



La medición de intangibles, de los procesos y de los servicios: Caso específico del software y de sus servicios asociados



El siguiente documento ha sido elaborado por Juan Carlos Torres, Director de Calidad y Organización de Avalon Tecnologías de la Información S.L, como vocal de la comisión de ingeniería de calidad de la Comunidad AEC Calidad.

Este Foro de relación trabaja para posicionar la Calidad y a sus profesionales como valor central de las organizaciones como un motor de transformación. Entre sus objetivos, la Comunidad se ha marcado la promoción y divulgación de la calidad como un elemento clave para la mejora de la competitividad de las empresas y los profesionales.

Para ello, fomenta el intercambio de conocimientos y buenas prácticas, a través de sus reuniones, eventos, comisiones de trabajo y elaboración de documentos e informes, como éste.

La AEC promueve la difusión de los trabajos elaborados por sus Comunidades y Comités, respetando al máximo su independencia y el criterio de sus miembros, sin tener que compartir en todo momento sus opiniones. Asociación Española para la Calidad (AEC).

Diciembre 2020

01 | **MEDIDA DEL SOFTWARE**

02 | **MEDICIÓN DEL PROCESO SOFTWARE**

03 | **METODOLOGÍA PARA DETERMINAR MÉTRICAS: GOAL-QUESTION-METRIC GQM**

04 | **MEDICIÓN DE UN SERVICIO DE MANTENIMIENTO DE SOFTWARE**

01 | MEDIDA DEL SOFTWARE

¿Por qué debemos preocuparnos del tamaño del software?

El objetivo principal del tamaño del software es la estimación del proyecto/servicio que lo desarrolla/mantiene.

El desconocimiento del tamaño del software conduce a malas estimaciones en los proyectos que lo desarrollan o los servicios que lo mantienen.

Sin el tamaño del software es difícil estimar:

- Cuánto durará el proyecto que lo desarrolla
- Cuánto costará el software
- Cuántos recursos serán necesarios
- Cuántos errores se pueden esperar durante la actividad de prueba del software
- Cuál será la productividad probable durante el desarrollo del software o el mantenimiento del mismo.

Y todo esto porque existe una relación no lineal entre el tamaño del software y la duración, el esfuerzo y los errores.

La estimación del tamaño del software no es fácil. Las personas que estiman necesitan usar diferentes métodos de cálculo del tamaño dependiendo del momento en que se encuentra el proyecto en su ciclo de vida y de la información de la que se dispone.

Etapas del ciclo de vida del software

De manera genérica podemos dividir el ciclo de vida del software en cuatro etapas:

- Qué
- Cómo
- Hacer
- Desplegar/arreglar

Las metodologías de desarrollo de software tienen sus propios nombres para las etapas básicas:

METODOLOGÍA	QUÉ	CÓMO	HACER	DESPLEGAR/ ARREGLAR
CASCADA (clásica)	Concepto	Requisitos y Diseño	Construcción y Pruebas	Despliegue
AGILE	Iniciación	Iteración de planificación	Iteración de desarrollo	Producción
PAQUETE COMERCIAL	Preparación del proyecto	Plan de negocio	Realización y preparación final	Puesta en marcha

En cada etapa somos capaces de estimar el tamaño con nuevos métodos y con mayor claridad, por eso podemos referir al proceso de cálculo del tamaño del software como una **elaboración progresiva**.

Como estaremos usando diferentes métodos de cálculo del tamaño conjuntamente, es necesaria una forma de normalizar sus resultados o convertirlos a una unidad de medida común. Esto es importante porque permite comparaciones entre diferentes tecnologías, proyectos, industrias y organizaciones.

Tipos de tamaño del software

Existen dos tipos principales de tamaño de software: **Tamaño funcional** y **Tamaño técnico**.

- **Tamaño funcional:** es la cantidad de funcionalidades del software que se entregan al usuario final del mismo. **EL USUARIO SOLO SE PREOCUPA DEL TAMAÑO FUNCIONAL.**
- **Tamaño técnico:** es la cantidad de elementos técnicos del software que llevan a cabo las funcionalidades del mismo. **LOS DESARROLLADORES DE SOFTWARE SE PREOCUPAN DEL TAMAÑO TÉCNICO.**

Bajo el punto de vista de desarrollo de software eficaz y eficiente, se debe **MAXIMIZAR EL TAMAÑO FUNCIONAL** a la vez que se **MINIMIZA EL TAMAÑO TÉCNICO**.

Para el tamaño funcional podemos convertir todas nuestras medidas obtenidas en cada método distinto de cálculo de tamaño en unidades base llamadas **PUNTOS FUNCIÓN**, método que es ampliamente usado en los estándares ISO para determinar el tamaño funcional.

Para el tamaño técnico podemos convertir todas nuestras medidas en unidades base llamadas **UNIDADES DE IMPLEMENTACIÓN**. Una unidad de implementación es equivalente a escribir una línea de código o a un paso técnico en la configuración de un paquete de software comercial.

Finalmente los puntos función pueden ser convertidos en unidades de implementación utilizando técnicas de retro disparo en las que se indica cuantas unidades de implementación se necesitan para obtener un punto función, estadísticamente hablando.

De esta manera si se estima en las etapas iniciales el tamaño funcional del software en puntos función, el tamaño funcional final del software desarrollado se puede calcular calculando el tamaño técnico del software final en unidades de implementación y aplicando sobre éstas el retro disparo.

Unidades de medida de tamaño

Las unidades de tamaño más comunes por etapa son:

- **QUÉ:** orden de magnitud del tamaño del software mediante analogía con proyectos similares.
- **CÓMO, HACER, DESPLEGAR/ARREGLAR: TAMAÑO FUNCIONAL** (normalizado a **PUNTOS FUNCIÓN** y luego a **UNIDADES DE IMPLEMENTACIÓN**):

o Requisitos funcionales también conocidos como capacidades funcionales: Son “declaraciones” comprobables que describen la intención de las funciones de software y a menudo mantenidas en una Matriz de trazabilidad de requisitos. Requieren una definición consistente. Es el método más utilizado en el desarrollo tradicional de software.

o Requisitos de negocio: Son un mayor nivel de abstracción que comprende múltiples requerimientos funcionales; pueden usarse para hacer una estimación aproximada asumiendo un número dado de requisitos funcionales por requisito de negocio. Es el método más utilizado en el desarrollo tradicional de software.

o Puntos de historia de usuario: Las historias de usuario son la formalización de las conversaciones entre desarrolladores y usuarios: equivalen a un escenario (es decir, hilo de funcionalidad) en el software que satisfará un objetivo del usuario; requiere una definición consistente. Están asociados con las metodologías ágiles. Existe un método formal para el cálculo de los Puntos de historia de usuario.

o Puntos de casos de uso: Los casos de uso son contenedores de múltiples escenarios (es decir, historias de usuarios) que describen las interacciones entre un humano (o sistema externo) y el software para lograr un objetivo común de usuario. Pueden usarse para hacer una estimación aproximada, pero requieren una definición consistente; Se usan tanto en proyectos tradicionales como en proyectos ágiles. Existe un método formal para el cálculo de los Puntos de caso de uso.

o Puntos función (IFPUG, MARK II, COSMIC, FISMA, NESMA): Son los métodos más robustos pero que requieren mucho tiempo para medir el tamaño funcional; pueden usarse para estimar en etapas tempranas, pero solo se pueden contar de manera fiable después de establecer los requisitos. Estos métodos están descritos en las correspondientes normas ISO (ver el apartado REFERENCIAS Y LECTURAS COMPLEMENTARIAS).

- **HACER, DESPLEGAR/ARREGLAR: TAMAÑO TÉCNICO** (normalizado a UNIDADES DE IMPLEMENTACIÓN)

o Objetos RICE (Reports, Interfaces, Conversions, y Enhancements Extensions) y Configuraciones de procesos de negocio: Se emplean en las configuraciones de paquetes comerciales de software. Trata la cantidad de configuraciones de procesos de negocio de alto nivel y detallados. El tamaño de las adaptaciones del paquete comercial se calcula contando el número de informes, interfaces, conversiones y extensiones. Este método requiere que los componentes estén consistentemente definidos.

o **Componentes técnicos:** Cuenta las pantallas, informes, formularios, tablas, módulos, etc. Este método requiere que los componentes estén consistentemente definidos.

o **Ficheros de código fuente:** Es una estimación aproximada de la cantidad de líneas de código fuente (SLOC), asumiendo una cantidad media de SLOC por cada fichero de código.

o **Líneas de código fuente SLOC:** El número de sentencias lógicas entregadas al usuario y es específico para el tipo de proyecto y el lenguaje de programación. Solo es medible después de la codificación.

o **Otras medidas del software cuyo objetivo no es la estimación.** Además de las medidas de tamaño técnico que se han descrito anteriormente, existen otros tipos de medidas cuyo objetivo es verificar el grado de cumplimiento de ciertos requisitos no funcionales que se le exige al software, tales como: Mantenibilidad, Usabilidad, Seguridad, Estandarización. Ejemplos de estas medidas, que son tomadas una vez construido el software con herramientas software construidas para tal fin, son:

OBJETIVO DE LA MEDIDA	MEDIDA
Comprobar el grado de mantenibilidad del software	<ul style="list-style-type: none"> • Complejidad ciclomática • Deuda Técnica • Código duplicado • Complejidad • Índice de cohesión • Índice de acoplamiento
Comprobar el grado de seguridad del software	<ul style="list-style-type: none"> • Quebrantamiento de normas de seguridad • Cantidad de vulnerabilidades
Comprobar el grado de estandarización del software	<ul style="list-style-type: none"> • Incumplimientos de buenas prácticas de programación

Usabilidad del software

- Cumplimiento de las normas W3C

Arquitectura: eficiencia del diseño

- Número de clases que hacen uso de la clase que estamos estudiando (fan-in)
- Número de clases que usa la clase que estamos estudiando (fan-out)

NOTA: Estas dos medidas también pueden ser tomada en la etapa de CÓMO

CUADRO RESUMEN DE MEDIDAS POR ETAPA DEL CICLO DE VIDA

FASE	QUÉ	CÓMO	HACER	DESPLEGAR/ ARREGLAR
UNIDAD DE MEDIDA	<ul style="list-style-type: none"> • Orden de magnitud 	<ul style="list-style-type: none"> • Requisitos funcionales • Requisitos de Negocio • Historias de usuario • Casos de usos • Puntos función (Estándares de ISO) 	<ul style="list-style-type: none"> • Requisitos funcionales • Requisitos de Negocio • Puntos de historia de usuario • Puntos de casos de usos • Puntos función (Estándares de ISO) • Objetos RICE • Componentes Técnicos • Ficheros de código fuente • Líneas de código fuente 	<ul style="list-style-type: none"> • Requisitos funcionales • Requisitos de Negocio • Puntos de historia de usuario • Puntos de casos de usos • Puntos función (Estándares de ISO) • Objetos RICE • Componentes Técnicos • Ficheros de código fuente • Líneas de código fuente

TIPO

Requisitos de alto nivel

Requisitos refinados

Medidas físicas

Medidas físicas refinadas

ESTIMACION vs VARIABILIDAD EN EL TIEMPO

INCERTIDUMBRE DE LA ESTIMACIÓN

ALTA

MEDIA

BAJA

LA MÁS BAJA

FASE

QUÉ

CÓMO

HACER

**DESPLEGAR/
ARREGLAR**

OBJETIVO DE LA MEDIDA: COMPROBACIÓN DE CUMPLIMIENTO DE REQUISITOS NO FUNCIONALES

UNIDAD DE MEDIDA

- Número de clases que hacen uso de la clase que estamos estudiando (*fan-in*)
- Número de clases que usa la clase que estamos estudiando (*fan-out*)

- Complejidad cliclomática
- Deuda Técnica
- Código duplicado
- Complejidad
- Índice de cohesión
- Índice de acoplamiento
- Quebrantamiento de normas de seguridad
- Cantidad de vulnerabilidades
- Incumplimientos de buenas prácticas de programación
- Cumplimiento de las normas W3C
- Número de clases que hacen uso de la clase que estamos estudiando (*fan-in*)
- Número de clases que usa la clase que estamos estudiando (*fan-out*)

- Complejidad cliclomática
- Deuda Técnica
- Código duplicado
- Complejidad
- Índice de cohesión
- Índice de acoplamiento
- Quebrantamiento de normas de seguridad
- Cantidad de vulnerabilidades
- Incumplimientos de buenas prácticas de programación
- Cumplimiento de las normas W3C
- Número de clases que hacen uso de la clase que estamos estudiando (*fan-in*)
- Número de clases que usa la clase que estamos estudiando (*fan-out*)

02 | MEDICIÓN DEL PROCESO SOFTWARE

¿Qué es un proceso software?

En el contexto del desarrollo y/o mantenimiento de software, se puede definir proceso como la organización lógica de personas, materiales, energía, equipos y procedimientos en actividades de trabajo diseñada para producir un resultado final específico.



Gestión del Proceso Software

Un proceso Software es el proceso que se aplica en el desarrollo, mantenimiento y soporte de productos software.

La gestión efectiva del proceso software produce que los productos y servicios entregados por el proceso:

- sean totalmente conformes con los requisitos internos y externos del cliente
- logran los objetivos de negocio de la organización responsable de producir los productos.

El concepto de gestión de procesos aquí descrito está fundamentado en los principios del Control Estadístico de Procesos. Estos principios mantienen que el

establecimiento de niveles de variabilidad sostenible hace que el rendimiento del proceso sea predecible.

Según Shewhart, se dice que un fenómeno está controlado cuando mediante el uso de la experiencia pasada se puede predecir, al menos entre límites, como se espera que el fenómeno varíe en el futuro.

Los procesos controlados son procesos estables, y los procesos estables nos permiten predecir resultados. Esto, a su vez, nos permite:

- preparar planes alcanzables.
- cumplir con las estimaciones de coste y los compromisos de calendario.
- entregar productos con la funcionalidad y calidad de aceptable y razonable consistencia.

Si un proceso controlado no es capaz de cumplir con los requisitos del cliente y los objetivos de negocio, el proceso debe ser mejorado o reorientado.

Las cuatro responsabilidades centrales de la gestión de procesos son:

1. Definir el proceso.
2. Medir el proceso.
3. Controlar el proceso (asegurar que la variabilidad es estable y por tanto los resultados futuros son predecibles).
4. Mejorar el proceso.

Estas responsabilidades son análogas al ciclo PDCA de Shewhart, popularizado por Deming.

Medir el proceso

Las mediciones son la base para detectar desviaciones del rendimiento aceptable del proceso; también son la base para identificar oportunidades de mejora del proceso. Los tres objetivos clave de la medición del proceso son:

1. Recoger datos que miden el rendimiento de cada proceso.
2. Analizar el rendimiento de cada proceso.

3. Retener y usar los datos para:

- Valorar la estabilidad y capacidad del proceso.
- Interpretar los resultados de las observaciones y de los análisis.
- Predecir costes futuros y rendimiento.
- Proporcionar líneas base y benchmark.
- Trazar tendencias.
- Identificar oportunidades de mejora.

Controlar el proceso

Controlar un proceso significa mantener el proceso dentro de sus límites de rendimiento normales o inherentes, es decir que el comportamiento del proceso sea consistente. Esto supone:

1. Medir (obtener información del rendimiento del proceso)
2. Detectar (analizar la información con el objeto de identificar variaciones en el proceso que son debidas a causas asignables)
3. Corregir (tomar medidas para eliminar del proceso las causas asignables y eliminar los resultados no deseables del proceso que se derivan en el producto)

Medir el comportamiento del proceso

Con el objeto de conseguir que las actividades de medida sean efectivas y eficientes éstas deben ser diseñadas y orientadas para soportar los objetivos de negocio de la organización y proporcionar información para la toma de decisiones de una manera efectiva y eficiente.

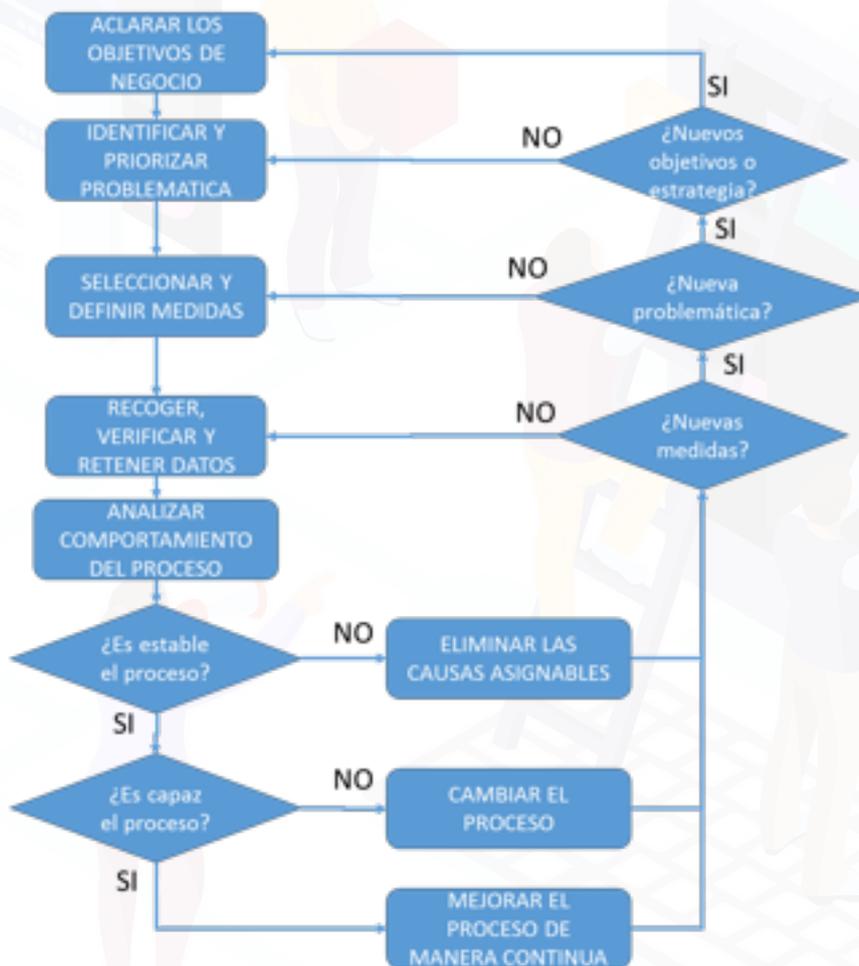
Para ayudar a abordar los objetivos de negocio, se propone agrupar las funciones de gestión del software en tres clases: gestión del proyecto, gestión del proceso e ingeniería del producto. Estas funciones de gestión abordan diferentes cuestiones, cada una de ellas con sus propios objetivos y problemática.

1. **Gestión del proyecto.** Los objetivos de la gestión del proyecto son establecer compromisos aceptables a cerca del coste, planificación, calidad y funcionalidades entregadas.
2. **Gestión del proceso.** Los objetivos de la gestión del proceso son asegurar que los procesos de la organización se realizan según lo esperado, asegurar que los procesos definidos son seguidos y llevar a cabo mejoras de los procesos para poder conseguir los objetivos de negocio.
3. **Ingeniería del producto.** Los objetivos de la ingeniería del producto son asegurar la aceptación y satisfacción del producto por parte del cliente. En esta función, la principal preocupación está relacionada con atributos físicos y dinámicos del producto: arquitectura, fiabilidad, usabilidad, sensibilidad, estabilidad, rendimiento.... La información sobre estos atributos y la satisfacción del cliente son importantes para valorar el logro de los objetivos de la ingeniería del producto.

Con objeto de ilustrar la relación entre Objetivos de negocio, problemática y medidas proponemos la siguiente tabla con objetivos típicos de negocio:

OBJETIVOS DE NEGOCIO	PROBLEMÁTICA DEL PROYECTO	PROBLEMÁTICA DEL PROCESO	MEDIDAS DEL PRODUCTO Y ATRIBUTOS DEL PROCESO
Incrementar la funcionalidad	<ul style="list-style-type: none"> • Crecimiento del producto • Estabilidad del producto 	<ul style="list-style-type: none"> • Conformidad del producto 	<ul style="list-style-type: none"> • Numero de requisitos • Tamaño del producto • Complejidad del producto • Tasa de cambios • Porcentaje de no conformidades
Reducir el coste	<ul style="list-style-type: none"> • Presupuesto • Tasa de gastos 	<ul style="list-style-type: none"> • Eficiencia • Productividad • Re trabajo 	<ul style="list-style-type: none"> • Tamaño del producto • Complejidad del producto • Esfuerzo • Número de cambios • Estabilidad de los requisitos
Reducción de time-to-market	<ul style="list-style-type: none"> • Calendario • Avance 	<ul style="list-style-type: none"> • Tasa de producción • Capacidad de respuesta 	<ul style="list-style-type: none"> • Tiempo transcurrido normalizado según las características del producto
Mejorar la calidad del producto	<ul style="list-style-type: none"> • Rendimiento del producto • Exactitud del producto • Fiabilidad del producto 	<ul style="list-style-type: none"> • Predictibilidad • Reconocimiento de problemas • Análisis de causas raíces 	<ul style="list-style-type: none"> • Cantidad de errores introducidos • Efectividad de las actividad de detección de errores • Tiempo medio entre fallos.

Un marco de trabajo para la medida del comportamiento del proceso



Selección de medidas de rendimiento del proceso

Para la selección de medidas de rendimiento proponemos la siguiente relación de criterios a aplicar:

- Las medidas deberían estar relacionadas de una manera muy cercana con la problemática bajo estudio. Las problemáticas habituales son calidad, consumo de recursos o tiempo transcurrido.
- Deben tener un alto contenido de información. Se deberían recoger medidas de producto o proceso que son sensibles a la mayor cantidad de facetas de resultado del proceso.

- Deben pasar una prueba de verosimilitud: ¿las medidas reflejan realmente el grado con el que el proceso logra los resultados que son importantes?
- Deben permitir una recogida fácil y económica de sus valores.
- Deben permitir una recogida bien definida y consistente de datos.
- Deben mostrar variaciones medibles. Un número que cambia no proporciona información del proceso.
- Como conjunto deben tener proporcionar un diagnóstico de valor y ser capaces de ayudar a identificar no solo lo que pasa de manera no usual sino también sus causas.

Medidas del proceso software

Según ha establecido el Software Engineering Institute de la Carnegie Mellon University, las medidas “core” de un proceso son:

- **Tamaño** de lo producido.
- **Esfuerzo** empleado para producirlo.
- **Cantidad de errores** producidos.
- **Desviación de la planificación.**

Ejemplos de medidas del proceso software

A partir de las medidas “core” enunciadas en el punto anterior, de las entidades que aparecen en el proceso software y de los atributos medibles de estas entidades, se puede proponer los siguientes ejemplos:

ENTIDADES MEDIBLES EN EL PROCESO SOFTWARE

Cosas recibidas o usadas	Actividades y sus elementos	Cosas consumidas	Cosas mantenidas o retenidas	Cosas producidas
<ul style="list-style-type: none">• Procesos y controladores<ul style="list-style-type: none">o Análisis de requisitoso Diseñoso Codificacióno Pruebaso Control de configuracióno Control de cambioso Gestión de problemaso Revisióno Inspeccióno Integración• Flujos y caminos<ul style="list-style-type: none">o Caminos de productoo Camino de los recursoso Camino de los datoso Caminos de Control• Buffers<ul style="list-style-type: none">o Colaso Pilaso Contenedores	<ul style="list-style-type: none">• Recursos<ul style="list-style-type: none">o Esfuerzoo Materiales en brutoo Energíao Dineroo Tiempo	<ul style="list-style-type: none">• Crecimiento del producto• Estabilidad del producto	<ul style="list-style-type: none">• Personas• Instalaciones• Herramientas• Materiales• Trabajo en curso• Datos• Conocimiento• Experiencia	<ul style="list-style-type: none">• Productos<ul style="list-style-type: none">o Requisitoso Especificacioneso Diseñoso Unidadeso Móduloso Casos de pruebao Resultado de las pruebaso Componentes probadoso Documentacióno Erroreso Informes de erroreso Peticiones de cambioo Datoso Materiales adquiridoso Otros artefactos• Sub productos<ul style="list-style-type: none">o Conocimientoo Experienciao Perfileso Mejoras de procesoso Datoso Satisfacción del cliente

ATRIBUTOS MEDIBLES DE LAS ENTIDADES DEL PROCESO SOFTWARE

Cosas recibidas o usadas	Actividades y sus elementos	Cosas consumidas	Cosas producidas
<ul style="list-style-type: none">• Cambios<ul style="list-style-type: none">o Tipoo Fechao Tamañoo Cantidad• Requisitos<ul style="list-style-type: none">o Estabilidad de los requisitoso Cantidad de requisitos identificadoso % trazados al diseñoo % trazados al código• Informes de problemas<ul style="list-style-type: none">o Tipoo Fechao Tamañoo Origeno Severidado Cantidado Fondoso Dineroo Presupuestoo Estado Económico• Personas<ul style="list-style-type: none">o Años de experienciao Tipo de educacióno % formado en un aspecto en concretoo Categoría profesional	<ul style="list-style-type: none">• Flujos y caminos<ul style="list-style-type: none">o Tiempo de procesamientoo Tasa de rendimientoo Desviacioneso Retrasoso Backlogs: trabajo pendiente• Longitud, tamaño<ul style="list-style-type: none">o Colaso Bufferso Pilas	<ul style="list-style-type: none">• Esfuerzo<ul style="list-style-type: none">o Horas de desarrolloo Horas de re trabajoo Horas de soporteo Horas de preparacióno Horas de reuniones• Tiempo<ul style="list-style-type: none">o Fecha de inicioo Fecha de fino Duración del proceso o tareao Tiempo de espera• Dinero<ul style="list-style-type: none">o Coste a la fechao Variación del costeo Coste del re trabajo	<ul style="list-style-type: none">• Estado de las unidades de trabajo<ul style="list-style-type: none">o % diseñadoo % codificadoo % probado• Tamaño de las unidades de trabajo<ul style="list-style-type: none">o Cantidad de requintoso Cantidad de puntos funcióno Cantidad de líneas de códigoo Cantidad de móduloso Cantidad de objetoso Tamaño de la base de datos• Cuantificación de las salidas<ul style="list-style-type: none">o Cantidad de accioneso Cantidad de aprobacioneso Cantidad de errores encontrados• Resultado de las pruebas<ul style="list-style-type: none">o Cantidad de casos del prueba con ejecución exitosao Cobertura de las pruebas• Arquitectura<ul style="list-style-type: none">o Número de clases que hacen uso de la clase que estamos estudiando (fan-in)o Número de clases que usa la clase que estamos estudiando (fan-out)

ATRIBUTOS MEDIBLES DE LAS ENTIDADES DEL PROCESO SOFTWARE

Cosas recibidas o usadas

- Instalaciones y entorno
 - o Superficie por empleado
 - o Nivel de ruido
 - o Iluminación
 - o Cantidad de empleados por cubículo
 - o Cantidad de empleados que comparte oficina o cubículo
 - o Inversión en herramientas por empleado
 - o Horas de uso de ordenador
 - o % de Capacidad utilizada

Actividades y sus elementos

Cosas consumidas

Cosas producidas

- Cambios
 - o Tipo
 - o Fecha
 - o Tamaño
 - o Esfuerzo empleado
- Problemas y errores
 - o Cantidad de informes
 - o Densidad de errores
 - o Tipo
 - o Origen
 - o Distribución por tipo
 - o Distribución por origen
 - o Cantidad de errores abiertos
 - o Cantidad de errores cerrados
- Utilización de recursos críticos
 - o % de memoria utilizada
 - o % de capacidad de CPU utilizada
 - o % de capacidad de I/O utilizada

El proceso de medida

Una vez seleccionadas las medidas que se van a utilizar es necesario definir el proceso que nos va a permitir obtener los valores de estas medidas. En la definición de este proceso nos debemos asegurar que se satisfacen los siguientes tres criterios establecidos por Deming.

1. **Comunicación** ¿Los métodos usados para definir las medidas y las obtención de sus valores permite a otras personas saber de una manera precisa lo que debe ser medido y que ha sido incluido y excluido en resultados agregados? ¿Los usuarios de los datos saben cómo se recogen los datos y como pueden interpretar los resultados correctamente?.
2. **Repetitividad** ¿Puede cualquier persona ser capaz de repetir las medias y obtener los mismos resultados?.
3. **Trazabilidad** ¿Están identificados los orígenes de los datos en términos de tiempo, fuentes, secuencia, actividad, producto, estado, entorno, herramienta de medida usada y agente de recogida?.

El requisito de trazabilidad es especialmente importante a la hora de mejorar el rendimiento del proceso software. Dado que las medidas de rendimiento pueden señalar la inestabilidad del proceso, es importante que el contexto y las circunstancias de las medidas sean registradas: esto nos ayudará a identificar las causas de las inestabilidades.

Integración del proceso de medida en el proceso software

A la hora de definir el proceso de medida es importante tener en cuenta ese viejo principio de la filosofía de la ciencia que dice: “La observación del experimento modifica el experimento”.

Para ello debemos integrar el proceso de medida en el propio proceso software, de tal manera que las medidas se obtengan del propio proceso de una manera natural.

Esta integración nos lleva al análisis de las actividades de medida existentes y al diagnóstico de las medidas existentes desde el punto de cumplimiento de los objetivos impuestos a las actividades de medida.

Una vez realizado el análisis y el diagnóstico sus resultados deben trasladarse a acciones realizables. Las acciones para integrar las medidas son:

1. Identificar las fuentes de datos dentro del proceso software existente.
2. Definir los métodos que se usaran para recoger y reportar los datos.
3. Identificar y especificar las herramientas necesarias para recoger, reportar y almacenar los datos.
4. Determinar los requisitos de puntos del tiempo y frecuencias de las medidas.
5. Documentar en detalle los procedimientos de recogida de datos: responsables y determinar dónde, cómo y cuándo recoger datos y reportar.
6. Determinar quién usará los datos.
7. Definir cómo se analizarán los datos.
8. Preparar una guía del proceso de definición y recogida de datos.

Se deben analizar los temas relativos al almacenamiento de los datos y los requisitos de acceso:

- Necesidades de retención histórica de los datos
 - Quién recoge almacena, mantiene y accede a los datos
 - Nivel organizacionales a los que se da servicio
 - Granularidad de los datos
 - Los procedimiento para editar y verificar dinámicamente los datos introducidos en la base de datos
 - La cantidad de personas con acceso a los datos
 - La necesidad de registro de las definiciones asociadas con los datos para así poder relacionar los datos con la información descriptiva que es necesaria para usar los datos correctamente.
- También es importante prestar atención a la problemática de la privacidad de los datos. Aunque el objeto de las medidas no deber ser la evaluación de personas, es especialmente importante la confidencialidad de los datos que podrían ser usados (o que se perciben que pueden ser usados) para evaluar el rendimiento individual de personas o equipos.

Revisión y valoración de los datos recogidos

Previamente al análisis de los valores de las medidas es necesario que se satisfagan criterios que permitan conceder credibilidad a los análisis realizados.

Estos criterios son:

1. **Formalmente correctos.** Nos debe permitir asegurar que los datos han sido recogidos de acuerdo a las especificaciones y no contienen errores. Los datos deben cumplir las siguientes características:
 - Son del tipo correcto (numérico o alfanumérico)
 - Tienen el formato correcto
 - Están dentro de los rangos especificados
 - Son completos
 - Son aritméticamente correctos
2. **Sincronicidad.** Se puede pensar en medidas que están sincronizadas cuando los valores de dos o más atributos están relacionados respecto del tiempo en que ocurren. La noción de medidas sincronizadas es especialmente importante cuando se miden los atributos de un proceso o cuando se usan los atributos de productos o recursos para describir el rendimiento de un proceso. Las medidas que están basadas en marcos de tiempo arbitrarios son susceptibles de tener problemas de sincronización.
3. **Consistencia.** No se debería introducir datos en la base de datos hasta confirmar que su definición es consistente con otros valores registrados del mismo tipo.
4. **Validez.** En el nivel más básico, debemos ser capaces de demostrar que los valores usados para describir un atributo describen con veracidad el atributo en cuestión. Para que esto sea verdad las medidas deben estar bien definidas: las reglas para la medición tienen que estar explícitamente declaradas y esto debe hacerse de manera tal que no haya preguntas sobre cuáles son las reglas. Cualquier decisión que se deje a interpretaciones locales o al juicio de la persona que recoge los datos debe conducir, en el peor de los casos, a solo a diferencias inmatrimales en los valores medidos.

03 | METODOLOGÍA PARA DETERMINAR MÉTRICAS: GOAL-QUESTION-METRIC GQM

El enfoque de GQM

El enfoque de GQM se basa en el supuesto de que para medir de una manera intencionada primeramente se deben especificar los objetivos de la organización y sus proyectos, trazar estos objetivos a los datos que están destinados a definir los objetivos y finalmente proporcionar un marco para interpretar los datos con respecto a los objetivos declarados.

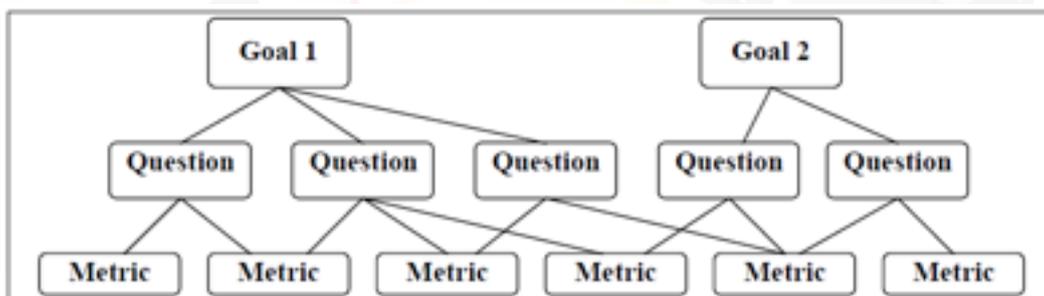
Por lo tanto, es importante determinar, al menos en términos generales, qué necesidades informativas tiene la organización de modo que estas necesidades de información puedan ser cuantificada siempre que sea posible, y que con la información cuantificada se puede analizar si se logran los objetivos.

El resultado de la aplicación de GQM es la especificación de un sistema de medición dirigido a un conjunto particular de problemáticas y un conjunto de reglas para la interpretación de los datos de medición. El modelo de medición resultante tiene tres niveles:

1. **Nivel conceptual (GOAL):** se define una meta para un objeto desde varios puntos de vista y en relación con un entorno en particular. Los objetos de medida son:
 - **Productos:** artefactos, entregas y documentos que se producen durante el ciclo de vida del software, por ejemplo, especificaciones, diseños, programas, conjuntos de pruebas...
 - **Procesos:** actividades relacionadas con la producción del software normalmente asociadas con el tiempo, por ejemplo tiempo empleado en especificar, tiempo empleado en probar....
 - **Recursos:** elementos utilizados por los procesos para producir sus resultados, por ejemplo personas, hardware, software, instalaciones...
2. **Nivel operativo (QUESTION):** se utiliza un conjunto de preguntas para caracterizar la forma en que se evalúa el logro de un objetivo específico. Las

preguntas intentan caracterizar el objeto de medición (producto, proceso, recurso) con respecto a la problemática seleccionada y para determinar su adecuación desde el punto de vista seleccionado.

3. Nivel cuantitativo (METRIC): se asocia un conjunto de datos con cada pregunta de manera que ésta pueda ser contestada de manera cuantitativa. Los datos pueden ser:
 - **Objetivos:** si dependen solo del objeto que se está midiendo y no del punto de vista desde el cual se toman, por ejemplo, número de versiones de un documento, horas de personal dedicadas a una tarea, tamaño de un programa.
 - **Subjetivos:** si dependen tanto del objeto que se está midiendo como del punto de vista desde el cual se toman, por ejemplo legibilidad de un texto, nivel de satisfacción del usuario.



Un modelo GQM es una estructura jerárquica que comienza con una meta (que especifica el propósito de la medición, objeto a medir, problema a medir y punto de vista desde donde se toma la medida). La meta se refina en varias preguntas que generalmente divide la problemática en sus componentes principales. Cada pregunta se refina en métricas, algunas de ellas objetivas y otras subjetivas. Se puede usar la misma métrica para responder diferentes preguntas bajo la misma meta. Otros modelos de GQM también pueden tener preguntas y métricas en común, asegurándose de que cuando la medida se toma realmente, los diferentes puntos de vista se tienen en cuenta correctamente (es decir, la métrica puede tener valores diferentes cuando se toma desde diferentes puntos de vista).

Ilustramos el modelo con el siguiente ejemplo:

META	PROPÓSITO	Mejorar la duración del procesado de cambios desde el punto de vista del Jefe de proyecto
	PROBLEMÁTICA	
	OBJETO	
	PUNTO DE VISTA	
PREGUNTA	¿Cuál es la velocidad actual de procesado de cambios?	
METRICA	Tiempo medio Desviación estándar % de valores fuera del límite superior	
PREGUNTA	¿Está mejorando el rendimiento del procesado?	
METRICA	(Tiempo medio actual/Tiempo medio de referencia) * 100 Valoración subjetiva de la Dirección	

El proceso GQM

Un modelo GQM se desarrolla identificando un conjunto de objetivos de calidad y/o productividad a nivel corporativo, de división o de proyecto (por ejemplo, satisfacción del cliente, entrega a tiempo...). A partir de esos objetivos y basados en los modelos del objeto de medición, se derivan preguntas que definen esos objetivos de la manera más completa posible. El siguiente paso consiste en especificar las medidas que deben recogerse para responder a esas preguntas y realizar un seguimiento de la conformidad de los productos y procesos con los objetivos. Una vez que se han especificado las medidas, necesitamos desarrollar los mecanismos de recopilación de datos, incluyendo mecanismos de validación y análisis.

Por lo tanto, el desarrollo de una meta se basa en tres fuentes básicas de información. La primera fuente es la política y la estrategia de la organización que aplica el GQM. De esta fuente derivamos tanto la problemática como el propósito de la Meta al analizar las políticas corporativas, planes estratégicos y, lo que es más importante, entrevistas y temas relevantes en la organización.

La segunda fuente de información es la descripción del proceso y los productos de la organización o, al menos, las que están dentro del alcance de la medición

que queremos realizar. Si, por ejemplo, queremos evaluar un proceso, necesitamos un modelo de ese proceso y de los subprocesos que lo componen. De esta fuente derivamos el objeto coordinado con la meta especificando los procesos y los modelos de productos, al mejor nivel de formalidad posible.

La tercera fuente de información es el modelo de la organización, que nos proporciona el punto de vista coordinado con la meta. Obviamente, no todas las problemáticas y procesos son relevantes para todos los puntos de vista en una organización, por lo tanto, debemos realizar un paso de análisis de relevancia antes de completar nuestra lista de metas para asegurarnos que las metas que tenemos definidas tienen la relevancia necesaria.

De esta manera, terminamos con una especificación de nuestras metas que tiene en cuenta la estructura y el objetivo de la organización. De la especificación de cada meta podemos derivar preguntas significativas que caracterizan esa meta de manera cuantificable. En general, haremos al menos tres grupos de preguntas:

Grupo 1. ¿Cómo podemos caracterizar el objeto (producto, proceso o recurso) con respecto a la meta general del modelo GQM específico?

Grupo 2. ¿Cómo podemos caracterizar los atributos del objeto que son relevantes con respecto a la problemática del modelo GQM específico?

Grupo 3. ¿Cómo evaluamos las características del objeto que son relevantes con respecto a la problemática del modelo GQM específico?

Una vez que se han desarrollado las preguntas, procedemos a asociar cada pregunta con métricas apropiadas. Los factores que consideramos al hacer esto variados, pero entre ellos podemos destacar:

- **Cantidad y calidad de los datos existentes:** intentaremos maximizar el uso de los datos existentes siempre y cuando las fuentes de datos estén disponibles y sean confiables.
- **Madurez de los objetos de medida:** aplicaremos medidas objetivas a los objetos de medición maduros y utilizaremos evaluaciones más subjetivas con los con objetos informales o inestables.
- **Proceso de aprendizaje:** los modelos GQM siempre necesitan refinamiento y adaptación, por lo tanto las medidas que definimos deben ayudarnos a evaluar no solo el objeto de medición, sino también la fiabilidad del modelo utilizado para evaluarlo.

Una vez que se haya desarrollado un modelo GQM, seleccionaremos los procedimientos, técnicas y herramientas para la recogida de datos. Los datos que se recopilan se mapean con el modelo y son interpretados según los esquemas definidos previamente por la organización.



04 | MEDICIÓN DE UN SERVICIO DE MANTENIMIENTO DE SOFTWARE

¿Qué es un servicio?

En ITIL, colección de buenas prácticas para la gestión de servicios de Tecnologías de la Información que se ha convertido en referente mundial en el ámbito de la gestión de Servicios de TI, se define:

“Un servicio es un medio de entregar valor a los clientes al facilitar los resultados que los clientes quieren lograr sin que éstos tengan la propiedad de los costes y de los riesgos específicos. Los servicios son un medio de entregar valor a los clientes al facilitar los resultados esperados.”

En CMMI, modelo de mejora de proceso ampliamente reconocido, se define:

“SERVICIO: Producto que es intangible y no almacenable.

Los servicios se proporcionan utilizando sistemas de servicio que han sido diseñados para satisfacer los requisitos de servicio.

SISTEMA DE SERVICIO: Combinación integrada e interdependiente de los recursos que lo componen que satisface los requisitos del servicio.

Un sistema de servicio engloba todo lo que se necesita para la prestación de servicios, incluyendo productos de trabajo, procesos, instalaciones, herramientas, consumibles y recursos humanos.

REQUISITOS DE SERVICIO: Conjunto completo de requisitos que afectan a la prestación de servicios y al desarrollo del sistema de servicio.

Los requisitos del servicio incluyen requisitos tanto técnicos como no técnicos. Los requisitos técnicos son características del servicio a prestar y del sistema de servicio que se necesita para posibilitar la prestación. Los requisitos no técnicos pueden incluir condiciones, disposiciones, compromisos y términos adicionales identificados en acuerdos y regulaciones, así como las capacidades y condiciones que se necesitan, derivadas de los objetivos de negocio.”

Objetivo de la medición del servicio

Teniendo en cuenta las definiciones dadas de servicio, el objetivo de la medición del servicio debe ser la comprobación de la entrega de valor y del cumplimiento de los requisitos del servicio.

En el ámbito de la gestión de servicios es habitual ligar el cumplimiento de los requisitos del servicio a los indicadores, o medidas, de nivel de servicio.

Medidas del servicio

A modo de ejemplo, proponemos la siguiente relación de medidas del servicio relacionadas con los objetivos de negocio.

OBJETIVO DE NEGOCIO	NECESIDAD DE INFORMACIÓN	OBJETIVO DE MEDICIÓN	CATEGORÍA DE LA INFORMACIÓN MEDIDA	EJEMPLOS DE MEDIDAS BASE	EJEMPLOS DE MEDIDAS DERIVADAS
<p>Proporcionar la continuidad del servicio acordada</p>	<p>¿Pueden recuperarse los servicios de los desastres o de las grandes alteraciones dentro de los plazos acordados?</p>	<p>Proporcionar visibilidad sobre si los planes de continuidad del servicio se ejecutarán de forma exitosa para proveer la continuidad del servicio acordada</p>	<p>Continuidad del servicio</p>	<ul style="list-style-type: none"> • Número de servicios con pruebas de recuperación fallidas • Número total de servicios en el catálogo de servicios 	<ul style="list-style-type: none"> • Ratio de confianza en la continuidad del servicio
<p>Proporcionar capacidad apropiada para cubrir las necesidades de negocio</p> <p>Prevenir incidencias relacionadas con la capacidad</p>	<p>¿Hay suficientes recursos (o demasiados) para cubrir la demanda de servicios?</p>	<p>Proporcionar visibilidad sobre utilización de recursos, recursos ociosos, y capacidad inadecuada para cubrir la demanda</p>	<p>Capacidad</p>	<ul style="list-style-type: none"> • Número total de peticiones de servicio • Horas disponibles del personal del proveedor de servicios • Tiempo de servicio 	<ul style="list-style-type: none"> • Tiempo de servicio medio • Uso del personal del proveedor de servicios
<p>Proporcionar servicios con costes eficientes</p>	<p>¿Se está proporcionando un servicio competitivo mediante una planificación de capacidad precisa?</p>	<p>Proporcionar visibilidad sobre los gastos debidos a la capacidad no planificada</p>	<p>Capacidad</p>	<ul style="list-style-type: none"> • Gasto total debido a la capacidad no planificada • Coste total de los recursos 	<ul style="list-style-type: none"> • Porcentaje del coste de los recursos de servicio originado por capacidad no planificada
<p>Mejorar el nivel de calidad en el servicio</p>	<p>¿Está mejorando el nivel de calidad en el servicio?</p>	<p>Proporcionar visibilidad sobre si la calidad del servicio que se está prestando está mejorando por medio de observar cuántos errores se repiten</p>	<p>Calidad del servicio</p>	<ul style="list-style-type: none"> • Número total de errores que se repiten • Número total de errores 	<ul style="list-style-type: none"> • Ratio de errores que se repiten

OBJETIVO DE NEGOCIO	NECESIDAD DE INFORMACIÓN	OBJETIVO DE MEDICIÓN	CATEGORÍA DE LA INFORMACIÓN MEDIDA	EJEMPLOS DE MEDIDAS BASE	EJEMPLOS DE MEDIDAS DERIVADAS
Proporcionar servicios eficaces	¿Cuál es la eficacia de los servicios?	Proporcionar visibilidad sobre qué porcentaje de peticiones de servicio se están rehaciendo	Rendimiento del servicio	<ul style="list-style-type: none"> • Número de peticiones de servicio rehechas • Número total de peticiones de servicio 	<ul style="list-style-type: none"> • Ratio de servicios rehechos
Proporcionar la apropiada disponibilidad de servicios acordada	¿Se está proporcionando la apropiada disponibilidad de servicios acordada?	Proporcionar visibilidad sobre la disponibilidad de los servicios	Disponibilidad	<ul style="list-style-type: none"> • Tiempo de servicio acordado • Tiempo sin servicio 	<ul style="list-style-type: none"> • Tiempo de servicio medio • Uso del personal del proveedor de servicios
	¿Es el servicio tan fiable como se acordó?	Proporcionar visibilidad sobre la fiabilidad del servicio	Disponibilidad	<ul style="list-style-type: none"> • Tiempo disponible (en horas) • Total de tiempo sin servicio (en horas) • Número de paradas en el servicio (el servicio normal se interrumpe) 	<ul style="list-style-type: none"> • Fiabilidad como tiempo medio entre fallos (MTBF)
Cumplir los indicadores de nivel de servicio	¿La duración del tratamiento de las peticiones es el acordado con el cliente?	Proporcionar visibilidad sobre el tiempo que se tarda en atender una petición end-to-end	Duración	<ul style="list-style-type: none"> • Tiempo de primera respuesta a las peticiones • Tiempo de duración del tratamiento de la petición 	
Mejora del servicio	¿Las acciones de mejora del servicio son eficaces desde el punto de vista de reducción de peticiones?	Proporcionar visibilidad sobre la eficacia de la mejora del servicio	Volumen del servicio	<ul style="list-style-type: none"> • Cantidad de peticiones al servicio 	<ul style="list-style-type: none"> • Variación por periodos de tiempo de la cantidad de peticiones al servicio

REFERENCIAS Y LECTURAS COMPLEMENTARIAS:

- **SOFTWARE SIZE MATTERS**

- **QSM** www.qsm.com

- **MÉTODO DE CÁLCULO DE PUNTOS DE CASO DE USO**

- <https://www.mountaingoatsoftware.com/articles/estimating-with-use-case-points>

- **MÉTODO DE CÁLCULO DE PUNTOS DE HISTORIA DE USUARIO**

- <https://www.atlassian.com/agile/project-management/estimation>

- **MÉTODO DE CÁLCULO DE PUNTOS FUNCIÓN**

- **ISO Standards:**

- **FiSMA:** ISO/IEC 29881:2010 Information technology – Systems and software engineering – FiSMA 1.1 functional size measurement method.
- **IFPUG:** ISO/IEC 20926:2009 Software and systems engineering – Software measurement – IFPUG functional size measurement method.
- **Mark-II:** ISO/IEC 20968:2002 Software engineering – MI II Function Point Analysis – Counting Practices Manual
- **Nesma:** ISO/IEC 24570:2018 Software engineering – Nesma functional size measurement method version 2.3 – Definitions and counting guidelines for the application of Function Point Analysis
- **COSMIC:** ISO/IEC 19761:2011 Software engineering. A functional size measurement method

- **MEASURING THE SOFTWARE PROCESS**

William A. Florac, Anita D. Carleton

Ed. Addison Wesley

- **THE GOAL QUESTION METRIC APPROACH**

Victor R. Basili(1) Gianluigi Caldiera(1) H. Dieter Rombach(2)

(1)Institute for Advanced Computer Studies. Department of Computer Science. University Of Maryland. College Park, Maryland

(2) FB Informatik. Universität Kaiserslautern. Kaiserslautern, Germany

- **THE GOAL QUESTION METRIC METHOD: a practical guide for quality improvement of software development**

Rini Van Soligen, Egon Bernhout

Ed. MacGraw-Hill Companies

- **ITIL VERSION 3 SERVICE STRATEGY**

Office of Government Commerce-United Kingdom

- **CMMI® PARA SERVICIOS, VERSIÓN 1.3**

CMMI Institute