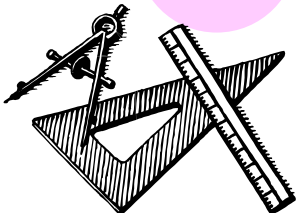
Métricas Técnicas del Software

Capítulo 19
Roger, Pressman
"Ingeniería de Software"
Quinta Edición



Motivación

- A diferencia de otras disciplinas, la ingeniería del software no está basada en leyes cuantitativas básicas de la Física.
- Se intenta obtener un conjunto de medidas indirectas que dan lugar a métricas que proporcionan una indicación de la calidad de algún tipo de representación del software.
- Según Fenton [FEN91]:
  - La medición es el proceso por el que se asignan números o símbolos a los atributos de las entidades en el mundo real, de tal manera que las definan de acuerdo con unas reglas claramente definidas.
  - En las ciencias físicas, medicina y, más recientemente, en las ciencias sociales, somos ahora capaces de medir atributos que previamente pensábamos que no eran medibles...
  - Por supuesto, tales mediciones no están tan refinadas como las de las ciencias físicas..., pero existen [y se toman importantes decisiones basadas en ellas].
  - Sentimos que la obligación de intentar «medir lo no medible» para mejorar nuestra comprensión de entidades particulares es tan poderosa en la ingeniería del software como en cualquier disciplina.

Definiciones

- **Medida**: indicación cuantitativa de la extensión, la cantidad, la dimensión o el tamaño de algún atributo de producto o proceso. Tomado en un punto del tiempo o lugar.
- **Métrica**: medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo determinado. Reune y relaciona medidas en varios puntos (promedio, mediana, tasa, razón)

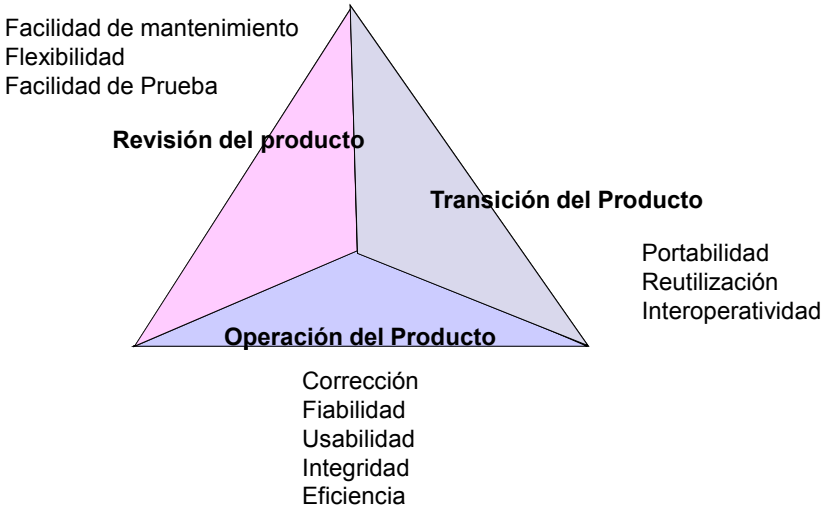
19.1 Calidad de Software

- En la búsqueda de calidad de software se enfatiza tres puntos importantes:
  1. Los requisitos del software son la base de las medidas de la calidad. La falta de concordancia con los requisitos es una falta de calidad.
  2. Unos estándares específicos definen un conjunto de criterios de desarrollo que guían la manera en que se hace la ingeniería del software. Si no se siguen los criterios, habrá seguramente poca calidad.
  3. Existe un conjunto de requisitos implícitos que a menudo no se nombran (por ejemplo, facilidad de mantenimiento). Si el software cumple con sus requisitos explícitos pero falla en los implícitos, la calidad del software no será fiable.

19.1.1 Factores de calidad de McCall

- McCall y sus colegas proponen, en 1977, una serie de factores de calidad de software:
  - Se concentran en tres aspectos:
    - Características Operativas
    - Capacidad de Cambios
    - Adaptabilidad a nuevos entornos

Factores de Calidad de McCall



Factores de Calidad de McCall

- Se desarrollan medidas indirectas como sigue:

$$Fq = c_1 \times m_1 + c_2 \times m_2 + \ldots + c_n \times m_n$$

  - Fq: factor; m métricas; c: cohef. correlación
- La relación entre los factores de calidad y las métricas se muestra en la tabla siguiente.

Métrica de Calidad \ Factor de Calidad	Corrección	Fiabialidad	Eficiencia	Integridad	Mantenimiento	Flexibilidad	Capacidad de pruebas	Portabilidad	Reusabilidad	Interoperativilidad	Usabilidad
Facilidad de auditoría				X			X				
Exactitud		X									
Estandarización de comunicaciones										X	
Compleción	X										
Complejidad		X				X	X				
Concisión			X		X	X					
Consistencia	X	X			X	X					
Estandarización de datos										X	
Tolerancia a errores		X									
Eficiencia de ejecución			X								
Capacidad de expansión						X					
Generalidad						X		X	X	X	
Independiencia del hardware								X	X		
Instrumentación				X	X		X				
Modularidad		X			X	X	X	X	X	X	
Operatividad			X								X
Seguridad				X							
Autodocumentación					X	X	X	X	X		
Simplicidad		X			X	X	X				
Independencia del sistema								X	X		
Trazabilidad	X										
Facilidad de formación											X

19.1.2 FURPS (Propuestas por Hewlett – Packard en 1987)

- **F**unctionality (Funcionalidad). Valora el conjunto de capacidades del programa, la generalidad de las funciones y seguridad global.
- **U**sability (Facilidad de uso). Valora aspectos como: estética, consistencia y documentación general.
- **R**eliability (Fiabilidad). Valora: fallos (frecuencia, gravedad y recuperación), exactitud de salidas.
- **P**erformance (Rendimiento). Tiempo de procesamiento y respuesta, consumo de recursos, efectividad y eficacia.
- **S**upport (Soporte). Facilidad de mantenimiento, compatibilidad y facilidad de: prueba, instalación y localización de problemas.

9.1.3 Factores de Calidad ISO 9126

- Este estándar identifica seis atributos de calidad:
  - Funcionalidad. Satisface necesidades.
    - Idoneidad, corrección, interoperabilidad, conformidad y seguridad.
  - Confiabilidad. Cantidad de tiempo disponible.
    - Madurez, tolerancia a fallos, facilidad de recuperación.
  - Usabilidad. Grado de facilidad de uso.
    - Facilidad de: comprensión, aprendizaje y operatividad.

19.1.3 Factores de Calidad ISO 9126

- Eficiencia. Grado en que optimiza los recursos de cómputo.
  - Tiempo de uso y recursos utilizados.
- Facilidad de mantenimiento. Que tan fácil corregir errores.
  - Estabilidad, facilidad de: análisis, cambios y prueba.
- Portabilidad. Facilidad de cambiar de entorno.
  - Facilidad de: instalación, ajuste y adaptación al cambio.

19.2 Una estructura para las métricas técnicas del software

- Para conseguir dar valores cuantitativos a las métricas, es necesario establecer un modelo de medición.
- Según Fenton
  - Desarrollar una métrica única sería semejante a la búsqueda imposible del *santo grial*.
- Según Shari Pfleeger
  - Lo mismo que las medidas de temperatura comienzan con un índice de referencia y evolucionan por sofisticadas escala, herramientas y técnicas, las medidas del software están evolucionando.

19.2.1 El reto de las métricas técnicas

- Según Fenton:
  - A medida que crece el número de mediciones de distintos atributos, se tienen que satisfacer objetivos incompatibles. Lo que es contrario a la teoría de la representación de la medición.
  - Hay pocos intentos científicos sobre medición…Se menciona a menudo la imposibilidad de llevar a cabo experimientos.
- Sin embargo, la medición es necesaria si se desea conseguir calidad.

19.2.2. Principios de medición

- Las métricas técnicas:
  - ayudan a la evaluación de los modelos de análisis y diseño,
  - proporcionan una indicación de la complejidad de los diseños procedimentales y del código fuente y
  - ayudan en el diseño de pruebas más efectivas.

Roche [ROC94] propone un proceso de medición se puede caracterizar por cinco actividades:

1. Formulación:
   - obtención de medidas y métricas del software apropiadas para la representación del software.
2. Colección:
   - mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
3. Análisis:
   - cálculo de las métricas y aplicación de herramientas matemáticas.
4. Interpretación:
   - evaluación de resultados de métricas en un esfuerzo por conseguir una visión interna de la calidad de la representación.
5. Realimentación (feedback):
   - recomendaciones obtenidas de la interpretación de métricas técnicas transmitidas al equipo que construye el software.

Principios asociados con la métricas técnicas

- Los objetivos de la medición deberán establecerse antes de empezar la recogida de datos.
- Todas las técnicas sobre métricas deberían definirse sin ambigüedades.
- Las métricas deberían obtenerse basándose en una teoría válida para el dominio de aplicación
- Hay que hacer las métricas a medida para acomodar mejor los productos y procesos específicos.

Principios asociados con las actividades de medición

- Siempre que sea posible, la recogida de datos y su análisis deben automatizarse.
- Se deberían aplicar técnicas estadísticas validas para establecer las relaciones entre los atributos internos del producto y las características externas de la calidad
- Se deberían establecer una directrices de interpretación y recomendaciones para todas las métricas.

Consideraciones

- El éxito de una actividad de medición esta ligada al soporte de gestión.
- Se deben considerar los fondos, la formación y la promoción si se quiere establecer y mantener un programa de medición técnica.

19.2.3. Características fundamentales de las métricas del software (Ejiogu)

- Simples y fáciles de calcular.
  - Deberían ser relativamente fácil aprender a obtener la métrica y su cálculo no debería demandar un esfuerzo o cantidad de tiempo inusuales.
- Empírica e intuitivamente persuasivas.
  - Satisfacer las nociones intuitivas del ingeniero sobre el atributo del producto en cuestión
- Consistentes y objetivas.
  - Deberían siempre producir resultados sin ambigüedad. Un tercer equipo debería ser capaz de obtener el mismo valor de métrica usando la misma información del software.

19.2.3. Características fundamentales de las métricas del software (Ejiogu)

- Consistentes en el empleo de unidades y tamaños.
  - El cálculo matemático de la métrica debería emplear medidas que eviten extrañas combinaciones de unidades.
- Independientes del lenguaje de programación.
  - Deberían basarse en el modelo de análisis, diseño o en la estructura del programa.
- Eficaces en el mecanismo para la realimentación de la calidad.
  - Proporcionar al desarrollador de software información que le lleve a un producto final de mayor calidad.

19.3 Métricas del modelo de análisis

- En esta fase las métricas técnicas proporcionan una visión interna a la calidad del modelo de análisis.
- Estas métricas examinan el modelo de análisis con la intención de predecir el tamaño del sistema resultante. Es probable que el tamaño y la complejidad del diseño estén directamente relacionadas.

Metodología de Puntos de Función

The International Function Point Users Group

Release 4.1

Ángeles Sumano López

Cálculo de Puntos de Función

- 1° Calcular Puntos de función sin ajustar
  - Establecer los 5 indicadores y su dificultad
- 2° Establecer Modificadores con su grado de influencia
- 3° Aplicar fórmula
- 4° Interpretar

Indicadores de datos en Puntos de Función

- Archivos Lógicos Internos (ALI).
  - Grupo identificable de datos relacionados lógicamente o de información de control que pertenece al usuario
  - es mantenido dentro de las fronteras del sistema.

Indicadores de datos en Puntos de Función

- Archivo de Interfaz Externa (AIE).
  - Grupo identificable de datos relacionados lógicamente o de información de control que pertenece al usuario,
  - es referido por la aplicación, pero mantenido dentro de las fronteras de otra aplicación.

Indicadores de transacciones en Puntos de Función

- Entradas Externas (EE).
  - Es un proceso elemental que procesa datos o información de control que viene de fuera de la frontera de la aplicación para mantener uno o más ALI y/o
  - alterar el comportamiento del sistema.

Indicadores de transacciones en Puntos de Función

- Salidas Externas (SE).
  - Es un proceso elemental lógico que debe contener al menos una fórmula matemática, cálculo o
  - crear datos derivados que envía datos o información de control fuera de la frontera de la aplicación.

Indicadores de transacciones en Puntos de Función

- Consultas Externas (CE).
  - Es un proceso elemental que envía datos o información de control fuera de la frontera de la aplicación.

Asignación de nivel de dificultad para los Archivos Lógicos Internos o Archivos de Interfaz Externos

	1 -19 TDE	20 - 50 TDE	51 ó + TDE
0 – 1 TRE	simple	simple	mediano
2 – 5 TRE	simple	mediano	complejo
6 ó + TRE	mediano	complejo	complejo

Cálculo de Puntos de Función sin ajustar

- Una vez señalado identificadores, su complejidad y modificadores se asignan pesos que ya están dados por el método.

Puntos de Función sin ajustar.

Indicador	simple	mediano	complejo	SUMA
ALI	_ * **7**	_ * **10**	_ * **15**	___
AIE	_ * **5**	_ * **7**	_ * **10**	___
EE	_ * **3**	_ * **4**	_ * **6**	___
SE	_ * **4**	_ * **5**	_ * **7**	___
CE	_ * **3**	_ * **4**	_ * **6**	___
			T =	____

Estimadores en Puntos de Función

1. Comunicación de datos.
   - Describe el grado con el cual la aplicación se comunica directamente con el procesador.
2. Procesamiento Distribuido de Datos.
   - Mide el grado con el que la aplicación transfiere datos entre componentes de la aplicación
3. Rendimiento.
   - El rendimiento será crítico y tendrá influencia sobre cómo diseñar, desarrollar o implementar.

Estimadores en Puntos de Función

4. Configuración Altamente Usada.
   - El software será implementado en un entorno existente y fuertemente utilizado.
5. Promedio de Transacciones.
   - Un alto promedio de transacciones influenciará al diseño, desarrollo, implantación y soporte.
6. Entrada de Datos en Línea.
   - El software requerirá entradas interactivas.

Estimadores en Puntos de Función

7. Eficiencia para el Usuario Final.
   - Las funciones en línea proveídas tendrán que enfatizar un diseño para la eficiencia del usuario final.
8. Actualización en Línea.
   - Se necesitará la actualización de archivos maestros en forma interactiva
9. Procesamiento Complejo.
   - Describe el grado en el cual el procesamiento lógico influencia el desarrollo de la aplicación.

Estimadores en Puntos de Función

10. Reusabilidad.
    - Describe el grado en el cual la aplicación y su código han sido específicamente diseñados, desarrollados y soportados para que se puedan reutilizar.
11. Facilidad de Instalación.
    - Describe el modo en que la conversión desde medios ambientes previos influenciarán el desarrollo de la aplicación.

Estimadores en Puntos de Función

- 12.Facilidad de Operación.
  - Describe el grado en el cual las aplicaciones atienden los aspectos operacionales, tales como:
    - salvar y recuperar datos y recuperación de procesos.
  - La facilidad de operación es una característica de la aplicación. Minimizando la necesidad de actividades manuales, tales como:
    - montaje de cintas, manejo de papel e intervención manual directa en el lugar.

Estimadores en Puntos de Función

13.Varios Sitios.

- Describe el grado en el cual la aplicación será diseñada, desarrollada e implantada en múltiples localizaciones y organizaciones de usuarios

14.Facilidad de Cambios.

- Describe el grado en el cual la aplicación ha sido desarrollada para la modificación fácil del procesamiento lógico o las estructuras de datos.

Estimadores en Puntos de función

- A cada estimador se le asigna un grado de influencia, la cual puede ser:
  - 0=sin influencia,
  - 1=accidental,
  - 2=moderado,
  - 3=medio,
  - 4=significativo,
  - 5=esencial.
- Ya calificados los estimadores, se suman en la variable M.

Asignación del grado de influencia para "Actualización en línea"

Grado	Razón
0	No hay actualización en línea
1	Se incluye la actualización en línea de uno a tres archivos de control. El volumen de actualización es bajo y la recuperación es fácil.
2	Se incluye la actualización en línea de cuatro o más archivos de control. El volumen de actualización es bajo y la recuperación es fácil.
3	Se incluye la actualización en línea de la mayoría de los archivos lógicos internos.
4	Además, la protección contra pérdida de datos es esencial y tendrá que ser especialmente diseñada y programada en el sistema.
5	Además, se consideran dentro de los procesos de recuperación volúmenes altos. Se incluyen procesos de recuperación altamente automatizados con intervención mínima del operador.

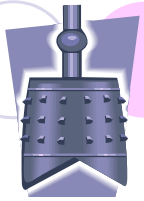
Aplicación de fórmula de PF

- pf = T * (0.65 + 0.01 * M)
- Posible interpretación sobre el proyecto:
  - Complejidad
    - sencilla hasta 100 puntos
    - media 101 a 300 puntos
    - difícil de 301 a 500 puntos
  - Costo de $50 a $100 dólares EEUU por punto de función
  - Considerando datos históricos de productividad por punto de función, se puede calcular:
    - Tiempo requerido
    - Personas requeridas

Conclusión sobre Puntos de Función

- El método de Puntos de Función sirve para:
  - detallar los requerimientos no funcionales o restricciones del sistema
  - estimar el costo, tiempo y personal requerido para el desarrollo de un sistema
- Es un método cuantitativo que apoya a la estimación de esfuerzo

19.3.2. La Métrica BANG



- Se emplea para desarrollar una indicación del tamaño del software a implementar como consecuencia del modelo de análisis.
- Desarrollada por DeMarco.
- Es una indicación independiente de la implementación del tamaño del sistema.
- El desarrollador de software debe evaluar un conjunto de primitivas (elementos del modelo de análisis que no se subdividen más en este nivel).

… Métrica BANG

- Primitivas
  - Primitivas funcionales (PFu). Transformaciones (burbujas) que aparecen en el nivel inferior de un diagrama de flujo de datos
  - Elementos de datos (ED). Los atributos de un objeto de datos,
    - los elementos de datos son datos no compuestos y aparecen en el diccionario de datos.
  - Objetos (OB). Objetos de datos (Ej. Entidad)
  - Relaciones (RE). Las conexiones entre objetos de datos
  - Estados (ES) .El número de estados observables para el usuario en el diagrama de transición de estados
  - Transiciones (TR). El número de transiciones de estado en el diagrama de transición de estados.

… Métrica BANG

- Además se determinan las cuentas adicionales para:
  - Primitivas modificadas de función manual (PMFu). Opciones que caen fuera del límite del sistema y que deben modificarse para acomodarse al nuevo sistema.
  - Elementos de datos de entrada (EDE). Aquellos elementos de datos que se introducen en el sistema.
  - Elementos de datos de salida (EDS). Aquellos elementos de datos que se sacan del sistema.
  - Elementos de datos retenidos (EDR). Aquellos elementos de datos que son retenidos (almacenados) por el sistema.
  - Muestras (tokens) de datos (TCJ). Las muestras de datos (elementos de datos que no se subdividen dentro de una primitiva funcional que existen en el límite de la i-ésima primitiva funcional (evaluada para cada primitiva).
  - Conexiones de relación (REJ). Las relaciones que conectan el i-ésimo objeto en el modelo de datos con otros objeto.

… Métrica BANG

- DeMarco sugiere que la mayoría del software se puede asignar a uno de los dos dominios siguientes:
  - Las aplicaciones de *dominio de función*
    - encontradas comúnmente en aplicaciones de ingeniería y científicas.
    - hacen hincapié en la transformación de datos y no poseen generalmente estructuras de datos complejas.
  - Las aplicaciones de *dominio de datos*
    - encontradas comúnmente en aplicaciones de sistemas de información.
    - tienden a tener modelos de dato complejos.

… la Métrica BANG

- Usando la relación RE/PFu:
  - RE / PFu **<** 0.7 implica una aplicación de dominio de funcional
  - 0.8 **<** RE / PFu **<** 1.4 indica una aplicación híbrida
  - RE / PFu **>** 1.5 implica una aplicación de dominio de datos
- DeMarco sugiere que se emplee una cuenta media de muestra (token) por primitiva

$$TC_{avg} = \Sigma TC_i/PFu$$

- para controlar la uniformidad de la partición a través de muchos diferentes modelos dentro del dominio de una aplicación.

… la Métrica BANG

- Para calcular la métrica Bang para aplicaciones de dominio de función se emplea el siguiente algoritmo:
  - Bang = 0
  - Mientras haya primitivas funcionales hacer:
    - Calcular cuenta-token de la primitiva i
    - Calcular el incremento PFu corregido (IPFuC)
    - Asignar la primitiva a una clase
    - Evaluar la clase y anotar el peso valorado
    - Bang = Bang + valoración IPFuC
  - Fin Mientras

… la Métrica BANG

- La cuenta-token se calcula determinando cuántos símbolos léxicos (tokens) diferentes son "visibles" dentro de la primitiva.
  - Es posible que el número de símbolos léxicos (tokens) y el número de elementos de datos sea diferente, si los elementos de datos pueden moverse desde la entrada a la salida sin ninguna transformación interna.
- La IPFuC corregida se determina de una tabla publicada por DeMarco.

Tci	IPFuC
2	1.0
5	5.8
10	16.6
15	29.3
20	43.2

… la Métrica BANG

- Para aplicaciones de dominio de datos, se calcula la métrica Bang con el algoritmo:
  - Bang = 0
  - Mientras haya objetos de datos
    - Calcular la cuenta de relaciones del objeto i
    - Calcular el incremento OB corregido (IOBC)
    - Bang = Bang + IOBC
  - Fin Mientras

… la Métrica BANG

- El IOBC corregido se determina también de una tabla publicada por DeMarco.

Rei	IOBC
1	1
3	4
6	9

… la Métrica BANG

- Una vez que se ha calculado la métrica Bang, se puede emplear el historial anterior para asociarla al esfuerzo y tamaño.
- Se sugiere que cada empresa construya sus propias versiones de las tablas IPFuC e IOBC para calibrar la información del proyecto completo.

19.3.3 Calidad de la especificación

- Propuesta por Alan Davis en 1993
- Corresponde a la calidad de la especificación de requerimientos.
- Incluyen las características siguientes:
  - Especificidad, compleción, corrección, comprensión, verificación, consistencia, logro, concisión, trazabilidad, modificación, exactitud, reutilización.
- Cualitativas,
  - pero podrían algunas podrían medirse.

… Calidad de la especificación (Ejemplo)

- Sea
  - Nr = Nf +Nnf, donde
    - Nr = Número de requerimientos
    - Nf = Número de requerimientos funcionales
    - Nnf = Número de requerimientos NO funcionales
- Especificidad (ausencia de ambigüedad)
  - Q1 = Nui / Nr, donde
    - Nui = número de requerimientos donde todos los revisores tuvieron interpretaciones idénticas.

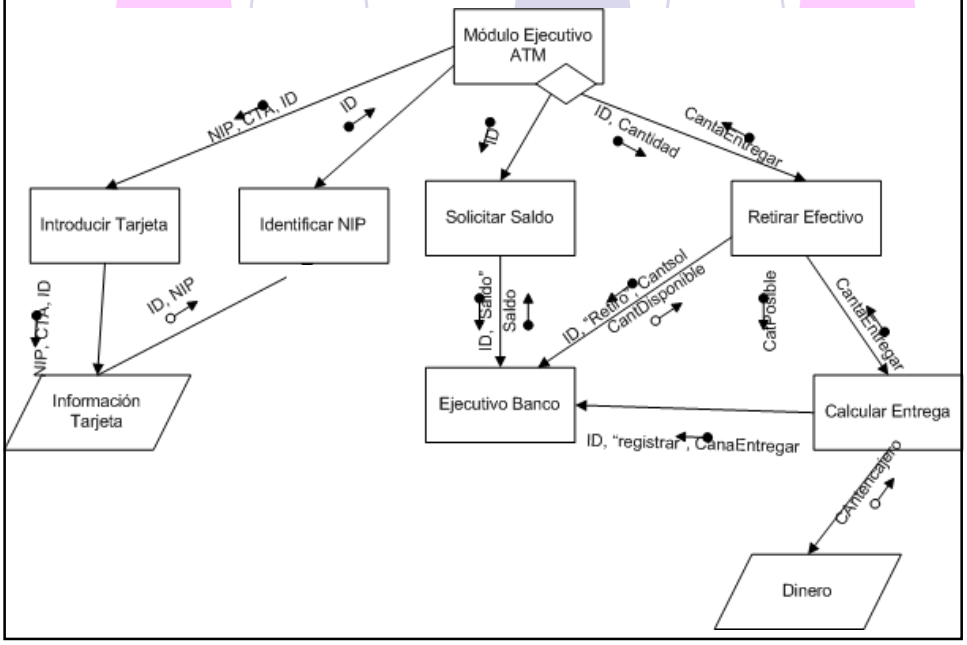
19.4 Métricas del Modelo de Diseño

- Proporcionan al diseñador una mejor visión interna
- Ayudan a que el diseño evolucione a un nivel superior de calidad
- Se dividen en:
  - Métricas de diseño arquitectónico
  - Métricas de Diseño a nivel de componente

19.4.1 Métricas de diseño arquitectónico

- Métricas de complejidad propuestas por Card y Glass en 1990
- Complejidad estructural
  - $S(i) = f^2_{out}(i)$, donde
    - f_{out} (i) = expansión del módulo i (número de módulos inmediatamente subordinados al módulo i)
- Complejidad de Datos. Sobre la interfaz interna del módulo.
  - $D(i) = v(i) / [f_{out}(i) + 1]$, donde
    - v(i) es el número de variables que entran o salen del módulo.
- Complejidad Total del Sistema.
  - $CS = \Sigma_i (S(i) + D(i))$
- Complejidad Relativa del Sistema
  - CRS = Promedio((S(i) +D(i)))
- A medida que crecen los valores de complejidad, crece la complejidad arquitectónica del sistema. CRS>26.5, malo

Suponga el siguiente DE de un cajero automático



Calcular la complejidad total del sistema

Módulo	Expansión del módulo	Complejidad Estructural	Complejidad de Datos
Módulo Ejecutivo ATM	4	16	1.6
Introducir tarjetas	0	0	6
Identificar NIP	0	0	3
Solicitar Saldo	1	1	2
Retirar Efectivo	2	4	3
Calcular Entrega	1	1	2.5
TOTALES		22	17.9

Complejidad total del sistema= 39.9

Complejidad Relativa = 6.65

… Métricas de diseño arquitectónico

- Henry y Kafura proponen en 1981 la métrica de complejidad de expansión – concentración, que considera las estructuras de datos que recoge (concentran) o actualizan (expansión).
  - MHK = longitud(i) x $[f_{in}(i) + f_{out}(i)]^2$, donde
    - longitud(i) = número de sentencias en lenguaje de programación
- En 1991, Fenton propone medidas de morfología simples basadas en la estructura de módulos jerárquicos del sistema:
  - Tamaño = n + a (número de nodos + número de aristas)
  - Profundidad
  - Anchura
  - Relación arco-nodo, r = a/n

19.4.2 Métricas de Diseño a nivel componentes

- Se centran en las características internas de las componentes del software e incluyen las medidas de las “3C”
  - Cohesion (Cohesión)
  - Coupling (Acoplamiento)
  - Complexity (Complejidad, en el capítulo 17)
- Pueden aplicarse antes o después de tener el código.

Métrica de Cohesión

- Para medir la cohesión se puede usar el trabajo de Bieman y Ott, el cual define varios conceptos:
  - número de elementos (tokens),
  - rebanada de datos (data slice),
  - adhesivo (glue) y
  - superadhesivo (superglue).
- Con ellas se calcula la cohesión funcional fuerte
  - CFF = número de superadhesivos / número de elementos
- Cohesión Funcional Débil, se calcula así:
  - CFD = número de adhesivos / número de elementos
- Mientras más cercano sean CFF ó CFD a 1, mayor será la cohesión del módulo.

Métrica de Cohesión

- **Número de elementos**. Son todas las referencias a variables y constantes que se utilizan en el módulo. Si una variable aparece varias veces, cada una se toma como un elemento.
  - Para distinguirlos se les puede agregar como subíndice un número que dice si es la primera aparición, la segunda u otra.
- **Rebanada de datos.** Es el conjunto de variables y constantes que afectan el valor de una variable dentro del módulo. Se forma como sigue:
  a) se comienza por una variable de resultado (ya sea regreso de una función o variable de salida), usualmente al final, donde guarda su valor definitivo y se agrega a la rebanada, incluyendo el subíndice que le tocó al revisar los elementos.
  b) Se analiza el lado derecho de la asignación y se anotan las variables y constantes que intervienen en el cálculo de su valor.
  c) Para cada elemento de la rebanada, se revisan las líneas anteriores disminuyendo una a una, repitiendo el proceso del paso b.

Métrica Cohesión

- **Adhesivo.** Se le llamará adhesivo a un elemento que aparece en dos o más rebanadas.
- **Superadhesivo.** Se denomina superadhesivo a un elemento que está en todas las rebanadas de un módulo.

Ejemplo

```
1  procedure SumAndproduct (N:integer; var SumN, ProdN: integer);
2  var I:integer;
3  begin
4     SumN := 0;
5     ProdN := 1;
6     for I := 1 to N do
7     begin
8       SumN := SumN + I;
9       ProdN := ProdN * I;
10   end;
11 end.
```

- **Elementos**: {N_1, $SumN_1$, $ProdN_1$, I1, $SumN2$, 0_1, $ProdN2$, 1_1, I2, 12, N2, SumN3, SumN4, I3, ProdN3, ProdN4, I4}
- **Número de elementos** = 17
- **Rebanada de SumN**: {SumN3, SumN4, I3, I2, 12, N2, SumN2, 01, N1, I1, SumN1}
- **Rebanada de ProdN:** {ProdN3, ProdN4, I4, I2, 12, N2, ProdN2, 11, N1, I1,Prod1}
- **Adhesivos**: I2, 12, N2, N1, I1
- **Superadhesivos:** I2, 12, N2, N1, I1
- **CFF** = 5/ 17 = 0.294
- **CFD** = 5 / 17 = 0.294

Métrica de Acoplamiento

- Proporciona una indicación de la conectividad de un módulo con otros.
- Medida propuesta por Dhama en 1995
- Medidas para el acoplamiento de flujo de datos y de control
  - d_i = número de parámetros de datos de entrada
  - c_i = número de parámetros de control de entrada
  - d_o = número de parámetros de datos de salida
  - c_o = número de parámetros de control de salida

… Métrica de Acoplamiento

- Medidas para el acoplamiento global
  - g_d = número de variables globales usadas como datos
  - g_c = número de variables globales usadas como control
- Medidas para el acoplamiento de entorno:
  - w = número de módulos llamados
  - r = número de módulos que llaman al módulo en cuestión

… Métrica de Acoplamiento

- Indicador de acoplamiento de módulo
  - m_c = k / M, donde
    - k = 1, constante de proporcionalidad
    - M = d_i + a x c_i + d_o + b x c_o + g_d + c x g_c + w + r, con
    - a = b = c = 2
- Cuando mayor es m_c, menor es el acoplamiento del módulo.

19.4.3 Métricas de diseño de Interfaz

- En 1993, Sears propone la métrica de *Conveniencia de la Representación* (CR)
  - mide la transición de una entidad de representación a otra dentro de una GUI[1].
- Entre las entidades de representación (er) de una GUI están:
  - *Iconos, campos de edición, menús y ventanas*
- Se asigna el costo a cada secuencia de acción como sigue:
  - Costo = Σ [frecuencia de transición (k) x costo de transición (k)], donde
    - k es la transición específica de una entidad de representación a la siguiente cuando se realiza una tarea específica

[1] Graphic User Interface

… Métricas de diseño de Interfaz

- La conveniencia de representación se calcula como:
  - CR = 100 x [(costo de representación óptima) / (costo de representación propuesta)]
  - CR sería igual a 100 como mejor medida
- Para el cálculo de la representación óptima se hace:
  - Dividir la pantalla en una cuadrícula en donde cada cuadro representa una **er.**
  - Si se tiene una cuadrícula de N posiciones y K diferentes er, entonces el número de posibles distribuciones es:
    - [N! / (K! × (N – K)!] × K!
- Alternativamente, se puede hacer un algoritmo de búsqueda de árbol.

[1] Graphic User Interface

… Métricas de diseño de Interfaz

- CR se utiliza para evaluar diferentes distribuciones.
- Se puede utilizar una ΔCR
- Por otro lado, Nielsen y Levy en 1994 proponen mejor consultar al usuario sobre cómo le gustaría su GUI.
  - NOTA: El trabajo de consulta del usuario también es laborioso y cansado.

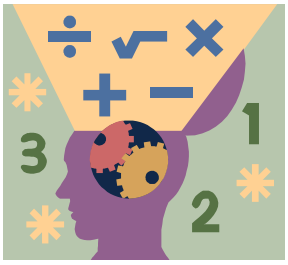
[1] Graphic User Interface

19.5 Métricas del Código Fuente

- Nota: Líneas de código representa la medida más simple e inmediata para el código fuente.
- Otra métrica que ha sido fuertemente estudiada, pero que no se le ha encontrado un significado práctico es la de Halsted, propuesta en 1977.
- Teóricamente, deben existir varias medidas para un algoritmo.

… Métricas del Código Fuente

- La forma de calcularlas es con las siguientes cantidades:
  - n_1 = número de operadores diferentes en el programa
  - n_2 = número de operandos distintos en el programa
  - N_1 = número total de veces que aparecen los operadores
  - N_2 = número total de veces que aparecen los operandos



… Métricas del Código Fuente

- Las métricas de Halstead, serían:
  - Longitud global del programa
    - $N = n_1 \log_2 n_1 + n_2 \log_2 n_2$
  - Volumen del programa
    - $V = N \log_2 (n_1 + n_2)$
  - Volumen compacto (ya que V varía dependiendo del lenguaje)
    - $L = 2 / n_1 \times n_2 / N_2$

19.6 Métricas de Prueba

- La que existen se concentran en el proceso de prueba y no el las características técnicas de la prueba.
- Los responsables de la prueba, se guían por las métricas de análisis, diseño y código.
- Puntos de Función
- Bang
- Halstead
- Complejidad ciclomática

19.7 Métricas de Mantenimiento

- En el estándar IEEE 982.1-1998, se propone un Índice de Madurez del Software (IMS).
  - Proporciona una indicación de la estabilidad del producto de software y se expresa como:
  - IMS = $[M_t - (F_c + F_a + F_d)] / M_t$, donde
    - M_t = número de módulos en la versión actual
    - F_c = número de módulos en la versión actual que han cambiado
    - F_a = número de módulos en la versión actual que se han añadido
    - F_d = número de módulos en la versión actual que han eliminado