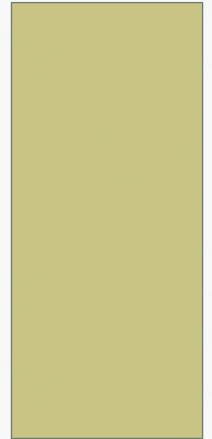


# INGENIERÍA DEL SOFTWARE

UNIDAD 2: MEDIDAS, MÉTRICAS E INDICADORES DEL PROCESO  
CICLO LECTIVO 2013



# OBJETIVOS DE LA CLASE

- **Mediciones del software.**
- **¿Por qué medir?**
- **Medidas, Mediciones, Indicadores y Métricas**
- **¿Qué medir?**

# TEMAS PARA LA PRÓXIMA CLASE

- LDC: Que y como mide. Software para medir. Ventajas y desventajas.
- Métricas orientadas a Casos de Uso: Que y como mide. Ventajas y Desventajas.

# INTRODUCCIÓN

- La medición es fundamental en cualquier ámbito de la ingeniería, y la IS no es una excepción.
- La medición nos permite tener una visión mas objetiva, y profunda proporcionando información para una evaluación fundamentada.
- Cuando se puede medir, y pasar a números lo que se esta haciendo, se conoce algo mas. Cuando no se puede medir, el conocimiento es precario y deficiente.

# MÉTRICAS Y MEDICIONES

- Las métricas del software se refieren a un amplio elenco de mediciones para el software de computadora.
- La medición se puede aplicar al proceso del software con el intento de mejorarlo sobre una base continua.
- Se puede utilizar en el proyecto del software para ayudar en la estimación, el control de calidad, la evaluación de productividad y el control de proyectos.
- Finalmente, el ingeniero de software puede utilizar la medición para ayudar a evaluar la calidad de los resultados de trabajos técnicos y para ayudar en la toma de decisiones táctica a medida que el proyecto evoluciona

# RAZONES PARA MEDIR

- Hay cuatro razones para medir los procesos del software, los productos y los recursos:
  - caracterizar
  - evaluar
  - predecir
  - mejorar

# MEDIR PARA ...

- Caracterizamos para comprender mejor los procesos, los productos, los recursos y los entornos y para establecer las líneas base para las comparaciones con evaluaciones futuras.
- Evaluamos para determinar el estado con respecto al diseño. Las medidas utilizadas son los sensores que nos permiten conocer cuándo nuestros proyectos y nuestros procesos están perdiendo la pista, de modo que podamos ponerlos bajo control. También evaluamos para valorar la consecución de los objetivos de calidad y para evaluar el impacto de la tecnología y las mejoras del proceso en los productos y procesos.

# MEDIR PARA ...

- Predecimos para poder planificar. Realizar mediciones para la predicción implica aumentar la comprensión de las relaciones entre los procesos y los productos y la construcción de modelos de estas relaciones. Hacemos esto porque queremos establecer objetivos alcanzables para el coste, planificación, y calidad de manera que se puedan aplicar los recursos apropiados. Evaluamos para determinar el estado con respecto al diseño. Las medidas utilizadas son los sensores que nos permiten conocer cuándo nuestros proyectos y nuestros procesos están perdiendo la pista, de modo que podamos ponerlos bajo control. También evaluamos para valorar la consecución de los objetivos de calidad y para evaluar el impacto de la tecnología y las mejoras del proceso en los productos y procesos.



# MEDIR PARA ...

- Medimos para mejorar cuando recogemos la información cuantitativa que nos ayuda a identificar obstáculos, problemas de raíz, ineficiencias y otras oportunidades para mejorar la calidad del producto y el rendimiento del proceso.

# DEFINICIONES

*No todo lo que se puede contar cuenta, ni todo lo que cuenta se puede contar. A. Einstein*

- Dentro del contexto de la ingeniería del software, una medida proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto. La medición es el acto de determinar una medida. El IEEE Standard Glossary of Software Engineering Terms [IEEE93] define métrica como «una **medida** cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado».

# DEFINICIONES

- Una **métrica** del software relata de alguna forma las medidas individuales sobre algún aspecto (por ejemplo: el número medio de errores encontrados por revisión o el número medio de errores encontrados por persona y hora en revisiones').
- Un **indicador** es una métrica o una combinación de métricas que proporcionan una visión profunda del proceso del software, del proyecto de software o del producto en sí.

# ¿QUÉ MEDIR?

- La medición es algo común en el mundo de la ingeniería. Se mide el consumo de energía, el peso, las dimensiones físicas, la temperatura, el voltaje, la relación señal-ruido... la lista es casi interminable.
- La medición es mucho menos común en el mundo de la ingeniería del software. Existen problemas para ponerse de acuerdo sobre qué medir y las medidas de evaluación de problemas recopilados.

# INDICADORES DEL PROYECTO

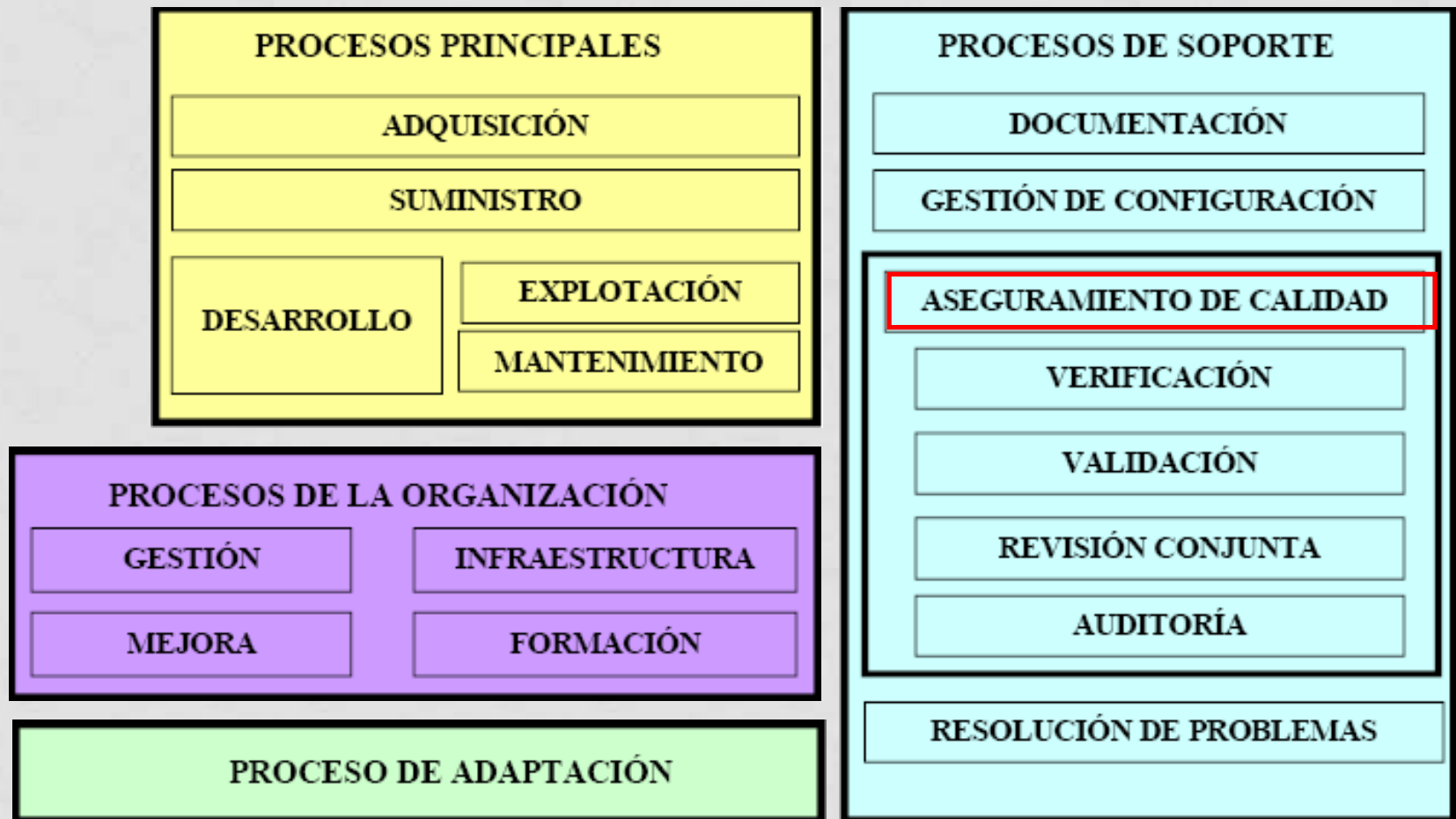
Los indicadores de proyecto permiten al gestor de proyectos del software:

- evaluar el estado del proyecto en curso;
- seguir la pista de los riesgos potenciales;
- detectar las áreas de problemas 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 del software.

# CALIDAD DEL SOFTWARE

- El **Aseguramiento de Calidad (AC)** del software es el conjunto de actividades planificadas y sistemáticas necesarias para controlar que el **producto** (proceso,...) satisface los requisitos de **Calidad**.
- El AC debe estar presente en:
  - Métodos y herramientas de Análisis, Diseño, Programación y Testeo
  - Control de la Documentación
  - Registro de Auditorías e Informes
  - Métodos para Medición y Evaluación

# ESTÁNDAR ISO/IEC 12207



# CALIDAD DEL SOFTWARE

- **Medición de software:** necesidad de obtener datos cuantitativos que ayuden a mejorar la calidad.
  - A partir de **Métricas**
- **Creación de Modelos de Calidad:** útiles para discutir, planificar y obtener indicadores de calidad.
- **Aplicación de Estándares de Calidad:** directrices (prescripciones) para el aseguramiento externo e interno de la calidad.
  - ISO 9126, etc.



# MEDICIÓN DE SOFTWARE (PRESSMAN)

- La medición se utiliza a lo largo de un proyecto de software como apoyo en:
  - La estimación,
  - El control de calidad,
  - La valoración de la productividad,
  - El control del proyecto.
- La medición es aplicada por los ingenieros de software para auxiliar la evaluación de la calidad de los productos de trabajo y para apoyar la toma de decisiones

# EJEMPLOS DE MÉTRICAS, INDICADORES, MEDIDAS

- **Medida:**
  - un programa tiene 10.000 LDC (líneas de código).
- **Métrica:**
  - la productividad del proyecto X fue de 250 (LDC/persona-mes).
- **Indicador:**
  - la productividad media de nuestra empresa es de 500(LDC/pm) y en el último proyecto ha sido de 250(LDC/pm).

Productividad  $\geq$  500(LDC/pm)  $\rightarrow$  OK

Productividad  $<$  500(LDC/pm)  $\rightarrow$  ¿?

# A PENSAR....

- Sugiera tres medidas, tres métricas y los indicadores que se podrían utilizar para evaluar un automóvil.

# MEDICIONES DE SOFTWARE

Mejora objetiva: midiendo atributos.

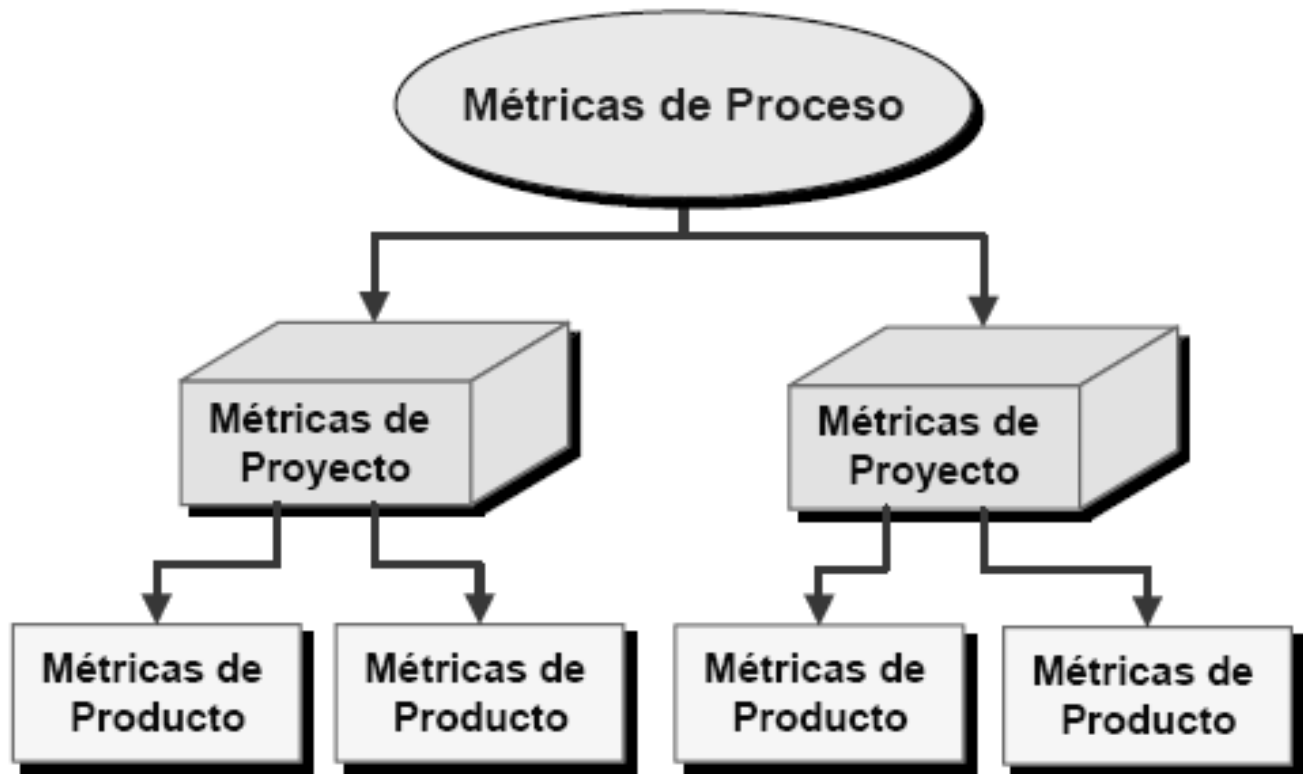
- Identificar atributos.
- Desarrollar métricas para esos atributos.
- Utilizar las métricas para obtener indicadores.
- Planificar e implementar el proceso de mejora.

# MÉTRICAS: BENEFICIOS

- Comunicación efectiva basada en hechos y datos y no en opiniones o cuestiones subjetivas.
- Identificación temprana de problemas y riesgos.
- Seguimiento y control de los objetivos.
- Soporte a las decisiones.

❑ Se debe garantizar el buen uso de las métricas, de lo contrario, pueden ocasionar más problemas que beneficios. ¿Por qué? Pensemos...

# MÉTRICAS: SUS DIMENSIONES

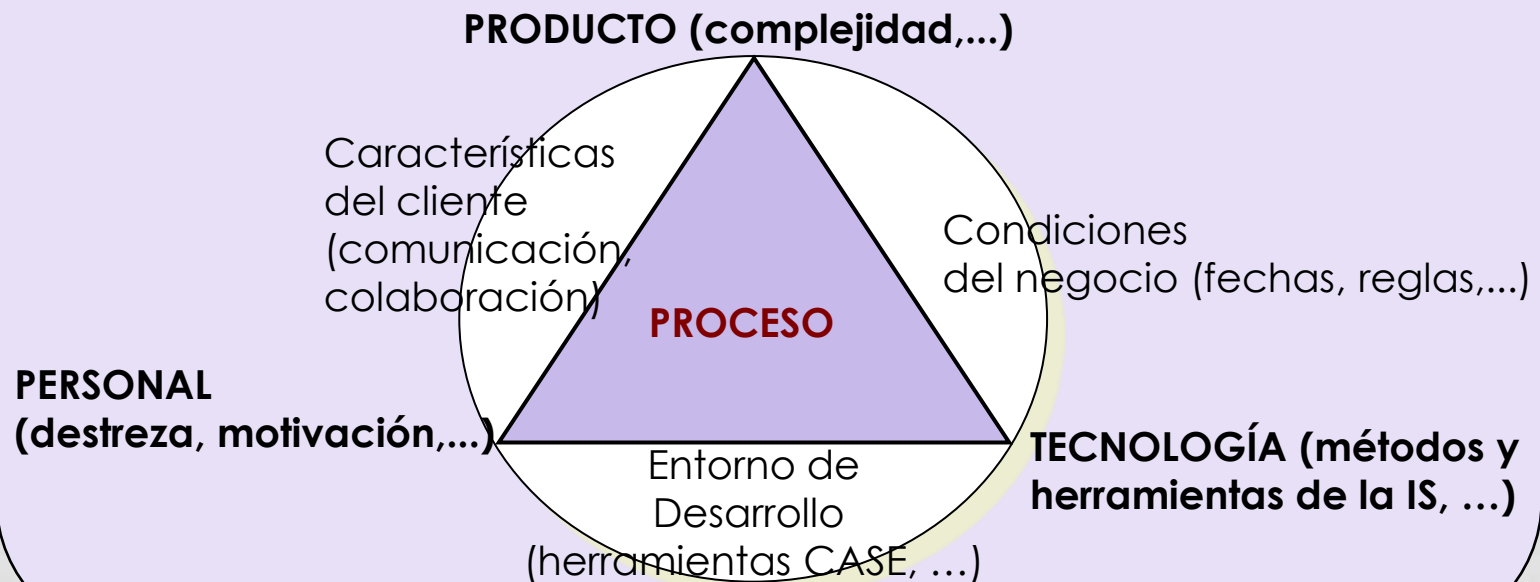


# MÉTRICAS: DIMENSIONES [SEI, 1997]

- Métricas del Proceso:
  - Para evaluar si los procesos de la organización se están llevando adelante según lo esperado.
  - Para asegurar que los procesos definidos están siendo considerados.
  - Para implementar mejoras de manera de lograr los objetivos de negocio (por ej., minimizar riesgos y generar productos de calidad).
  - ¿Ejemplo?

# MÉTRICAS DEL PROCESO [PRESSMAN, 2005]

## DETERMINANTES DE LA CALIDAD DEL SOFTWARE Y DEL DESEMPEÑO DE LA ORGANIZACIÓN





# MEDICIONES DEL PROCESO

- Única forma racional de mejorar procesos: medir sus atributos específicos, desarrollando y empleando métricas.
- Métricas del Proceso:
  - Proporcionan *Indicadores de Proceso*.
  - Se recopilan de varios proyectos.
  - Se utilizan a largo plazo.
  - Permiten tener visión de la eficacia de un proceso existente (actividades, productos de trabajo, hitos, esfuerzo, etc.).

# MÉTRICAS DEL PROCESO

- Las métricas del proceso pueden ser muy útiles, pero hay que saber interpretarlas.
- Normas básicas de interpretación:
  - Utilizar el sentido común al interpretar los datos.
  - Proporcionar una realimentación regular a personas y equipos.
  - No utilizar métricas para evaluar a personas.
  - Establecer métricas claras y objetivos para alcanzarlas.
  - No utilizar métricas para amenazar a particulares o equipos.
  - Si una métrica identifica un área problemática no se debería considerar como negativa.
  - Hay que interpretar todas las métricas en su conjunto, y no hacer prevalecer una en particular.

# MÉTRICAS: DIMENSIONES [SEI, 1997]

- Métricas del Proyecto.
  - Planificar objetivamente.
  - Conocer el estado y el progreso de los productos.
  - Se focalizan en cuestiones clave que comprometen el proyecto (tiempo, tamaño, esfuerzo, costo, calidad, estado).
  - Se utilizan para adaptar el flujo de trabajo del proyecto y las actividades técnicas.

# MEDICIONES DEL PROYECTO

- Métricas del Proyecto → Indicadores del Proyecto:
  - Evaluar estado del proyecto actual.
  - Analizar riesgos potenciales.
  - Detectar áreas o cuestiones problemáticas.
  - Ajustar flujos y tareas de trabajo.
  - Evaluar habilidad del equipo en mantener bajo control la calidad de los productos.

# MÉTRICAS DEL PROYECTO

- Primera aplicación de las métricas del proyecto → ¿Cuándo?
- A medida que avanza el proyecto, las medidas del esfuerzo y el tiempo se comparan con las de la planificación.
- El líder de proyecto utiliza estos datos para supervisar y controlar el avance.
- Ejemplos:
  - tiempo y esfuerzo medios de corrección de errores,
  - errores detectados antes de la entrega del software,
  - defectos detectados e informados por los usuarios finales, productos de trabajo entregados, etc.
  - ...

# MÉTRICAS DEL PROYECTO

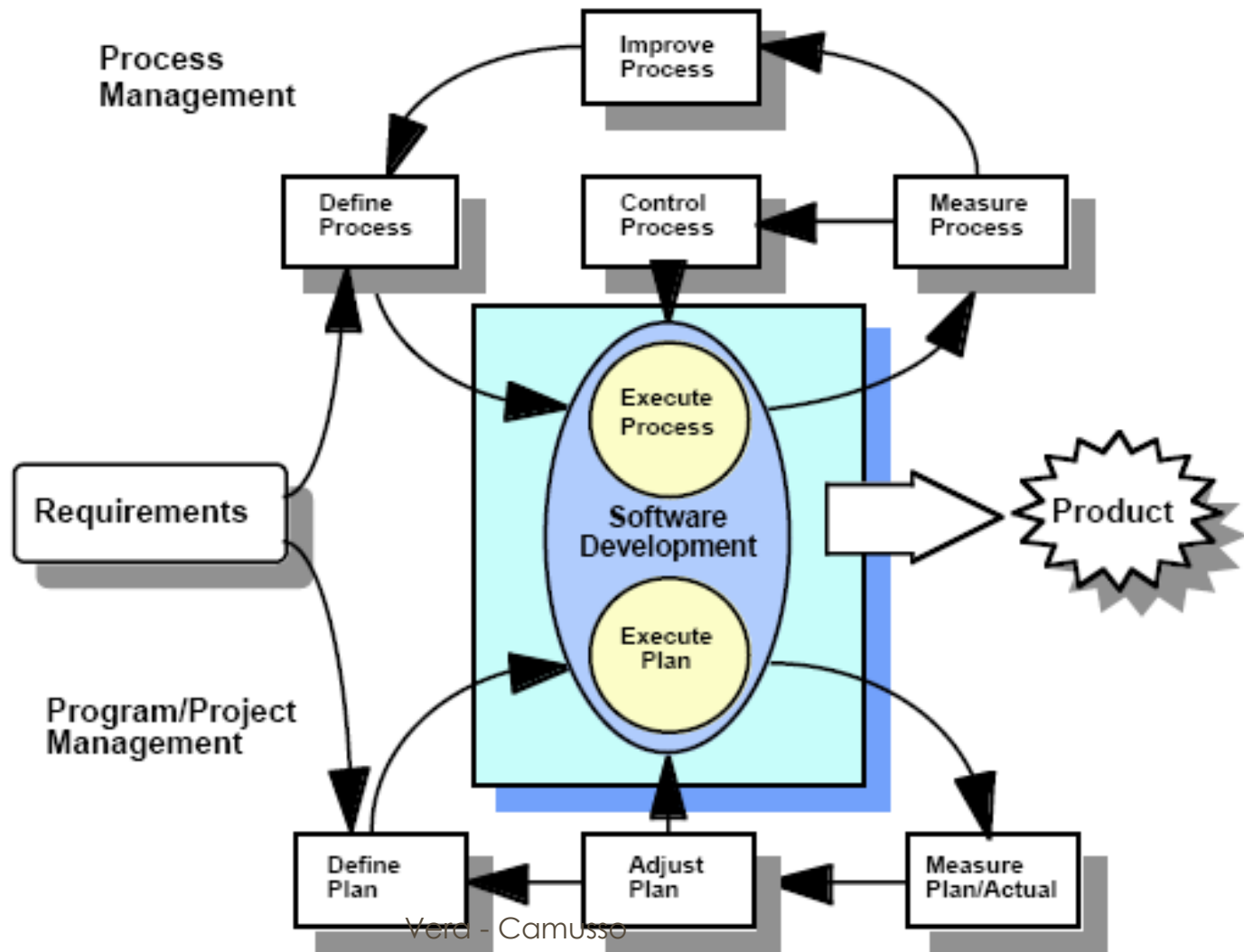
➔ La utilización fundamental de las métricas del proyecto son dos:

- Optimizar la planificación, guiando los ajustes necesarios que eviten retrasos y mitiguen problemas y riesgos potenciales.
- Evaluar la calidad de los productos en el momento actual, modificando el enfoque técnico para mejorar la calidad, si es necesario.

# MÉTRICAS: DIMENSIONES [SEI, 1997]

- Métricas del Producto.
  - Asegurar la aceptación del cliente.
  - Cuestiones de mayor peso: principalmente, las relativas a los atributos físicos y dinámicos del producto (arquitectura, productividad, confiabilidad, usabilidad, estabilidad, performance, etc.)

## MEDICIONES DEL PROCESO Y PROYECTO [SEI, 1997]





# MÉTRICAS DEL PROCESO VS. MÉTRICAS DEL PROYECTO

- Técnicamente no existe gran diferencia entre las métricas del proyecto y del proceso.
- Las métricas del proceso pueden concebirse como recopilaciones de métricas del proyecto.
- Las métricas del proceso son *estratégicas*: determinan el curso del *proceso de desarrollo de software*.
- Las métricas del proyecto son *tácticas*: determinan el curso del *proyecto actual*.

➔ Muchas métricas se utilizan en ambos dominios.

# MÉTRICAS DE PROCESO Y PROYECTO

- Las métricas de proyecto se consolidan con el fin de crear métricas del proceso que sean públicas para la organización como un todo.
- Ejemplo:
  - Un equipo A encontró 342 errores durante el proceso de software. El equipo B encontró 184 errores. Si todas las demás medidas se mantienen iguales, ¿qué equipo es más eficiente?
    - ➔ No tenemos información sobre el tamaño ni la complejidad de los proyectos.
    - ➔ Es necesario normalizar medidas para poder comparar.

# MÉTRICAS ORIENTADAS AL TAMAÑO

Proyecto	LDC	Esfuerzo	Coste £(000)	Pag. Doc.	Errores	Defectos	Personas
Alfa	12,100	24	168	365	134	29	3
Beta	27,200	62	440	1224	321	86	5
Gamma	20,200	43	314	1050	256	64	6

- Líneas de código (LDC) → valor de normalización.
- Métricas simples orientadas al tamaño:
  - Errores por KLDC, Defectos por KLDC, Costo por KLDC, Páginas de documentación por KLDC....
- Otras:
  - Errores por persona/mes, KLDC por persona/mes, costo por página de documentación, ...

# MÉTRICAS ORIENTADAS A LA FUNCIÓN

- Valor de normalización: funcionalidad entregada por la aplicación.
- Métrica más utilizada: Punto de Función (PF).
- PF se calcula en base a características del dominio y de la complejidad esperada del software.
- PF es independiente del lenguaje de aplicación.
- PF puede conocerse en etapas tempranas del proyecto.

# MÉTRICAS ORIENTADAS AL TAMAÑO Y A LA FUNCIÓN

- Las métricas basadas en PF y en LDC son indicadoras del esfuerzo y costo del desarrollo de software.
- Existe relación entre ambas medidas.
- La estimación de la funcionalidad de la aplicación está relacionada con las LDC y con el esfuerzo de desarrollo necesario.
- Ejemplos:
  - C: 162 LDC por PF.
  - C++: 66 LDC por PF.
  - Java: 63 LDC por PF.
  - VisualBasic: 47 LDC por PF.
  - FoxPro: 32 LDC por PF.

# MÉTRICAS ORIENTADAS A OBJETOS

- *Nº de guiones de escenario*: directamente relacionado con el tamaño de la aplicación y con el número de casos de prueba necesarios.
- *Nº de clases clave*: “componentes independientes”, centrales en el dominio del problema. Indican la cantidad de esfuerzo necesario para desarrollar el software.
- *Nº de clases de apoyo*: no están relacionadas con el dominio. Son necesarias en la implementación.
- *Nº de subsistemas*: agregados de clase que apoyan una función visible para el usuario final.

# MÉTRICAS ORIENTADAS A CASOS DE USO

- El CU se define en etapas tempranas del proceso → útil en estimación.
- Describen funciones y características visibles al usuario que son requerimientos básicos.
- Es independiente del lenguaje de programación.
- Es directamente proporcional al tamaño de la aplicación en LDC y al nro. de casos de prueba necesarios para probar completamente la aplicación.
- Problema: grado de abstracción → la importancia como medida de normalización es sospechosa.

# MÉTRICAS DE PROYECTOS DE INGENIERÍA WEB

- Proyectos web: énfasis en contenido + funcionalidad.
- Métricas para evaluación de complejidad relativa y esfuerzo de desarrollo:
  - N° de páginas web estáticas.
  - N° de páginas web dinámicas.
  - N° de vínculos internos de páginas.
  - N° de objetos de datos persistentes.
  - N° de sistemas externos.
  - N° de objetos de contenido estático y dinámico.
  - N° de funciones ejecutables.
- Índice de personalización:  $C = N_{pd} / (N_{pd} + N_{pe})$



# ¿POR QUÉ MEDIR?

- Si no se mide, no existe forma real de determinar si se están haciendo bien las cosas (proyecto) o si se está mejorando (proceso).
  - Sin embargo, la mayoría de los desarrolladores de software todavía no miden.
- ➔ Medición: sirve para establecer una línea base a partir de la cual se evalúan mejoras.**

# OBJETIVOS DE LA CLASE

- ✓ Mediciones del software.
- ✓ ¿Por qué medir?
- ✓ Medidas, Mediciones, Indicadores y Métricas
- ✓ ¿Qué medir?

¿Dudas, consultas?

