CC51A – Ingeniería de Software

Requisitos y Calidad

Sergio Ochoa D.

Estructura de la Presentación

- Introducción y principios básicos.
- La calidad.
- Calidad medible a través de los atributos.
- Requisitos de Calidad
- Atributos críticos.
- Especificación de atributos.
- Uso de Métricas.

Aspectos Relevantes en Calidad de Software

Cuando se habla de “calidad de software”, hay 2 aspectos que es necesario considerar:

- **Especificación de Requisitos:** Trata toda la problemática asociada a la especificación de requisitos completos, sin ambigüedades, y verificables.
- **Procesos de Supervisión:** Trata la problemática asociada al proceso de supervisión de procesos y productos.

En esta presentación nos abocaremos al primero de estos aspectos. El segundo espectro lo veremos en el capítulo 4 (SQA).

Ideas Relevantes.....

- La calidad es la clave del éxito en el negocio del software.
- Para mejorar la productividad del software construido, hay que mejorar su calidad.
- Para mejorar la calidad del producto, hay que mejorar la calidad del proceso.
  - La gestión es tan importante como la tecnología.

¿Qué es Calidad ?

La calidad:

“La totalidad de características y atributos de un producto o servicio, que están relacionados con satisfacer necesidades expresas o implícitas”.

Calidad = Satisfacción

La calidad depende de la expectativas...

¿Por Qué Necesito Calidad ?

La calidad:

- Es un asunto de competitividad
- Es esencial para sobrevivir
- Es esencial para exportar
- Es rentable
- Retiene clientes y aumenta las utilidades

Fuentes de Baja Calidad

- Requisitos imprecisos, mal entendidos o incompletos.
- Defectos en el software construido.
- Defectos en el soporte prestado.
- Falta de monitoreo del proceso y del producto.
- Etc.

Problemas de la Ing.de Soft.

El principal problema es: "*falta de claridad en los requisitos*"

Si queremos tener *éxito*, debemos definir éxito en función de:

- Costos
- Plazos
- Recursos
- Satisfacción de Requisitos
- Etc.

"Si los objetivos no están claros, no los alcanzaremos claramente".

Ejemplos de Requisitos Vagos....

- “La funcionalidad del nuevo sistema debe ser **mejor** que la del sistema anterior”.
- “El nuevo sistema debe tener una interfaz en **Visual Basic** que lo haga **más fácil** de usar, en particular sin las dificultades de uso del sistema actual”.
- “El sistema brindará **mejores reportes** de modo que se pueda aprovechar al máximo la Base de Datos”.
- “El sistema asegurará que los **datos** sean **correctos**”.

Requisitos Vagos....

Consecuencias:

- No podemos demostrar que hemos logrado los objetivos
- No podemos demostrar que no los hemos logrado
- No podemos evaluar alternativas de diseño
- Se termina especificando medios y no fines
- Si hay más de una forma de expresar un objetivo, tal vez no es un objetivo, sino un medio para lograr algo. Por ej. la interfaz en VisualBasic

Tipos de Requisitos

- **Requisitos de Usuario:** Expresan las necesidades del usuario.
- **Requisitos de Software:** Expresan las capacidades que debe tener el software, para poder satisfacer los requisitos del usuarios.

Por otra parte también están:

- Requisitos Funcionales
- Requisitos de Calidad
- Requisitos de Restricción

Tipos de Requisitos

- **Requisitos Funcionales:** indican ***¿Qué?***.
  Deben ser alcanzados si o sí.
- **Requisitos de Calidad:** indican ***¿Cuán bien?***.
  La calidad final dependerá del logro de estos objetivos
- **Requisitos de Restricción:** indican ***¿Con qué restricciones?***. En función de costos, tiempos, personal, etc.

- Los requisitos más difíciles de especificar son los req. de calidad.
- Los requisitos funcionales y los de restricción, suelen especificarse bien.

Tipos de Requisitos

- de Usuario
  - de Funcionalidad
  - de Restricción
- de Software
  - de Funcionalidad
  - de Calidad
  - de Restricción

Requisitos ->Atributos del Soft.

"todos los requisitos de calidad pueden y deben ser especificados sin ambigüedades"

"los requisitos determinan los atributos del software"

- Si un requisito está claramente expresado (por ej. terminar antes del 3/5/2001), se le da prioridad por sobre otros requisitos no tan claros ("más fácil", "mejor", "consistente", "amigable")
- Si queremos que los datos sean consistentes (por ej.), debemos indicar a qué le llamamos "consistente".
- Veamos el siguiente ejemplo...

Ejemplo de Especificac. de Req.

NOMBRE: Consistencia de Mediciones

ESCALA: Probabilidad de que dos mediciones básicas sean consistentes

PRUEBA: 1000 mediciones básicas comparadas usando 3 cifras significativas

ACTUAL: entre 90% y 95%

PEOR: 98%

PLANIFICADO: 99.5%

AUTORIDAD: minuta del 20/12/2004

Otro Ejemplo de Especificación de Requisitos

NOMBRE: Consistencia de Datos

ESCALA: Probabilidad de que se muestre un dato inconsistente, sin que se advierta su inconsistencia.

PRUEBA: Sobre una base de datos 100% consistente, se ingresan inconsistencias hasta hacerla 98% inconsistente. Luego se cuentan los datos inconsistentes no detectados.

ACTUAL: 100% (no hay detección).

PEOR: 5%.

PLANIFICADO: 1%.

AUTORIDAD: Entrevista con Usuario (05/12/2004).

Ejemplos de Requisitos de Calidad

- Estos atributos deberían ser revisados a la hora de especificar un sistema. Debido a que aparecen en la mayoría de ellos.

- Entre los *Requisitos de Calidad Típicos* figuran:

Funcionalidad:	Usabilidad:
Pertinencia	Entendibilidad
Precisión	Aprendibilidad
Interoperabilidad	Operabilidad
Adherencia	Aceptación de Uso
Seguridad	

Ejemplos de Requisitos de Calidad (cont...)

- Entre los *Requisitos de Calidad Típicos* figuran:

Mantenibilidad: Analizabilidad Cambiabilidad Estabilidad Demostrabilidad	Portabilidad: Adaptabilidad Instanciabilidad Adecuación Reemplazabilidad
Eficiencia: Rendimiento Uso de Recursos	Confiabilidad: Madurez Tolerancia a Fallas Recuperabilidad

Requisitos Usuales Importantes

- Los requisitos de calidad antes mostrados, casi siempre deben ser especificados.
- Esa lista de requisitos debería formar parte de la plantilla estándar.
  - Así se disminuye el riesgo de no especificar requisitos obvios.
  - Es más fácil acordarse de qué preguntar, o de negociar con el cliente en caso de ser necesario.
  - Los nuevos empleados pueden aprender de cómo hacemos las cosas aquí.

Requisitos Usuales Importantes (cont...)

- Funcionalidad
- Confiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Portabilidad

- Estos requisitos son conocidos como las "*ilidades*" de un software.

Funcionalidad

- Conjunto de requisitos referidos a la existencia las capacidades de un software.
- Las capacidades que satisfacen necesidades expresas o implícitas de los usuarios.
- Sus sub-requisitos son:

Pertinencia: la existencia de funciones apropiadas para las tareas especificadas.

Precisión: la capacidad de entregar resultados correctos o con un grado de error acotado.

Funcionalidad (cont…)

Interoperabilidad: la habilidad de interactuar con determinados sistemas (No confundir con reemplazabilidad).

Adherencia: la compatibilidad con estándares, convenciones o regulaciones.

Seguridad: la habilidad de prevenir uso no autorizado, tanto intencional como accidental, de programas y datos.

Confiabilidad

- La capacidad de mantener un nivel adecuado de servicio, bajo ciertas condiciones y por cierto tiempo.
- Sus sub-requisitos son:

Madurez: atributos relacionados con la frecuencia de fallas por errores del software, o bien con el porcentaje de tiempo que el sistema estará disponible.

Tolerancia a Fallas: habilidad de funcionar aún después de ciertas fallas.

Recuperabilidad: capacidad, tiempo y costo para reestablecer un nivel de servicio, y recuperar datos después de una fallas.

Usabilidad

- Atributos (o requisitos) relacionados con el esfuerzo de uso, y la evaluación del uso, realizada por los usuarios.
- Entre sus sub-requisitos están:

Entendibilidad: posibilidad de que los usuarios reconozcan los conceptos y su aplicabilidad

Aprendibilidad: esfuerzo necesario para adquirir un determinado nivel de destreza

Operabilidad: esfuerzo necesario para operar y controlar la operación del software

Agrado de uso: evaluación subjetiva (encuesta) hecha por los usuarios

Mantenibilidad

- Atributos (o requisitos) relacionados con el esfuerzo de hacer modificaciones.
- Entre sus sub-requisitos están:
  - **Analizabilidad:** esfuerzo de diagnosticar deficiencias o causas de fallas, o de identificar las partes a modificar
  - **Cambiabilidad:** esfuerzo de hacer un cambio
  - **Estabilidad:** riesgo de efectos inesperados por realizar un cambio
  - **Demostrabilidad:** esfuerzo de comprobar o validar la corrección

Portabilidad

- Habilidad de ser transferido de un ambiente a otro.
- Entre sus sub-requisitos están:

  Adaptabilidad: capacidad de adaptarse a otros ambientes usando los medios del software

  Instalabilidad: esfuerzo de instalación en un determinado ambiente

  Adecuación: adherencia a estándares o convenciones de portabilidad

Portabilidad (cont…)

Reemplazabilidad: posibilidad de ser usado en lugar de otro software, en el ambiente del otro.

NOTAS:

1- No debe confundirse con interoperabilidad (algunos llaman a ambos “compatibilidad”)

2- La reemplazabilidad está relacionada con la instalabilidad y la adaptabilidad. Debido a su importancia, se pone por separado.

Eficiencia

- Relación entre el nivel de rendimiento, y la cantidad de recursos usados, bajo ciertas condiciones.

  Uso del Tiempo: requisitos relacionados con tiempos de respuesta, tiempos de procesamiento, o tasas de servicio (throughput)

  Uso del Recursos: requisitos relacionados con el uso de recursos

  - Cantidad de recursos
  - Duración del uso

Requisitos Críticos

- Muchas veces hay requisitos tan importantes y conocidos, que por *obvios* nadie los dice.
- Principio de lo obvio:

  "Las cosas obvias, que todos saben, no se pueden dejar que se cuiden solas..."
- Un *requisito crítico* es aquel que si está fuera de control, amenaza la viabilidad de la solución.
  - El olvidarse de 1 sólo requisito crítico puede ser suficiente para un desastre.
- Un requisito crítico puede ser de *calidad*, *funcional*, o de *restricción*.

Soluciones, Medios y Fines

- Una ***solución*** es un conjunto de ideas que si se implementan, tienen un impacto positivo en alguna parte del problema
  - Las soluciones son los medios
  - Los requisitos son los fines.
- Las soluciones suelen tener impactos negativos
  - Una solución es interesante sólo si sus impactos positivos son mayores que los negativos
  - Para evaluar una solución necesitamos:
    - Requisitos claros.
    - Identificar los impactos positivos sobre los requisitos
    - Identificar los impactos negativos y efectos laterales
    - Uso de recursos

Soluciones y Requisitos

Requisito Funcional: es un requisito absoluto, con dos posibles estados: verdadero o falso.

Requisito de Calidad: es un requisito que se puede expresar en una escala de medición

Solución: es una idea que de ser implementada, debería impactar en forma positiva a los requisitos

La labor del “Administrador de Proyectos” es hacer todo lo posible, para que la *solución* encontrada satisfaga los *requisitos*.

Objetivos Escondidos

- Muchas especificaciones mencionan soluciones, pero esconden objetivos
  - "Programa escrito en C", tal vez significa "*bajo costo de portar el software*"
  - "*Arriendo de un computador central en menos de $x*", tal vez significa que "*se cuenta con $x mensuales, para ese ítem*".
  - "*Servidor de BD replicado*", tal vez significa "*que la probabilidad de que los datos estén inaccesibles, sea menor que z*".
  - "*Función automatizada*", tal vez significa "*rápida y con poco errores*", o tal vez significa "*bajo costo de operación*".

Principios Generales

- La especificación de requisitos es o debería ser la piedra angular de la administración
- Expresa los objetivos de un sistema
  - Se puede usar para describir sistemas existentes y sistemas futuros.
- Un atributo es un concepto de calidad que describe cualitativamente un sistema
- Una especificación de atributos es una lista de atributos, donde se especifica el nivel con el que se quiere alcanzar cada uno.

Especificación de Requisitos sirve para...

- Representar multiplicidad de vistas del sistema (desde el pto. de vista del usuario, del cliente, etc).
- “Registrar” la historia de las necesidades (cambios y refinamientos).
- “Registrar” las decisiones en cuanto a uso de recursos, prioridades, etc.
- “Registrar” objetivos (requisitos de calidad).
- “Registrar” restricciones (atributos de recursos).

¿Qué Requisitos Especificamos?

- Todos.... especialmente los críticos
- A veces “los atributos dependen mucho del problema”
- Normalmente debemos hacer una lista de atributos relevantes para cada proyecto
- Es importante revisar proyectos similares, que hayamos hecho, para chequear si nos falta algo importante.

Ventajas de Especificar Requisitos

La especificación de requisitos es engorrosa, pero tiene varias ventajas:

Para los Clientes:

Certeza: Tienen más probabilidad de obtener lo que quieren.

Comparación: Las ofertas de distintos proveedores son comparables.

Clarificación: Se promueve una discusión temprana de los temas.

Demostrabilidad: Será más fácil determinar si los requisitos fueron o no cumplidos.

Ventajas de Especificar Requisitos (cont…)

Para la Fuerza de Ventas:

Certeza: Menor probabilidad de perder una propuesta por no entender los requisitos

- Sabremos antes que la competencia qué es lo que se necesita

Imagen: Se nos percibirá como realmente interesados en saber qué es lo que se necesita

Manejo del Riesgo: Sabemos qué? y cuánto? se necesita. Se detectan riesgo en forma temprana.

Control de Cambios: Podemos demostrar que nuevas solicitudes, deben ser tratadas como *cambios*

Claridad: Podemos hablar sin ambigüedades y sin doble sentido, sobre:

- Las fortalezas de nuestra propuesta
- Los resultados esperados, en lugar de promesas vagas.

Ventajas de Especificar Requisitos (cont…)

Para Desarrolladores y Subcontratistas:

- Podemos basar las estimaciones en un entendimiento firme de los requisitos.
- El cliente no nos puede sorprender con requisitos más exigentes, sin estar dispuesto a pagar/negociar la diferencia.
- Podemos diseñar por objetivos, lo cual tiene más posibilidades de éxito, que hacer algo y ver si sirve.
- Puedo ejercer un “control de cambios”.
  - Nosotros debemos tener la sartén por el mango, no el cliente.

Conclusiones

- “Especificar la calidad” no significa que mi software será de “buena calidad”.
- “Especificar la calidad” significa que conozco lo que debo entregar al cliente.
- “Especificar la calidad” significa que el cliente sabe lo que debo entregar.
- “Especificar la calidad” significa que conozco mis limitaciones, por lo tanto puedo trabajar para superarlas.
- “Especificar la calidad” me ayuda a presentar presupuestos realistas, y a no perder dinero.

Conclusiones

- No existe la posibilidad de desarrollar o comprar “un software de calidad”.
- Siempre hay que especificar la calidad…. pero que eso no sea el centro de mi proyecto.
- No caer en una “parálisis por análisis”.
- Revisar los requisitos usuales, y preguntar por ellos !!!, siempre antes de entregar un presupuesto.
- No se permitan ambigüedades… el cliente puede ser más vivo que ustedes…