

Estimaciones y métricas

Administración de Recursos
Ing. en Sistemas de Información
FRBA – UTN – Argentina - 2014

Importancia de Medir

- Estimación
- Evaluación de productividad
- Control de proyectos
- Calidad del Software
- Tomar decisiones tácticas

Conceptos

- La ***Métrica*** es una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado
- Se recopilan medidas y desarrollan ***métricas*** para obtener ***indicadores***
- El ***Indicador*** es una métrica o una combinación de métricas que proporcionan una visión profunda del proceso del software, del proyecto de software, o el producto en sí

Utilidad de las métricas

- Evaluar el estado del proyecto en curso
- Seguir la pista de los riesgos potenciales
- Detectar las áreas de problema antes de que se conviertan en "***críticas***"
- Ajustar el flujo y las tareas del trabajo
- Evaluar la habilidad del equipo del proyecto en controlar la calidad de los productos de trabajo de la ingeniería del software.

Métricas de Software

- ***Medidas directas:***
 - líneas de código (LDC) producidas
 - velocidad de ejecución
 - tamaño de memoria utilizado
 - defectos informados durante un período de tiempo
- ***Medidas indirectas:***
 - Funcionalidad
 - Calidad
 - Complejidad
 - Facilidad de mantenimiento

Clasificación de Métricas

- Orientadas al tamaño
- Orientadas a la función

Métricas orientadas al tamaño

- Errores por KLDC (miles de línea de código)
- Defectos por KLDC
- \$ por KLDC
- Páginas de documentación por KLDC
- Errores por personas-mes
- LDC por persona-mes
- \$ por página de documentación

Las métricas orientadas al tamaño no están aceptadas universalmente como el mejor modo de medir el proceso de desarrollo.

Métricas orientadas a la función

Las métricas orientadas a la función miden la funcionalidad que ofrece una aplicación.

Debido a que la funcionalidad no se puede medir en forma directa, debe derivarse de otras métricas directas.

La técnica de *Puntos de Función* aplica este enfoque. Los puntos de función se derivan utilizando una relación empírica basada en métricas directas del dominio y de la complejidad del software.

La técnica de *Puntos de Caso de Uso* utiliza el mismo enfoque.

Métricas orientadas a la función

Con los puntos de función se pueden elaborar métricas análogas a las de KLDC:

- Errores por PF (punto de función)
- Defectos por PF
- \$ por PF
- Páginas de documentación por PF
- PF por persona-mes

Puntos de Función

Técnica de puntos de función

Esta técnica se basa **en el diseño lógico del sistema** y en características fundamentalmente “Externas” de la aplicación a desarrollar.

Los objetivos de los puntos de función son:

- **Medir independientemente de la tecnología** utilizada en la implantación del sistema.
- Proporcionar una métrica de tamaño que dé soporte al análisis de la calidad y la productividad.
- Proporcionar un medio para la estimación del software.
- **Proporcionar un factor de normalización para la comparación de distintos software.**

Puntos de Función

Elementos de la técnica:

1. **Número de Entradas** (EI, External Input).
2. **Número de Salidas** (EO, External Output).
3. **Número de Consultas** (peticiones de usuario) (EQ, External Query).
4. **Número de Archivos** (Grupos de datos lógicos) **internos** (ILF, Internal Logic File).
5. **Número de interfaces externas** (Grupos de datos lógicos externos) (EIF, External Interface File).

Puntos de Función

Entradas (EI, External Input): son datos que se introducen a la aplicación desde fuera de sus límites. Estos datos mantienen un fichero lógico interno.

Una entrada externa es considerada única si tiene un formato distinto de las demás o el diseño lógico requiere una lógica de procesamiento diferente de otra entrada externa del mismo formato.

Ejemplos:

- Las transacciones: datos introducidos para mantener ficheros lógicos internos.
- Las pantallas de entrada: hay que añadir una unidad a entradas externas por cada función que mantiene un fichero lógico interno. Por ejemplo, en un ABM, se contarían tres entradas externas.

En cambio, **no son entradas externas:**

- Los datos referenciados utilizados por la aplicación pero no mantenidos como ficheros lógicos internos.
- La entrada de una consulta.
- Las pantallas de conexión que no mantengan un fichero lógico interno.

Puntos de Función

Salidas (EO, External Output): Las salidas externas son datos o información de control que sale de los límites de la aplicación. Esta salida debe ser considerada única si tiene un formato único o si el diseño lógico requiere un proceso lógico distinto de otras salidas del mismo formato. El procesamiento de una salida debe contener al menos un cálculo o fórmula matemática o crear datos derivados. Además puede actualizar uno o más ILFs o alterar el comportamiento del sistema.

Ejemplos:

- La transferencia de datos a otras aplicaciones: datos que residen en un fichero lógico interno que son procesados para ser utilizados por otra aplicación.
- Los informes: cada informe producido por la aplicación se cuenta como una salida externa.

Dos informes que tengan el mismo formato pero los datos de origen sean distintos, ya que se generan en distintos procesos, se contarán como dos salidas.

Dos informes idénticos, producidos en diferentes soportes debido a requisitos específicos de los usuarios, se cuentan como salidas distintas.

Puntos de Función

Salidas (EO, External Output):

Ejemplos:

- Los informes *on-line* que no corresponden a la salida de una consulta, se contarán también como una salida.
- Los gráficos: cada gráfico distinto, solicitado por el usuario, debería ser contado como una salida. Así, si unos datos estadísticos se presentan en formato de tabla, diagrama de barras, y trata se contarán como tres salidas.
- Los generadores de informes: una salida desarrollada por el usuario con un generador de informes debería ser contada como una salida para cada tipo de informe especificado.

Puntos de Función

Salidas (EO, External Output):

Ejemplos:

No se deben contar como salidas:

- **Las ayudas.**
- **Las distintas formas de invocar la misma salida lógica.**
- **Los mensajes de error/confirmación asociados con tipos de función distintos de entradas externas.**
- **Por ejemplo, no se contabilizarán como salida los mensajes de error/confirmación asociados a una consulta externa.**
- **Las totalizaciones: los informes de totales no constituyen una salida.**

Puntos de Función

Consultas (EQ, External Query): representan los requisitos de información a la aplicación en una combinación única de entrada/salida que se obtiene de una búsqueda de datos, no actualiza un fichero lógico interno. Una consulta se considera única si tiene un formato distinto de otras consultas, ya sea en entrada o salida, o si el diseño lógico requiere ediciones distintas a las de otras consultas. El procesamiento de una consulta no debe contener cálculos ni fórmulas matemáticas ni crear datos derivados. Tampoco puede actualizar ningún ILF ni alterar el comportamiento del sistema.

Puntos de Función

Ejemplos de consultas son:

- La búsqueda inmediata de datos.
- Las consultas no explícitas: las pantallas de modificación/borrado que proporcionan capacidad de búsqueda de datos antes de la funcionalidad de cambio/borrado se consideran como consultas.
- Pantallas de conexión: las pantallas de logon que proporcionarían seguridad se cuentan como una consulta.
- Las ayudas: son una consulta donde la entrada y la salida (texto) son únicas.

No se consideran como consultas:

- Los mensajes de error/confirmación.
- La utilización de distintos métodos de llamada a la misma consulta.

Puntos de Función

- **Archivos internos** (ILF, Internal Logic File):
Un fichero lógico interno es un grupo de datos lógicamente relacionados, identificables por los usuarios o información de control, mantenidos y utilizados dentro de los límites de la aplicación.

Ejemplos de ILF son:

- Ficheros maestros.
- Aplicaciones de seguridad de datos.
- Datos de auditoría.
- Mensajes *help*.
- Mensajes de error.
- Datos de *back-up*
- Ficheros internos lógicos mantenidos por más de una aplicación.

Puntos de Función

Interfaces externas (EIF, External Interface File):

Representan un grupo de datos relacionados lógicamente identificables por el usuario o información de control utilizada por la aplicación, pero mantenida por otra aplicación.

Ejemplos de EIF son: archivos de datos de cinta o disco que se utilizan para transmitir información a otro sistema.

Puntos de Función

VALORACIÓN DE LA COMPLEJIDAD

Para cada uno de los parámetros externos se ha de indicar su complejidad como baja, media o alta. Para las entradas, salidas y consultas, se puede evaluar su complejidad en función del número de campos que contengan y del número de ficheros a los que hagan referencia.

Para los ficheros, por el contrario, su complejidad vendrá dada en función del número de registros y de campos que tengan.

Puntos de Función

Computación de métricas de punto de funciones					
		Factor de ponderación			
Parámetro de medición	Cuenta	Simple	Media	Compleja	Resultado
• Número de entradas del usuario	*	3	4	6	=
• Número de salidas del usuario	*	4	5	7	=
• Número de consultas del usuario	*	3	4	6	=
• Número de archivos	*	7	10	15	=
• Número de Interfaces externas	*	5	7	10	=
Cuenta total puntos de función sin ajustar					

Cálculo de Punto de Función

Calculo del TDI (*Total Degree of Influence*)

Una vez calculados el total de puntos de función sin ajustar, debe realizarse **un ajuste teniendo en cuenta las características generales** del sistema. A cada una de estas características se las pondera con un valor de 0 a 5 (0 No influencia, 1 Incidental, 2 Moderado, 3 Medio, 4 Significativo, 5 Esencial)

Características generales del sistema

1. Comunicación de datos.
2. Funciones distribuidas.
3. Rendimiento.
4. Configuraciones fuertemente utilizadas.
5. Frecuencia de transacciones.
6. Entrada *on-line* de datos.
7. Diseño para la eficiencia del usuario final.
8. Actualización *on-line*.
9. Procesos complejos.
10. Utilización en otros sistemas.
11. Facilidad de instalación.
12. Facilidad de operación.
13. Instalación de múltiples sitios.
14. Facilidad de cambio.

Cálculo de Punto de Función

Una vez calculado el grado de influencia, TDI (del inglés *Total Degree of Influence*), se calcula el valor del factor de ajuste:

$$AF = (TDI \times 0,01) + 0,65$$

El valor de los puntos de función ajustados será:

$$FPA = FP \times AF$$

Computación de Punto de Función

- ¿Cómo estimar el esfuerzo a partir de los PFA ?

Esfuerzo(HS) = PFA * Promedio_Organización(Lenguaje)

Esfuerzo(HS) = PFA * DatosEstadisticos

<http://www.isbsg.org/>

<http://www.softwaremetrics.com>

Conclusión de Punto de Función

Size	Hours/FP
50	1.3
100	1.4
500	1.6
1,000	2.0
2,000	2.7
3,000	3.6
4,000	4.9
5,000	6.6
6,000	9.0
7,000	12.1
8,000	16.3
9,000	22.1
10,000	29.8
11,000	40.2
12,000	54.3
13,000	73.3
14,000	98.9
15,000	133.6



Algunas organizaciones utilizan un valor de HS/FP en base a estadísticas de acuerdo al tamaño del sistema obtenido por el calculo de los FP.

Se podría asumir que la media general de todos los proyectos está en 11,35 horas-hombre por punto-función.

<http://www.softwaremetrics.com>

Medida de Calidad del Software

- **Corrección:** La corrección es el grado en el que el software lleva a cabo su función, la medida más común de corrección son los defectos por KLDC
- **Facilidad de mantenimiento:** es la facilidad con la que se puede corregir un programa si se encuentra un error o un cambio solicitado por el usuario, para ello se deben utilizar medidas indirectas, como el tiempo medio de cambio (TMC)
- **Integridad:** Este atributo mide la habilidad de un sistema para resistir ataques (tanto accidentales como intencionados). La integridad del sistema se puede definir como: $\text{integridad} = \text{Sumatoria } [1 - \text{amenaza} * (1 - \text{seguridad})]$
- **Facilidad de uso:** es un intento de cuantificar «lo amigable que puede ser con el usuario» y se puede medir en función de:
 - habilidad intelectual y/o física requerida para aprender el sistema
 - el tiempo requerido para llegar a ser moderadamente eficiente en el uso del sistema
 - aumento neto en productividad
 - valoración subjetiva de la disposición de los usuarios hacia el sistema.

Métrica de Calidad

Una métrica de la calidad que proporciona beneficios tanto a nivel del proyecto como del proceso, es la eficacia de la eliminación de defectos (EED).

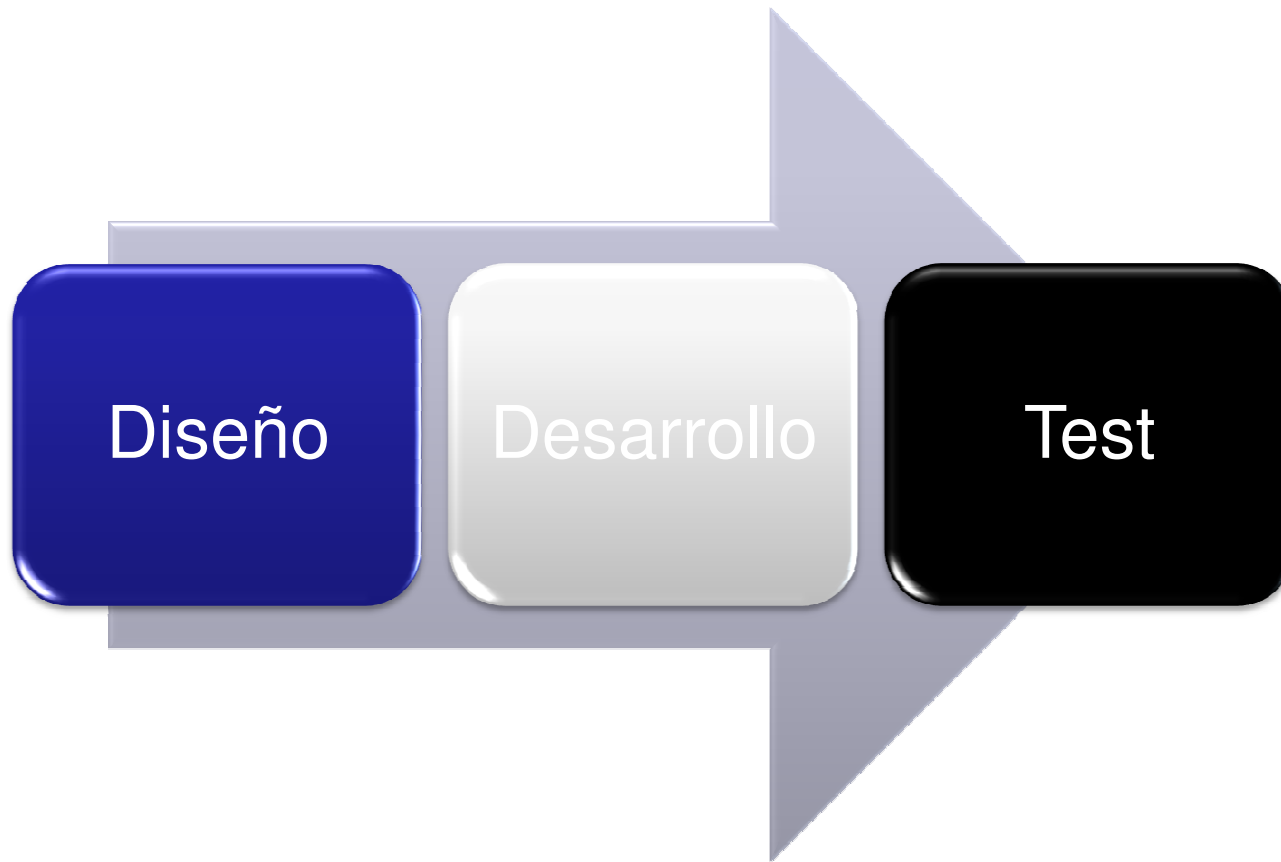
Se define como: **$EED = E / (E + D)$**

E = número de errores encontrados antes de la entrega del software al usuario final

D = número de defectos encontrados después de la entrega.

El valor ideal de EED ES 1

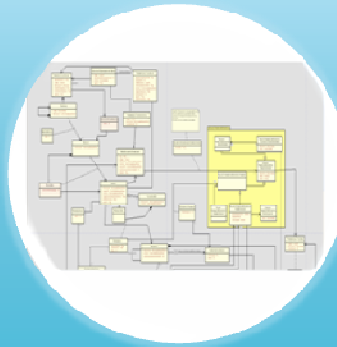
Etapas a estimar



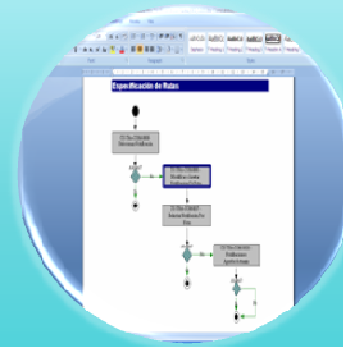
Requerimientos para estimar



Casos de
uso



Modelo
de datos



Definición
de
procesos



Herramientas de Estimación

- **Juicio de Expertos**
- **Estimación Análoga (Top Down)** : de arriba hacia abajo)
Basada en datos históricos de proyectos anteriores similares
- **Estimación Detallada (Bottom Up)**: de abajo hacia arriba)
Trasladan hacia arriba los estimados individuales para obtener el total del proyecto.
- **Estimación Paramétrica**: Definidas por una fórmula en función del esfuerzo por unidad de trabajo
- **Estimación por Tres Valores**

Herramientas de Estimación

- **Estimación Análoga (Top Down)**
- **Ventajas**
 - Rápida y de menor costo
 - Las actividades no necesitan estar desagregadas
 - Proveen idea a nivel gerencial
- **Desventajas**
 - Poco precisa
 - Se preparan con conocimiento limitado del proyecto
 - Tiene que ser desarrollada por personal experimentado
 - No toma en cuenta la diferencia entre proyectos

Herramientas de Estimación

- **Estimación Detallada (Bottom Up)**
- **Ventajas**
 - Mayor precisión que la ascendente
 - Basada en un análisis detallado del proyecto
 - Provee una base para el monitoreo, control , medición del rendimiento y gestión del proyecto
- **Desventajas**
 - Tiempo y costo para desarrollarla
 - Tendencia a generar “colchones”
 - El proyecto debe estar claramente definido antes de realizar la estimación

Herramientas de Estimación

- **Estimación Parametrica**

- Muy utilizado cuando se están implementando proyectos que son similares unos a otros, proyectos con la misma tecnología, o proyectos en áreas funcionales similares.
- Consiste en detectar variables clave del proyecto, indicadores, parámetros, que son los principales determinantes del tamaño del proyecto
- Utiliza una relación estadística entre los datos históricos y otras variables según el proyecto
- Pueden lograrse niveles superiores de exactitud dependiendo de la calidad de los datos históricos

Herramientas de Estimación

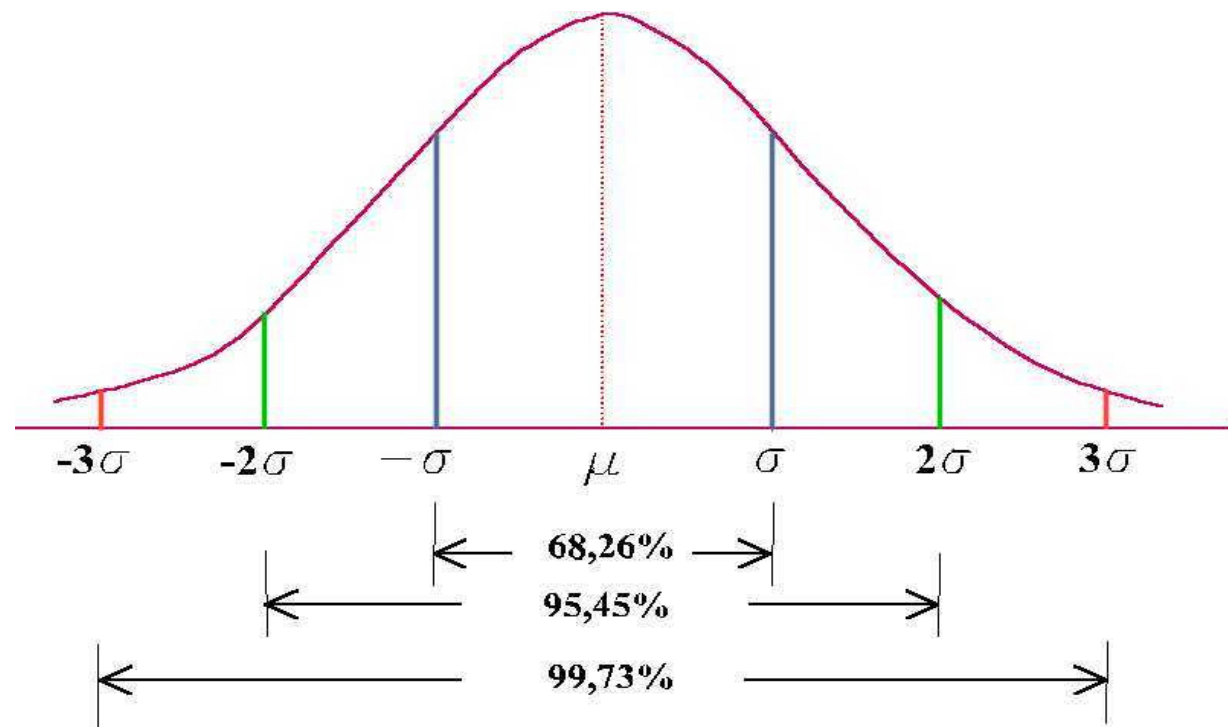
- **Estimación a 3 Valores (PERT)**
- Se utiliza el promedio ponderado de estimaciones para calcular la duración **de una actividad**.
- Se base en 3 valores
 - **Tm**: valor mas probable
 - **To**: valor optimista
 - **Tp**: valor pesimista
- Tiempo esperado
$$\mathbf{Te = (To + 4Tm + Tp) / 6}$$

Herramientas de Estimación

- **Estimación a 3 Valores (PERT)**

- Desvío estándar

$$\sigma = \frac{T_p - T_o}{6}$$



Herramientas de Estimación

- **Estimación a 3 Valores (PERT)**

- **Ejemplo**

TP= 14 Tm=11 To=2

Te=10 Desvío estándar=2

¿Qué significa esto?

- Hay un 68,26% de probabilidades que la tarea dure entre 8 y 12
- Hay un 94,45% de probabilidades que la tarea dure entre 6 y 14
- Hay un 99,73% de probabilidades que la tarea dure entre 4 y 16

Herramientas de Estimación

- **Estimación de un proyecto con PERT**
- Las tareas a considerar son las del camino critico
- La duración esperada del proyecto es la suma de la duración esperada de las tareas del camino critico
- Para el calculo del desvío estándar NO se debe sumar los desvíos de las tareas

$$DESVIÓ ESTÁNDAR PROYECTO = \sqrt{\sum \left(\frac{(P - O)^2}{6} \right)}$$

Validación de la estimación

- Experiencia
- Analogía
- Puntos de Función
- Registros históricos

Bibliografía

- *Software Engineering, A Practitioner's Approach, 5th Edition*, Pressman, Roger
- *Function Point Counting Practices Manual, Release 4.2.1*, International Function Point Users Group