NORMA IEEE 830 PARA ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

IEEE 830

- El estándar 830-1998 fue generado por un equipo de trabajo del IEEE (Instituto de Ingenieros Eléctricos y Electrónicos), su finalidad es la integración de los requerimientos del sistema desde la perspectiva del usuario, cliente y desarrollador.

- El propósito principal es ayudarnos a elaborar un documento muy útil: el SRS (Especificación de Requerimientos de Software).

- Es esencialmente una guía para redacción.

IEEE 830

- Quién la puede usar:
  - Un cliente/usuario que vaya a definir requerimientos (características) de un software que necesite
  - Un desarrollador (interno/externo) que haga software "a la medida" mediante proyecto
  - Un desarrollador que haga software "de paquete" que se venda masivamente

IEEE 830 SIRVE PARA:

- Un cliente describa claramente lo que quiere
- Un proveedor entienda claramente lo que el cliente quiere
- Se establezcan bases para un contrato de desarrollo (o de compra-venta)
- Se reduzca el esfuerzo de análisis, diseño, y programación (evitando re-trabajos)
- Se tenga una base o referencia para validar o probar el software solicitado
- Se facilite el traspaso del software a otros clientes/usuarios
- Se le puedan hacer mejoras (o innovaciones) a ese software

CONSIDERACIONES PARA REDACTAR EL SRS

- Su naturaleza
- Su ambiente
- Características deseables del documento
- Preparación conjunta del SRS
- Evolución del documento
- Prototipos
- Diseño "implícito" en el SRS
- Requerimientos de proyecto "implícitos"

NATURALEZA DEL SRS

- El SRS es una especificación para un producto de software en particular, ya sea un sólo programa, o un conjunto de programas, que realicen ciertas funciones en un ambiente específico
- A veces el usuario no sabe si necesitará un solo programa o más de uno
- El SRS puede escribirse por uno o más representantes del proveedor, uno o más del cliente, o por ambos
- Lo más recomendable es que haya representantes de ambas partes
- El usuario/cliente puede redactar un borrador inicial y después revisarlo con el proveedor

NATURALEZA DEL SRS

- Funcionalidades deseadas
- Interfaces externas
- Desempeño
- Atributos
  (seguridad, portabilidad, mantenibilidad, etc.)
- Restricciones de diseño impuestas a la implementación *(estándares técnicos propios o internacionales, lenguaje de progr., sistema operativo, límites de recursos, políticas internas)*.

AMBIENTE DEL SRS

- El SRS es la fuente principal para hacer el plan detallado de un proyecto de software
- Un SRS puede referirse a los requerimientos deseados de todos los componentes de un sistema grande, o a componentes (módulos) individuales del mismo
- Si se hacen SRS por separado para varios módulos, tiene que mantenerse la consistencia en los documentos
- Si un software necesita interactuar con otro, tienen que especificarse los requerimientos de esa interacción (interfaces), definiendo sus funcionalidades y el nivel de desempeño deseado

CARACTERÍSTICAS DE UN BUEN SRS

- Correcto
- No ambiguo
- Completo
- Consistente
- Ordenado con base en importancia y/o estabilidad
- Verificable
- Modificable
- Rastreable

CARACTERISITICAS DE UN BUEN SRS

CORRECTEZ

- El SRS es correcto si los requerimientos escritos son aquellos que el software deberá cumplir.
- No hay un método para saber si el SRS es correcto; lo importante es que se pida lo que realmente se necesita.

NO AMBIGUO

- Un SRS es *no* ambiguo si cada requerimiento establecido en él tiene una sola interpretación posible, tanto por el cliente/usuario como por el desarrollador
- En casos donde alguna palabra pudiera tener múltiples significados, se debe incluir su definición precisa en un glosario que se adicione al SRS.
  - Ejemplos:
    planta, obra, maestro, carga, flecha

CARACTERISTICAS DE UN BUEN SRS

COMPLETO

- El SRS es completo si incluye:
  - Todos los requerimientos significativos sobre funcionalidades, desempeño, restricciones de diseño, atributos, o interfaces externas.
  - Las respuestas que debería dar el software a todas las posibles entradas de datos en todas las situaciones posibles (entradas aceptables o no aceptables: validación).
  - Especificación de unidades de medida (si son aplicables).
  - En caso de que el SRS tenga diagramas o tablas informativas, hay que darles número o identificación.

CONSISTENTE

- Los diversos requerimientos escritos tienen que ser compatibles entre sí; no debe haber contradicciones ni conflictos entre ellos.
- Para lograr la consistencia deben evitarse los siguientes conflictos:
  - En características especificadas de objetos del mundo real.
  - De lógica o de tiempo entre dos actividades.
  - Referencia a un mismo objeto del mundo real pero usando diferentes palabras para el mismo objeto.

CARACTERISITICAS DE UN BUEN SRS

ORDENADO CON BASE DE IMPORTANCIA O ESTABILIDAD

- Cada requerimiento especificado debe tener alguna identificación (número, letra, secuencia alfanumérica) para indicar su grado de importancia o de estabilidad.
- Algunos requerimientos son más importantes que otros.
- Grado de estabilidad: número de cambios que se podrían esperar para un requerimiento, debido a eventos futuros que afecten a la organización, las responsabilidades, y las personas que usarán el software.
- Grado de necesidad:
  - Esencial
  - Condicional
  - Opcional

VERIFICABLE

- Un requerimiento es verificable si existe algún método rentable mediante el cual se pueda analizar si el software cumple ese requerimiento.
- Si no existe algún método para saber si el software cumple o no un requerimiento, entonces ese requerimiento debe ser revisado o eliminado.

EVOLUCIÓN DEL SRS

- Un SRS puede necesitar cambios mientras el software está en etapas de diseño o de desarrollo.
- Los cambios pueden estar motivados por: deficiencias, errores, omisiones o imprecisiones en el documento original.
- Cada requerimiento debe documentarse tan completo como sea posible, aún si pudiera necesitar cambios posteriormente.
- Los cambios en los requerimientos tienen que documentarse con el propósito de: identificarlos, controlarlos, rastrearlos, y reportarlos.
- Tanto el cliente como el proveedor deben designar a su respectivo responsable de autorizar (o rechazar) cambios en los requerimientos.

CREACIÓN DE PROTOTIPOS

- Un prototipo es un pequeño programa parecido al software solicitado que sirve de ejemplo, muestra o modelo para que el cliente vaya especificando sus requerimientos en forma progresiva junto con el desarrollador.

PROTOTIPOS

- El prototipo es útil para:
  - Que el cliente/usuario vea y describa más fácilmente las funcionalidades que desea.
  - Prever aspectos de la conducta del sistema, haciendo que el SRS sea más completo y preciso.
  - Reducir la cantidad de cambios durante las etapas de diseño o desarrollo.
- Un prototipo puede ayudar al usuario a definir detalles específicos de la interfaz humana o de las funcionalidades que requiera.
- Puede facilitarle al desarrollador el diseño y la programación del software.

DISEÑO IMPLÍCITO EN EL SRS

- Aunque el SRS *no* constituye un documento de diseño, implícitamente está diciéndole a los desarrolladores lo que se espera que ellos diseñen
  - Establece restricciones

- El SRS tiene que especificar las funcionalidades que se aplicarán sobre ciertos datos para producir resultados en cierto lugar para determinados usuarios

REQUERIMIENTOS DE PROYECTO IMPLÍCITOS

- El SRS se enfoca en el software como producto, no en su proceso de creación.
- Implícitamente establece restricciones sobre la planeación y administración del proyecto correspondiente.
- El SRS da origen a otros documentos (por separado) relacionados con el ciclo de vida de un software
  - Estimación de costos
  - Fechas de entrega
  - Reportes de avances
  - Métodos de desarrollo
  - Aseguramiento de la calidad
  - Criterios de validación y verificación
  - Procedimientos de aceptación

ORGANIZACIÓN DE LOS REQUERIMIENTOS ESPECÍFICOS

Para lograr un mejor entendimiento de los requerimientos, conviene organizarlos con base en alguno de los siguientes criterios:

- Por modo de operación del sistema
- Por clase de usuario
- Por objetos
- Por características
- Por estímulos
- Por respuestas
- Por jerarquía funcional