  + Describe la interacción entre el sistema y el ambiente. Como debería reaccionar ante determinados estímulos.
  + Detallan específicamente lo que debe hacer el sistema, incluso como **NO** debería comportarse.
  + Describe con detalle la funcionalidad del mismo.
  + Son independientes de la implementacion de la solucion

.

* No funcionales: Representan las limitaciones o restricciones del sistema y limita nuestras soluciones a la hora de la resolucion del problema .

Especificación de requerimientos:

* Permitir a los desarrolladores explicarle a los clientes como han entendido el problema.
* Explicarle a los diseñadores las funcionalidades y características que tendrá el sistema.
* Permitir a los encargados de las pruebas a los clientes que el sistema satisface las necesidades requeridas.
* Documento de **definición** de requerimientos: Listado de las funcionalidades que el cliente espera que haga el sistema.
* Documento de **especificación** de requerimientos: Mismo listado pero en términos técnicos.
* Documento de **especificación** de requerimientos de **software**: Describe contenidos y la cualidades de una buena especificación.

Aspectos basicos de una **especificacion de requerimientos**:

* **Funcionalidad**: que debe hacer el sistema.
* **Interfaces externas**: Como interactúa con el mundo exterior.
* **Rendimiento**: Disponibilidad, velocidad, etc.
* **Atributos**: Portabilidad, Seguridad, etc.
* **Restricciones de diseño**: Lenguaje de programación, limite de recursos.

Validación de requerimientos:

* Es el proceso de certificar la corrección del modelo con las intenciones del usuario.
* Trata de mostrar que el sistema lleva a cabo los requerimientos definidos.

IEEE:

* **Validación**: Al final del desarrollo se evalúa el software para ver si este cumple con los requerimientos.
* **Verificación:** Determina si un producto de software de una fase cumple con los requerimientos de la fase anterior.

**Tecnicas de especificacion de requerimientos dinamicas:**

**Historias de Usuario (HU):** ??

**Casos de Uso (CU):**

* Captura de requerimientos funcionales.
* Descompone el sistema en piezas más manejables.
* Medio de comunicación con los usuarios.
* Permite estimar el alcance del proyecto y los esfuerzos a realizar.
* Define una línea base para la documentación del sistema.
* Provee de una herramienta para el seguimiento de requisitos.

**Diagrama de transición de estado(DTE):** ???

**Redes de Petri:**

* Utilizadas para especificar sistemas de tiempo real, donde se deben representar aspectos de concurrencia.
* Los sistemas concurrentes estan diseñados para permitir la ejecucion de muchos procesos o tareas en varios procesadores o en uno solo intercalados.
* Los sistemas concurrentes deben estar sincronizados para poder permitir una comunicacion entre ellas.
* Pueden ejecutarse varias tareas en paralelo, pero son ejecutadas en un orden imprescindible.
* No son secuenciales.

**Gestion de los Requerimientos**

**¿Por que cambian los requerimientos?**

* Porque hicieron preguntas inadecuadas a las personas inadecuadas.
* Porque clientes y usuarios son distintos.
* Porque cambió el problema.
* Porque los usuarios cambiaron su forma de pensar.
* Porque el negocio tomó otro rumbo.

**Evolucion:**

* **Requerimientos duraderos:** Se mantienen a lo largo del tiempo, son relativamente estables.
* **Requerimientos Volátiles:** Cambian durante el desarrollo del sistema, o después de que se puso en operación.
* **Requerimientos cambiantes:** Estos cambian por que se modifica el ambiente o entorno.
* **Requerimientos consecuentes:** Surgen como consecuencia de la puesta en marcha del sistema.
* **Requerimientos de compatibilidad:** Cambia por que interactúa con otros sistemas que cambiaron.

**Calidad:**

Se divide en **calidad del producto** y **calidad del proceso**

* **Calidad del producto:** Un producto es de buena calidad si satisface las necesidades del usuario para los objetivos para el que fue concebido.
* **Calidad del proceso:** Un proceso mal llevado a cabo deriva en productos de mala calidad, en cambio uno bien llevado a cabo, lo contrario.

**FIN**

**???**