

Especificación de Requerimientos

Parte 1

Sistemas de Información Administrativos
Departamento de Ingeniería Industrial
Universidad de Chile
Derechos Reservados (c)

Algunas definiciones

- El *IEEE Standard Glossary of software Engineering Terminology*, define un requerimiento como:
 - Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
 - Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente para satisfacer un contrato, estándar, especificación u otro documento impuesto formalmente.
 - Una representación documentada de una condición o capacidad como en el 1 y 2.

Algunas definiciones (2)

- **La ingeniería de requerimientos de sistemas** se encarga de descubrir, analizar y describir funciones y capacidades de los sistemas, requerimientos del negocio, organizativos y del usuario; protección, seguridad, ingeniería de factores humanos, interfaz, operaciones y requerimientos de mantenimiento; restricciones de diseño y requerimientos de calidad.
- **La ingeniería de requerimientos de software** se encarga del análisis y documentación de los requerimientos del software. Incluye la transformación de los requerimientos del sistema en una descripción de requerimientos de software, parámetros de desempeño y configuración del sistema, a través del uso iterativo de un proceso de definiciones, análisis, estudios de costo/beneficio y prototipos.

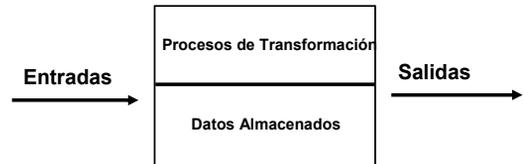
Requerimientos

- La identificación y especificación de los requerimientos de mecanización es uno de los productos más importantes de un proyecto de desarrollo de sistemas de información
- Se necesita en cualquier enfoque metodológico alternativo que se utilice
- Sin tener claridad en cuales son en detalle los requerimientos de mecanización (ej: una buena especificación de los requerimientos) no puede haber diseño físico y construcción del SIA y menos satisfacción de las necesidades del cliente

Requerimientos (2)

- Los requerimientos de mecanización se derivan del análisis del sistema, ya sea por el diseño lógico, construcción de prototipos, etc.
- En relación a la arquitectura de sistemas computacionales, la especificación de requerimientos corresponde a los resultados del diseño de la segunda fila (nivel lógico o del SIA) que pasan como input al diseño de la tercera fila (diseño físico o computacional)

Requerimientos (3)



- Además:
- Interfaces
 - Eventos (triggers)
 - Transacciones

Origen de cada requerimiento

(mirando SPD como caja negra)



Definición de Salida

● Tiene tres áreas:

- Especificación (de los parámetros): Nombre, volumen, nivel de complejidad, tiempo de respuesta, modo de obtención, generación.
- Distribución (donde se define como va a estar distribuida la información en el espacio disponible)
- Contenido (se identifican todos y cada uno de los datos que van a ir en la salida. Si hay mucha información, es importante definir su orden ("llaves")).

Definición de Salida (2)

- **Especificación (de los parámetros):**
 - Nombre, volumen, nivel de complejidad, tiempo de respuesta, modo de obtención, generación.
 - Tiempo de respuesta.
 - Tiempo real blando (segundos): existen variaciones aceptables que no causan daño. Ej. cobro cheque. Puede durar más o menos seg. El cliente puede esperar xx.
 - Tiempo real duro (milisegundos): importa timing exacto.
 - **Ej. vuelos aviones. errores pueden causar choques.**

Especificación de Salidas (3)

- **Modo de Obtención:**
 - **En línea: datos reales en ese momento**
 - **En línea diferido**
 - **Batch**
- **Generación:**
 - **Puede ser automática o a pedido. Lo mejor es que sea a pedido.**
- **Nodo:**
 - **Lugar donde se necesita el requerimiento**

Definición de la semántica de la interfaz del usuario

- Considere las tareas en abstracto
- Organice gruesamente las tareas en categorías
- Para cada categoría considere varias alternativas de flujo para las tareas
- Cree y describa tareas y objetos computacionales en el ámbito del sistema
- Haga referencias cruzadas al SPD
- Puede usar términos del Diccionario de Datos
- Evalúe el diseño semántico- gestión y feedback del usuario

Decidiendo por un método apropiado de presentación

- ¿Quién recibirá la información?
- ¿Bajo que circunstancias se recibirá la información?
- ¿Qué uso le darán los receptores de la información a los informes de salida o las consultas por pantalla??
- ¿Cuándo y con que frecuencia será necesaria la información de salida?

Tipos de medio de salida de información

- Impresa
- Pantalla (CRT, PDA, celulares, etc.)
- Audible
- Fax
- E-Mail
- Conexión a páginas web Web
- Formas de salida especializada (pulsos a sistemas físicos analógico-digital)

Especificación de requerimientos en acción

- Todo proyecto informático, tiene una etapa de "levantamiento de requerimientos".
- En esta etapa, los distintos actores del proyectos interactúan para acordar "qué se va a desarrollar".
- Desde el punto de vista de los constructores del SIA, se aclaran roles y responsabilidades.
- Desde el punto de vista del usuario, se aclara "qué quiere y cómo lo quiere"

En un proyecto informático: Definición de roles

- Gerente del proyecto. Es el encargado de las relaciones políticas con la alta gerencia. Es quién "le pone precio" al proyecto y negocia los atrasos.
- Jefe de proyecto. Ingeniero de software que sabe modelar un sistema, define plazos, cartas gantt, estándares. Idealmente, que sea un líder.
- Diseñador. Encargado del modelamiento del sistema
- Ingenieros de software. Son los encargados de desarrollar la solución en forma óptima, siempre buscando el camino corto y eficiente.
- Programadores. Son los que programan :)!!
- Hay más roles en un proyecto!!

¿Para que sirve?

- Ayuda al jefe de proyecto a especificar la función y el rendimiento del software, indicar la interfaz con otros elementos del sistema y establecer las restricciones que debe cumplir el software.
- Permite al gerente del proyecto "cubicar" el esfuerzo de desarrollo para luego saber cuanto cobrar.
- Permite al jefe de proyecto refinar la definición del software y construir los modelos de los dominios de datos, funcional y de comportamiento que van a ser tratados.

¿Para que sirve? (2)

- Proporciona modelos al diseñador del software que puede traducirse en el modelo de datos, arquitectónico, de interfaz y procedural.
- Proporciona al diseñador y al cliente los medios para valorar la calidad una vez que se ha construido el software.
- Permite solucionar dudas antes de comenzar el proyecto.

¿Para que sirve? (3)

- A veces el cliente no sabe lo que quiere, por lo tanto se orienta al especificar los requerimientos.
- Permite clarificar que trabajos están dentro del proyecto y cuales no.
- Si el cliente cambia las reglas del juego, se le muestra el documento de "levantamiento de requerimientos funcionales" y se negocia el esfuerzo adicional.
- Recuerde que "las cosas terminan mal cuando no se habla claro"

Elementos claves

- Reconocimiento del problema (tal como lo percibe el usuario/cliente)
- Evaluación y síntesis (describe el problema de forma que pueda sintetizarse un enfoque o solución global)
- Modelo (construcción de un prototipo)
- Especificación (se intenta proporcionar una representación del software que pueda ser revisada y aprobada por cliente)
- Revisión (casi siempre produce modificaciones en la función, en el rendimiento, en la representación de la información, en las restricciones o en los criterios de validación)

Obtención de requerimientos

- Es recomendable obtener requerimientos de varias fuentes individuales, ver si las necesidades son consistentes y factibles y validar si son reflexión precisa de las necesidades.
- Una especificación de requerimientos debe ser:
 - No ambigua
 - Completa
 - Verificable
 - Consistente
 - Modificable
 - Rastreadable
 - Usable
 - Necesario

Obtención de requerimientos (2)

- Al cliente le interesa lo bueno que traerá el sistema y no lo bueno que es el sistema.
- El cliente no tiene porqué saber de tecnicismos, pero si es experto en los procesos de negocio de su empresa.
- La especificación no apunta a cambiar la forma de operación del proceso de negocio, sino a entenderlo.
- Cuidado con el problema de API con un cliente!!!

Recopilando la información

- Entrevista
- Cuestionarios
- Muestreo e investigación de datos

La entrevista

- Lectura de antecedentes .- Consulte y comprenda el mayor número posible de antecedentes de los entrevistados y de su organización.
- Haga uso de la "cadena de contactos" es decir si conoces a alguien dentro de la organización o contacto alternativo, has preguntas sobre su visión.
- Establecimiento de los objetivos de la entrevista:
 - Las fuentes de información
 - La calidad de la información
 - Los formatos de información
 - El estilo de la toma de decisiones
 - La frecuencia de la toma de decisiones

La entrevista (2)

- Selección de los entrevistados.- Incluya a gente clave de todos los niveles del sistema, es importante muestrear a los miembros de la organización.
- Preparación del entrevistado.- preparar a la persona que entrevistará, avisándole con tiempo para que pueda pensar acerca de la entrevista.
- Selección del tipo y estructura de las preguntas.- Redactar las preguntas que cubran los aspectos fundamentales de la toma de decisiones, detectados al planear los objetivos de la entrevista.

Tipos de preguntas: abiertas

- Son las opciones que el entrevistado tiene para responder.
- Ejemplos
 - ¿Cuál es su opinión del sistema de cómputo actual?
 - ¿Cómo ve los objetivos de este departamento?
 - ¿Cuáles son algunos de los problemas que percibe respecto a recibir oportunamente la información?
 - ¿Cuáles son los errores más comunes en la captura de los datos en este departamento?
 - Describa el sistema más frustrante con el cual haya trabajado

Tipos de preguntas: abiertas (2)

- Ventajas:
 - Simplifican las cosas para el entrevistado
 - Permiten al entrevistador, seleccionar el vocabulario del entrevistado (educación, valores, creencias, etc.)
 - Proporciona una gran riqueza de detalle
 - Revela nuevas alternativas sobre preguntas no consideradas
 - Hacen más interesante la entrevista
 - Permite una mayor espontaneidad
 - Facilitan el estilo del entrevistador, sin interrumpirlo
 - Se utilizan como una alternativa, cuando el entrevistado no está preparado

Tipos de preguntas: abiertas (3)

- Desventajas de utilizar preguntas abiertas:
 - Permiten preguntas que pueden generar demasiada información irrelevante
 - La posible pérdida del control de la entrevista
 - Respuestas muy largas para poca información
 - Puede dar la apariencia de que el entrevistador no se preparó

Tipos de preguntas: cerradas

- Estructura básica ¿Cuántos usuarios tendrá el sistema?
- Las respuestas son limitadas y específicas.
- Ventajas:
 - Ahorrar tiempo
 - Facilitan la comparación entre entrevistas
 - Llegan al punto de interés
 - Mantienen el control de la entrevista
 - Cubren rápidamente diversos aspectos
 - Obtienen datos de relevancia

Tipos de preguntas: cerradas (2)

- Desventajas
 - Aburren al entrevistado
 - Pierden la riqueza del detalle
 - Se pueden perder ideas centrales por el punto anterior
 - No favorecen un clima de armonía entre el entrevistado y el entrevistador
- Ejemplos de preguntas cerradas
 - ¿Cuántos reportes generan al mes?
 - ¿Durante cuanto tiempo ha trabajado en la empresa?
 - ¿Quién recibe este reporte?
 - ¿Utiliza la computadora?
 - ¿Desea recibir mensualmente un reporte de su status contable?
 - ¿Desea un reporte igual a este?

Requerimientos funcionales

- Definen cómo trabajará cada parte del sistema.
- La idea es identificar las funciones o tareas de cada uno de los subsistemas y descomponerlos hasta identificar un sistema esencial.
- Es muy útil identificar los flujos de datos, sobre todo en los proyectos de desarrollo para el web.
- Ejemplo. "se requiere una interfaz que muestre al usuario la cartola de su cuenta corriente"

Requerimientos funcionales (2)

- Pregunta: "¿Cómo se valida un usuario?" R. Con su rut y una clave secreta.
- Entonces, en la interfaz descrita se encuentran las siguientes funcionalidades.
 - Validador del rut
 - Validador de clave secreta.
 - Consulta por cartola
 - Formateo de la respuesta
 - Presentación gráfica.

Requerimientos funcionales (3)

- La idea es llegar a un entendimiento que permita pensar en cuan complejo será implementar el código que soportará la funcionalidad.



Requerimientos no funcionales

- Son más bien características subjetivas que debe tener el sistema.
- Por ejemplo: Que sea rápido, que sea seguro, que sea bonito, que sea de buena calidad.
- Como no existe un "calidómetro" es decir un instrumento mágico que defina las expectativas subjetivas del cliente, es mejor, presentar prototipos que clarifiquen al cliente.

Prototipos

- Es mostrarle al cliente cómo quedará el sistema al estar operativo.
- Naturalmente, el cliente no entenderá de código.
- Muéstrole cómo serán las interfaces y reportes y su interacción dentro del sistema
- Ejemplo: El caso de los cajeros automáticos

Validando los modelos

- Básicamente nos encontraremos con:
 - Modelo de datos. Probablemente existirá una base de datos que habrá que modelar.
 - Diseño de la interfaz. Si fuera un sitio web, tendríamos que definir el aspecto del mapa, de navegación.
 - Flujos de datos. Hay que tener en claro como opera el negocio.
- En el primer modelo, se dice que es consistente si responde a todos los requerimientos funcionales planteados.

Validando los modelos (2)

- El segundo, se puede validar con un prototipo que muestre la interfaz "operando"
- El tercero, junto con los usuarios se les muestra cómo vemos el negocio y se solicita su apoyo para que esclarezca dudas. No dejes el modelo para retirarlo luego, siéntate con el usuario y empáptate del negocio.

Documento con el análisis funcional

- Finalmente, se genera un documento donde se especifican los requerimientos funcionales, los diagramas de flujos de datos, los prototipos de interfaces y reportes.
- Ese documento será parte del contrato.
- ¿Se cobra el documento? Si!!!, porque eres un profesional cuyo tiempo tiene valor, cuando vas al médico y no te encuentra nada malo, ¿te cobra?
- ¿Se entrega el modelo de datos? No, pues aun está muy primitivo, todavía no comenzamos el proyecto y el documento sólo define el sistema a groso modo, es decir, le dice al cliente lo que recibirá como mínimo.

¿Cuánto cobro?

- Con el documento podemos proyectar el esfuerzo en desarrollo y definir el equipo de trabajo que creará el sistema.
- Para calcular el esfuerzo en desarrollo, debemos estimar la cantidad de HH involucradas en desarrollar alguna de las funcionalidades descritas en el documento.
- Lo anterior se logra con mucha experiencia y conociendo muy bien las herramientas que intervienen en el desarrollo así como el equipo humano con que se trabajará.
- Lo veremos más adelante, así que muy atentos!! :)