Calidad en el Desarrollo de Software

Modelos de calidad de software

Pablo R. Fillottrani

Depto. Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

Segundo Cuatrimestre 2007

Medición de calidad de software

- la calidad, al igual que la belleza, está en el ojo de quien lo mira
- sin embargo, desde el punto de vista de medición, se debe tener una definición precisa en términos de atributos del software que sean dññe interés al usuario
- en general, éstos son atributos externos
- sin embargo, muchas propuestas miden y analizan atributos internos porque los consideran predictores de aquellos externos
- los atributos internos tienen dos ventajas:
  - están disponibles para medición más temprano
  - son más fáciles de medir

Modelos iniciales de calidad

- desde el principio de la ingeniería de software, se observó que la calidad está compuesta por una composición de muchas características
- un modelo de calidad describe entonces estas características y sus relaciones
- muchos modelos hacen difusa la distinción entre atributos internos y externos, lo que dificulta la comprensión del concepto de calidad
- los modelos que se presentarán a continuación son los que han ganado mayor popularidad en la comunidad, pero no tienen sustento científico
- extrayendo los factores comunes a todos ellos, es posible derivar modelos propios adaptados a usos específicos

Modelo de McCall

- el modelo de McCall fue el primero en ser presentado en 1977, y se originó motivado por US Air Force y DoD
- se focaliza en el producto final, identificando atributos claves desde el punto de vista del usuario
- estos atributos se denominan factores de calidad y son normalmentes atributos externos
- pero también se incluyen algunos atributos posiblementes internos

Modelo de McCall

- los factores de calidad son demasiados abstractos para ser medidos directamente, por lo que por cada uno de ellos se introduce atributos de bajo nivel denominados criterios de calidad
- algunos criterios de calidad son atributos internos, reflejando la creencia de McCall que el atributo interno tiene un efecto directo en el atributo externo correspondiente
- un nivel más de descomposición es necesario, mapeando cada criterio de calidad en un conjunto de métricas de calidad que son atributos (tanto del producto como del proceso) de muy bajo nivel, medibles directamente

Factores de calidad

- McCall propone tres perspectivas para agrupar los factores de calidad
  - revisión del producto habilidad para ser cambiado
  - transición del producto adaptabilidad al nuevo ambiente
  - operación del producto características de operación

Factores de calidad de revisión

- la revisión del producto incluye los siguientes factores de calidad:
  - mantenibilidad esfuerzo requerido para localizar y corregir fallas
  - flexibilidad facilidad de realizar cambios
  - testeabilidad facilidad para realizar el testing, para asegurarse que el producto no tiene errores y cumple con la especificación

Factores de calidad de transición

- la transición del producto incluye los siguientes factores de calidad:
  - portabilidad esfuerzo requerido para transferir entre distintos ambientes de operación
  - reusabilidad facilidad de reusar el software en diferentes contextos
  - interoperabilidad esfuerzo requerido para acoplar el producto con otros sistemas

Factores de calidad de operación

- la operación del producto incluye los siguientes factores de calidad:
  - correctitud el grado en el que el producto cumple con su especificación
  - confiabilidad la habilidad del producto de responder ante situaciones no esperadas
  - eficiencia el uso de los recursos tales como tiempo de ejecución y memoria de ejecución
  - integridad protección del programa y sus datos de accesos no autorizados
  - usabilidad facilidad de operación del producto por parte de los usuarios

Criterios de calidad: factor mantenibilidad

- según McCall el factor mantenibilidad incluye los siguientes criterios:
  - consistencia
  - simplicidad
  - concisidad
  - auto-descripción
  - modularidad
- pero la mantenibilidad ha cambiado bastante desde 1977; encontrar y corregir errores es sólo un aspecto más

Criterios de calidad: factor mantenibilidad

- Ghezzi la divide en tres categorías:
  - correctiva concerniente a remover pequeñas fallas remanentes después del testeo
  - adaptativa concerniente al cambio del producto necesario por el cambio de sus requerimientos
  - perfectiva busca solo mejorar los algoritmos usados para hacerlos más eficientes

Criterios de calidad: factor mantenibilidad

- mantenibilidad está muy influenciado por el uso de buenas prácticas a lo largo de todo el ciclo de desarrollo
- algunas de estas buenas prácticas son:
  - seguir una metodología bien definida
  - usar buenas técnicas de diseño, tanto de procedimientos como de datos, para aumentar cohesión y reducir acoplamiento
  - observar la documentación interna
  - usar buenas prácticas de programación: nombres significativos, código legible, etc

Criterios de calidad: factor flexibilidad

- según McCall el factor flexibilidad incluye los siguientes criterios:
  - expandibilidad
  - generalidad
  - auto-descripción
  - modularidad

Criterios de calidad: factor flexibilidad

- con el correr de los años este criterio se ha fusionado con mantenibilidad
- de hecho, en la definición original, dos de los criterios de flexibilidad estaban compartidos con mantenibilidad

Criterios de calidad: factor testeabilidad

- según McCall el factor testeabilidad incluye los siguientes criterios:
  - simplicidad
  - instrumentación
- dado su ubicación en tradicionales modelos de ciclo de vida de software, la facilidad de testing se define claramente como un criterio de calidad

Criterios de calidad: factor testeabilidad

- ISO 9000-3 divide el testeo en cuatro etapas
  - testeo de unidad se testea los componentes individuales, generalmente realizado por los programadores
  - testeo de integración se testean los módulos compuestos por diversos componentes
  - testeo de sistema se testea el sistema completo tal como lo usaría un usuario normal, pero sin su presencia
  - testeo de aceptación el usuario ejecuta el sistema completo para asegurarse que cumpla con los requerimientos. También llamado *alpha testing*

Criterios de calidad: factor testeabilidad

- el testeo interactua con otros criterios de calidad, por ejemplo correctitud y eficiencia
- debe ser llevado a cabo siguiendo planes pre-definidos, con datos conocidos y cuyos resultados sean predeterminados
- la testeabilidad puede ser maximizada usando herramientas automáticas, buenas estrategias de cohesión y de diseño, y buenas prácticas de programación
- McCall definió originalmente métricas para testeabilidad consistentes en una matriz de com plejidad que involucra número y tamaño de módulos, tamaño de procedimientos, profundidad de anidamiento, número de errores por unidad de tiempo, etc.

Criterios de calidad: factor portabilidad

- según McCall el factor testeabilidad incluye los siguientes criterios:
  - auto-descripción
  - modularidad
  - independencia de la máquina
  - independencia del sistema operativo

Criterios de calidad: factor portabilidad

- algunos autores (Sommerville) lo consideran parte de la reusabilidad
- se favorece mediante el seguimiento de estándares, tanto de procedimientos (X Windows) como de datos (XML)
- la existencia de compiladores cruzados favorece la portabilidad

Criterios de calidad: factor reusabilidad

- según McCall el factor reusabilidad incluye los siguientes criterios:
  - generalidad
  - modularidad
  - auto-descripción
  - independencia de la máquina
  - independencia del sistema operativo

Criterios de calidad: factor reusabilidad

- se puede favorecer la reusabilidad usando librerías de software, y técnicas de programación orientada a objetos
- hay que tener en cuenta que el desarrollo de código reusable cuesta más tiempo y dinero
- existe un factor económico difícil de medir: el costo de código reusable y la ganancia por reusar código ya desarrollado

Criterios de calidad: factor interoperabilidad

- según McCall el factor interoperabilidad incluye los siguientes criterios:
  - modularidad
  - interoperabilidad en comunicación
  - interoperabilidad en datos

Criterios de calidad: factor interoperabilidad

- la interoperabilidad está relacionada con la reusabilidad
- en la actualidad su importancia ha crecido debido al creciente interés de conectarse con sistemas *legacy*
- se favorece mediante la adopción de estándares

Criterios de calidad: factor correctitud

- según McCall el factor correctitud incluye los siguientes criterios:
  - trazabilidad
  - completitud
  - consistencia

Criterios de calidad: factor correctitud

- correctitud es un factor muy difícil de identificar debido a la falta de terminología estándar
- se lo pueden confundir con otros factores, tales como confiabilidad e integridad
- para medirlo es necesario tener disponible una especificación formal de los requerimientos, cosa muy rara salvo en proyecto de alto presupuesto y sistemas críticos
- las técnicas para verificarlo pueden ser: inspecciones de código, verificación matemática y analizadores estáticos de programas

Criterios de calidad: factor confiabilidad

- según McCall el factor confiabilidad incluye los siguientes criterios:
  - tolerancia a errores
  - consistencia
  - simplicidad
  - exactitud

Criterios de calidad: factor confiabilidad

- combina la tolerancia tanto a errores de hardware como de software
- técnica de programación tales como tolerancia a las fallas, manejo de excepciones y programación defensiva ayudan
- puede ser medido con medidas como
  - tiempo medio entre fallas
  - tiempo medio antes de mantenimiento
  - tiempo medio antes de recuperación
  - probabilidad de falla

Criterios de calidad: factor eficiencia

- según McCall el factor eficiencia incluye los siguientes criterios:
  - eficiencia en tiempo
  - eficiencia en espacio
- muchas técnicas favorecen este factor: el lenguaje de programación, el sistema operativo, optimización de algoritmos, normalización de datos

Criterios de calidad: factor integridad

- según McCall el factor integridad incluye los siguientes criterios:
  - control de acceso
  - auditoría de acceso
- involucra tanto evitar el acceso malintencionado, asi como los daños causados por errores involuntarios de usuarios autorizados

Criterios de calidad: factor usabilidad

- según McCall el factor usabilidad incluye los siguientes criterios:
  - operabilidad
  - entrenamiento
  - comunicación
  - volumen de E/S
  - tasa de E/S

Criterios de calidad: factor usabilidad

- la usabilidad ha cambiado mucho desde la época de McCall
- incluye aspectos tales como adaptabilidad, aprendizaje, adecuación al contexto
- algunos autores consideran por ejemplo que facilidad de aprendizaje es un factor de calidad independiente
- se puede subdividir en
  - ergonomía general el equipo es adecuado para el uso previsto
  - ergonomía de software estilos de diálogoes, metáforas, diseño de pantallas, etc

Métricas de calidad

- la medición de cualquiera de estos factores está definida en este modelo en base a 41 métricas
- para cada criterio existe una lista de condiciones que se deben cumplir en distintas etapas: requerimientos (R), diseño (D), implementación (I)
- se cuentan las condiciones que se satisfacen en cada una de las etapas, sobre el total posible
- eso da una medida del criterio, que se pondera en partes iguales para medir el factor con los otros criterios asociados al factor

Métricas de calidad: ejemplo

- para medir el criterio completitud del factor correctitud McCall sugiere las siguientes condiciones:
  1. referencias no ambiguas [R,D,I]
  2. referencias a datos bien definidas, o externas [R,D,I]
  3. todas las funciones definidas son usadas [R,D,I]
  4. todas las condiciones y procesamientos están definidos para cada punto de decisión [R,D,I]
  5. todos los parámetros formales y actuales coinciden [D,I]
  6. todos los reportes de problemas han sido resueltos [R,D,I]
  7. el diseño concuerda con los requerimientos [D]
  8. el código concuerda con el diseño [I]

Métricas de calidad: ejemplo

- entonces se cuentan la cantidad de sí en cada etapa, resultando en la métrica de completitud:

$$(\frac{\text{sí en R}}{6} + \frac{\text{sí en D}}{8} + \frac{\text{sí en I}}{8})/3$$

- luego la correctitud se mide como la media entre las medidas de sus criterios

$$\frac{(\textit{completitud} + \textit{trazabilidad} + \textit{consistencia})}{3}$$

Modelo de Boehm

- el segundo modelo de calidad más conocido es el presentado por Barry Boehm en 1978
- este modelo introduce características de alto nivel, características de nivel intermedio y características primitivas, cada una de las cuales contribuye al nivel general de calidad

Características de alto nivel

- las características de alto nivel representan requerimientos generales de uso
- pueden ser:
  - utilidad per-se cuan (usable, confiable, eficiente) es el producto en sí mismo
  - mantenibilidad cuan fácil es modificarlo, entenderlos y retestearlo
  - utilidad general si puede seguir usándose si se cambia el ambiente

Características de nivel intermedio

- las características de nivel intermedio representan los factores de calidad de Boehm:
  - portabilidad (utilidad general)
  - confiabilidad (utilidad per-se)
  - eficiencia (utilidad per-se)
  - usabilidad (utilidad per-se)
  - testeabilidad (mantenibilidad)
  - facilidad de entendimiento (mantenibilidad)
  - modificabilidad o flexibilidad (mantenibilidad)

Características primitivas

- el nivel más bajo corresponde a características directamente asociadas a una o dos métricas de calidad
- de portabilidad
  - independencia de dispositivos
  - auto-contención
- de confiabilidad:
  - auto-contención
  - exactitud
  - completitud
  - consistencia
  - robustez/integridad

Características primitivas

- de eficiencia
  - accesibilidad
  - eficiencia de uso de dispositivos
- de usabilidad
  - robustez/integridad
  - accesibilidad
  - comunicación
- de testeabilidad
  - comunicación
  - auto descripción
  - estructuración

Características primitivas

- de entendibilidad
  - consistencia
  - estructuración
  - concisidad
  - legibilidad
- de modificabilidad
  - estructuración
  - aumentabilidad

Comparación de modelos McCall-Boehm

- aunque parezcan similares, la diferencia está en que McCall focaliza en medidas precisas de alto nivel, mientras que Boehm presenta un rango más amplio de características primarias
- la mantenibilidad está más desarrollada en Boehm

Comparación de modelos McCall-Boehm

Criterio	McCall	Boehm	Criterio	McCall	Boehm
correctitud	+	+	confiabilidad	+	+
integridad	+	+	usabilidad	+	+
eficiencia	+	+	mantenim.	+	+
testeabilidad	+		interoperab.	+	
flexibilidad	+	+	reusabilidad	+	+
portabilidad	+	+	claridad		+
modificab.		+	document.		+
entendib.		+	validez		+

Evaluación de los modelos de McCall y Boehm

- estos modelos tienen sus límites:
  - es difícil que las características y subcaracterísticas sean siempre perfectamente independientes
  - falta una asociación explícita entre los modelos y el proceso de software, ie *cómo* realizar software de calidad
  - las características son en general propiedades abstractas medible mediante métricas. No siempre existe una relación perfectamente lineal entre los valores de las métricas y las características que deben estimar

Modelos ad-hoc

- para monitorear la calidad de software, se pueden tomar dos caminos:
  - adoptar un modelo fijo se supone que todos los factores de calidad importantes son un subconjunto de los de un modelo publicado; se acepta el conjunto de criterios y métricas asociados al modelo
  - desarrollar un modelo propio de calidad se acepta que la calidad está compuesta por varios atributos, pero no se adopta lo impuesto por modelos existentes
- en este último caso, se debe consensuar el modelo con los clientes antes de empezar el proyecto
- se deciden cuáles atributos son importantes para el producto, y cuáles medidas específicas los componen

Modelos ad-hoc

- Gilb y Kitchenham-Walker fueron los pioneros en la filosofía de calidad evolutiva
- Gilb propone la identificación de medidas objetivas de calidad, en complemento con su filosofía de desarrollo evolutivo
- el producto es entregado incrementalmente al cliente, basado en la importancia de las diferentes funcionalidades
- la propuesta COQUAMO de Kitchenham y Walker extiende las ideas de Gilb con el soporte de herramientas automáticas

Normas ISO para el software

- la ISO ha emitido algunas normas que definen un modelo de calidad del software, en varios contextos de uso
  - ISO 9126-1 - define 6 características de calidad principales, y 27 subcaracterísticas. Incluye 3 reportes técnicos (ISO/IEC 9126-2, 3 e 4)
  - ISO/IEC 9241 - define las características de un software usable
  - ISO 12119 - define las características de calidad para un software COTS (Commercial off the shelf)
  - la ISO también ha publicado la norma 14598 que guía en el proceso de valoración de la calidad del software según los criterios de la 9126

Introducción al modelo ISO 9126

- durante muchos años se buscó en la Ingeniería de Software un modelo único para expresar calidad
- la ventaja era obvia: poder comparar productos entre sí
- en 1992, una variante del modelo de McCall fue propuesta como estándar internacional para medición de calidad de software
- ISO 9126 Software Product Evaluation: Quality Characteristics and Guidelines for their Use es el nombre formal
- la última revisión ha sido realizada en el 2004; está en proceso de una nueva revisión
- no se preveen certificados de calidad por esta norma

La calidad en el ciclo de vida del software

- el foco en la calidad cambia durante el ciclo de vida:
  - al principio, durante la recopilación de requerimientos y análisis, la calidad es especificada por los requisitos del usuarios, sobre todo desde el punto de vista externo
  - en la fase de diseño e implementación, la calidad externa se traduce en un disño técnico, confrontándose con el punto de vista de los desarrolladores sobre la calidad interna y complementandose con los requisitos implícitos que el software debe cumplir
  - la calidad final (la del uso) debe ser apropiada para los usuarios y el contexto de uso
- no existe una calidad perfecta o absoluta. Existe solamente una calidad necesaria y suficiente para un dado contexto

Características de calidad internas y externas

- en ISO 9126 se reconocen seis factores de calidad que se pueden considerar tanto internos como externos
  - funcionalidad
  - confiabilidad
  - eficiencia
  - usabilidad
  - mantenibilidad
  - portabilidad

Características de calidad y puntos de vista

	Usuario	Desarrollador	Administr.	Comisionista
funcion.	alta	baja	baja	media
confiab.	media	media	alta	media
usabil.	alta	baja	media	media
eficiencia	media	media	alta	media
mantenib.	baja	alta	media	media
portab.	media	alta	alta	media

Características de calidad de uso

- en ISO 9126 se reconocen cuatro factores de calidad de uso:
  - eficacia
  - productividad
  - seguridad
  - satisfacción

Subcaracterísticas de calidad

- se pretende que estas características sean comprensivas, es decir cualquier componente de la calidad de software se puede describir como combinación de aspectos de estos factores
- a su vez estas características incluyen un conjunto de subcaracterísticas que pueden ser refinado mediante múltiples niveles
- ISO 9126 también define un proceso para evaluar la calidad del software

Subcaracterísticas de funcionalidad

- funcionalidad conjunto de atributos que relacionan la existencia de un conjunto de funciones con sus propiedades especificadas. Las funciones satisfacen necesidades especificadas o implícitas
  - adecuación atributos que determinan si el conjunto de funciones son apropiadas para las tareas especificadas
  - exactitud atributos atributos que determinan que los efectos sean los correctos o los esperados
  - seguridad atributos que miden la habilidad para prevenir accesos no autorizados, ya sea accidentales o deliberados, tanto a programas como a datos
  - interoperabilidad atributos que miden la habilidad de interactuar con sistemas especificados
  - cumplimiento atributos que hacen que el software adhiera a estándares relacionados con la aplicación, y convenciones o regulaciones legales

Subcaracterísticas de confiabilidad

- confiabilidad conjunto de atributos que se relacionan con la capacidad del software de mantener su nivel de performance bajo las condiciones establecidas por un período de tiempo
  - madurez atributos que se relacionan con la frecuencia de fallas por defectos en el software
  - tolerancia a las fallas atributos que miden la habilidad de mantener el nivel especificado de performance en caso de fallas del software
  - recuperación atributos que miden la capacidad de reestablecer el nivel de performance y recuperar datos en caso de falla, y el tiempo y esfuerzo necesario para ello
  - cumplimiento idem en funcionalidad

Subcaracterísticas de usabilidad

- usabilidad conjunto de atributos que se relacionan con el esfuerzo necesario para usar, y en la evaluación individual de tal uso, por parte de un conjunto especificado o implícito de usuarios
  - entendimiento atributos que miden el esfuerzo del usuario en reconocer el concepto lógico del software y su aplicabilidad
  - aprendizaje atributos que miden el esfuerzo del usuario en aprender la aplicación (control, operación, entrada, salida)
  - operabilidad atributos que miden el esfuerzo del usuario en operarar y controlar el sistema
  - atractivo
  - cumplimiento idem funcionalidad

Subcaracterísticas de eficiencia

- eficiencia conjunto de atributos que se relacionan con el nivel de performance del software y la cantidad de recursos usados, bajo las condiciones establecidas
  - en tiempo atributos que miden la respuesta y tiempos de procesamiento de las funciones
  - en recursos atributos que miden la cantidad de recursos usados y la duración de tal uso en la ejecución de las funciones
  - cumplimiento idem funcionalidad

Subcaracterísticas de mantenibilidad

- mantenibilidad conjunto de atributos que se relacionan con el esfuerzo en realizar modificaciones
  - analizabilidad atributos que miden el esfuerzo necesario para el diagnóstico de deficiencias o causas de fallas, o para identificación de las partes que deben ser modificadas
  - facilidad para el cambio atributos que miden el esfuerzo necesario para realizar modificaciones, remoción de fallas o cambios en el contexto
  - estabilidad atributos que se relacionan con el riesgo de efectos no esperados en las modificaciones
  - testeabilidad atributos que miden el esfuerzo necesario para validar el software modficado
  - cumplimiento idem funcionalidad

Subcaracterísticas de portabilidad

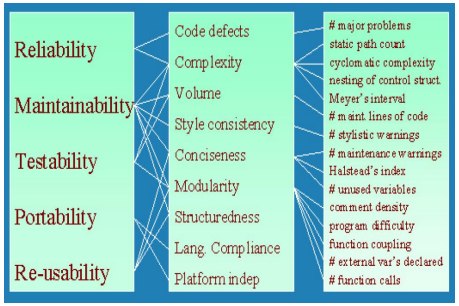
- portabilidad conjunto de atributos que se relacionan con la habilidad del software para ser transferido de un ambiente a otro
  - adaptabilidad atributos que miden la oportunidad de adaptación a diferentes ambientes sin aplicar otras acciones que no sean las provistas para el propósito del software
  - instalabilidad atributos que miden el esfuerzo necesario para instalar el software en el ambiente especificado
  - conformidad atributos que miden si el software se adhiere a estándares o convenciones relacionados con portabilidad
  - reemplazo atributos que se relacionan con la oportunidad y esfuerzo de usar el software en lugar de otro software en su ambiente

Características de calidad de uso

- eficacia capacidad de ayudar al usuario a realizar sus objetivos con exactitud y completitud, en un dado contexto
- productividad capacidad de ayudar al usuario en emplear una apropiada cantidad de recursos en obtener sus resultados
- satisfacción capacidad de satisfacer un usuario en un dado contexto de uso
- seguridad capacidad de lograr aceptables niveles de riesgo para las personas, el ambiente de trabajo, y la actividad, en un dado contexto de uso

Métricas de calidad en ISO 9126

- ISO 9126 provee 3 conjuntos de métricas, para medir respectivamente las características externas (en ISO 9126-2), las internas (en ISO 9126-3), y las de uso (en ISO 9126-4)



Métricas de calidad en ISO 9126

- todas las métricas están caracterizadas por los siguientes elementos:
  - nombre de la métrica
  - objetivo de su uso
  - método con el cual se usa
  - fórmula y elementos de cálculo
  - interpretación de la métrica
  - escala
  - tipo de métrica
  - fuente de los datos de entrada
  - beneficiarios de la métrica

Métricas de calidad en ISO 9126

- ejemplo de una métrica de funcionalidad

Fórmula y elementos de cálculo	$X = 1(A/B)$ $A =$ no. funcionalidad faltante desc. en eval. $B =$ no. funcionalidad descripta en los requisitos
Objetivo de uso	medir la completitud de la funcionalidad ofrecida
Método de medida	uso de un test de tipo caja negra
Interpretación	$0 \leq X \leq 1$ el mejor valor es 1
Escala	absoluta
Tipo de medida	$A =$ número $B =$ número $X =$ número
Fuentes	especificación de requerimientos reporte de evaluación

Objetivos del uso de métricas para medir características externas

- representar la calidad de un producto de software respecto a las características y subcaraterísticas del modelo 9126, durante el testeo
- validar el cumpliemto del software respecto a los requisitos de calidad externa
- predecir el nivel de calidad de uso del producto
- describir el grado de respuesta del producto respecto a los requisito explícitos e implícitos de su uso

Objetivos del uso de métricas para medir características internas

- representar la calidad de un producto de software, en los estados de evolución intermedios y finales no ejecutables, respecto a las características y subcaracterísticas del modelo 9126
- predecir el nivel de calidad externo del producto
- prevenir problemas en el uso del producto, descubriendo anticipadamente potenciales defectos
- las métricas internas son en general combinación de métricas elementales aplicadas a código fuente, diagramas UML o DFD, gráficos, etc (medidas mediante análisis estático o con inspección de código)

Objetivos del uso de métricas para medir características de uso

- verificar la capacidad de un producto de satisfacer las exigencias de los usuarios en un dado escenario de uso, en relación con los objetivos previstos
- estas métricas son en general combinación de métricas elementales aplicadas a la interacción entre usuario y sistema (medidas mediate field tests, inspecciones, walkthrough, etc)

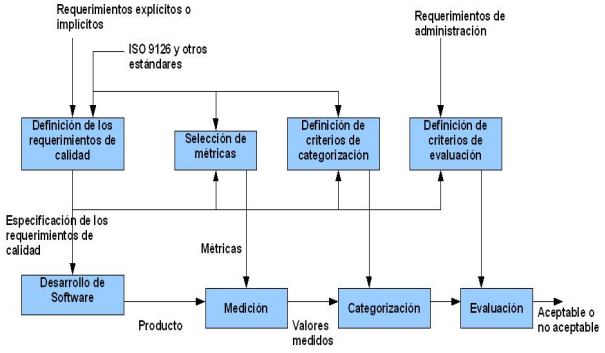
Uso integrado de las métricas

- existen diversas relaciones e interdependencias entre las tres clases de métricas
- muchas de las características del modelo 9126 pueden ser medidas contemporáneamente por métricas internas y externas
- por ejemplo, la *confiabilidad* puede ser medida externamente relevando el número de errores durante la ejecución del producto en un período de tiempo, e internamente inspeccionando el código fuente para verificar el nivel de tolerancia a los errores

Uso integrado de las métricas

- un problema detectado en el uso del producto (ejemplo la imposibilidad de un usuario de completar correctamente una operación) puede ser referido tanto como una característica de calidad externa (confiabilidad o usabilidad) como una interna (error en la estructura de decisión de un código)
- la calidad debe entonces ser medida como una combinación de las métricas de los tres aspectos, de modo de cubrir los distinto puntos de vista

Proceso de evaluación de la calidad



Uso de las métricas en el ciclo de vida

- un comisionista puede valorar la conveniencia de elegir un determinado producto usando métricas para la confiabilidad
- un desarrollador deberá disponer de métricas de funcionalidad para verificar la correcta implementación de productos semielaborados
- en el mantenimiento se podrá evaluar el esfuerzo y el riesgo de modificar un programa usando métrica de mantenibilidad
- se deberan considerar métricas de portabilidad antes de decidir la migración de un producto a otros ambientes
- los usuarios deberán poder medir la usabilidad y la eficacio del producto que se les entrega

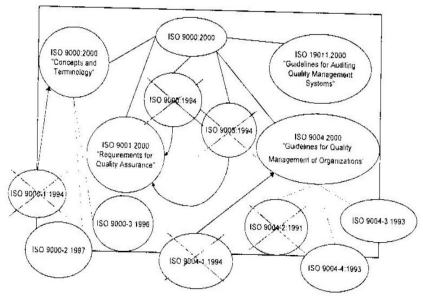
Futuro de la norma ISO 9126

- la norma ISO 9126 está en proceso de revisión
- el modelo de calidad del software confluirá en el sistema de normas ISO 25000 en la norma ISO 25010, sin modificaciones
- las métricas serán definidas por normas en las secciones 2502X, compuestas de más documentos que soportaran la medición del software según distintos puntos de vista (calidad interna, calidad externa, calidad de uso) è in corso di revisione
- el modelo proveerá
  - un modelo de referencia para relevamiento de métricas
  - definición matemática de varias métricas, primitivas de medición y métricas derivadas
  - una guía práctica para el proceso de revisión

ISO 9126 vs. otras normas ISO

- ISO ha emitido una batería de normas bajo el nombre ISO 9000 referidas a la gestión de calidad en todo tipo de organización
- permite controlar los procesos haciendales dirigiéndolos a la satisfacción del cliente
- las ISO 9000 actualmente (desde el año 2000 en adelante) se dividen en
  - ISO 9000 que describe la terminología y los principios esenciales del sistema de gestión de calidad y su organización
  - ISO 9001 para la definición de los requisitos de calidad
  - ISO 9004 que es una guía para el mejoramiento de la prestación en una organización
- la única certificación posible es por adherir a la norma ISO 9001; las otras son guías tentativas para favorecer la correcta aplicación e interpretación del sistema de calidad

ISO 9126 vs. otras normas ISO



ISO 9126 vs. otras normas ISO

- las normas ISO 9000 son universales, y su aplicación prescinde de las dimensiones o del sector de la empresa
- definen principios genéricos que la empresa debe seguir pero no el modo en la cual debe producir. Por lo tanto no sólo son aplicables a los productos sino la organización productora
- la ISO 9001 garantiza el control del proceso productivo y su eficacia, pero no su eficiencia
- actualmente las ISO 9000 son usadas en la industria como modelo de referencia para la calificación y selección de proveedores y clientes

ISO9126 vs. otras normas ISO

- ISO 9001 propone un manejo de la calidad orientado al proceso
- es decir, diseñar, documentar, implementar, soportar, monitorear, controlar y mejorar las siguientes actividades:
  - proceso de administración de calidad
  - proceso de administración de recursos
  - proceso de investigación sobre regulaciones
  - proceso de investigación de mercado
  - proceso de diseño de productos
  - proceso de compras
  - proceso de producción
  - proceso de provisión de servicios
  - proceso de protección del producto
  - proceso de evaluación de las necesidades del cliente

ISO9126 vs. otras normas ISO

- es decir, diseñar, documentar, implementar, soportar, monitorear, controlar y mejorar las siguientes actividades (cont.):
  - proceso de comunicaciones con el cliente
  - proceso de comunicaciones interno
  - proceso de control de documentación
  - proceso de registro de actividades
  - proceso de planificación
  - proceso de entrenamiento
  - proceso de auditoría interna
  - proceso de monitoreo y medición
  - proceso de mejora contínua

Otros modelos de calidad

- FURPS (Robert Grady, 1992, extendido por Rational Software en FURPS+) incluye funcionalidad, usabilidad, confianza, performance, soportabilidad
- Dromey (Robert Dromey, 1996) describe la idea de relacionar atributos del producto con atributos de calidad para su evaluación