

## INGENIERIA DE SOFTWARE I

### CONCEPTOS DE INGENIERIA DE SOFTWARE

- **Software:** es el conjunto de programas de cómputo, procedimiento, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación (IEEE).

#### - Tipos de producto de software:

Genéricos: sistemas aislados producidos por organizaciones desarrolladoras de software y que se venden en un mercado abierto.

Personalizados: sistemas requeridos por un cliente en particular. Desarrollados por la propia organización interesada o un contratista.

#### - Clasificación de software:

De sistema (sirve a otros programas)

De gestión (proceso de información comercial)

Científico (algoritmo de manejo de números)

Empotrado (reside en memoria)

De tiempo real (coordina/analiza/controla sucesos del mundo real)

Basados en la web (sitios)

De inteligencia artificial (uso de algoritmos no numéricos para resolver problemas complejos)

#### - Historia sobre el software:

Década 1950-1960: Hardware de propósito general, con continuos cambios. "Documentación inexistente".

Década 1960-1970: Multiprogramación, multiusuario, minicomputadoras. Tiempo real. Nace el "producto de software". "Mantenimiento del software"

Década 1970-1990: Complejidad. Microprocesadores. Redes. Baja espectacular del costo del hardware. "Productividad"

Década 1990-2000: Tecnologías O.O, procesamiento paralelo. "Ingeniería"

Década 2000-actual: Aplicaciones web. Sistemas expertos. Computación Ubicua. Código abierto. "Comunicación y distribución"

#### - Características del software:

1. El software se desarrolla, no se fabrica como otros productos.

2. El software no se desgasta.

3. No sigue una curva clásica de envejecimiento.

4. Es inmune a los males que desgastan al hardware.

5. Aunque la industria tiende a ensamblar componentes, la mayoría del software se construye a medida.

- **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 este, luego que se comienza a ejecutar.

Es el estudio de técnicas relacionadas con el uso de métodos sistemáticos, disciplinados y cuantificables para el desarrollo, operación y mantenimiento de software (IEEE).

Surgió como reacción a las dificultades de desarrollar sobre la base de habilidad, experiencia o intuición individuales. Trata de dar principios y métodos que permitan producir software confiable y eficiente, al menor costo posible, para esto establece métodos, desarrolla herramientas automáticas o semiautomáticas y define procedimientos que establecen la relación de métodos y herramientas.

- **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.

Debe tener una combinación de conocimientos científicos, metodológicos, tecnológicos y administrativos.

Debe estar familiarizado con la aplicación de métodos formales: lógica, estadística, simulación y con el uso de notaciones de modelización, especificación, diseño y programación.

Debe poder aplicar metodologías de documentación, análisis, especificación, diseño, implementación y prueba. Debe conocer las ventajas y limitaciones de cada notación y técnica. Debe saber cómo y cuándo aplicarlas.

Debe conocer las tecnologías y productos: SO, lenguajes, herramientas CASE, BD, sistemas generados de interfaces, bibliotecas de código.

Debe conocer técnicas de administración de proyectos: planificación, análisis de riesgo, control de calidad, seguimiento de proyectos, control de subcontratistas, etc.

**- Responsabilidad profesional y ética:** la IS se desarrolla en un marco económico, socio y legal. Los IS deben aceptar responsabilidades más amplias que las responsabilidades técnicas. No debe utilizar sus capacidades y habilidades de forma deshonestas, o de forma que deshonre la profesión.

Confidencialidad (respetar esto de sus empleados y clientes).

Competencia (no falsificar este nivel y aceptar responsabilidades fuera de su capacidad).

Derechos de la propiedad intelectual (conocer las leyes vigentes sobre las patentes y copyright)

Uso inapropiado de las computadoras.

Existen organizaciones como ACM o IEEE que sugieren diferentes códigos de ética a respetar.

## **TECNICAS DE COMUNICACIÓN**

Hay que saber lo que el usuario quiere, como lo quiere, cuando y porque.

### **- La comunicación:**

La comunicación es la base para la obtención de las necesidades del cliente. Al hablar de necesidades en términos más técnicos, hablamos de requerimientos.

### **- Requerimientos o requisitos:**

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.

La IEEE-Std-610 la define como condición o capacidad que necesita el usuario para resolver un problema o alcanzar un objetivo.

### **- Fuentes de requerimientos:**

Documentación, stakeholders, especificación de sistemas similares.

**- Stakeholders:** el término se utiliza para referirse a cualquier persona o grupo que se verá afectado directamente o indirectamente por el sistema. (Usuarios finales, ingenieros, gerentes, expertos del dominio, etc.)

**- Puntos de vista:** es una forma de clasificar a los stakeholders.

1. 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.

2. Indirecto: representan a los stakeholders que no utilizan el sistema ellos mismos pero que influyen en los requerimientos de algún modo.

3. Del dominio: representan las características y restricciones del dominio que influyen en los requerimientos del sistema.

## **ELICITACION DE REQUISITOS**

Es el proceso de adquirir ("eliciting") todo el conocimiento relevante necesario para producir un modelo de los requerimientos de un dominio del problema.

Es una actividad de carácter social, mucho más que tecnológico. Los problemas que se plantean son por tanto de naturaleza psicológica y social, más que técnicos.

### **- Problemas de comunicación:**

- Dificultad para expresar claramente las necesidades.
- No ser conscientes de sus propias necesidades.
- No entender como la tecnología puede ayudar.
- Miedo a parecer incompetentes por ignorancia tecnológica.
- Cultura y vocabulario diferentes.
- Intereses distintos en el sistema a desarrollar.
- Medios de comunicación inadecuados.
- Conflictos personales o políticos.

### **- Técnicas de elicitation:**

#### **- Muestro de la documentación, los formularios y los datos existentes:**

Recolección de hechos a partir de la documentación existen. Organigramas. Memos, notas internas, registros contables. Solicitudes de proyectos de sistemas de información anteriores. Diagramas. Documentos de diseño.

#### **- Investigación y visitas al lugar:**

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

#### - Observación del ambiente del trabajo:

El analista se convierte en observador de las personas y actividades con el objeto de aprender acerca del sistema. Algunas ventajas son los datos confiables, se puede ver exactamente lo que se hace, análisis de disposiciones físicas, tránsito, iluminación, ruido y es económica en comparación con otras técnicas. Algunas desventajas son que la gente se siente incómoda siendo observada, las tareas están sujetas a interrupciones, algunas tareas pueden ser realizadas en horarios incómodos.

#### - 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. Se deben utilizar cuando las personas están dispersas geográficamente, cuando hay muchas personas involucradas, cuando queremos saber opiniones o problemas generales. Algunas ventajas son la respuesta rápida, son económicos, anónimos y estructurados de fácil análisis. Algunas desventajas son número de bajo respuestas, no responde a todas las preguntas, preguntas rígidas, no se puede realizar el análisis corporal, difíciles de preparar.

#### Tipos:

*Abiertos (formato libre):* diseñado para ofrecer al encuestado más flexibilidad en la respuesta. (¿Qué reportes recibe actualmente? ¿Hay problemas con los reportes?)

*Cerrados (formato fijo):* requieren la selección de una respuesta entre respuestas predefinidas anteriormente. (¿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)

#### Diseño:

Son difíciles de desarrollar, procedimientos: 1) Determinar que hechos, opiniones y de quien, quieren recolectarse. 2) Determinar qué tipo de preguntas dan mejores resultados a los hechos buscados. 3) Escribir las preguntas y examinarlas en cuanto a errores o malas interpretaciones. 4) Ensayar en una pequeña muestra de encuestados. 5) Duplicar y distribuir el cuestionario.

Deje espacio en blanco y suficiente para responder, hay que facilitar el marcado de las respuestas. Hay que mantener el estilo, el orden de las preguntas, agrupar por contenido similar y plantear primero los temas menos controvertidos.

#### Redacción:

Debe ser autocontenida y precisa (claridad), debe tener un hilo conductor entre las preguntas (flujo), debe usar el lenguaje de los encuestados (vocabulario), debe evitar preguntas muy específicas (redacción sencilla) y no hay que subestimar a las personas que lo realizan (preguntas breves), se debe evitar la parcialidad, preguntas ofensivas.

#### - Entrevistas:

Técnica de exploración mediante la cual el analista de sistema 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. Algunas ventajas son que el entrevistado se siente incluido en el proyecto, es posible obtener retroalimentación, es posible adaptar las preguntas, y se obtiene información no verbal observando las acciones y expresiones. Algunas desventajas es que son costosas, toma tiempo y recursos humanos, no es aplicable a distancia y la entrevista depende en gran parte del entrevistador.

#### Tipos:

*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. Las ventajas son que ahorran tiempo, se mantiene más fácil el control de la entrevista y se consiguen datos relevantes. Las desventajas son que puede aburrir al entrevistado y no se obtienen detalles.

*No estructuradas (abiertas):* el encuestador lleva a un tema en general. Sin preparación de preguntas específicas. Iniciar con preguntas que no dependan del contexto, para conocer el problema, la gente involucrada, etc. Las ventajas son que revelan nueva línea de preguntas, hacen más interesante la entrevista y permiten espontaneidad. Las desventajas son que se pueden dar muchos datos irrelevantes, se puede perder el control de la entrevista y parece que el entrevistador no tiene los objetivos claros.

### Tipo de información obtenida:

Opiniones, objetivos, procedimientos informales y sentimientos.

### Tipos de preguntas:

*Abiertas:* permiten 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:* permiten obtener más detalle sobre un tema puntual (¿Podría dar detalles sobre...? ¿Podría dar un ejemplo de...?)

### Organización:

Piramidal (inductivo): preguntas cerradas a abiertas.

Embudo (deductivo): preguntas abiertas a cerradas.

Diamante (combinación): preguntas cerradas a abiertas y a cerradas.

### Conducción:

Selección del entrevistado: según el requerimiento a analizar, conocer sus fortalezas, prejuicios y motivaciones, hacer una cita, respetar el horario de trabajo, establecer la duración de la entrevista, obtener el permiso del supervisor o jefe, la entrevista es personal y privada.

### Preparación de la entrevista:

Informar al entrevistado el tema a tratar, definir un “guion de entrevista”, evitar preguntas sesgadas o con intención, amenazantes o críticas, usar lenguaje claro y conciso, no incluir opinión como parte de la pregunta, evitar realizar preguntas largas y complejas. *Preparación previa:* 1) Leer los antecedentes. Poner atención en el lenguaje. Buscar un vocabulario en común. Imprescindibles para poder entender al entrevistado. 2) Establecer los objetivos de la entrevista. 3) Seleccionar los entrevistados (minimizar el número). 4) Planificación de la entrevista y preparación del entrevistado. 5) Selección del tipo de preguntas a usar y su estructura.

### Como escuchar:

Llegue con actitud positiva, haga que la otra persona se tranquilice, haga ver que está escuchando lo que dice, haga preguntas sobre lo que dice, no haga suposiciones, tome nota.

### El lenguaje corporal:

Información no verbal que comunicamos. Expresiones faciales. Contacto visual. Postura.

### **- 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 intenso entrenamiento, reduce el tiempo de exploración de requisitos, amplía la participación de los integrantes y se trabaja sobre lo que se va generando. Algunas *ventajas* pueden ser ahorro de tiempo, usuarios involucrados y desarrollos creativos. Algunas *desventajas* son que es difícil organizar los horarios de los involucrados y es complejo encontrar un grupo de participantes integrados y organizados.

Participantes: Facilitador (dirige las sesiones y tiene amplias habilidades de comunicación y negociación). Patrocinador (miembro de la dirección con autoridad sobre los departamentos que participan, responsable del proyecto, toma las decisiones finales). Secretarios (llevan el registro de la sesión y van publicando los resultados). Equipos de TI (escuchan y toman nota de los requerimientos). Usuarios y Gerentes (los usuarios comunican los requerimientos y los gerentes los aprueban).

Como planear las sesiones de JRP: 1) Selección de una ubicación para las sesiones. 2) Selección de los participantes. 3) Preparar la agenda.

Beneficios: JRP involucra activamente a los usuarios y la gerencia en el proyecto de desarrollo, reduce el tiempo de la etapa de requerimientos y 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.

Principios: “Cuanto más ideas se sugieren, mejores resultados se conseguirán”. La producción de ideas en grupo 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.

Fases de aplicación: Descubrir hechos, producir ideas, descubrir soluciones.

Es 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 y ayuda a entender al usuario y al analista.

## **PROCESO DE SOFTWARE**

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

### **Actividades fundamentales:**

- Especificación del software
- Desarrollo del software
- Validación del software
- Evolución del software

**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.

Permite gestionar las necesidades del proyecto en forma estructurada, mejora la capacidad de predecir cronogramas de proyectos, disminuye los costos y retrasos del proyecto, mejora la calidad del software y la comunicación entre equipos y evita rechazos de usuarios finales.

**Propiedades:** Necesario (su omisión provoca una deficiencia). Conciso (fácil de leer y entender). Completo (no necesita ampliarse). Consistente (no contradictorio con otro). No ambiguo (tiene una sola implementación). Verificable (puede testearse a través de pruebas).

### **Tipos:**

*Requerimientos funcionales:* describen una interacción entre el usuario y su ambiente. Como debe comportarse el sistema ante determinado estímulo. Describen lo que el sistema debe hacer, o incluso como 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, pueden ser requerimientos del producto, organizacionales, externos, etc.

*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.)

### **Especificación:**

*Objetivos:* permitir que los desarrolladores expliquen cómo han entendido lo que el cliente pretende del sistema. Indicar a los diseñadores que funcionalidad y características va a tener el sistema resultante. Indicar 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 pedido.

*Documento de definición:* listado completo de todas las cosas que el cliente espera que haga el sistema propuesto.

*Documento de especificación:* definición en términos técnicos.

*Aspectos básicos:* Funcionalidad (¿Qué debe hacer el software?). Interfaces externas (¿Cómo interactuar el software con el medio externo?). Rendimiento (velocidad, disponibilidad, tiempo de respuesta, etc.). Atributos

(portabilidad, seguridad, mantenibilidad, eficiencia). Restricciones de diseño (estándares requeridos, lenguaje, límite de recursos, etc.)

#### Técnicas de especificación:

*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, etc.)

*Dinámicas:* se considera a un sistema en función de los cambios que ocurren a lo largo del tiempo. Se considera que el sistema está en un estado particular hasta que un estímulo lo obliga a cambiar su estado. Ejemplos (tablas de decisión, diagramas de transición de estados, diagramas de persianas, redes de Petri, etc.)

#### Validación:

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

Comprenden verificaciones de validez (para todos los usuarios), de consistencia (sin contradicciones), de completitud (todos los requerimientos) de realismo (se pueden implementar) y verificabilidad (se puede diseñar conjunto de pruebas).

Algunas técnicas de validación pueden ser manuales o automatizadas, revisiones de requerimientos (formales o informales), construcción de prototipos y generación de casos de prueba.

### **TABLAS DE DECISION**

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

Las especificaciones pueden ser completas (aquellas que determinan acciones para todas las reglas posibles), redundantes y contradictorias (aquellas que marcan para reglas que determinan las mismas condiciones acciones iguales).

### **REQUERIMIENTOS**

Verificamos y validamos los requerimientos:

Correctitud. Consistencia. Completitud. Verificabilidad. Comprensibilidad. Adaptabilidad. Trazabilidad.

Gestión de los requerimientos:

*Cambios en los requerimientos:*

Estos cambian porque al analizar el problema, no se hacen las preguntas correctas a las personas correctas. Porque los clientes y los usuarios son distintos. Porque cambio el problema que se estaba resolviendo. Porque los usuarios cambiaron su forma de pensar o sus percepciones. Porque cambio el ambiente de negocios.

*Evolución:*

Requerimientos duraderos: relativamente estables, se derivan de la actividad principal de la organización.

Requerimientos volátiles: cambian durante el desarrollo del sistema o después de que se puso en operación. Ejemplos: Req. Cambiantes (cambian porque se modifica el ambiente, el entorno). Req. Emergentes (surgen como ampliación). Req. Consecuentes (surgen por la introducción del sistema. Pueden cambiar los procesos de la organización por desarrollar nuevas formas de trabajo). Req. De compatibilidad (cambian porque interactúan con otros sistemas que cambian).

*Pasos a realizar:*

1) Identificación de requerimientos. 2) Gestión del cambio (análisis del problema y especificación del cambio, análisis del cambio y cálculo del costo, implementación del cambio). 3) Políticas de rastreo (fuente, requerimientos, diseño). 4) Ayuda de herramientas CASE (almacenar requerimientos, gestionar el cambio, gestionar el rastreo).

### **DTE (DIAGRAMA DE TRANSICION DE ESTADOS)**



Es una máquina de estado finito, y estas describen al sistema como un conjunto de estados donde el sistema reacciona a ciertos eventos posibles (externos o internos).

Construcción de un DTE:

1) Identificar los estados. 2) Si hay un estado complejo se puede explotar. 3) Desde el estado inicial, se identifican los cambios de estados con flechas. 4) Se analizan las condiciones y las acciones para pasar de un estado a otro. 5) Se verifica la consistencia, que se hayan definido, se puedan alcanzar y se pueda salir de todos los estados.

### **REDES DE PETRI**

Fueron inventadas por Carl Petri en la universidad de Bonn, Alemania Occidental (1962). Utilizadas para especificar sistemas de tiempo real en las que son necesarios representar aspectos de concurrencia.

Los sistemas concurrentes se diseñan para permitir la ejecución simultánea de componentes de programación, llamadas tareas o procesos, en varios procesadores o intercalados en un solo procesador.

Las tareas concurrentes deben estar sincronizadas para permitir la comunicación entre ellas (pueden operar a distintas velocidades, deben prevenir la modificación de datos compartidos o condiciones de bloqueo). Pueden realizarse varias tareas en paralelo, pero son ejecutadas en un orden impredecible.

Los eventos se presentan como transiciones (T).

Los estados se presentan como lugares o sitios (P).

Los arcos indican, a través de una flecha, la relación entre sitios y transición y viceversa. A los sitios se les asignan tokens (fichas) que se representan mediante un número o puntos dentro del sitio y son ilimitados. Esta asignación de tokens a sitios constituye la marcación.

Luego de una marcación inicial se puede simular la ejecución de la red.

El conjunto de tokens asociado a cada estado sirve para manejar la coordinación de eventos y estados. Una vez que ocurre un evento, un token puede “viajar” de uno de los estados a otro.

Las reglas de disparo provocan que los tokens “viajen” de un lugar a otro cuando se cumplen las condiciones adecuadas.

La ejecución es controlada por el número y distribución de los tokens y se realiza disparando transiciones habilitadas.

Una transición está habilitada cuando cada lugar de entrada tiene al menos tantos tokens como arcos hacia la transición.

Disparar una transición habilitada implica remover tokens de los lugares de entrada y distribuir tokens en los lugares de salida (teniendo en cuenta la cantidad de arcos que llegan y la cantidad de arcos que salen de la transición).

La ocurrencia de los eventos (transiciones) depende del estado del sistema.

Una condición puede ser V (con token) o F (sin token)

La ocurrencia de un evento está sujeta a que se den ciertas condiciones (pre) y al ocurrir el evento causa que se hagan verdaderas las post-condiciones.

Las RP son asincrónicas y el orden en que ocurren los eventos es uno de los permitidos.

- La ejecución es NO DETERMINÍSTICA

Se acepta que el disparo de una transición es instantáneo.