



Universidad de Chile
Departamento de Ciencias de la Computación



Especificación de Requisitos de Calidad y Otros Requisitos

Requisitos de calidad o atributos

Taller: Jerarquías de atributos

Requisitos de Calidad Importantes

Requisitos de calidad o atributos

- ❑ Todos los requisitos de calidad se pueden y deben expresar sin ambigüedades.

- ❑ Si un requisito se expresa claramente
 - ❑ “terminar antes del 31/3/1998”se le da prioridad por sobre requisitos no tan claros
 - ❑ “más fácil de usar”
 - ❑ “información consistente”
 - ❑ “user friendly”
 - ❑ “mejor que el sistema anterior”



Ejemplo de requisito de calidad cuantificado

- Si queremos que los datos sean consistentes podemos especificar el grado de consistencia aceptable
- Ejemplo

ESCALA	Fracción de mediciones básicas consistentes
PRUEBA	Tomar 1000 mediciones y comparar a ± 50 metros
ACTUAL	Entre 90% y 95%
PEOR	98%
PLAN	99.5%
AUTORIDAD	Minuta del 25/8/95



Otro requisito de calidad cuantificado

- Si queremos que el sistema detecte inconsistencias, debemos indicar la calidad de la detección

- Ejemplo

ESCALA Probabilidad que se muestre un dato inconsistente sin advertencia de inconsistencia

PRUEBA Sobre una base de datos artificial 98% consistente se cuentan los datos inconsistentes sin advertencia

ACTUAL 1,00 (no hay detección)

PEOR 0,05

PLAN 0,01

AUTORIDAD Ninguna (sugerencia no autorizada)



Cómo se describe un requisito de calidad

- ❑ Escala Qué se mide (una dimensión)
- ❑ Prueba Cómo se mide (un procedimiento)
- ❑ Peor Valor apenas aceptable
- ❑ Plan Valor que deseamos

- ❑ Autoridad Quién, qué o de dónde sale esto
- ❑ Actual Valor alcanzado ahora en el sistema
- ❑ Mejor Valor mejor alcanzado por alguien en alguna parte (también llamado récord)



La escala y la prueba

- ❑ La **escala** es la medición que se hará de un requisito de calidad
 - ❑ La escala es una medición teórica (ej., kilogramos)
- ❑ La medición práctica asociada se llama **prueba** (ej., uso de una pesa)

- ❑ La escala dice relación con qué se mide
- ❑ La prueba dice relación con cómo se mide
 - ❑ Más vale una prueba mediocre por escrito que la prueba perfecta que no sabemos describir



Escala y prueba

❑ Facilidad de aprendizaje

ESCALA = minutos para aprender

PRUEBA = se hará una demo de 5 minutos y se contestarán dudas, midiendo el tiempo hasta que la persona diga que aprendió

❑ Productividad

ESCALA = tareas por hora

PRUEBA = el sistema promediará el número de transacciones procesadas por persona por hora en la segunda semana de uso

❑ Ventajas de especificar la prueba

- ❑ Se reconoce el costo de la medición (para el presupuesto y el plan)
- ❑ Todos sabrán que el objetivo se puede y se va a medir
- ❑ Si no se dice lo contrario, la prueba es aplicable durante las pruebas de aceptación



El peor nivel

- ❑ El **peor** nivel es el peor nivel aceptable bajo cualquier circunstancia
 - ❑ Cualquier valor menor significa que hubo fracaso
 - ❑ Para tener éxito no podemos tener todos los requisitos de calidad en este nivel
 - ❑ La idea es llegar al nivel planificado
- ❑ El peor nivel de cada requisito de calidad debe ponerse en un contrato legal
 - ❑ Si no se cumple el proveedor está en deuda
 - ❑ Se podrían incluso definir multas
 - ❑ Todas las partes interesadas deben firmar

El peor nivel puede usarse como un punto de control del proyecto si se usa Control Estadístico de Procesos (SPC)

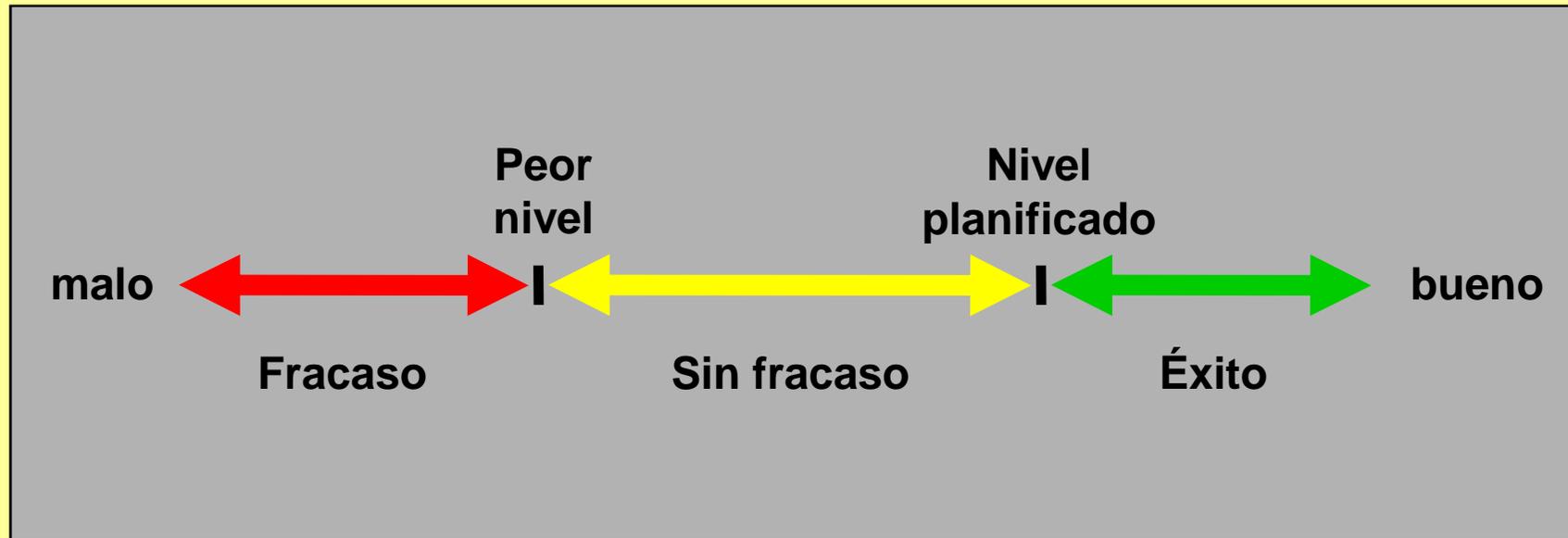


El nivel planificado

- El nivel ***planificado*** es aquel nivel que se desea alcanzar, en conjunto con los demás requisitos de calidad
 - Este es el límite de la definición de éxito
 - Es el nivel en donde si se alcanza ya no ponemos más recursos
 - El nivel planificado no significa perfección: es un nivel alcanzable y realista
 - Debe ser consistente con los niveles planificados de los demás requisitos de calidad



Los límites de la especificación de atributos



Niveles Actual y Mejor

- El nivel **actual** es desde donde partimos
- El nivel **mejor** es el nivel más alto que alguna vez se ha medido

- Usualmente el nivel Planificado es mejor que el Actual
 - Excepción: un atributo demasiado bien logrado actualmente
- Usualmente el nivel Planificado es peor que el Mejor nivel
 - Excepción: si se quiere hacer avanzar el estado del arte

La autoridad

- ❑ La autoridad es una indicación que sustenta el requisito
 - ❑ Puede ser una orden de un jefe
 - ❑ Puede ser un manual de procedimientos o estándares
 - ❑ Puede ser un requisito ya aprobado como válido
 - ❑ Puede ser un contrato
 - ❑ Incluso puede ser un tincada
- ❑ La autoridad es opcional y puede haber varias
- ❑ Ejemplos
 - AUTORIDAD = análisis de Jorge (¡debe ser revisado!)
 - AUTORIDAD = anexo A del contrato, sección A.3.b



Jerarquía de requisitos de calidad

- ❑ Suele ser conveniente expresar los requisitos de calidad como una jerarquía de requisitos y sub-requisitos de calidad

- ❑ Rendimiento del sistema
 - ❑ capacidad de procesamiento semanal
 - ❑ disponibilidad del sistema
 - ❑ tiempo de respuesta
 - ❑ confiabilidad
 - ❑ utilización de la CPU del servidor
 - ❑ largo medio de la cola del scheduler



Otro ejemplo de jerarquía: Usabilidad

- La usabilidad es difícil de medir directamente porque tiene muchas facetas:
 - Nivel de habilidades prerequisites para un curso del sistema
 - Tiempo de entrenamiento necesario para un cierto nivel de uso
 - Cantidad de trabajo producido por una persona así entrenada
 - Tasa de errores cometidos por un usuario entrenado, trabajando a velocidad normal
 - Puntaje en una encuesta con las opiniones subjetivas de los usuarios en relación al agrado y facilidad de uso



Cómo hacer la jerarquía de atributos

□ Por dónde partimos

- de arriba a abajo (*top-down*)
- de abajo hacia arriba (*bottom-up*)
- del medio hacia arriba y hacia abajo

□ De rendimiento del sistema hacia arriba

- ¿por qué? Mejor servicio al cliente
- ¿por qué? Mantener la clientela
- ¿por qué? Proteger nuestra base de ingresos
- . . .



Análisis de requisitos

- ❑ A veces al subir en la jerarquía preguntando por qué no hay respuestas claras, lo que sugiere que tal vez el supuesto requisito no es tal
- ❑ Cuando sí hay respuestas claras, podemos bajar de los objetivos de alto nivel a otros objetivos de bajo nivel
 - ❑ ¿de qué otro modo podemos mejorar el servicio al cliente?
 - ❑ ¿mantener la clientela?
 - ❑ ¿proteger nuestra base de ingresos?
- ❑ Esto es análisis de requisitos de verdad, no sólo enumeración de requisitos



Taller de Requisitos de Calidad

Jerarquía de Atributos Especificación de Atributos

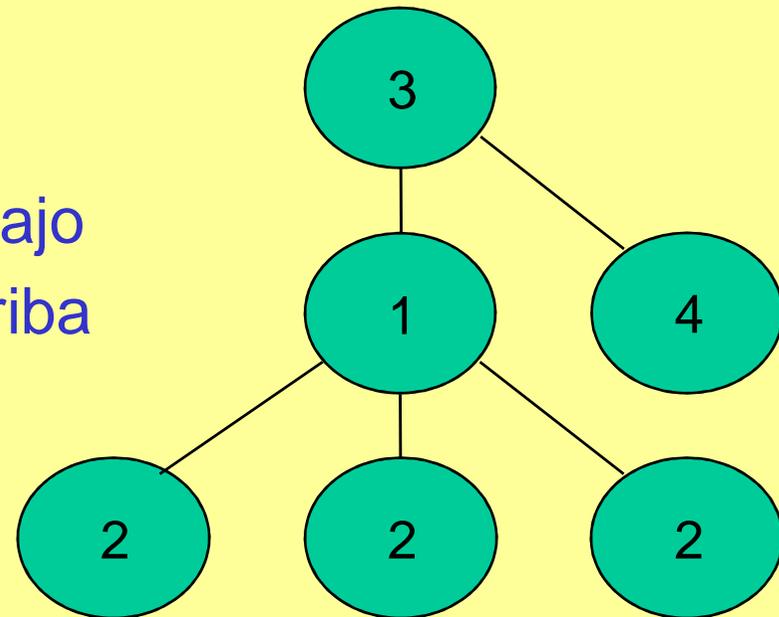
Taller Parte 1: Identificar proyecto

- ❑ Organizarse en grupos de tres o cuatro personas
- ❑ Elijan un proyecto o sistema real en el que haya participado alguno de los miembros del equipo
- ❑ Quien conoce el proyecto les explica a los demás de qué se trata y cuáles eran los requisitos de calidad críticos para el éxito



Taller Parte 2: Expandir una jerarquía de atributos

- 1 Elegir un requisito de calidad entre
 - Mantenibilidad
 - Portabilidad
 - Facilidad de probar
- 2 Expandir la jerarquía hacia abajo
- 3 Expandir la jerarquía hacia arriba preguntando ¿por qué?
- 4 Identificar modos alternativos de obtener lo identificado en 3



Taller Parte 3: Especificación de atributos

- Especifique en detalle un atributo identificado en el punto 2 del taller 2, indicando:
 - Escala
 - Prueba
 - Valor Planificado
 - Valor Peor

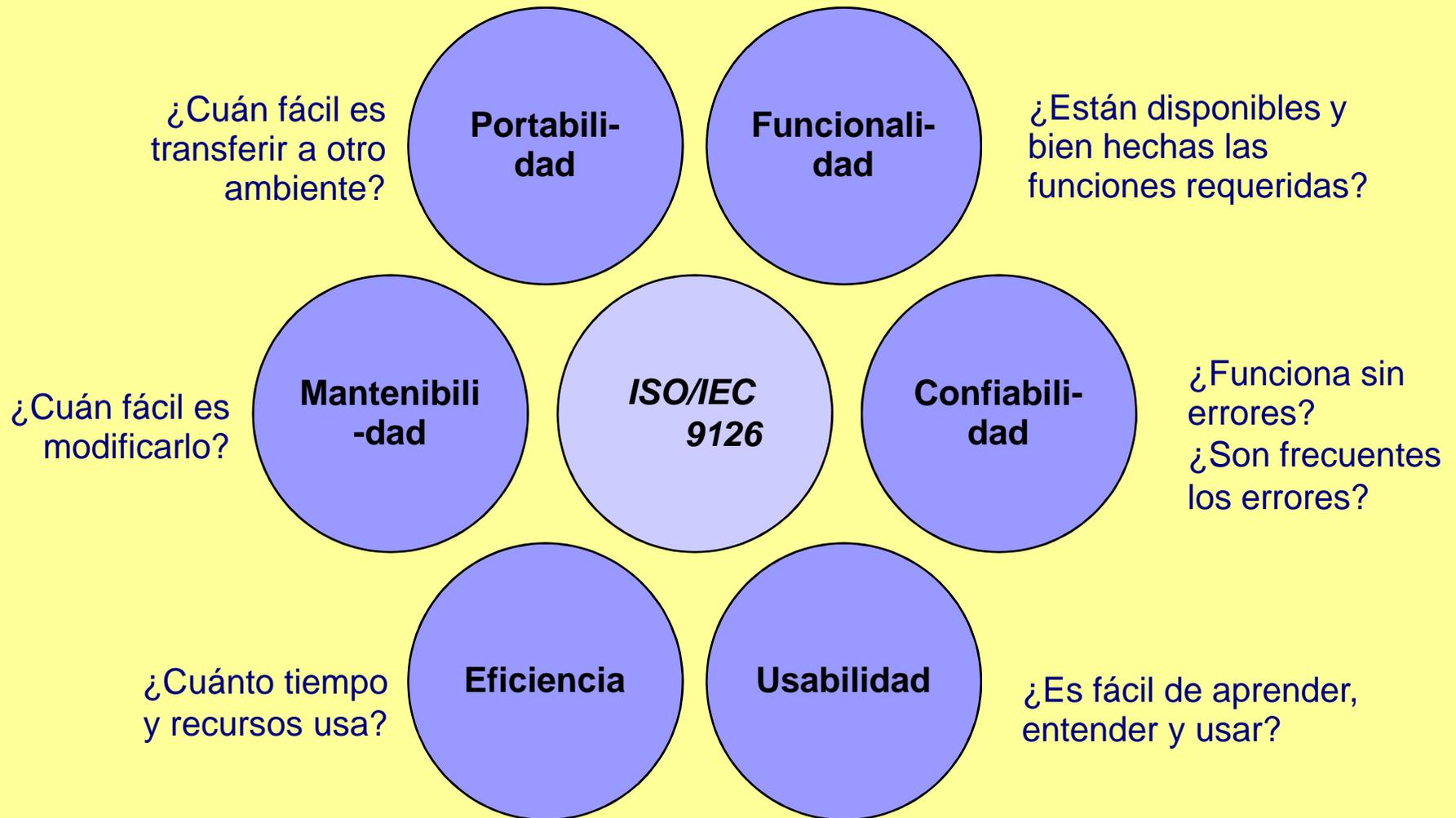




Requisitos de Calidad Importantes

Norma ISO/IEC 9126 Parte 1

Clases de atributos según ISO/IEC 9126



Requisitos usuales importantes

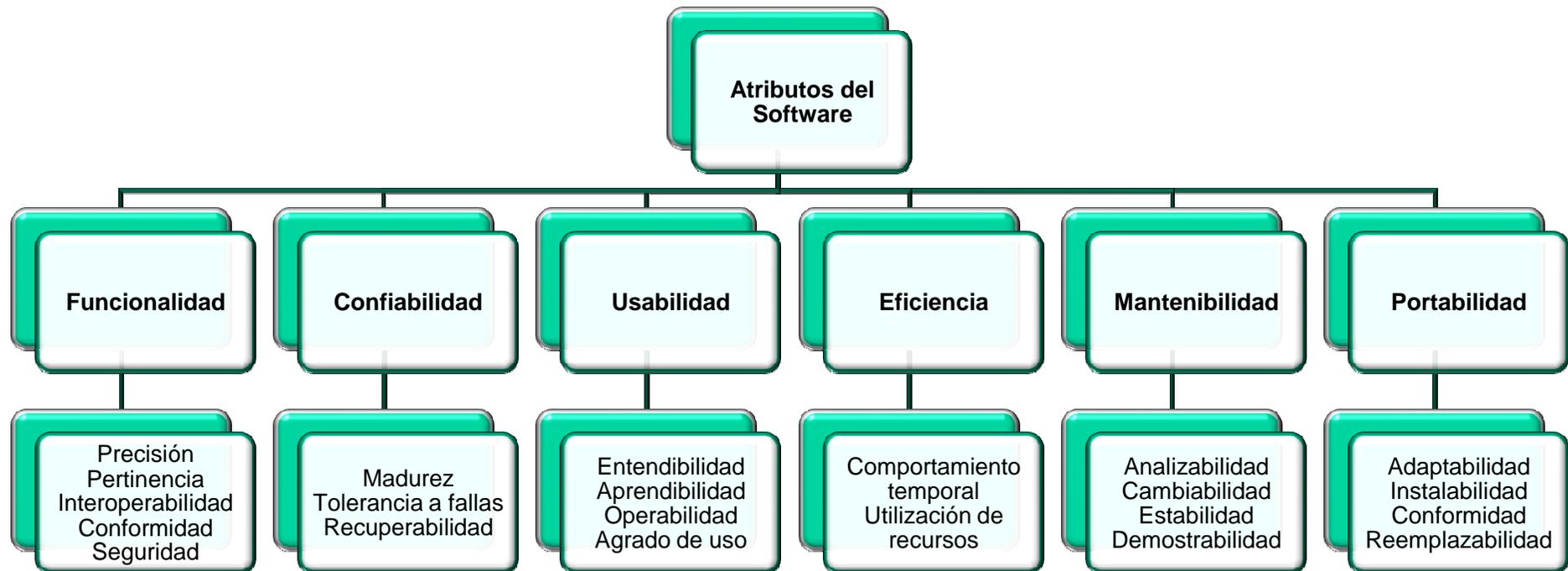
- 1 [Calidad de] Funcionalidad
- 2 Confiabilidad
- 3 Usabilidad
- 4 Eficiencia
- 5 Mantenibilidad
- 6 Portabilidad

En conjunto la jerga las llama las ***ilidades***

Vamos a inventar palabras terminadas en **ilidad**



ISO/IEC 9126-1



¡Advertencia!

- No confundir *Requisitos de Calidad Asociados a la Funcionalidad* con **Requisitos Funcionales**
- Los primeros se refieren a cuán bien se logran los requisitos funcionales, los segundos a qué hace el sistema



1 Funcionalidad

- Conjunto de atributos sobre la existencia de un conjunto de funciones y sus propiedades. Las funciones satisfacen necesidades expresas o implícitas de los usuarios.
 - *Pertinencia*: la existencia de funciones apropiadas para las tareas especificadas.
 - *Precisión*: la capacidad de entregar resultados correctos o con un grado de incorrección acotado.
 - *Interoperabilidad*: la habilidad de interactuar con determinados sistemas (no confundir con reemplazabilidad)
 - *Conformidad*: la compatibilidad con estándares, convenciones o regulaciones.
 - *Seguridad*: la habilidad de prevenir uso no autorizado, tanto intencional como accidental, de programas y datos.



2 Confiabilidad

- La capacidad de mantener un nivel adecuado de rendimiento bajo ciertas condiciones y por cierto tiempo
 - *Madurez*: atributos relacionados con la frecuencia de fallas por errores de software
 - *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 falla



3 Usabilidad

- ❑ Atributos relacionados con el esfuerzo de uso y la evaluación sobre el uso hecha por los usuarios
 - ❑ *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



Pseudo-atributos de usabilidad

- ❑ Amistosidad con el usuario (totalmente indefinido)
- ❑ Manipulación directa (es más bien un estilo de solución)
- ❑ Interfaz intuitiva (casi nada en los computadores es de hecho intuitivo)

- ❑ ¿Se nos ocurre otro?



4 Eficiencia

- Relación entre el nivel de rendimiento y la cantidad de recursos usados, bajo ciertas condiciones
 - *Uso de tiempo*: atributos relacionados con tiempos de respuesta, tiempos de proceso, o tasas de servicio (throughput)
 - *Uso de recursos*: atributos relacionados con el uso de recursos
 - Cantidad de recursos
 - Duración del uso



5 Mantenibilidad

- Atributos relacionados con el esfuerzo de hacer modificaciones especificadas
 - *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 un cambio
 - *Demostrabilidad*: esfuerzo de comprobar o validar la corrección



6 Portabilidad

- Habilidad de ser transferido de un ambiente a otro
 - *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
 - *Reemplazabilidad*: posibilidad de ser usado en lugar de otro software en el ambiente del otro (no confundir con interoperabilidad)



Críticas a ISO/IEC 9126

- ❑ No es claro si el modelo es completo
 - ❑ No incluye reusabilidad
- ❑ Mantenibilidad abarca mejoras y corrección de defectos, sin diferenciación
- ❑ No es claro si algunas subcaracterísticas deben ser características
- ❑ Interoperabilidad v/s portabilidad puede ser confuso
- ❑ Recomendación:
 - ❑ Usar la norma como posibles buenas ideas
 - ❑ No es verdad revelada



Referencias

- Tom Gilb. *Principles of Software Engineering Management*, Addison-Wesley, 1988.
- ISO/IEC 9126, Parte 1.



Anexo: Caso práctico

M-DTV™ Module Simulator



Caso: M-DTV™ Module Simulator

- ❑ Contexto: Un nuevo módulo para diseñar televisores digitales modificando levemente el diseño de un televisor análogo
 - ❑ El módulo recibe órdenes del microcontrolador actual del TV
 - ❑ No tiene iniciativa por sí solo
 - ❑ Se comunica con el microcontrolador vía puerta serial
- ❑ Producto: M-DTV™ Module MCT5100
- ❑ Problema: El software del TV actual debe modificarse y probarse antes de conectar el módulo
- ❑ Solución
 - ❑ Un simulador del módulo para probar el software del TV
 - ❑ Comportamiento del módulo descrito en un archivo script



Atributos relevantes del simulador (en inglés)

- ❑ Usability
 - ❑ Ease of learning
 - ❑ Ease of operation
- ❑ Time accuracy
- ❑ Maximum size of the script file
- ❑ Completeness of initial delivery
- ❑ Cost
 - ❑ Calendar time
 - ❑ Resources (people)



Ease of learning

- Scale Time in hours required to learn the use of the application
- Test A programmer of TV software will be taught how to use the application and then will use it in specific tasks
- Worst 24
- Plan 8
- Authority none



Ease of operation

- Scale Time in hours required to operate the application once learned
- Test The trained representative user will be given the task to create a specific 2-state, 10-transaction script and test its operation in automatic mode
- Worst 24
- Plan 8
- Authority none



Time accuracy

- Scale Difference in timing in milliseconds between the specified response time and actual response time.
- Test By the second system delivery the testing computer will have a software that sends a command and measures the time it takes for the response to arrive.
- Worst 5
- Plan 1
- Authority Time will be measured in milliseconds, because maximum response time is specified to be 100ms in Host-MDTV Express Module Interaction.



Completeness of initial delivery

- Scale A measure of the functionality ready for delivery after the first coding phase
- Test Apply these definitions:
 - **Complete:** All functional requirements are implemented. Errors in scripts are detected.
 - **Basic:** The executable includes both modes of operation, parsing may impose formatting constraints; the following are optional: detect errors in scripts, macros, explanations, comments, saving/hiding the log.
 - **Operational:** In addition to the constraints for basic functionality, the following are optional: manual execution mode, timers, and log.
 - **Non-operational:** The executable does not run.
- Worst operational
- Plan basic
- Authority As agreed by the software coordinators on the meeting held September 1st, the deadline cannot be moved but the level of functionality is flexible.



Resources

- Scale Working man-days of development for first version of the system
- Test Project management tracking
- Worst 7 for initial delivery
10 for full functionality
- Plan 6 for initial delivery
8 for full functionality
- Authority One developer will be assigned full time, this developer is needed for other applications as well.

