

GUÍA AVANZADA DE MEDICIÓN Y ANÁLISIS

**Laboratorio Nacional de Calidad del
Software**

NOTA DE EDICIÓN

Esta guía ha sido desarrollada por el Laboratorio Nacional de Calidad del Software de INTECO. Esta primera versión ha sido editada en Mayo del 2009.

El presente documento cumple con las condiciones de accesibilidad del formato PDF (Portable Document Format).

Se trata de un documento estructurado y etiquetado, provisto de alternativas a todo elemento no textual, marcado de idioma y orden de lectura adecuado.

Para ampliar información sobre la construcción de documentos PDF accesibles puede consultar la guía disponible en la sección [Accesibilidad > Formación > Manuales y Guías](#) de la página <http://www.inteco.es>.

AVISO LEGAL

- CMMI® es una marca registrada en la Oficina de Marcas y Patentes de EEUU por la Universidad Carnegie Mellon.
- Las distintas normas ISO mencionadas han sido desarrolladas por la International Organization for Standardization.
- PMBOK® es una marca registrada por el Project Management Institute, Inc.

Todas las demás marcas registradas que se mencionan, usan o citan en la presente guía son propiedad de los respectivos titulares.

INTECO cita estas marcas porque se consideran referentes en los temas que se tratan, buscando únicamente fines puramente divulgativos. En ningún momento INTECO busca con su mención el uso interesado de estas marcas ni manifestar cualquier participación y/o autoría de las mismas.

Nada de lo contenido en este documento debe ser entendido como concesión, por implicación o de otra forma, y cualquier licencia o derecho para las Marcas Registradas deben tener una autorización escrita de los terceros propietarios de la marca.

Por otro lado, INTECO renuncia expresamente a asumir cualquier responsabilidad relacionada con la publicación de las Marcas Registradas en este documento en cuanto al uso de ninguna en particular y se eximen de la responsabilidad de la utilización de dichas Marcas por terceros.

El carácter de todas las guías editadas por INTECO es únicamente formativo, buscando en todo momento facilitar a los lectores la comprensión, adaptación y divulgación de las disciplinas, metodologías, estándares y normas presentes en el ámbito de la calidad del software.

ÍNDICE

1. PROPÓSITO	9
2. CONCEPTOS Y PRINCIPIOS	10
2.1. ¿Por qué medir?	10
2.2. Conceptos básicos	13
3. DEFINICIÓN DE LAS MÉTRICAS	21
3.1. Identificar necesidades de información	25
3.2. Diseño de las métricas	27
4. CLASIFICACIÓN/TIPOS DE MÉTRICAS	31
4.1. A nivel de proyecto	31
4.2. A nivel de proceso	33
4.3. A nivel de producto y servicio	35
4.3.1. Métricas de calidad de producto final	37
4.3.2. Métricas de mantenimiento de software	37
5. MODELO DE FICHA INDICADOR	39
6. TÉCNICAS DE ANÁLISIS	45
6.1. Métodos de análisis	46
6.1.1. Checklist	46
6.1.2. Diagrama de Pareto	47
6.1.3. Histograma	48
6.1.4. Diagrama de dispersión	49
6.1.5. Gráfico de ejecución	50
6.1.6. Gráfico de control	52
6.1.7. Diagrama de causa-efecto	53
7. MECANISMOS PARA HACER EL REPORTING DE LOS INDICADORES	55
7.1. Cuadros de mando	55
7.2. Mejores prácticas para construir cuadros de mando	57
8. INTEGRAR LAS MÉTRICAS EN UN PROCESO ORGANIZATIVO	61
8.1. Establecer y mantener compromisos	62
8.2. Planificar el proceso de medición y análisis	62

8.3.	Ejecución de la medición y análisis	65
8.4.	Evaluación del proceso de medición	66
9.	ASPECTOS CLAVE EN EL PROCESO DE MEDICIÓN	68
10.	ENFOQUE MODELOS	71
10.1.	CMMI®	71
10.2.	ISO/IEC 15504	73
11.	ARTEFACTOS RELACIONADOS CON LA MEDICIÓN Y ANÁLISIS	75
12.	GLOSARIO	77
13.	REFERENCIAS	80
14.	ANEXO: EJEMPLOS DE INDICADORES	82
14.1.	Producto físico	82
14.1.1.	Tamaño	82
14.1.2.	Complejidad	83
14.1.3.	Calidad	84
14.2.	Gestión de proyectos (incl. proyectos de mejora)	85
14.2.1.	Planificación de proyecto	85
14.2.2.	Ejecución de proyecto	86
14.2.3.	Costes	86
14.2.4.	Productividad	87
14.2.5.	Seguimiento	88
14.3.	Satisfacción del cliente	89
14.3.1.	Problemas en explotación	89
14.3.2.	Percepción del cliente	90
14.4.	Proceso	90

ÍNDICE DE FIGURAS

Figura 1	Relación conceptos medición I	17
Figura 2	Relación conceptos medición II	20
Figura 3	Descripción de los 6 pasos del proceso GQM.....	23
Figura 4	Ejemplo de preguntas y métricas relacionadas con una meta	25
Figura 5	Evaluación de una necesidad de información a un plan de medición	27
Figura 6	Niveles de una construcción de medida	28
Figura 7	Relación conceptos medición III	30
Figura 8	Ejemplo Diagrama de Pareto para errores	47
Figura 9	Ejemplo histograma	48
Figura 10	Ejemplo diagrama de dispersión	50
Figura 11	Ejemplo gráfico de ejecución	51
Figura 12	Curva S del progreso de pruebas	51
Figura 13	Ejemplo diagrama causa-efecto.....	53
Figura 14	Ejemplo datos sin contexto	58
Figura 15	Ejemplo datos con contexto	59
Figura 16	Modelo del proceso de medición y análisis.....	62
Figura 17	Equilibrio necesidades de información.....	68
Figura 18	Diagrama de contexto de MA.....	72

ÍNDICE DE TABLAS

Tabla 1	Beneficios del proceso de medición a nivel de proyecto y organizativo	13
Tabla 2	Conceptos en el proceso de medición I.....	16
Tabla 3	Conceptos en el proceso de medición II.....	19
Tabla 4	Ejemplo de métricas durante la gestión de proyectos	33
Tabla 5	Ejemplos de métricas a nivel de producto y servicio	36
Tabla 6	Plantilla ficha indicador	43
Tabla 7	Tipos de correlación	49
Tabla 8	Comparativa entre dashboard y scorecard.....	57
Tabla 9	Métricas del proyecto en un plan de medición	64
Tabla 10	Actividades y tareas de PSM	67
Tabla 11	Área de proceso MA	72
Tabla 12	Plantilla indicador tamaño.....	82
Tabla 13	Plantilla indicador complejidad	83
Tabla 14	Plantilla indicador calidad	84
Tabla 15	Plantilla indicador planificación del proyecto	85
Tabla 16	Plantilla indicador ejecución del proyecto	86
Tabla 17	Plantilla indicador costes	86
Tabla 18	Plantilla indicador productividad	87
Tabla 19	Plantilla indicador seguimiento	88
Tabla 20	Plantilla indicador problemas en explotación.....	89
Tabla 21	Plantilla indicador percepción del cliente	90

Tabla 22	Plantilla indicador a nivel de proceso	90
----------	--	----

1. PROPÓSITO

Con esta guía avanzada se pretende principalmente dar a conocer en qué consiste el proceso de medición y análisis, es decir, los pasos a seguir para definir y utilizar de forma eficiente las métricas. Va dirigida a empresas del sector de las TIC como ayuda y referencia para que puedan definir sus indicadores como apoyo para su gestión interna.

Primero se comenzará definiendo algunos conceptos y principios clave para entender el proceso de medición y análisis, explicando la importancia de dicho proceso y los beneficios que aporta. A continuación, y tomando como base el método “Goal Question Metrics”, se explicará la técnica para una buena definición de métricas.

También se hará una clasificación de las métricas en función de su nivel de uso (métricas de producto, proyecto, proceso organizativo) dando ejemplos de distintas métricas en los diferentes modelos, y se propondrá un modelo que contemple los contenidos que debería tener una ficha que recoja y muestre los contenidos de cada una de las métricas o indicadores.

Luego se describirán distintas técnicas de análisis de métricas basándose en diferentes métodos de análisis, y diferentes mecanismos para comunicar los resultados de las métricas y su análisis. A continuación, se describirá cómo definir un proceso que integre todas las actividades descritas con anterioridad, es decir, un proceso que aglutine todas las actividades relacionadas con las métricas.

Por último se dará el enfoque que dos importantes metodologías como son SPICE y CMMI® dan a la medición y análisis y un listado de artefactos relacionados con este proceso.

2. CONCEPTOS Y PRINCIPIOS

En general, la medición persigue tres **objetivos fundamentales** [1]: entender qué ocurre durante el desarrollo y el mantenimiento, controlar qué es lo que ocurre en nuestros proyectos y mejorar nuestros procesos y productos.

Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento. En definitiva la medición nos ayuda a:

- **Analizar**, Comprender (los atributos de un ente)
- **Controlar** (la calidad del producto, ...)
- **Estimar** (el tiempo y coste de un proyecto...)
- **Mejorar** (la calidad de un producto, proceso ...)

Un modelo de métricas proporciona un mecanismo formal de unir necesidades de información ya definidas con procesos y productos de ingeniería del software que puedan ser medidos. Desarrollar un plan de medición para un proyecto particular requiere un modelo que sea específico para las necesidades de información del proyecto. El modelo sirve como recurso primario del proceso de medición ayudando en la planificación e implementación de las actividades de recolección y análisis de datos.

El modelo de métricas establece una estructura definida de relación de diferentes términos de medición, estableciendo a su vez unos principios para la definición consistente de las métricas y los métodos de análisis, y reportaje de los resultados. En el mundo de la medición no existe un consenso en cuanto a términos.

El definir e implementar un programa de medición implica tomar muchas decisiones acerca de cómo recoger, organizar y analizar datos. De ahí la importancia de una terminología concisa y consistente, ya que sin ella la comunicación entre analistas de métricas, los proveedores de los datos y los usuarios de dichas métricas sería imposible. Otra razón para esta necesidad de términos bien definidos es que las personas que toman las decisiones en los proyectos de una organización necesitan entender las necesidades de información.

2.1. ¿POR QUÉ MEDIR?

Dada la gran inversión realizada por las compañías para desarrollar y mantener información crítica, hay una demanda creciente de **más evaluaciones objetivas y gestión de**

proyectos software. Todas las organizaciones software exitosas tienen implementadas métricas como parte de sus actividades diarias tanto de gestión como técnicas. Las métricas proporcionan **información objetiva** necesaria para **tomar decisiones** que tengan un impacto positivo en el negocio y en el rendimiento. En estas organizaciones, la información derivada de las métricas es tratada como **un recurso importante** utilizado a todos los niveles de gestión.

La manera en que la medición es llevada a cabo y utilizada en una organización software determina el valor que la organización recibirá en términos de rendimiento. La medición es **más efectiva** cuando es implementada de tal forma que dé soporte a los **objetivos técnicos y de negocio de la organización** y cuando se integra con el resto de actividades de gestión y técnicas ya definidas en la organización. Las métricas son de gran utilidad sobre todo cuando proporcionan **información objetiva** relacionada con los riesgos y problemas que pueden impactar con los objetivos definidos para un proyecto.

El éxito de cada organización también depende de la capacidad de sus empleados de **realizar pronósticos y establecer compromisos claros** en cuanto a los productos que desarrolla, y una vez más las métricas actúan como factor clave.

Las métricas de software son importantes a nivel de proyecto ya que ayudan al jefe de proyecto a hacer mejor su trabajo. Las métricas de software proporcionan información objetiva al jefe de proyecto ayudándole a:

- **Comunicar de forma efectiva:** las métricas proporcionan información objetiva a través de toda la organización, reduciendo así la ambigüedad que a menudo rodea a los proyectos software. Las métricas ayudan a los responsables a identificar, priorizar, seguir y comunicar objetivos y temas asociados a todos los niveles de la organización. También es importante mantener una comunicación entre las organizaciones proveedoras y adquiridoras.
- **Asignar de una forma adecuada recursos** para llevar a cabo las planificaciones definidas previamente.
- **Controlar el progreso y el rendimiento contra los planes establecidos** de la forma adecuada
- **Realizar seguimiento de los objetivos específicos del proyecto:** las métricas describen con mayor precisión el estado de los procesos de proyecto y productos software. Es clave representar de forma objetiva el progreso de las actividades del

proyecto y la calidad de productos software asociados a través del ciclo de vida del proyecto. Las métricas ayudan a responder a preguntas cruciales como: ¿El proyecto está dentro de la planificación?, ¿El software está listo para ser entregado al usuario/cliente?

- **Identificar y corregir problemas de manera temprana:** las métricas facilitan una estrategia de gestión pre activa porque ayudan a la identificación y la gestión de los riesgos. Las métricas fomentan la temprana recuperación y corrección de problemas técnicos y de gestión que pueden ser más difíciles y costosos de resolver en fases más avanzadas del ciclo de vida de desarrollo. Los responsables de la organización utilizan métricas como fuente para anticiparse a los problemas tomando así un enfoque proactivo.
- **Tomar decisiones informadas o basadas en datos:** todo proyecto software está sujeto a ciertas restricciones de presupuesto, agenda, calidad técnica... que deben ser compensadas entre sí y gestionadas para cumplir los objetivos establecidos del proyecto. Las decisiones que se toman sobre un área casi siempre impactan en otras, incluso aunque parezca que no están relacionadas. Las métricas ayudan a la persona que tiene que tomar las decisiones a evaluar estos impactos de forma objetiva y a optimizar el rendimiento del proyecto.
- **Justificar decisiones:** las métricas dan soporte a los responsables para defender las bases de sus estimaciones y planificaciones.
- **Relacionar e integrar la información derivada de otros proyectos.** Esto ayudará al jefe de proyecto a **tomar decisiones utilizando información objetiva.**
- **Definir e implementar planificaciones más realistas**

En definitiva, un proceso efectivo de medición y análisis (MA) proporciona una base adecuada de entendimiento de las capacidades de desarrollo, lo que permite **definir planes viables** para el desarrollo de productos y la prestación de servicios de calidad. Las medidas permiten **detectar tendencias y anticipar problemas** y, por lo tanto, permiten establecer un mejor control de los costes, una reducción de los riesgos, mejorar la calidad y asegurar la consecución de los objetivos de negocio.

El proceso de medición y análisis da soporte al negocio de tal forma que facilita la identificación de los problemas más relevantes sobre los que enfocarse, la decisión de los recursos a utilizar, la definición de acciones correctivas y la mejora de los procesos.

La siguiente tabla muestra los **beneficios típicos** resultantes de un buen proceso de medición y análisis a distintos niveles.

Tabla 1 Beneficios del proceso de medición a nivel de proyecto y organizativo

A nivel de proyecto	A nivel de la organización
<ul style="list-style-type: none"> • Reducir las actividades que no aportan valor al producto que se está desarrollando. • Mejorar el control de costes de desarrollo y mantenimiento de los productos. • Mejorar la gestión de los recursos. • Mejorar la comunicación entre los grupos de trabajo y los departamentos. • Aumentar la eficiencia de los servicios prestados. • Saber a qué nos comprometemos al planificar los proyectos • Conocer cuál es la probabilidad de alcanzar ese compromiso • Saber si estamos en el camino correcto para conseguir los compromisos • Saber si estamos consiguiendo los compromisos de manera efectiva en costes 	<ul style="list-style-type: none"> • Identificar las oportunidades de mejora. Mejorar el flujo de trabajo de los procesos. • Fomentar la visión de la empresa. • Aumentar el Retorno de Inversión. • Mejorar la satisfacción del cliente. • Mejorar la posición de la empresa en el mercado. • Saber si hemos mejorado • Mejorar la visibilidad sobre la situación de los proyectos basada en datos objetivos

2.2. CONCEPTOS BÁSICOS

La medición de software es una disciplina relativamente joven y no existe consenso general sobre la definición exacta de los conceptos y terminología que maneja. En este apartado se explicarán ciertos conceptos clave del proceso de medición que pueden ser de ayuda en la comprensión de la guía.

Para comenzar, empezaremos explicando un concepto que es el origen de las actividades de medición, que es el de cubrir las **necesidades de información** de la organización.

Para dar soporte a las necesidades de información de una organización es importante desarrollar y establecer una capacidad de medición que ayude a proporcionar resultados objetivos que sean útiles para la toma de decisiones y acciones correctivas necesarias.

En todos los niveles de la organización se necesita diferente tipo de información. Las necesidades de información de una organización pueden ser, por ejemplo, la información necesaria para gestionar un proyecto, es decir, sus objetivos, hitos, riesgos y problemas. Otros ejemplos de necesidades de información podrían ser los siguientes:

- Conocer el nivel de productividad del proyecto en comparación con lo habitual en otros proyectos en la organización.
- Determinar si los recursos del proyecto son adecuados para satisfacer sus objetivos.
- Evaluar el rendimiento de la actividad de codificación.
- Evaluar si un producto software satisface los criterios acordados con el cliente.

Las necesidades de información de los responsables de la toma de decisiones a todos los niveles de la organización dirigen la selección de medidas de software y técnicas de análisis asociadas. Estas necesidades normalmente se derivan de unos **objetivos** establecidos por la dirección o de problemas (riesgos, estado de los proyectos) que dificultan la consecución de estos objetivos. La medición no tiene sentido si no hay un responsable de toma de decisiones con una necesidad de información que lo motive.

Una vez que se han definido las necesidades de información de la organización, se puede identificar un conjunto de medidas común que cubra estas necesidades. Una necesidad de información se asocia a un **concepto medible**.

Un concepto medible es una relación abstracta entre **atributos** de una o más **entidades**, y una necesidad de información [ISO 15939]. Es decir, es la relación entre una propiedad medible, física o abstracta de una o más entidades, y una necesidad de información. Una **entidad** puede ser un proceso, un producto, un servicio, un proyecto, o un recurso concreto.

En software hay tres clases de entidades cuyos atributos podemos querer medir:

- **Procesos:** Son actividades software que normalmente conllevan el factor tiempo. Algunos atributos internos interesantes serían por ejemplo el tiempo (duración del proceso), el esfuerzo (asociado al proceso) y el número de incidentes o defectos de un tipo específico que se dan durante el proceso (por ejemplo: el número de errores de requisitos encontrados durante la construcción de la especificación).

- **Productos:** son entregables, artefactos o documentos generados en el ciclo de vida del software. Ejemplos de **atributos externos** son la fiabilidad del código, la inteligibilidad de un documento de especificación, la mantenibilidad del código fuente.... También podemos hablar de **atributos internos** como la longitud, funcionalidad, modularidad o corrección sintáctica de los documentos de especificación.
- **Recursos:** son todos aquellos elementos que hacen de entrada a la producción de software. Por ejemplo, el personal, los materiales, las herramientas y los métodos. Un atributo interesante es el esfuerzo o coste. En el caso del personal, además del esfuerzo y coste, se suele medir la productividad.

Las métricas se concentran en atributos específicos de los productos de trabajo de ingeniería del software y se recopilan a medida que realizan las etapas técnicas para conseguir software de mayor calidad.

Para dejar claro la diferencia entre estos términos vamos a ver un ejemplo.

- Una necesidad de información para una empresa podría ser evaluar la fiabilidad de los enlaces de una página web.
- La entidad por lo tanto sería el producto, la página web.
- El concepto medible podría ser la confiabilidad de un enlace.

Otros ejemplos de conceptos medibles son la calidad, coste, accesibilidad, calidad en uso, relación de productividad de un equipo de desarrollo frente a un grado de productividad objetivo, idoneidad de la tecnología...

Tabla 2 Conceptos en el proceso de medición I

Concepto	Relaciones	Ejemplos
Necesidad de información	<ul style="list-style-type: none"> -Se asocia a un concepto medible -Se satisface con uno o más indicadores. 	Evaluar si un producto SW satisface las expectativas del cliente.
Entidad	<ul style="list-style-type: none"> -Una entidad puede pertenecer a una o más categorías de entidad. -Una medición se realiza sobre los atributos de una entidad. 	Una entidad puede ser un proceso, un producto, un servicio, un proyecto o un recurso en concreto.
Categoría de entidad	<ul style="list-style-type: none"> -Una categoría puede incluir a una o varias categorías de entidades. -Tiene uno o varios atributos 	Procesos, servicios, proyectos, recursos, programas, componentes software...
Atributo	<ul style="list-style-type: none"> -Solo puede pertenecer a una categoría de entidad. -Una medición se realiza sobre los atributos de una entidad -Está relacionado con uno o más conceptos medibles. 	"Tamaño de código fuente" podría ser atributo de la categoría de entidad "programas".
Concepto medible	<ul style="list-style-type: none"> -Se asocia a una o varias necesidades de información. -Puede incluir otros conceptos medibles. -Relaciona uno o más atributos. 	Idoneidad tecnológica, relación de productividad equipo de desarrollo respecto al objetivo.

Otro concepto clave en el proceso de medición es la definición de **métrica**. IEEE (*Institute of Electrical and Electronics Engineers*) define una métrica como una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo determinado. La diferencia entre métrica y **medida**, es que medida es el número o categoría asignada a un atributo de una entidad mediante una medición [ISO 14598-1:1999], es decir, es el valor de la métrica. Por ejemplo, siendo la métrica "líneas de código" la medida podría ser "40.000 líneas de código". Otras medidas relacionadas con otras métricas relacionadas con el tamaño, podrían ser "500 páginas" u "80 clases".

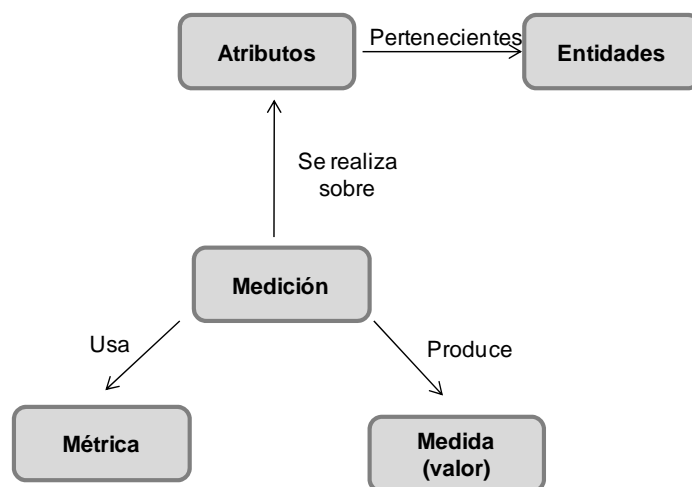


Figura 1 Relación conceptos medición I

Una métrica puede estar definida para uno o más atributos y puede expresarse en una **unidad**. Por ejemplo, la métrica “líneas de código” puede ser definida para realizar mediciones del “tamaño” de un “módulo en C” y para realizar mediciones del “tamaño” de un “programa en Java”.

Una **unidad** es una cantidad particular, definida y adoptada por convención, con la que se pueden comparar otras cantidades de la misma clase para expresar sus magnitudes respecto a esa cantidad particular [ISO 15939]. Por ejemplo, podemos considerar como unidades: Líneas de código, Páginas, Persona-mes, LOC, bytes, palabras, links...

Las métricas del software es un término que se asigna a un amplio rango de actividades diversas, por ejemplo:

- Medidas y modelos de estimación de coste y esfuerzo
- Aseguramiento y control de calidad
- Recogida de datos
- Modelos de fiabilidad
- Modelos y evaluación de ejecución
- Cálculo de complejidad computacional o algorítmica.

Las métricas pueden clasificarse en dos categorías: **Métricas directas e indirectas**.

Una **métrica directa** es una métrica de un atributo que no depende de ninguna métrica de otro atributo, es decir una métrica de la cual se pueden realizar mediciones sin depender de ninguna otra métrica y cuya forma de medir es un método de medición.

Por ejemplo, como métricas directas podemos enumerar:

- **LCF** (líneas de código fuente escritas).
- **HPD** (horas-programador diarias).
- **CHP** (coste por hora-programador, en unidades monetarias).
- **Cantidad de Imágenes con Texto Alternativo:** Medido por la presencia de la etiqueta ALT (con texto no nulo) en cada una de las imágenes vinculadas a las páginas de un sitio Web.

Por el contrario, una **métrica indirecta** es una métrica de un atributo que se deriva de una o más métricas de otros atributos. Las mediciones de dicha métrica utilizarán las medidas obtenidas en mediciones de otras métricas directas o indirectas.

La forma de medir una métrica indirecta es una **función de cálculo**. Además, una métrica indirecta puede usarse en una función de cálculo. Por ejemplo, como métricas indirectas podemos enumerar:

- **HPT** (horas-programador totales).
- **LCFH** (líneas de código fuente por hora de programador).
- **CTP** (coste total actual del proyecto, en unidades monetarias).
- **CLCF** (coste por línea de código fuente).

Las métricas no pueden interpretar por sí solas un concepto medible. De ahí la necesidad de **indicadores**.

Un indicador es una métrica o una combinación de métricas que proporcionan conocimientos acerca de los aspectos de un proyecto de software.

Los indicadores pueden servir de base para cuantificar conceptos medibles para una necesidad de información, y ofrecen información para la toma de decisiones. Un indicador podría ser, por ejemplo, la productividad de los programadores.

Otro concepto muy relacionado con los términos que se acaban de definir es **construcción de medición**. La **construcción de medición** describe cómo los atributos de software relevantes son cuantificados y convertidos en **indicadores** que proporcionan una **base** para tomar decisiones. Una construcción de medida única puede involucrar: métricas directas, métricas indirectas e indicadores.

Hemos estado tratando hasta ahora conceptos medibles y qué medir, pero ¿cómo medir? Pues bien, la forma de medir está a elección de la organización. Se podría seguir un **método de medición** que es una secuencia lógica de operaciones, descritas de forma

genérica, usadas para realizar mediciones de un atributo respecto de una **escala** específica. (Cuando hacemos operativa una definición y derivamos indicadores de medición debemos considerar una escala de medición.)

Además, un método de medición puede usar **instrumentos de medición**. Por ejemplo una herramienta que sirva para contar líneas de código es un instrumento de medición que asiste al método de medición contar líneas de código.

El tipo de método de medición va a depender de la naturaleza de las operaciones utilizadas para cuantificar el atributo: la cuantificación está basada en métodos numéricos.

Algunos ejemplos podrían ser: contar líneas de código, anotar cada día las horas dedicadas por los programadores al proyecto o valorar el grado de dificultad de un problema.

Estos últimos conceptos que se acaban de definir, se recogen en lo que se llama **procedimiento de medición**. Un **procedimiento de medición** define el mecanismo para recoger y organizar los datos requeridos para instanciar una construcción de medida. El procedimiento de medición también contempla los métodos de medición a seguir y el instrumento de medición a utilizar.

Tabla 3 Conceptos en el proceso de medición II

Concepto	Relaciones	Ejemplos
Métrica	<ul style="list-style-type: none"> -Está definida para uno o más atributos -Se expresa en una unidad 	La métrica "líneas de código" puede definirse para realizar mediciones del tamaño de un módulo.
Método de medición	<ul style="list-style-type: none"> -Define una o más métricas directas -Puede usar instrumentos de medición 	Contar líneas de código
Instrumentos de medición	<ul style="list-style-type: none"> -Asiste a uno o más métodos de medición 	Herramienta de inspección de código.
Escala	<ul style="list-style-type: none"> -Una escala y una forma de medir están definidas para realizar medición de uno o varios atributos 	<ul style="list-style-type: none"> -El nivel de madurez CMMI: 1, 2, 3, 4, 5. -Tamaño de código (LOC), conjunto de números naturales.

Por último, el siguiente gráfico muestra la relación de la mayoría de los conceptos vistos en este apartado:

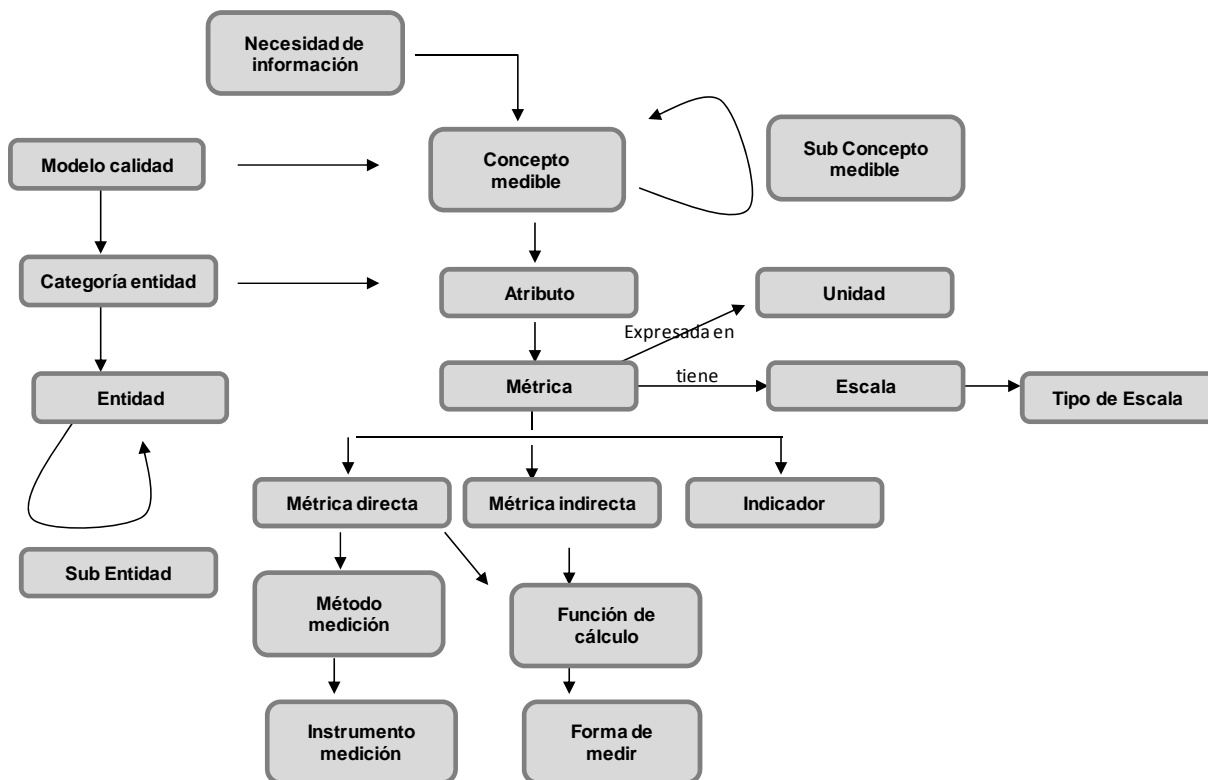


Figura 2 Relación conceptos medición II

3. DEFINICIÓN DE LAS MÉTRICAS

Últimamente ha aparecido un gran número de métricas para capturar atributos del software de una forma cuantitativa. Sin embargo, muy pocas métricas han sobrevivido a la fase de definición y por lo tanto de utilización en la industria. Esto se debe a múltiples **problemas**, entre ellos:

- Las métricas no siempre se definen en un contexto apropiado en el que el objetivo de interés se pueda alcanzar.
- Incluso si el objetivo es explícito, las hipótesis experimentales a menudo no se hacen explícitas.
- Las definiciones de métricas no siempre tienen en cuenta el entorno o contexto en el que serán aplicadas.
- No siempre es posible realizar una validación teórica adecuada de la métrica porque el atributo que queremos medir no siempre está bien definido
- Un gran número de métricas nunca se han validado empíricamente.

Esta situación ha conducido frecuentemente a cierto grado de ambigüedad en las definiciones, propiedades y asunciones de las métricas, haciendo que el uso de las mismas sea difícil, la interpretación peligrosa y los resultados contradictorios.

Para evitarlo es necesario contar con un **método de definición de métricas** y con una base para su formalización. Existen distintos métodos que ayudan en la definición de métricas. En este apartado nos centraremos en el enfoque que **GQM** da a la definición de métricas.

El **enfoque GQM (Goal-Question-Metric)** proporciona una manera útil para definir métricas tanto del proceso como de los resultados de un proyecto. Según este método, un programa de medición puede ser más satisfactorio si se diseña orientado a las metas u objetivos que se quieren alcanzar.

Este método fue originariamente definido por Basili y Weiss (1984) y extendido posteriormente por Rombach (1990) como resultado de muchos años de experiencia práctica e investigación académica.

La diferencia entre metas y objetivos es la siguiente:

- Las **metas** son generales, abstractas e intangibles. Responden a la pregunta de ¿qué queremos alcanzar?, y la respuesta es **cualitativa**. P.ej.: Reducir el tiempo de entrega.
- Los **objetivos** son precisos, tangibles y concretos. Las metas se descomponen en un conjunto de objetivos bien definidos, con la intención de que alcanzando los objetivos alcanzaremos la meta. Responden a la pregunta de ¿cuánto queremos alcanzar?, y la respuesta es **cuantitativa**. P.ej.: Reducir el tiempo de entrega en un 20% al final del año.

GQM define una meta, refina esta meta en preguntas y define métricas que intentan dar información para responder a estas preguntas. Las preguntas ayudarán a medir si se está alcanzando la meta definida, por lo tanto se considerarán preguntas que sean potencialmente medibles.

GQM se puede aplicar a todo el ciclo de vida del producto, procesos y recursos.

GQM sigue un proceso de seis pasos donde, los tres primeros tratan de identificar las métricas a partir de las metas del negocio y los tres últimos se basan en la recopilación de los datos de las medidas y su utilización eficaz en la toma de decisiones.

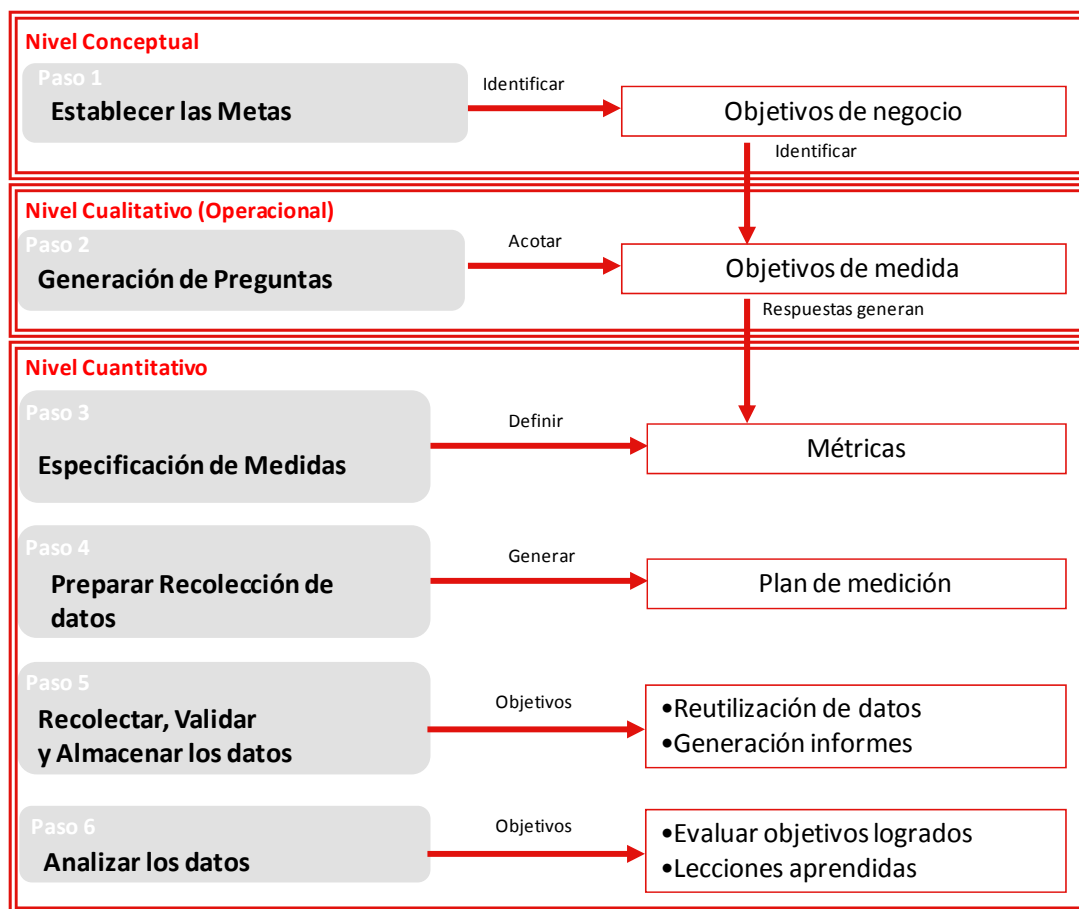


Figura 3 Descripción de los 6 pasos del proceso GQM

En la figura anterior se mencionan distintos niveles, que son los que caracterizan al método GQM:

- **Nivel Conceptual – Goals:** Los objetivos identifican lo que queremos lograr respecto a los productos, procesos o recursos.

Objetos de la medición:

- Productos: entregables y documentos que se producen durante el ciclo de vida de un sistema.
 - Procesos: actividades relacionadas con el software y asociadas generalmente al tiempo.
 - Recursos: elementos que los procesos utilizan para producir sus salidas.
- **Nivel Operacional– Questions:** Las preguntas nos ayudan a comprender cómo satisfacer el objetivo. Tratan de caracterizar al objeto de la medición con respecto a

un aspecto de calidad concreto y tratan de determinar la calidad de dichos objetos desde el punto de vista seleccionado, analizando el grado de cumplimiento de los objetivos específicos. Habrá que tener en cuenta qué atributos tiene el objeto con respecto al objetivo planteado y qué características de los atributos del objeto son importantes con respecto al aspecto de calidad, además de cómo se van a evaluar dichas características.

- **Nivel Cuantitativo – Metrics:** Se asocia un conjunto de datos a cada pregunta, con el fin de proporcionar una respuesta de manera cuantitativa.

Los datos pueden ser:

- **Objetivos:** si dependen únicamente del objeto que se está midiendo y no del punto de vista desde el que se captan (por ejemplo, el número de versiones de un documento).
- **Subjetivos:** si dependen tanto del objeto que se está midiendo como del punto de vista desde el que se captan (por ejemplo, el nivel de satisfacción del usuario).

Como resultado se seleccionan medidas existentes o se definen nuevas medidas.

Para cada meta, puede haber varias preguntas y la misma pregunta se puede ligar a múltiples metas. De la misma forma, para cada pregunta puede haber múltiples métricas y una métrica puede ser aplicable a más de una pregunta.

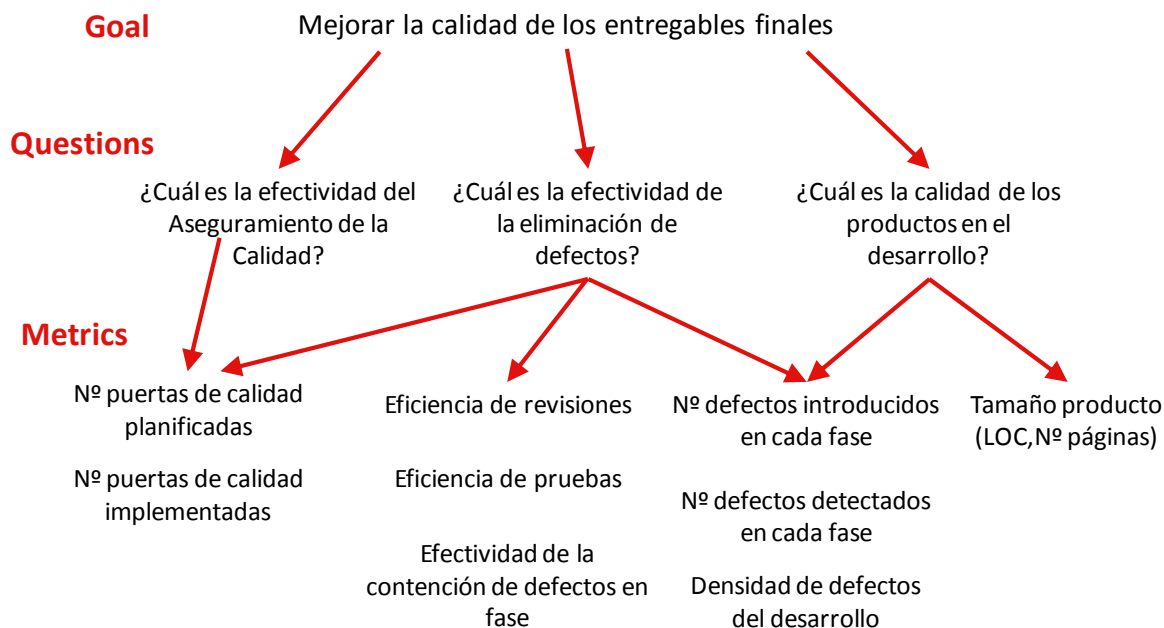


Figura 4 Ejemplo de preguntas y métricas relacionadas con una meta

3.1. IDENTIFICAR NECESIDADES DE INFORMACIÓN

El primer paso del proceso de definición de métricas es el de **identificar las necesidades de información** de la organización. Para dar soporte a las necesidades de información de una organización es importante desarrollar y establecer una capacidad de medición que ayude a proporcionar resultados objetivos que sean útiles para la toma de decisiones y acciones correctivas necesarias.

Según **PSM (Practical Software Measurement)** [2], las necesidades de información se pueden clasificar en **siete categorías**:

- **Calendario y progreso:** esta categoría trata los objetivos de los hitos del proyecto y completitud de unidades de trabajo individuales. Un proyecto que se sale de la agenda, normalmente puede cumplir sus objetivos de entrega sólo eliminando funcionalidades o sacrificando la calidad del producto.
- **Recursos y costes:** esta categoría hace referencia al equilibrio entre el trabajo a realizar y los recursos de personal asignado al proyecto. Un proyecto que exceda el esfuerzo presupuestado normalmente puede recuperarse sólo reduciendo la funcionalidad del software o sacrificando la calidad del producto.

- **Tamaño de producto y estabilidad:** esta categoría trata la estabilidad de la funcionalidad o capacidad requerida del software. También relaciona el volumen de software entregado para proporcionar la capacidad requerida. La estabilidad incluye cambios en el alcance de la funcionalidad o cantidad. Un aumento en el tamaño del software normalmente requiere aumentar los recursos aplicados o la agenda del proyecto.
- **Calidad de producto:** esta categoría trata la habilidad del producto software entregado para dar soporte a las necesidades del usuario sin fallar. Si se entrega un producto de baja calidad, la carga de hacerlo funcionar normalmente recae sobre la organización de mantenimiento asignada.
- **Ejecución de proceso:** esta categoría se refiere a la capacidad de la organización relativa a las necesidades del proyecto. Una organización con un proceso de desarrollo de software pobre o de baja productividad puede tener dificultad al cumplir los objetivos de coste y la programación del proyecto planificada.
- **Efectividad de la tecnología:** esta categoría trata la viabilidad del enfoque técnico propuesto. Trata enfoques de ingeniería como reutilización de software, utilización de componentes software comerciales, fiabilidad sobre procesos de desarrollo de software avanzados, e implementación de arquitecturas de software común. Si no se consiguen los elementos clave del enfoque técnico propuesto puede resultar en aumentos de costes y retrasos en la agenda.
- **Satisfacción del cliente:** esta categoría se ocupa del grado en que los productos o servicios entregados por el proyecto cumplen las expectativas del cliente. Los indicadores de satisfacción pueden obtenerse de la retroalimentación del cliente y de los niveles que sean requeridos el soporte de clientes.

Se han utilizado diferentes enfoques para implementar métricas sobre proyectos software aunque no todos son efectivos. Algunos de estos enfoques se han basado en la **definición detallada de métricas** que se puedan aplicar de forma ecuaníme a cada proyecto de la organización. Otros sin embargo optan por la compra de herramientas de análisis y métricas automatizadas de un proveedor externo sin conocimiento alguno acerca con los procesos de la organización. Pero realmente para que una métrica tenga éxito y ayude a cumplir los objetivos del proyecto en un entorno cambiante, hay que tener muy en cuenta las **necesidades de información**.

La recolección, análisis y los informes que recogen los datos obtenidos de las métricas están relacionados directamente con las **necesidades de información de la organización**. Es decir, las métricas están definidas e implementadas para tratar las necesidades de información específicas del proyecto tal y como se establece en los objetivos del proyecto. A medida que el proyecto avanza, las necesidades de información cambian, y por lo tanto también lo harán las métricas aplicadas.

3.2. DISEÑO DE LAS MÉTRICAS

Las necesidades de información resultan de los esfuerzos de ingenieros y responsables de la organización por mejorar las salidas de los procesos y actividades de software. Las salidas deseables de estos procesos normalmente son definidas en términos de **objetivos**, tal y como se vio en el apartado anterior. Las necesidades de información relacionan directamente los objetivos establecidos con las áreas con más probabilidad de que impacten negativamente en el éxito de dichos objetivos.

El gráfico siguiente muestra cómo una **necesidad de información** puede evolucionar en un plan para generar los productos medibles del proyecto.

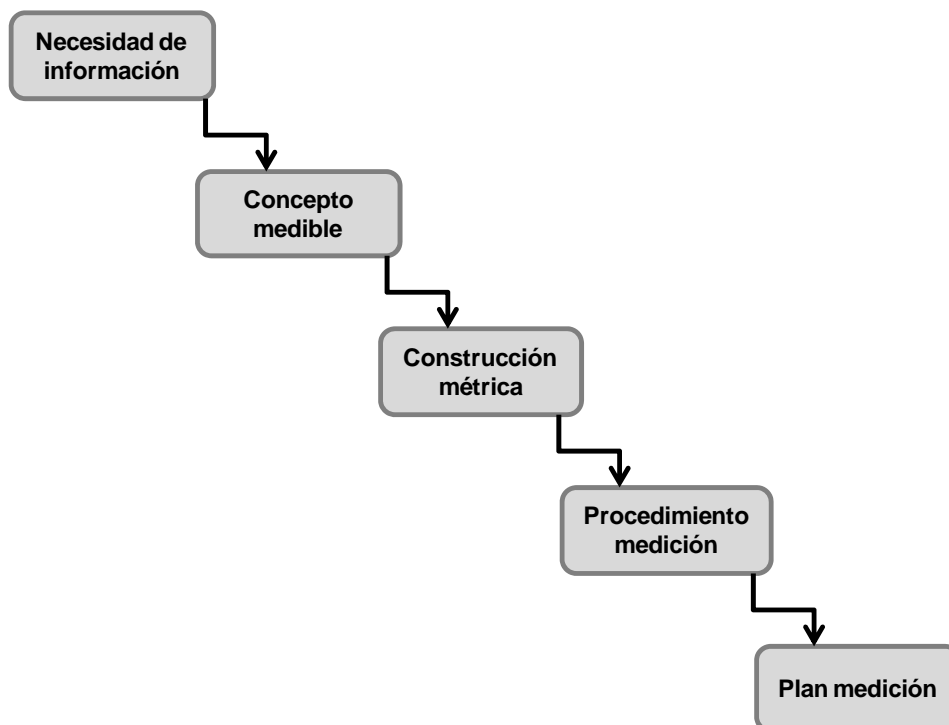


Figura 5 Evaluación de una necesidad de información a un plan de medición

La planificación de la medición comienza con el reconocimiento por parte de algún responsable o jefe de proyecto de una necesidad específica necesaria para la toma de alguna decisión.

Tal y como se definió en el apartado de conceptos, el **concepto medible** es una percepción sobre las entidades que deberían ser medidas con el objetivo de satisfacer una determinada necesidad de información. El concepto medible será formalizado como una **construcción de medida** que especifica exactamente qué será medido y cómo los datos serán combinados para producir resultados que satisfagan la necesidad de información. Dos medidas aplicables a este ejemplo podrían ser el tamaño del software y el esfuerzo.

El siguiente gráfico muestra la estructura básica de una **construcción de medida**.

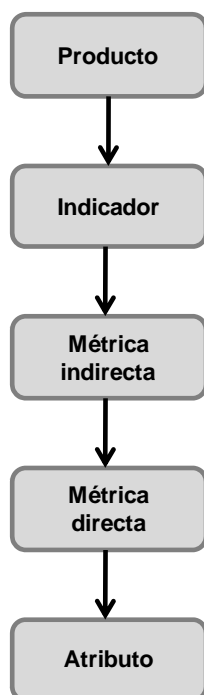


Figura 6 Niveles de una construcción de medida

Todo lo que puede realmente ser medido incluye atributos específicos de procesos y productos software, como tamaño, esfuerzo y número de defectos.

La persona encargada de la planificación del programa de medición especificará los detalles de las construcciones de medición que se van a usar, al igual que los procedimientos para recoger los datos, analizarlos y realizar informes, y plasmarlo en el plan de medición. Cuanto mejor sea el **diseño de la construcción de medición**, mejor se relacionarán los atributos

medibles del software con las necesidades de información identificadas y más sencillo será para el jefe de proyecto tomar decisiones objetivas.

El formalizar un **concepto medible** en una **construcción de medición** implica considerar los detalles de los procesos y productos del proyecto software y una descripción detallada de la necesidad de información asociada. Estos detalles ayudarán en la implementación de un programa de medición efectivo y eficiente. Los **beneficios** más destacados del uso de construcciones de medición bien definidos incluyen:

- **Aumentar la exactitud**, asegurando que todos los aspectos esenciales del enfoque de medición estén definidos de una manera adecuada.
- **Maximizar el valor de las métricas base** creando patrones de métricas derivadas e indicadores que puedan ser fácilmente reconocidos, reutilizados y adaptados.
- **Documentar la relación entre las necesidades de información y cómo se satisfacen estas necesidades.**
- **Reducir la redundancia de métricas**, ayudando a identificar un conjunto de métricas base (primitivas) que puedan servir para distintos propósitos.

Todas las necesidades de información, las construcciones y los procedimientos de medición son combinados en un **plan de medición** del que se hablará en apartados posteriores. La ejecución de este plan de medición produce **productos de información** que responden a las necesidades de información del proyecto. Los productos de información son un conjunto de indicadores, interpretaciones y recomendaciones proporcionadas a la persona encargada de tomar la decisión como salida del proceso de medición.

Un plan de medición podría tratar una única necesidad de información, pero en la mayoría de los casos dirige varias de ellas. El plan define:

- Qué necesidades de información son aplicables para un proyecto en particular
- De qué manera se medirá el software para satisfacer estas necesidades de información
- Cómo será gestionado el proceso de medición y con qué recursos se cuenta

En definitiva, desarrollar un **programa de medición efectivo** requiere un entendimiento exhaustivo de las necesidades de información de software del proyecto, un conjunto de conceptos medibles bien definidos y relacionados, y un conocimiento total de las entidades de software disponibles para ser medidas dentro del proyecto.

El siguiente gráfico representa los conceptos clave dentro de la construcción de métricas:

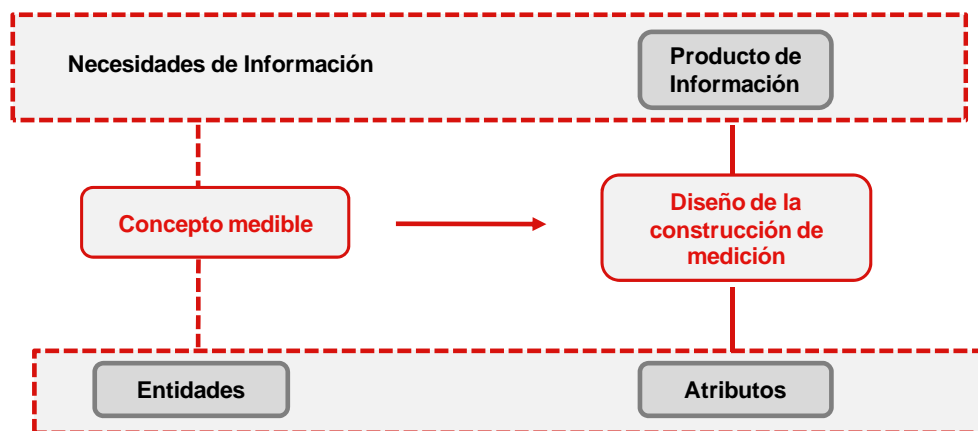


Figura 7 Relación conceptos medición III

4. CLASIFICACIÓN/TIPOS DE MÉTRICAS

Existen distintas clasificaciones de las métricas del software. A más alto nivel las métricas del software se pueden clasificar en 3 niveles: *métricas de producto y servicio*, *métricas de proceso* y *métricas de proyecto*.

Las **métricas de producto y de servicio** describen las características del producto, como tamaño, complejidad, características de diseño o rendimiento, y el nivel de calidad del servicio prestado.

Las **métricas de proceso** se pueden utilizar para mejorar el desarrollo y el mantenimiento del software. Las métricas del proceso dependen esencialmente del entorno de desarrollo. Un ejemplo de este tipo de métricas es el tiempo empleado para desarrollar un elemento software que dependa de factores externos tales como la capacidad del personal, la metodología empleada...

Las **métricas de proyecto** describen las características propias del proyecto y de su ejecución. Algún ejemplo puede ser el tamaño en punto de función (PF), esfuerzo estimado, distribución de esfuerzo por fase del ciclo de vida, distribución de duración por fase del ciclo de vida.

Vamos a entrar en cada uno de estos niveles en más profundidad clasificando distintos tipos de métricas en cada uno de dichos niveles:

4.1. A NIVEL DE PROYECTO

Las métricas a nivel de proyecto son **métricas tácticas**. Estas métricas van a facilitar a los responsables de los proyectos el adaptar el flujo de trabajo del proyecto y las actividades técnicas a los resultados obtenidos, pudiendo hacer un seguimiento del estado del proyecto, y de los cambios que vaya sufriendo.

La calidad de la gestión de los proyectos se apoya, en primer lugar, en la base de una buena planificación. Los proyectos se deben ejecutar de acuerdo con lo establecido en sus planes de proyecto, y al objeto de comprobar que progresan adecuadamente, se efectuará sobre ellos un seguimiento periódico y completo a lo largo de toda su vida, de tal forma que se cubran todos los factores críticos (eficacia y eficiencia) para su éxito y se asegure que las posibles desviaciones puedan ser detectadas y corregidas a tiempo.

Una de las aplicaciones principales de las métricas de proyectos son las **estimaciones**. Las métricas recopiladas de proyectos anteriores, por ejemplo, se podrían usar como base para realizar estimaciones de esfuerzo y tiempo del proyecto actual. A medida que un proyecto avanza, las medidas del esfuerzo y del tiempo consumido se comparan con las estimaciones originales, es decir, con la planificación del proyecto. El responsable del proyecto utilizará estos datos para supervisar y controlar el avance del mismo.

La utilización de métricas del proyecto tiene dos aspectos fundamentales [McDermid' 91]:

- Estas métricas se utilizan para facilitar el minimizar la planificación de desarrollo, guiando los ajustes necesarios que eviten retrasos y atenúen problemas y riesgos potenciales.
- Las métricas del proyecto se usan para evaluar la calidad de los productos en el momento actual, y cuando sea necesario, se modificará el enfoque técnico de la mejora de la calidad.

Otro modelo de métricas del proyecto [Hetzel, B.] sugiere que todos los proyectos deberían medir:

- **Entradas:** la dimensión de los recursos que se requieren para realizar el trabajo.
- **Salidas:** medidas de las entregas o productos creados durante el proceso de ingeniería del software.
- **Resultado:** medidas que indiquen la efectividad de las entregas.

En realidad este modelo se puede aplicar tanto al proceso como al proyecto. En el contexto del proyecto, el modelo se puede aplicar recursivamente a medida que aparece cada actividad estructural.

En definitiva, las métricas del proyecto permiten al gestor de proyectos del software [Pressman]:

- Evaluar el estado del proyecto en curso
- Seguir la pista de 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.

Tabla 4 Ejemplo de métricas durante la gestión de proyectos

Gestión de proyectos	
Planificación	Calidad del plan de proyecto
	Progreso adecuado
Ejecución	Cumplimiento de tiempos
	Cumplimiento de hitos
	Cumplimiento de esfuerzos
	Cumplimiento de costes
	Cumplimiento de alcance
	Cumplimiento de RR.HH.
Seguimiento	Existencia de seguimiento
	Continuidad de seguimiento
	Compleción del seguimiento
Productividad	Porcentaje de trabajo repetido (<i>rework</i>)

En realidad, las medidas que recopila un equipo de proyecto y que convierte en métricas para utilizarse durante un proyecto, también pueden transmitirse a los que tienen la responsabilidad de mejorar el proceso del software. Por esta razón, se utilizan muchas métricas similares tanto en el dominio del proceso como en el del proyecto.

4.2. A NIVEL DE PROCESO

Las métricas del proceso de software se utilizan para **propósitos estratégicos**, a diferencia de las orientadas al proyecto, que como se vio en el apartado anterior eran tácticas.

La única forma de mejorar cualquier proceso es:

- Medir sus atributos
- Desarrollar un conjunto de métricas significativas en función de estos atributos
- Finalmente utilizar estas métricas para proporcionar indicadores que conducirán a una estrategia de mejora.

Por ejemplo, la efectividad de un proceso de desarrollo de software se mide indirectamente, es decir, se extrae un conjunto de métricas según los resultados que provienen de la instanciación del proceso. Estos datos pueden venir de la ejecución de un proyecto, y a partir de esos datos consolidados de distintos proyectos se puede tener una visión del comportamiento del proceso. Entre los resultados se incluyen medidas de errores detectados antes de la entrega del software, defectos detectados y reportados por los

usuarios finales, productos de trabajo entregados (productividad), esfuerzo humano y tiempo consumido, cumplimiento de la planificación y otras medidas.

Las métricas del proceso de desarrollo del software pueden proporcionar beneficios significativos a medida que una organización trabaja por mejorar su nivel global de madurez del proceso.

Vamos a considerar el siguiente ejemplo para ver ejemplos de métricas a este nivel. Las pruebas son un proceso dentro de la organización, y como tal, se pueden tomar métricas para gestionar el estado de la calidad de ese proceso. Una vez que se desarrolla un plan de pruebas, las actividades y planificación tienen que ser revisados contra lo que realmente se está realizando.

Los datos necesarios para realizar el seguimiento del progreso de las pruebas se pueden recoger manualmente (p.ej.: contando los casos de prueba desarrollados cada día) o mediante alguna herramienta de gestión de pruebas. Estos datos de progreso también se utilizan para medir criterios de salida como la cobertura de pruebas (p.ej.: 50% de cobertura de requisitos alcanzada).

Algunas **métricas del proceso de pruebas** más comunes son:

- Porcentaje de trabajo realizado en la preparación de los casos de prueba (o porcentaje de casos de prueba planificados que han sido preparados).
- Porcentaje de trabajo realizado en la preparación del entorno de pruebas respecto a lo planificado.
- Ejecución de casos de pruebas (p.ej.: número de casos de prueba ejecutados/no ejecutados, y casos de prueba pasados/fallados).
- Información de defectos (p.ej.: número de defectos/PF, defectos encontrados y corregidos, índice de fallos y resultados de repetir las pruebas).
- Cobertura de pruebas de requisitos.
- Desviación de hitos en las pruebas.
- Costes de las pruebas.

A continuación se definen algunas **métricas a nivel de proceso**.

Una de ellas es la **eficiencia de las revisiones**. Esta métrica es de gran utilidad para obtener visibilidad sobre la eficiencia de las revisiones de los productos. Una mejor eficiencia ayuda a mejorar la productividad global del proyecto. La eficiencia de las revisiones se define como el número de defectos detectados en revisiones entre el esfuerzo que se ha dedicado para llevarlas a cabo, teniendo en cuenta otros factores tales como el tamaño o la complejidad del elemento a revisar.

Una manera de evaluar la efectividad de las revisiones y de las pruebas en los proyectos sería utilizando la métrica de **contención de defectos en fase**, que sería el número de defectos detectados en una determinada fase respecto al total de defectos que se han originado. Obtener un alto valor de la métrica indica la efectividad en las revisiones y en las pruebas de los productos.

Con el objetivo de obtener visibilidad sobre la efectividad de los mecanismos de verificación y validación se utiliza otra métrica, la **efectividad en la eliminación de defectos**. Esta métrica muestra la relación entre los defectos detectados durante las pruebas y las revisiones y los defectos detectados después de la entrega del producto.

4.3. A NIVEL DE PRODUCTO Y SERVICIO

La calidad del desarrollo de los productos software, debe basarse en la **satisfacción de las expectativas** de sus usuarios finales, por ello, se deberá asegurar que cumplen con los requisitos y atributos de calidad especificados para los mismos, ya que este debería ser uno de los objetivos principales de toda organización.

También debería asegurarse que la documentación desarrollada será útil para el usuario: completa (contendrá la información esperada), correcta (cumpliendo tanto en contenido técnico como formal con lo establecido en la normativa) así como fácil de utilizar y mantener.

De la misma forma, las aplicaciones desarrolladas deberán ser correctas, fiables, usables y fáciles de mantener y evolucionar.

Otro de los objetivos es proporcionar un **servicio de calidad** tanto en lo relativo a la explotación de los sistemas (conjunto de plataformas hardware, software y comunicaciones sobre las cuales se construyen y explotan las aplicaciones de usuario) como al mantenimiento de las aplicaciones (productos software que resuelven las necesidades del usuario final), garantizando que éstos se encuentren disponibles, que tengan un rendimiento adecuado, que sean seguros y que sobre ellos se realice un mantenimiento correcto, en los plazos establecidos y con los recursos asignados.

Tabla 5 Ejemplos de métricas a nivel de producto y servicio

Producto físico	
Tamaño	Número de componentes
	Líneas de código
	Puntos función
	Volumen de software
Reutilización	Número de componentes llamados más de n veces
	Número de llamadas
	Volumen de reutilización
	Ratio de reutilización
Desarrollo de productos	
Documentación	Calidad de documentación de proyecto
	Calidad de documentación de soporte
Calidad de código	Porcentaje de código comentado
	Complejidad ciclomática
	Código muerto
Calidad de pruebas	Cobertura de pruebas
	Densidad de defectos
Servicios	
Calidad de explotación	Disponibilidad del sistema
	Eficacia del sistema
	Seguridad del sistema
Satisfacción del cliente	
Grado de satisfacción	Reparos en explotación
	Grado de satisfacción del cliente

4.3.1. Métricas de calidad de producto final

Ahora vamos a describir algunas de las métricas que se podrían tomar en este nivel sobre la calidad de un producto:

La calidad de producto se mide normalmente por la **densidad de defectos** y el **tiempo medio de fallo**. La primera métrica mide los defectos relativos al tamaño del software (líneas de código, puntos función, etc.) y la segunda mide el tiempo que transcurre entre la detección de fallos.

La **densidad de defectos** de un producto o el número esperado de defectos en un determinado periodo de tiempo es importante para estimar el coste y recursos de la fase de mantenimiento del ciclo de vida del software. En esta métrica el numerador es el número de defectos. Sin embargo, desde el punto de vista del usuario, todos los problemas que encuentre al utilizar el software, no solamente los defectos, son problemas con el software. Estos problemas que no son defectos pueden ser problemas de usabilidad, documentación o información que no está clara, o incluso errores de los usuarios. Por eso, otra métrica importante relacionada con la calidad del producto es la que mide los **problemas encontrados por el cliente** al utilizar el producto.

También existe una métrica directamente relacionada con la **satisfacción del cliente**. La satisfacción del cliente, normalmente, se mide mediante encuestas realizadas al cliente, que puntúa en una escala de cinco puntos: muy satisfecho, satisfecho, neutral, insatisfecho, muy insatisfecho.

Algunos de los parámetros que se evalúan en las encuestas de satisfacción del cliente son: capacidad, funcionalidad, usabilidad, rendimiento, fiabilidad, mantenibilidad, documentación /información y servicio.

4.3.2. Métricas de mantenimiento de software

Todas las métricas que se mencionaron en el apartado anterior están relacionadas con el producto final, pero cuando finaliza el desarrollo de un producto software y se lanza al mercado, comienza la fase de mantenimiento dentro del ciclo de vida del producto.

Durante esta fase, las métricas que se suelen recoger son:

- la llegada de defectos por intervalo de tiempo
- las incidencias del cliente (que pueden ser defectos o no) por intervalo de tiempo

Sin embargo, el número de defectos o problemas que llegan viene determinado por el proceso de desarrollo antes de la fase de mantenimiento. Ya no se puede hacer mucho para alterar la calidad del producto en esta fase. Por tanto, estas dos métricas, aunque son importantes, no reflejan la calidad del mantenimiento del software. Lo que se debe hacer en esta fase de mantenimiento es corregir los defectos lo antes posible y hacerlo con buena calidad. Estas acciones, pueden mejorar en gran medida la satisfacción del cliente. En este sentido, las siguientes métricas de mantenimiento son muy importantes.

- **Tiempo de respuesta en las correcciones**

En muchas organizaciones de desarrollo de software, se establecen guías sobre el tiempo límite en el que las correcciones de los defectos registrados deben estar disponibles. La métrica de tiempo de respuesta en las correcciones se calcula para todos los problemas, según su nivel de severidad, como el tiempo medio desde que se registra un problema hasta que se cierra.

- **Porcentaje de correcciones atrasadas**

Una corrección se clasifica como atrasada si el tiempo empleado en ella supera el tiempo de respuesta requerido.

- **Calidad de las correcciones**

La calidad de las correcciones o el número de correcciones defectuosas es otra métrica importante de calidad para la fase de mantenimiento. Desde la perspectiva del cliente, es malo encontrar defectos funcionales en el software pero es peor aún si las correcciones de estos defectos no solucionan el problema registrado, o lo solucionan pero introducen un nuevo defecto.

La métrica del porcentaje de correcciones defectuosas es simplemente el porcentaje de todas las correcciones en un intervalo de tiempo que son defectuosas.

5. MODELO DE FICHA INDICADOR

Como se definió en el apartado de *Conceptos básicos*, un **indicador** es una métrica o combinación de métricas que proporcionan una visión exhaustiva del proceso, del proyecto o del producto software en sí.

En este apartado se propone una forma de recoger y mostrar los contenidos de un indicador mediante una ficha. Primero se hará una descripción de lo que debería contener la ficha, es decir, se identificarán y definirán los distintos campos en que la ficha está dividida. A continuación se mostrará una tabla a modo de plantilla con todos los campos de la ficha. Algunos de los campos tienen un asterisco “*” justo antes del nombre. Esto indica los campos que debería ser obligatorio rellenar.

Una vez más, esto es sólo una forma de representar una ficha de un indicador, pero cada organización debe utilizar la que crea más conveniente o ajustarla a sus necesidades.

Los **campos** que contiene la ficha indicador serían los siguientes:

Nombre indicador: Se podría utilizar cualquier nombre para el indicador, aunque éste debería ser claro y definir el significado del indicador para que cualquier persona pueda entender o hacerse una idea de lo que representa dicho indicador.

Vamos a poner un ejemplo que nos ayudará a entender las distintas partes de la ficha del indicador. Utilizaremos un indicador ampliamente utilizado en el ámbito de los procesos de medición como es el número de líneas de código. El nombre del indicador podría ser: **Número de Líneas de Código**.

Identificador o código del indicador: Conjunto de letras, símbolos y/o números que representan al indicador. Es la forma corta de hacer referencia a los indicadores. Normalmente el identificador de un indicador suele contener letras de su nombre original o incluso si el nombre del indicador está formado por varias palabras, el identificador puede ser la primera letra de cada palabra.

En este caso, y basándonos en el ejemplo anterior, el identificador más ampliamente utilizado para el indicador número de líneas de código es **LOC**. Este identificador proviene de las primeras letras del nombre compuesto del indicador en inglés (**Lines of Code**).

Tipo de indicador (eficacia, eficiencia...): Este campo indicará la categoría a la que pueda pertenecer el indicador, es decir, el nivel o grupo que contenga indicadores de su misma dimensión o que se toman con un objetivo similar.

En el caso del número de líneas de código, podría pertenecer a la categoría tamaño del producto, ya que ayuda a medir el tamaño de los productos. Dentro de esta categoría podría estar el indicador “puntos función” por ejemplo.

Objetivo o fundamento del indicador: Este campo hace referencia al objetivo que se persigue con el cálculo de ese identificador, es decir, el motivo por el cual se necesita conocer el valor de ese indicador.

En este caso el objetivo de querer conocer el número de líneas de código es medir el tamaño del software.

Descripción del indicador: En este campo se daría una descripción del indicador. Se podrían explicar, por ejemplo, los factores que pueden afectar al indicador o los elementos que forman parte del mismo, o simplemente qué se ha tenido en cuenta a la hora de crear el indicador. En el caso de que el indicador tenga una fórmula para su cálculo, se podrían explicar los componentes de la fórmula, el comportamiento del indicador en función de esos componentes...

En el caso de LOC, como no tiene una fórmula como tal para su cálculo, en este campo se podría explicar qué líneas de código se van a considerar a la hora del recuento, es decir, si se van a contar sólo las líneas de código ejecutables, o también los comentarios ya que la no especificación puede llevar a errores. Vamos a suponer en este caso que el código que estamos midiendo es un código en java y que solamente vamos a considerar como líneas de código aquellas líneas ejecutables para determinar su tamaño.

Ámbito del desempeño: En este campo se va a especificar el ámbito del indicador, es decir, dónde poder aplicar el indicador. En el caso de nuestro ejemplo podríamos decir que el ámbito del indicador líneas de código sería el propio código fuente a partir del cual se leen las líneas.

Fuente de información: El campo fuente de información contendrá el origen de los datos que hemos utilizado para obtener el indicador, es decir, se explicará de donde se han tomado los datos que se utilizaron y el método utilizado para obtener dichos datos. También se podría incluir la fiabilidad de la fuente de información utilizada.

La fuente de información del indicador líneas de código podría ser el recuento realizado por la herramienta de inspección de código XXX que tiene una fiabilidad de 100%.

Datos de entrada: El campo de la ficha datos de entrada contiene los datos que se utilizan en la fórmula de cálculo del indicador. Si se tratase de una métrica indirecta, en este apartado se indicarían las métricas directas utilizadas para su cálculo. En el caso del indicador líneas de código, los datos de entrada serían el propio código fuente, es decir, lo que se necesita para calcular el valor del indicador.

Definiciones y abreviaturas: En este campo se describen las variables que se usan en la fórmula del indicador y así poder entender con más facilidad la fórmula. También se podría incluir la definición de las abreviaturas utilizadas en una fórmula, en el caso de que las hubiera. Si, por ejemplo, la fórmula contiene variables cuyos nombres son excesivamente largos para escribirlos en la ficha del indicador en ocasiones se suelen utilizar abreviaturas. O incluso hay variables cuyas abreviaturas son ampliamente utilizadas para hacer referencia a las mismas. En estos casos que se usen abreviaturas, se definirán aquí.

En el caso del indicador que estamos utilizando como ejemplo este campo no aplicaría.

Fórmula de cálculo: En este campo se especificará la fórmula de cálculo del indicador. En caso de que el indicador no tenga fórmula para su cálculo, se describiría el algoritmo en el que se basa el indicador o técnicas que se utilizan para el cálculo del mismo.

En el caso del indicador líneas de código se podría especificar que la fórmula de recuento de líneas de código es **Sum (Líneas de código)**. También se podrían describir las técnicas que se están utilizando para su recuento o el algoritmo de recuento de líneas de código, ya que dependiendo del algoritmo se podrían considerar:

- Sólo líneas ejecutables
- Líneas ejecutables y definiciones de datos
- Líneas ejecutables, definiciones de datos y comentarios
- Líneas físicas en una pantalla de entrada de datos
- Líneas que terminan con un delimitador lógico...

Vamos a suponer, en nuestro caso, que el algoritmo de recuento de líneas que vamos a utilizar es el tener en cuenta sólo líneas que sean ejecutables.

Escala o unidad de medida: En este campo se define la escala o unidad utilizada para el cálculo de la fórmula del indicador. En el caso del indicador que tenemos como ejemplo, la escala o unidad sería las líneas de código.

Criterios de Análisis: En este apartado se describen pautas que ayudarán a entender el valor que se obtiene de la fórmula del indicador, es decir la forma en que se deben interpretar los datos obtenidos del cálculo de los indicadores. Incluso se pueden incluir recomendaciones describiendo las acciones que deberían tomarse en función de los resultados obtenidos, por ejemplo “para que el programa pueda ser fácilmente probado y mantenible se recomienda que ningún módulo tenga una complejidad ciclomática superior a 8 (valor del indicador)” o “si se obtiene un valor muy alto de defectos, el equipo del proyecto debería llevar a cabo un análisis de la causa raíz para poder tomar medidas de prevención de defectos para eliminar su aparición en un futuro”...

En el caso particular de nuestro ejemplo, en este campo podríamos especificar el siguiente criterio de análisis:

La densidad de defectos (Nº defectos/LOC) disminuye con el tamaño, pero cuando los módulos son muy grandes la densidad de defectos tiende a incrementarse. Hay, por tanto, un tamaño óptimo de un programa que implica el mínimo número de defectos. Este tamaño óptimo depende del lenguaje, el proyecto, el producto y el entorno y se necesitará una investigación empírica (p.ej.: basada en versiones previas de un mismo producto, basada en productos similares desarrollados por el mismo equipo) para poder utilizar esta información como guía para el desarrollo de un módulo nuevo.

Datos históricos: Este campo refleja la evolución del indicador a lo largo del tiempo. En este campo el indicador se puede desglosar en distintos campos, uno por cada variable de la fórmula del indicador, y así ver la evolución no sólo del indicador sino de las variables de las que depende el indicador. Para visualizar la evolución de las variables, y por lo tanto del indicador, de forma clara los datos históricos podrían trasladarse a un gráfico

Todos estos valores se pueden clasificar, por ejemplo, en 3 grupos: histórico (datos del pasado dividido en años o meses...), resultados para el año actual (primer trimestre, tercer trimestre), valores proyectados (para años o meses... futuros).

Basándonos en el ejemplo de líneas de código, este campo no tendría sentido desglosarlo porque este indicador simplemente depende del número de líneas de código, que es la variable. Entonces se puede ir registrando el número de líneas de código de cierto producto software a lo largo del ciclo de vida de desarrollo del producto para ver su evolución.

Meta para el año XXX: En este campo se pueden indicar los objetivos y las metas que se esperan para el año siguiente por ejemplo. Se puede establecer un umbral dentro del cual debe estar el indicador.

Para el caso del número de líneas de código este campo no aplicaría ya que no tiene sentido poner como meta llegar a un determinado número de líneas de código.

Frecuencia de reporte: Este campo recoge el periodo o fechas de actualización del indicador (anual, mensual, trimestral). Es decir, aquí se recoge la frecuencia de la medición. Por ejemplo, el número de líneas de código se podrían tomar mensualmente. Sería interesante indicar la audiencia a quien va dirigido ese indicador.

Información complementaria u observaciones: En este campo se especificaría cualquier tipo de observación o información que debería tenerse en cuenta en relación con el indicador. En nuestro ejemplo, en este campo se reflejaría la dependencia de este indicador al tipo de lenguaje de programación y los criterios utilizados para tomar este indicador. En este caso habíamos determinado que se iban a contar el número de líneas de código ejecutables, sin contar comentarios.

Versión: En este campo se especifica la versión del indicador. Por ejemplo, versión 1.0.

Fecha: Aquí se especificaría la fecha actual, para hacer constancia de la fecha en que se creó el indicador y poder realizar un seguimiento del indicador con más facilidad.

Responsables del indicador: En este campo se especificarían los responsables o área/s responsable/s del indicador o de su cumplimiento. Este campo se puede dividir a su vez en responsable del cálculo y responsable del seguimiento y análisis del indicador.

Firmas de los responsables: En este campo los responsables especificados en el campo anterior plasmarían sus firmas.

Tabla 6 Plantilla ficha indicador

PLANTILLA FICHA IDENTIFICADOR								
*Nombre Indicador:			*Identificador:			Tipo identificador:		
*Objetivo del indicador:						*Ámbito del desempeño:		
*Descripción del indicador								
Fuente de información:			Datos de entrada:			Definiciones y abreviaturas:		
*Fórmula de cálculo:						*Escala o unidad:		
*Criterios de análisis:								
*Datos históricos								
Variables	Datos Históricos			Resultados año actual				Valores proyectados
	Año 2006	Año 2007	Año 2008	Trimestre I	Trimestre II	Trimestre III	Trimestre IV	Año 2010
Nombre Variable I								
Nombre Variable II								
Nombre Variable III								
.....								
.....								
Nombre								

Indicador								
Metas para el año XXX								
Metas								
Trimestre I		Trimestre II		Trimestre III		Trimestre IV		
...			
Frecuencia del reporte:								
Información complementaria u observaciones:								
*Responsables y firmas:								
Versión:					*Fecha:			

6. TÉCNICAS DE ANÁLISIS

Una vez que se han definido unos objetivos a nivel de la organización y se han especificado unas métricas que proporcionarán información cuantitativa sobre los objetivos fijados, hay que desarrollar mecanismos para la recolección de datos.

Dentro del **plan de medición** ya definido se tienen que contemplar todos los procedimientos detallados para la recolección de datos:

- Definición y descripción de medidas
- Resultados posibles de las medidas
- Persona o rol encargado de la recolección de las medidas
- Cuándo se deben recoger las medidas
- Dónde se van a almacenar los datos
- Qué medios o herramientas automáticas se van a utilizar.

Es importante formar a las personas involucradas en la recolección de datos para asegurar que entienden por qué los datos son necesarios, cómo van a ser utilizados y cómo sus acciones contribuyen a la validación total del proceso de recolección.

Una vez que los datos se han recogido y son tratados, proporcionarán **realimentación** a los proyectos en una acción correctiva. La recolección de datos es un proceso inútil si no se hace nada con ellos. Si el análisis de los datos sólo se utiliza para mostrar lo que se ha conseguido será un proceso meramente informativo, pero no será más útil que leer el periódico del día anterior. Las métricas deben ser analizadas inmediatamente después de calcularlas, comparándolas con los objetivos fijados.

Si hay una desviación sobre los objetivos, ya sean organizativos, de proceso o de proyecto, o no se alcanzan las metas, se deben iniciar **acciones correctivas y preventivas**. Posibles acciones en este sentido son:

- Actualizar la documentación del proceso
- Utilizar distintos métodos o ciclos de vida en los proyectos
- Introducir controles adicionales mediante la verificación y validación
- Utilizar distintas tecnologías o herramientas
- Actualizar la infraestructura o sistema de medición
- Automatizar algunas actividades del proyecto
- Actualizar las habilidades del personal

Por todo ello será necesario definir en un **plan de análisis** cómo se van a organizar los datos, qué métricas o datos se van a analizar, los métodos que se van a utilizar para analizarlos, qué objetivos se pretenden satisfacer con el análisis o qué información se pretende obtener del análisis, a quién se va a presentar y cómo.

6.1. MÉTODOS DE ANÁLISIS

Las herramientas estadísticas promovidas por Ishikawa para el control de la calidad son ampliamente utilizadas y se han convertido en una parte importante dentro del control de la calidad. Se las conoce como las siete herramientas básicas de Ishikawa y se suelen utilizar para analizar las métricas de software.

Estas herramientas estadísticas se pueden aplicar a nivel de proyecto y de organización, por lo tanto, son útiles tanto para jefes de proyecto como para expertos en procesos pero no van a proporcionar información específica a los desarrolladores de software sobre cómo mejorar la calidad de sus diseños o sus implementaciones. Además, los beneficios de estas estadísticas puede que no se alcancen en el caso de proyectos pequeños, donde los patrones estadísticos de los parámetros de un proceso de desarrollo son poco obvios.

Las **siete herramientas básicas de Ishikawa** son:

- Checklist (o *check sheet*)
- Diagrama de Pareto
- Histograma
- Diagrama de dispersión
- Gráfico de ejecución
- Gráfico de control
- Diagrama de causa-efecto

En este apartado, se describirá la aplicación de estas herramientas para el control de calidad y procesos en el desarrollo de software.

6.1.1. Checklist

La checklist es un formulario con elementos que deben ser comprobados. Su **propósito** principal es ayudar en la recogida y organización de datos para facilitar su posterior análisis.

La checklist es una herramienta importante para asegurar la consistencia de la implantación de un proceso, y se puede aplicar a muchas fases y actividades a lo largo del proceso de desarrollo software. Por ejemplo, puede ayudar a los desarrolladores a asegurar que todas las tareas se completan y que se cubren los factores importantes o las características de calidad de cada tarea.

Algunas **características y consideraciones** a tener en cuenta con las checklists son:

- Las checklist no deberían ser largas.
- Deberían desarrollarse basándose en la experiencia del equipo.

- Las checklists deberían ser revisadas y actualizadas de forma periódica.
- Para derivar indicadores cuantitativos de las checklists, un equipo puede medir los porcentajes de los criterios de la checklist que han sido alcanzados para las actividades clave del proceso.

Algún tipo de checklist podría ser una hoja de confirmación de una serie de características de calidad de un proceso o producto, o una lista de errores frecuentes que podría utilizarse como parte de un proceso de prevención de defectos, en el cuál se analicen los defectos para encontrar su causa y se sugieran acciones para poder evitarlos.

6.1.2. Diagrama de Pareto

El **diagrama de Pareto** es un gráfico de barras de frecuencia en orden descendente. Las barras de frecuencia se asocian normalmente con tipos de problemas.

En el desarrollo de software, el eje **X** del diagrama de Pareto es normalmente la causa del defecto y el eje **Y** es el número de defectos. Si se ordenan las causas de los defectos por la frecuencia de los mismos, el diagrama nos permitirá identificar las principales causas que originan la mayoría de los defectos. De esta forma, nos indica qué problemas deberían solucionarse primero para obtener una mejora y un mayor retorno de inversión.

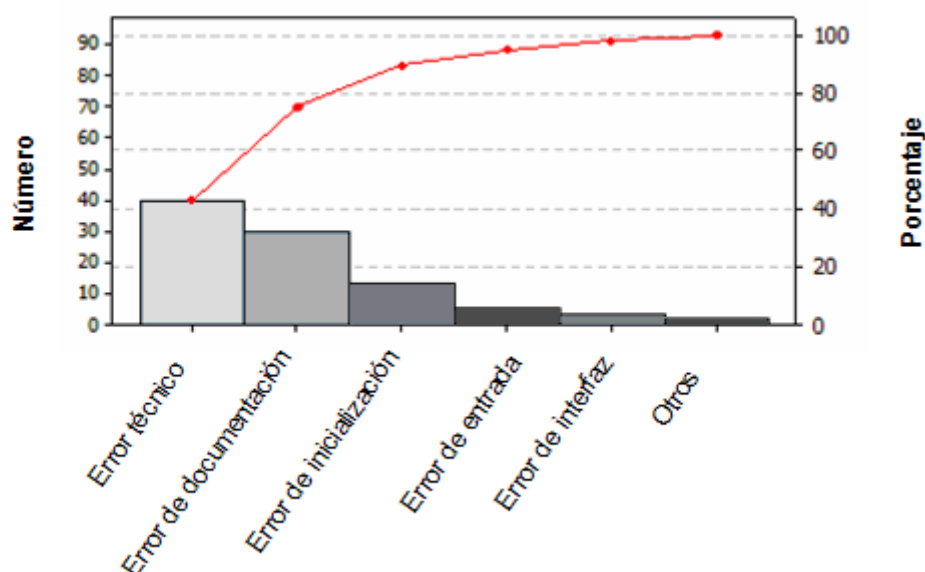


Figura 8 Ejemplo Diagrama de Pareto para errores

Algunos ejemplos de aplicaciones de esta herramienta son:

- Identificar las principales causas de los cambios de requisitos, permitiendo tomar acciones correctivas.
- Centrarse en los tipos de defectos más frecuentes, determinando posibles causas e implantando mejoras en los procesos, permitiendo alcanzar una mejora en la calidad.

6.1.3. Histograma

Un histograma es una representación gráfica de frecuencias dentro de una muestra o población. El eje **X** lista los intervalos de unidades de un parámetro (por ejemplo el nivel de severidad de defectos del software) ordenados de forma ascendente de izquierda a derecha, y el eje **Y** contiene números de frecuencias.

El propósito del histograma es mostrar las características de distribución de un parámetro, como tendencia central, dispersión y asimetría. El histograma ayudará a comprender mejor el parámetro de interés.

A continuación, se muestra un ejemplo de un histograma utilizado en proyectos de software y gestión de calidad que muestra la frecuencia de defectos de un producto por nivel de severidad (de 1 a 4, siendo 1 el más severo y 4 el menos). Los defectos con distintos niveles de severidad difieren en su impacto en los clientes:

- Los **defectos menos severos** normalmente tienen alguna solución alternativa disponible, y para los clientes simplemente suponen una molestia.
- Los **defectos de severidad alta** pueden ocasionar caídas del sistema y afectar al negocio del cliente.

Por eso, dado un mismo índice de defectos (o número de defectos), el histograma de severidad de los defectos nos dirá más sobre la **calidad del software**.

Distribución de defectos por nivel de severidad

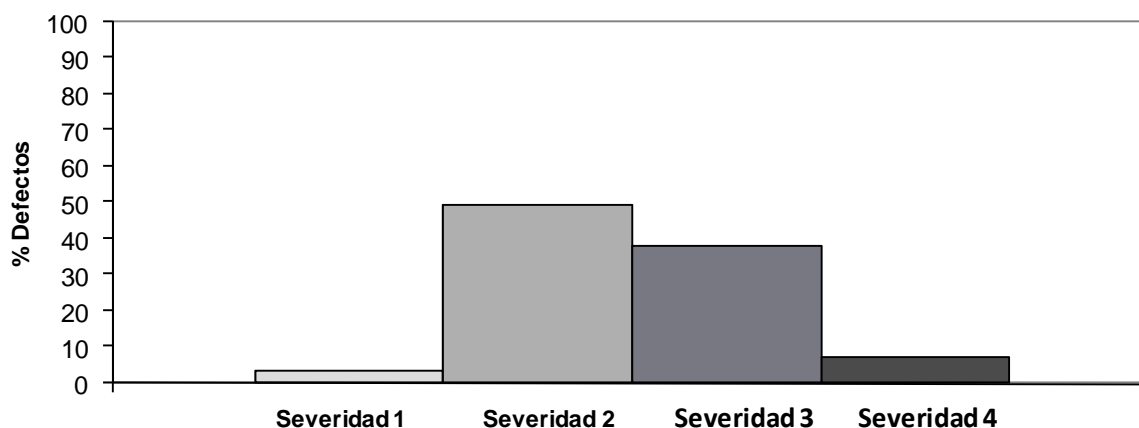


Figura 9 Ejemplo histograma

En un histograma, la escala de medición de los datos es ordinal, de intervalo, o índice. Si la escala es nominal (p.ej.: tipos de software, modelos de procesos de desarrollo), no tiene sentido ordenar el eje X. Por eso, estos gráficos se denominan gráficos de barras en lugar de histogramas.

6.1.4. Diagrama de dispersión

Un **diagrama de dispersión** es una herramienta útil para identificar una relación potencial entre dos variables. En una relación causa-efecto, el eje **X** se utiliza para la variable independiente y el eje **Y** para la variable dependiente.

Cada punto de un diagrama de dispersión representa una observación de ambas variables. Estos diagramas están orientados a la toma de decisiones basadas en datos (por ejemplo si se planifica una acción en la variable X y se espera algún efecto en la variable Y).

Se debe utilizar un diagrama de dispersión cuando se quiera presentar el **coeficiente de correlación** entre dos variables ya que es un método muy sensible a los valores atípicos y muestra claramente cualquier valor atípico en la relación.

Los tipos de correlación que se pueden producir son:

Tabla 7 Tipos de correlación

	X	Y
Correlación positiva	aumenta	aumenta
Posible correlación positiva	aumenta	aumentará algo pero parece condicionado por algún otro aspecto distinto de X
Sin correlación	no hay correlación aparente entre las dos variables	
Posible correlación negativa	provocará una tendencia de disminución en Y	parece condicionada por algún otro aspecto distinto de X
Correlación negativa	aumenta	disminuye

Algunos ejemplos de diagramas de dispersión incluyen relaciones entre defectos, índice de calidad de un mismo componente entre la versión actual y otras anteriores, la relación entre el índice de defectos en pruebas y en producción, etc.

En el desarrollo de software, la **reutilización** es uno de los factores más significativos para mejorar la productividad. Sin embargo, la calidad del software nuevo a menudo se ve afectada por limitaciones en el diseño o defectos latentes en el código heredado. Supongamos que tenemos dos plataformas y que algunos productos desarrollados en la segunda han reutilizando componentes de productos desarrollados en la primera. Para ver la relación entre el índice de defectos de los componentes reutilizados entre los dos sistemas, utilizaremos un diagrama de dispersión como el siguiente:

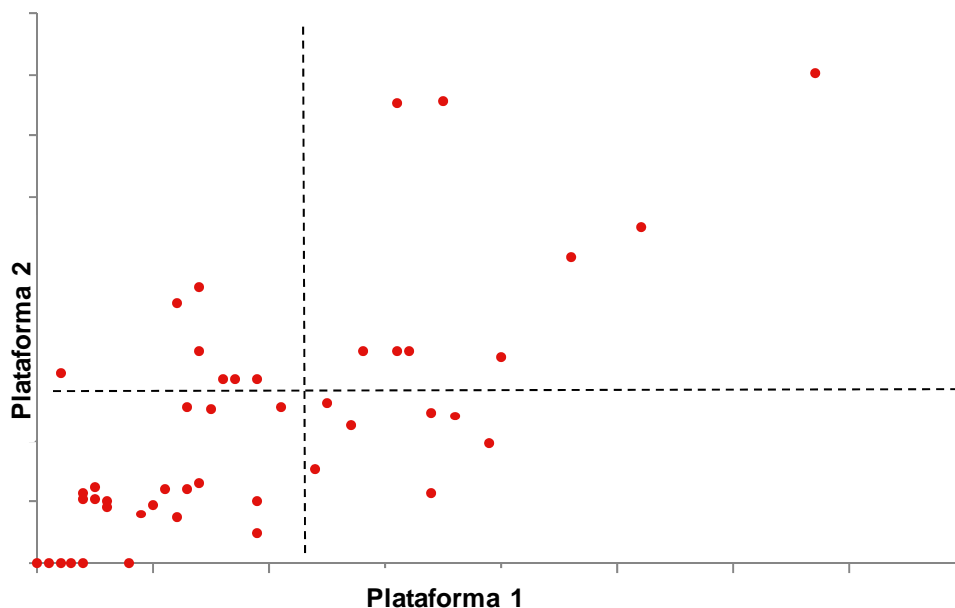


Figura 10 Ejemplo diagrama de dispersión

La figura anterior es el diagrama de dispersión de un producto. Cada punto representa un componente, con la coordenada **X** indicando su índice de defectos en la plataforma 1 y la coordenada **Y** indicando el índice de defectos en la plataforma 2.

6.1.5. Gráfico de ejecución

Un gráfico de ejecución registra el rendimiento de un parámetro de interés a lo largo del tiempo. El eje **X** es el tiempo y el eje **Y** es el valor del parámetro. Un gráfico de ejecución se utiliza en los análisis de tendencias, especialmente si hay datos históricos disponibles para realizar comparaciones con la tendencia actual. Ishikawa considera varios tipos de gráficos de ejecución: gráficos circulares, gráficos de barras, gráficos de barras compuestos.

Los gráficos de ejecución se utilizan frecuentemente en la gestión de proyectos software. Por ejemplo, la llegada de defectos en una semana y el trabajo atrasado en corrección de defectos durante las fases de pruebas del sistema se pueden monitorizar mediante estos gráficos.

Estos gráficos sirven como **indicadores** en tiempo real tanto de la calidad como de la carga de trabajo. A menudo, los gráficos de ejecución se comparan con datos históricos o modelos de forma que la interpretación se haga desde una perspectiva adecuada.

Otro ejemplo podría ser registrar el porcentaje de correcciones en el software que superan el tiempo de respuesta, según el criterio correspondiente. El objetivo es asegurar entregas a tiempo de las correcciones a los clientes.

En el siguiente gráfico, la línea horizontal sería el objetivo a alcanzar.

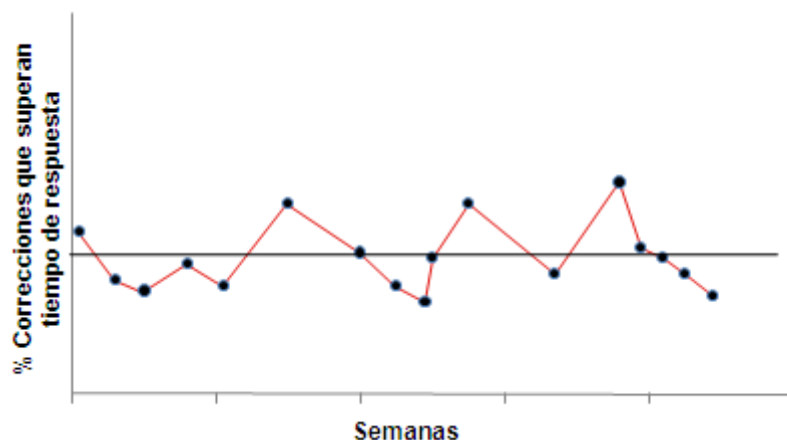


Figura 11 Ejemplo gráfico de ejecución

Otro tipo de gráfico de ejecución utilizado por muchas organizaciones de desarrollo de software para gestión de proyecto y calendario es la **curva S**, que registra el progreso acumulado de un parámetro de interés en el tiempo, comparado con el plan. El eje **X** de la curva S representa las unidades de tiempo y el eje **Y** representa valores del parámetro de interés. Por ejemplo, la curva S del progreso de pruebas debe contener la siguiente información:

- **Progreso planificado** en términos de número de casos de prueba completados por semana (o la unidad de tiempo que se considere).
- Número de **casos de prueba** que se han **abordado** por semana.
- Número de **casos de prueba** que se han **completado** por semana.

El siguiente gráfico sería un ejemplo:

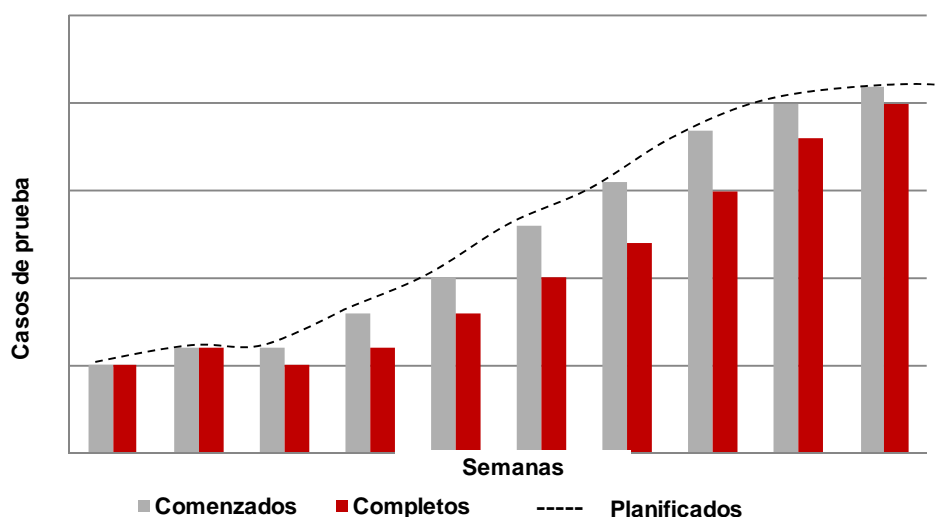


Figura 12 Curva S del progreso de pruebas

El propósito de esta métrica es seguir el progreso de las pruebas y compararlo con el plan para poder tomar acciones en el momento en que haya algún indicador de que las actividades de prueba no se están cumpliendo. Cuando hay riesgos de no cumplir el calendario del proyecto, se suelen sacrificar las pruebas del desarrollo. Al contar con una métrica formal del progreso de las pruebas, es más difícil que se ignore el problema. Además, desde el punto de vista de la planificación del proyecto, la curva S ayuda a realizar una mejor planificación.

6.1.6. Gráfico de control

Un gráfico de control puede considerarse como un tipo avanzado de gráfico de ejecución en situaciones donde se puede definir la capacidad de un proceso. Consiste en:

- Una línea central (valor medio de todos los datos).
- Un par de límites de control (a veces también un par de límites de peligro dentro de los límites de control).
- Valores del parámetro de interés trazados en el gráfico, que representan el **estado de un proceso**. Si todos los valores del parámetro:
 - o están dentro de los límites de control y no muestran una tendencia particular, se dice que el proceso está en un estado controlado.
 - o se encuentran fuera de los límites de control o se observa una tendencia, el proceso se considera fuera de control.

En estos casos, será necesario analizar las causas y tomar acciones correctivas. Permite ver patrones de comportamiento en el tiempo y proporciona una base para predecir el futuro.

El gráfico de control es una herramienta potente para:

- Realizar un control de procesos de forma estadística.
- Definir la capacidad de un proceso dentro del desarrollo de software.
- Mejorar la consistencia y estabilidad.

Hay muchos tipos de gráficos de control. Los más comunes son:

- Gráficos de **control de variables** (P.ej.: X Bar-R, X-MR Charts)
- Gráficos de **control de atributos**

Las **variables** y los **atributos** son los tipos de datos que puede contener un gráfico de control. Las variables son una medida (a menudo continua) de la calidad de un elemento o evento como, por ejemplo, tamaño de módulo o complejidad, puntos función por personas o días en cerrar informes de defectos. Los atributos son recuentos de elementos que poseen un atributo o recuento de eventos que ocurren en un área de oportunidad (medida discreta) como, por ejemplo, tanto por cierto de casos de prueba pasados en la primera ejecución, número de módulos con defectos o tanto por cierto de correcciones rechazadas.

6.1.7. Diagrama de causa-efecto

El diagrama causa-efecto es una herramienta que ayuda a identificar, clasificar y poner de manifiesto posibles causas, tanto de problemas específicos como de características de calidad. Ilustra gráficamente las relaciones existentes entre un resultado dado (efectos) y los factores (causas) que influyen en ese resultado. Es decir, ayuda a clasificar y relacionar las interacciones entre factores que están afectando al resultado de un proceso.

Algunas de las ventajas de este diagrama son:

- Permite que el grupo se concentre en el contenido del problema, no en la historia del problema ni en los distintos intereses personales de los integrantes del equipo.
- Ayuda a determinar las causas principales de un problema, o las causas de las características de calidad, utilizando para ello un enfoque estructurado.
- Estimula la participación de los miembros del grupo de trabajo, permitiendo así aprovechar mejor el conocimiento que cada uno de ellos tiene sobre el proceso.
- Incrementa el grado de conocimiento sobre un proceso.

Para desarrollar un diagrama causa-efecto, también denominado diagrama de espina de pescado (*fishbone diagram*) hay que decidir primeramente cual va a ser la característica de calidad que se va a analizar. Esta característica irá en la flecha central del diagrama. De la flecha principal, que es totalmente horizontal, saldrán una serie de flechas que representarán los factores causales más importantes y generales que puedan generar la fluctuación de la característica de calidad. Es decir, serán flechas secundarias que irán hacia la principal. A continuación se irán incorporando en cada rama factores más detallados que se puedan considerar causas de fluctuación.

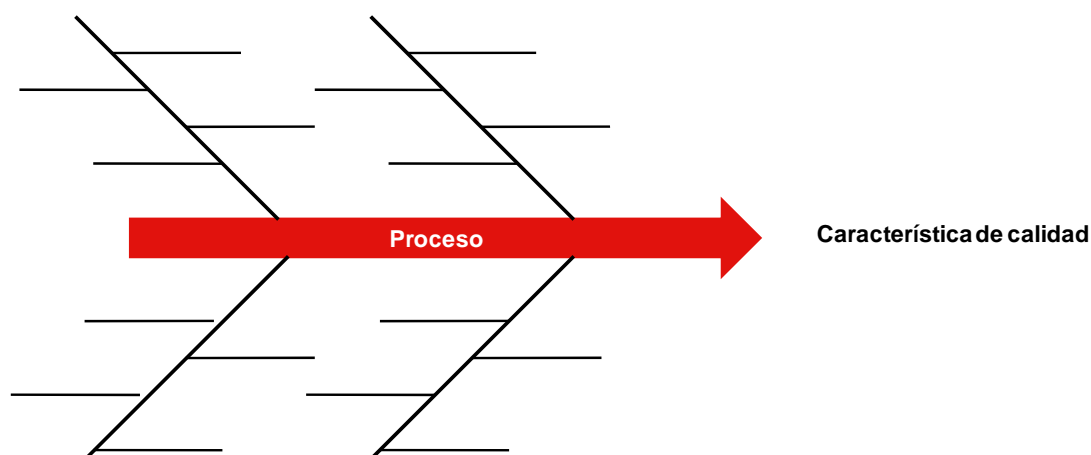


Figura 13 Ejemplo diagrama causa-efecto

Un diagrama de causa-efecto también puede verse como un diagrama educativo ya que sirve para que la gente conozca en profundidad el proceso con que trabaja, visualizando con claridad las relaciones entre los efectos y sus causas. Sirve también para guiar las discusiones, al exponer con claridad los orígenes de un problema de calidad y encontrar

más rápidamente las causas asignables cuando el proceso se aparta de su funcionamiento habitual.

7. MECANISMOS PARA HACER EL REPORTING DE LOS INDICADORES

Toda organización desea prosperar y alcanzar el éxito dentro de su área profesional. Para ello, debe establecer una serie de metas u objetivos de negocio a alcanzar, es decir, definir claramente qué es lo que la empresa debe conseguir para mejorar y garantizar que los procesos organizativos conducen a la empresa hacia el éxito buscado.

Una vez que han analizado su misión, identificado a los agentes implicados y definido sus metas, necesitan una forma de medir el progreso hacia dichas metas.

Una de las claves para la gestión efectiva es monitorizar regularmente los factores críticos que determinan el éxito del negocio. Para ello, las empresas necesitan una serie de medidas clave que les permitan saber cómo se está desarrollando el negocio. La forma en la que muchas empresas hacen esto es mediante los *Key Performance Indicators (KPIs)*. El mayor valor de los KPIs es la posibilidad de ayudar a crear una cultura de mejora continua y de trabajo en equipo dentro de la compañía. Los KPIs proporcionan una oportunidad para mantener a todos los implicados (dirección, empleados, proveedores) centrados en lo que se necesita hacer para mejorar el rendimiento y mantener el negocio.

Existen herramientas que permiten monitorizar, controlar y gestionar los procesos de una organización a través de indicadores de rendimiento alineados con los objetivos estratégicos, los **cuadros de mando**. En este apartado se abarcarán dichas herramientas y se detallarán algunos de los tipos de uso más extendido como son los **dashboards** y los **scorecards**. Por último se expondrán una serie de **mejores prácticas** que se pueden tener en cuenta a la hora de construir cuadros de mando.

7.1. CUADROS DE MANDO

Los cuadros de mando son herramientas de gestión del rendimiento que presentan al usuario una visualización de los indicadores empresariales. Permiten monitorizar, controlar y gestionar los procesos de una organización a través de códigos semafóricos que establecen alertas con las que disponer de una visión completa del rendimiento de la compañía. Permiten comprobar, por ejemplo, si la actividad diaria está alineada con la estrategia corporativa o interpretar lo que está ocurriendo y saber si debemos tomar medidas de mejora.

Algunos de los **beneficios** que proporcionan los cuadros de mandos incluyen:

- Presentación visual de medidas de rendimiento.
- Eliminación de entrada de datos duplicados.
- Capacidad de identificación y corrección de tendencias negativas.
- Medidas de eficacia/ineficacia.
- Capacidad de generación informes detallados para mostrar nuevas tendencias.

- Incremento de ingresos globales.
- Alineación de estrategias y metas organizacionales.

A continuación se presenta una **clasificación** de cuadros de mando en función de su contenido:

- **Dashboards:** se conocen así los cuadros de mando que muestran información sin compararla con objetivos propuestos. Es decir, con ellos únicamente se pueden visualizar los *KPI (Key Performance Indicator)*.

La percepción común de la industria es que un *dashboard* es, en tiempo real, como un cuadro de mandos de un automóvil que permite al conductor comprobar la velocidad actual, el nivel de gasolina, la temperatura del motor, todo de un vistazo. Un *dashboard* está unido directamente a los sistemas que capturan eventos en el momento que suceden y avisa a los usuarios de alertas o notificaciones de excepciones cuando el rendimiento se desvía de alguna de las métricas establecidas.

- **Scorecards:** Son cuadros de mando que muestran información estratégica y están orientados a mostrar objetivos, por lo tanto, además de ofrecer los indicadores KPI, permiten almacenar en el sistema los *KGI (Key Goal Indicador)*.

La percepción común de la industria sobre un *scorecard* es que muestra imágenes periódicas del rendimiento asociado a los objetivos y planes de la organización.

Los *scorecards* miden la actividad del negocio contra los objetivos predefinidos para ver si el rendimiento está dentro de los rangos aceptables. La selección de los KPIs ayuda a los ejecutivos a comunicar la estrategia y centrar a los usuarios en las tareas de mayor prioridad.

Mientras que un *dashboard* informa a los usuarios qué están haciendo, un *scorecard* les dice cómo de bien lo están haciendo. En conclusión, un *dashboard* es un sistema de monitorización del rendimiento mientras que un *scorecard* es un sistema de gestión del rendimiento, es decir, traza progreso.

Tabla 8 Comparativa entre dashboard y scorecard

	Dashboard	Scorecard
Propósito	Muestra el rendimiento	Muestra el progreso
Uso	Monitorización del rendimiento	Gestión del rendimiento
Actualizaciones	Alimentación en tiempo real	Imágenes mensuales
Datos	Eventos	Resúmenes
Medidas	Métricas	KPI
Contexto	Excepciones, alertas	Objetivos, umbrales
Fuente	Unido a los sistemas	Unido a los planes

7.2. MEJORES PRÁCTICAS PARA CONSTRUIR CUADROS DE MANDO

Las mejores prácticas son un conjunto de recomendaciones que ayuden a construir un cuadro de mandos más útil, eficiente y estéticamente satisfactorio. Estas prácticas comprenden todas las etapas de desarrollo del cuadro de mandos, desde su diseño a su implantación.

Estas mejores prácticas incluyen las siguientes recomendaciones:

1. Identificar el usuario final

La primera práctica y la más importante a la hora de construir un cuadro de mandos es identificar quién es el usuario a quien va dirigido. Un cuadro de mandos que esté dirigido a un ejecutivo de una compañía y uno dirigido a un director de marketing será muy diferente. Durante todo el proceso de desarrollo hay que tener en cuenta al usuario final, ya que la aplicación deberá hacerse a la medida de ese usuario. Este proceso de ‘hacer a medida’ puede incluir cosas muy simples como la colocación de controles o las fuentes de datos, pero también puede incluir cosas más complicadas como el flujo del cuadro de mandos entero, o la vista de datos seguros a través de una conexión segura.

Lo importante es que presente la información que interesa y sea útil al usuario final.

2. Utilizar la representación gráfica adecuada

Un requisito difícil a la hora de construir un cuadro de mandos es decidir qué tipo de visualización de datos usar para diferentes tipos de los mismos.

Los datos se pueden presentar de las siguientes formas:

- Mapas: Para datos geográficos (Compras por provincia).
- Gráficos: Datos en el tiempo, datos proporcionales, comparación de datos lineales.
- Indicadores: Datos instantáneos, valores simples (KPIs).

- OLAP (Procesamiento analítico en línea): Datos multidimensionales.
- Diagramas: Otros datos.

A partir de esta lista, debería ser fácil desglosar cualquier dato en uno de los grupos y mostrarlo con la apropiada representación gráfica.

También cabe la posibilidad de hacer un cuadro de mandos dinámico, de tal forma que el usuario sea libre de cambiar la visualización de los datos según las necesidades. Esto es una buena opción, ya que puede existir una perspectiva que el usuario desee y que no haya sido establecida durante el desarrollo.

3. Identificar los KPIs correctamente

Un KPI es una medida cuantificable que refleja los factores que contribuyen al éxito dentro de una compañía. Normalmente, ciertas personas dentro de la compañía o una consultora externa acuerdan estas medidas de antemano, pero hay ocasiones en las que estas medidas las define el desarrollador del cuadro de mandos.

Es imprescindible que los KPIs se elijan correctamente, ya que si no es así pueden ofrecer información incorrecta que induzca a malas decisiones. Hay que dedicar tiempo a investigar cuales son los indicadores de éxito de una organización o grupo para hacer un cuadro de mandos.

4. Dar un contexto a la información presentada

El contexto es una cuestión que está olvidada en muchos cuadros de mando. Esto es incomprensible, ya que sin contexto, muchos cuadros de mandos no son útiles.

Consideremos la siguiente figura:



Figura 14 Ejemplo datos sin contexto

Después de echar un vistazo rápido a la figura anterior, se podría hacer la siguiente afirmación: Pedro está haciendo muy buenas ventas y los demás no. Sin embargo, esto puede no ser del todo cierto.

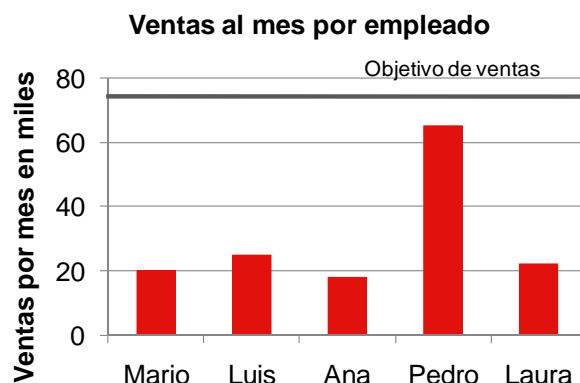


Figura 15 Ejemplo datos con contexto

Esta segunda figura muestra los mismos datos con contexto. En este caso, se ve de forma clara en el gráfico que ninguno de los empleados ha llegado este mes al objetivo de ventas propuesto.

Dar contexto a los datos puede parecer un paso obvio, pero es la práctica más omitida entre los desarrolladores de cuadros de mando. En la mayoría de los casos, el contexto se olvida porque la persona que crea el cuadro de mando ha trabajado tanto con los datos que sabe cuando éstos son buenos o malos. Esta suposición, sin embargo, no puede aplicarse al usuario final ya que puede traer falsas conclusiones.

5. Establecer la distribución de los datos

La distribución incluye el diseño y el emplazamiento de controles dentro de un cuadro de mandos. La distribución tiene un gran impacto ya que determina si un usuario puede o no usar el cuadro de mandos fácilmente, y debería ofrecer un orden lógico y fluido con respecto a los datos específicos que se muestran. El cuadro de mandos debería cumplir también con la regla de 3 y 10 segundos; en 3 segundos el usuario debería tener una idea de todo el rendimiento global, y en 10 segundos el usuario debería tener una idea general de por qué se está consiguiendo ese rendimiento. Por ejemplo, si un usuario estuviera mirando un cuadro de mandos de marketing bien diseñado, a los 3 segundos sabría qué campañas se están desarrollando en la empresa actualmente y cómo se está haciendo cada una, y en 10 segundos sabría qué campañas se están haciendo mejor y quizás tendría alguna idea de por qué. Si el cuadro de mandos no es demasiado intuitivo como para cumplir con la regla de 3 a 10 segundos entonces no será útil.

6. Establecer la estética visual

La estética visual incluye animación, paletas, efectos 2D y 3D, y la apariencia real de los controles. Está muy relacionado con la distribución de un cuadro de mandos pero afecta a la apariencia estética de cuestiones específicas dentro de un cuadro de mandos más que al diseño general. Mientras que la estética visual es importante a la hora de hacer un cuadro de mandos atractivo, los desarrolladores deben tener cuidado con que los efectos visuales no interfieran con la usabilidad y la eficiencia del cuadro de mandos.

7. Personalizar las vistas

Una vez que se ha decidido qué información es importante para un usuario en un cuadro de mandos, es una buena práctica el proporcionar algunas capacidades de personalización de las vistas. Este punto es especialmente cierto en cuadro de mandos basados en OLAP donde la información es multidimensional y la única forma de tener una dibujo coherente de la información es verla desde todos los ángulos. De la misma forma, dar al usuario la capacidad de cambiar la perspectiva de los datos permite a veces ver tendencias o cambios importantes dentro de ellos que el usuario no sería capaz de ver de otra manera.

8. INTEGRAR LAS MÉTRICAS EN UN PROCESO ORGANIZATIVO

En este apartado se va a describir la manera de integrar el proceso de medición y análisis en una organización, es decir, cómo integrar las métricas en un proceso organizativo. Para ello, nos hemos apoyado en un proceso de referencia como es **PSM** (*Practical Software and System Measurement*) que se basa en normas o estándares tales como ISO/IEC 15939, ISO 9000 y CMMI.

PSM es un proceso de medición orientado a información que fue desarrollado para cumplir los objetivos de negocio y técnicos de una organización. El proceso de medición es un proceso flexible, que se puede adaptar para satisfacer las necesidades de información y los objetivos tanto a nivel organizacional como a nivel de proyectos específicos.

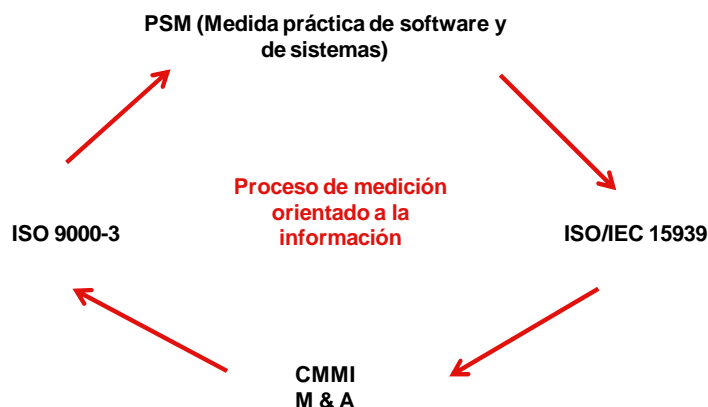


Figura 1 Normas y estándares en el proceso de medición y análisis

PSM proporciona una serie de mejores prácticas utilizadas por profesionales de la medición dentro de la adquisición de sistemas y del software y comunidades de ingeniería.

PSM define un proceso general de medición y análisis que se detalla a continuación en el siguiente gráfico:

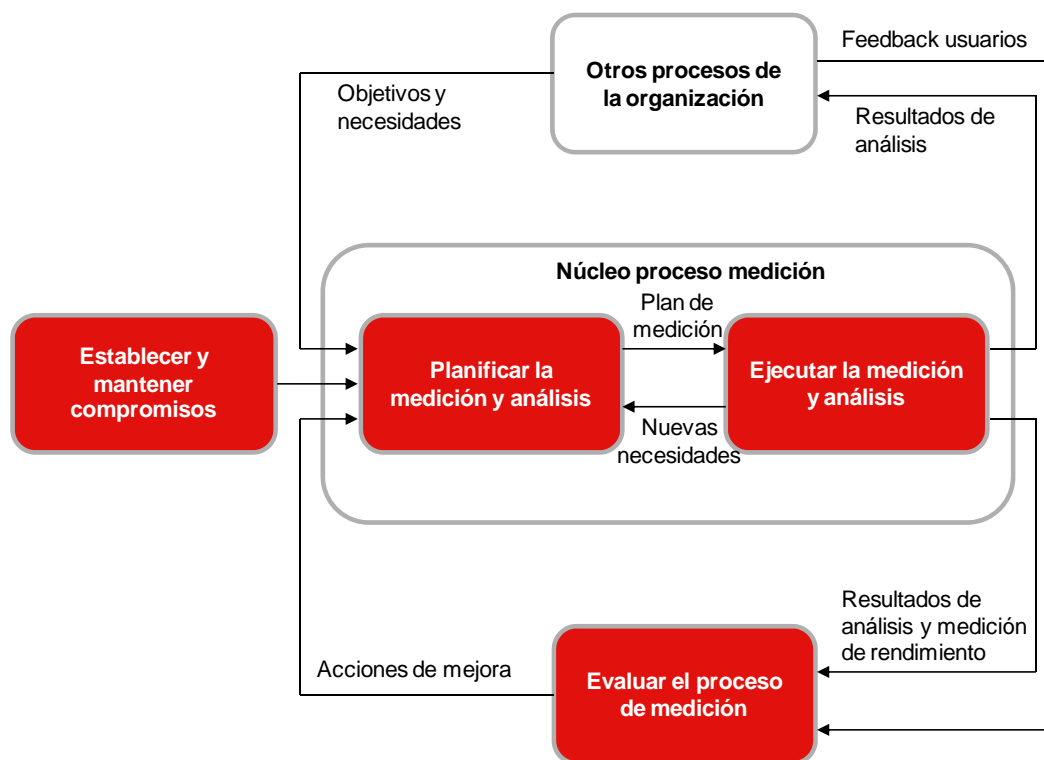


Figura 16 Modelo del proceso de medición y análisis

8.1. ESTABLECER Y MANTENER COMPROMISOS

Tal y como se muestra en la figura anterior, el primer paso es **establecer y mantener los compromisos** de la compañía relacionados con el proceso de medición y análisis. Entre las tareas que se llevan a cabo durante esta fase están:

- Definir roles y responsabilidades de las métricas.
- Proporcionar recursos, formación y herramientas para implementar de forma efectiva el proceso de medición y análisis.
- Establecer compromisos para utilizar la información que se genere durante la ejecución del proceso.
- Reforzar este compromiso continuamente.
- Revisar el programa de medición.
- Adoptar una orientación hacia las acciones a tomar.

8.2. PLANIFICAR EL PROCESO DE MEDICIÓN Y ANÁLISIS

El siguiente paso es el de **planificar el proceso de medición y análisis**. Un **proceso de medición** estructurado y repetible define las actividades de medición del proyecto. Este proceso debe ser flexible y adaptable para dar soporte a los procesos de gestión y técnicos del software existente y a las características del dominio específico de la aplicación. El

proceso de medición ha de ser iterativo hasta el fin del proyecto, de tal forma que los esfuerzos de las métricas se centren en los problemas más críticos.

Para ser efectivas, **las métricas** deben implementarse dentro de un proyecto u organización como un proceso de soporte de ingeniería de software. De esta forma, las métricas deben incluirse en todas las actividades asociadas con la planificación, rendimiento y evaluación de tareas de la organización o del proyecto.

Pero las métricas no se mantienen por sí mismas sino que será necesario definir qué información es necesaria para la toma de decisiones, y cómo esta información será recogida, analizada, presentada y utilizada. El **proceso de medición** combina distintos datos objetivos y subjetivos en productos de información integrados que dirigen directamente las necesidades de información del proyecto ya definidas.

Esta fase del proceso se centra en definir las métricas que proporcionan información acerca de las necesidades de los proyectos y/o de la organización. Las métricas se definen identificando qué necesitan saber los que toman decisiones y relacionando estas necesidades con entidades que se pueden medir.

- Las tareas que se llevan a cabo en esta fase son:
 - Identificar y priorizar las necesidades de información.
 - Seleccionar y especificar métricas.
 - Integrar el enfoque de medición en procesos del proyecto.
- Es importante recordar que para identificar las necesidades de información hay que:
 - Entender qué se necesita medir.
 - Documentar la trazabilidad desde las necesidades de información a los objetivos a nivel de organización y operativos.
 - Asegurar que todas las necesidades de información han sido cubiertas.

En todo proceso de medición será necesario un **plan de medición** que especifique qué se va a medir y cómo se va a realizar el proceso. Este plan de medición será desarrollado en esta fase aunque se irá retroalimentando con la información obtenida en la siguiente fase, es decir, a medida que se vayan tomando y analizando las métricas. El plan de medición se irá modificando a medida que vayan apareciendo nuevas necesidades de información.

A continuación se muestran los puntos o apartados que podrían contemplarse en un plan de medición:

- **Propósito:** Es importante que en el plan de medición se especifique el propósito del documento. En este plan se expondrán las métricas que se utilizarán a lo largo del proyecto asegurando que las métricas que se definen están alineadas con los objetivos del negocio y del proyecto y que se implementan de forma organizada y planificada.
- **Alcance:** Documentación del alcance de las métricas del proyecto indicando qué etapas del desarrollo y mantenimiento del ciclo de vida se cubren con este plan.

- **Objetivos:** Exposición de los objetivos del documento. Entre los objetivos de este plan pueden estar la identificación de métricas y la gestión del rendimiento del proyecto. Este plan detalla los procesos y las herramientas, en el caso en que se usen, necesarios para recolectar medidas, calcular métricas, analizar e informar de los resultados.
- **Roles y responsabilidades:** Se hará una revisión de los roles y responsabilidades relacionados con las métricas de acuerdo con los roles del proyecto. Estos datos se podrían exponer en una tabla para hacerlo más visual.
- **Dependencias del plan de medición:** En esta sección se han de exponer todos los planes y procesos que estén relacionados con el plan de medición. Entre ellos pueden estar: plan de proyecto, plan de gestión de la configuración, plan de gestión de riesgos, plan de gestión de la calidad.... Los cambios que se produzcan en alguno de estos planes o procesos puede implicar una revisión del que se está definiendo en este documento.
- **Objetivos de calidad de proyecto y rendimiento de proceso:** Definición de los objetivos de calidad y de rendimiento de proceso para el proyecto. Los objetivos han de ser medibles.
- **Selección de métricas:** Descripción del método de selección de métricas para este proyecto. Estas métricas han de estar alineadas con los objetivos de negocio. También se debería indicar de qué manera pueden estas métricas ayudar a conseguir los objetivos de negocio.
- **Métricas y medidas del proyecto:** este apartado se podría dividir en distintas partes:

Tabla 9 Métricas del proyecto en un plan de medición

Métricas	Descripción
Definición de métricas	<p>Definición detallada de las métricas que se van a usar en el proyecto indicando por ejemplo:</p> <ul style="list-style-type: none"> - nombre de la métrica - descripción de la métrica - fórmula - prioridad - fuente de los datos - nivel de análisis - informe de análisis
Captura de datos	<p>Definición de los métodos a seguir en la recogida de los datos usados en las métricas: de dónde se van a obtener los datos, dónde almacenarlos de manera que sean accesibles para su uso futuro...</p>

Gestión y análisis de datos	Definición de los procesos que se van a llevar a cabo para el análisis de los datos medidos. Especificar los procedimientos de análisis por adelantado asegura que el análisis se va a realizar conforme a los objetivos de medida.
Comunicación de resultados	Descripción del proceso de comunicación de los datos a todos los agentes involucrados en un tiempo adecuado para servir de apoyo a la toma de decisiones y ejecución de acciones correctivas.
Formación	Descripción de la formación necesaria para los involucrados en el proceso relacionado con las métricas.

8.3. EJECUCIÓN DE LA MEDICIÓN Y ANÁLISIS

Tras haber realizado la planificación del proceso ya se podría pasar a la siguiente fase que será la **realización o ejecución de la medición y del análisis**. Las tareas o actividades a realizar en esta fase constituyen el núcleo de la actividad de medición y análisis.

1. Recoger datos de medición

La recogida de los datos fuente es una práctica embebida en las actividades diarias, y suele realizarse por aquellos que realizan las actividades que generan los datos fuente. Por ejemplo, la recogida del esfuerzo dedicado a un proyecto se realiza cumplimentando la hojas de imputación (parte de horas) por cada participante en el proyecto. Es importante que antes de calcular los indicadores utilizando los datos fuente, se compruebe que los datos son completos y correctos. Una vez que disponemos de los datos fuente se podrá calcular los indicadores tal y como se definió en el plan de medición. Es muy recomendable automatizar el proceso de recogida de datos siempre y cuando sea posible.

2. Analizar los datos

Una vez que se han recopilado los datos se puede utilizar alguna técnica que ayude a analizarlos. Por ejemplo, se pueden utilizar técnicas de estimación o de análisis de viabilidad y de rendimiento.

Para analizar los datos también se pueden utilizar métodos como los siguientes:

- Análisis Pareto
- Análisis Causa-Efecto (*Fishbone/Ishikawa Diagram*)
- Análisis de tipos de fallos y efectos (*FMEA – Failure Mode and Effects Analysis*)
- QFD - *Quality Function Deployment*
- Diagrama de Flujo de Procesos
- Pruebas de Correlación
- Análisis estadístico de datos

Los indicadores recogidos son analizados, primero por cada responsable del indicador (también especificado en el plan de medición), para comprobar si es necesario recoger más datos. Y luego aquellos involucrados en la toma de decisiones soportada por los indicadores cuantitativos continuarán con el análisis de los datos.

3. Análisis y comunicación de los resultados

Tanto los datos fuente como los resultados del análisis necesitan estar disponibles para ser utilizados en el futuro, por lo que será necesario que estén almacenados en un repositorio accesible, definiendo claramente quien tiene acceso a los distintos repositorios. Es imprescindible mantener informados de los resultados del análisis a todos aquellos involucrados o afectados por la toma de decisiones así como de la manera en la que se verán afectados por las decisiones tomadas.

Otro punto importante a tener en cuenta durante esta fase es el establecer bien los mecanismos de presentación de informes tanto a nivel de proyectos como a nivel de organización.

8.4. EVALUACIÓN DEL PROCESO DE MEDICIÓN

Por último estaría la última fase del proceso de medición y análisis que es la fase de evaluación del proceso de medición. El proceso de medición y análisis se debería evaluar periódicamente e irlo mejorando al igual que cuando cambien las necesidades de información habrá que revisar las métricas definidas y realizar las acciones correspondientes de mejora. Por lo tanto, el proceso de medición y análisis es un **proceso iterativo**.

Esta fase comprende las siguientes tareas:

- Evaluar las métricas.
- Evaluar el proceso de medición.
- Actualizar la base de la experiencia con los resultados de la evolución realizada.
- Identificar e implementar las mejoras identificadas con el objetivo de mejorar las áreas sobre las que se han tomado los datos.
- Recoger las lecciones aprendidas tras el proceso.

La evaluación y acciones de mejora también sirven como retroalimentación a la fase de planificación de la medición y análisis, es decir, ayuda a mejorar el plan de medición iniciado al comienzo del proceso. Una vez más vuelve a aparecer el término de proceso iterativo.

Tabla 10 Actividades y tareas de PSM

ACTIVIDAD	TAREAS
Establecer y mantener compromisos de medición	Aceptar los requisitos de la medición
	Asignar recursos
Planificar el proceso de medición	Obtener las características de la organización
	Identificar las necesidades de información
	Seleccionar las medidas
	Definir los procedimientos de recolección de datos, análisis e informes
	Definir los criterios de evaluación de los productos de información y el proceso de medición
	Revisar, aprobar y proporcionar recursos para las tareas de medición
	Adquirir y utilizar tecnologías de apoyo
Ejecutar el proceso de medición	Recoger los datos
	Analizar los datos y desarrollar productos de información
	Comunicar los resultados
Evaluar la medición	Evaluar los productos de información y el proceso de medición
	Identificar las mejoras potenciales

9. ASPECTOS CLAVE EN EL PROCESO DE MEDICIÓN

Implementar un programa de medición es un esfuerzo importante, especialmente cuando los objetivos a nivel de proyecto y a nivel de la organización difieren o surgen algunos conflictos. Sin embargo, llegar a un equilibrio entre las necesidades de información de ambos niveles es posible.



Figura 17 Equilibrio necesidades de información

En este apartado vamos a ver una serie de acciones a llevar a cabo para tener éxito en la definición e implantación de métricas en una organización.

La primera tarea para planificar un programa de medición, como ya hemos visto, es **definir las necesidades de información** de la organización. Para ello, será necesario formar a la dirección y trabajar con ellos para identificar y priorizar sus necesidades de información.

Una vez que se han definido las necesidades de información de la organización, se puede identificar un conjunto de medidas comunes que cubran estas necesidades. Esto quedará recogido en un plan de medición organizacional que, además, describirá el repositorio donde se almacenarán los datos y el proceso a seguir por cada proyecto para incorporar los datos a este repositorio. Este conjunto de medidas será requerido para cada proyecto, pudiendo realizarse adaptaciones para proyectos específicos. Al tratarse de un conjunto común de medidas, se puede comenzar la implantación del programa a pequeña escala, a modo de piloto, para extenderlo posteriormente a todo el departamento de TI.

A la hora de decidir qué medidas recoger se pueden tener en cuenta varios factores:

- ¿Qué datos existen y están disponibles para ser analizados?
- ¿Qué datos nuevos serían relativamente fáciles de obtener?
- ¿Qué medidas contribuirían de forma inmediata a la toma de decisiones?

La segunda fase de un programa de medición **involucra a los proyectos**. Los responsables de proyectos tienen unas necesidades de información específicas a nivel de proyecto, aparte de las definidas por la organización. El proceso de medición, por tanto, debe cubrir tanto las medidas requeridas por la organización (adaptadas, si es necesario) como aquellas

orientadas a las necesidades de información específicas del proyecto. Tener un único proceso de medición simplifica la recogida de datos y reduce la duplicidad. Las medidas del proyecto se documentan en un plan de medición del proyecto. Los analistas del proyecto deben recopilar los datos de su propio proyecto a la vez que reportar las medidas comunes al repositorio de la organización para su consolidación y análisis.

La clave para equilibrar las necesidades de información de múltiples niveles de la organización es definir medidas que sean útiles tanto a nivel de proyecto como a nivel organizacional. Las medidas organizacionales normalmente están basadas en consolidaciones de datos de proyecto. Por ejemplo, si una necesidad de información para la organización es entender la calidad de producto, se requerirán datos sobre defectos en cada proyecto. A nivel de proyecto, se pueden generar medidas detalladas de los defectos encontrados y cerrados orientadas a satisfacer necesidades de información como “¿Estará listo el producto para comenzar las pruebas de aceptación en el tiempo planificado?”. A nivel organizacional, los datos sobre los defectos se pueden consolidar para cubrir necesidades de información del tipo “¿Cuántos defectos se generan, por término medio, en cada fase del proyecto?” o “Para un proyecto nuevo, ¿cuánto esfuerzo y tiempo se debe planificar para el re-trabajo derivado de la resolución de defectos?”.

Como conclusión, la implementación de un proceso objetivo de medición basado en hechos implica definir necesidades de información a nivel de la organización y de proyecto y seleccionar medidas que proporcionen información relativa a esas necesidades. La información debe ser comunicada en la organización y utilizada de forma regular en la toma de decisiones para que el proceso de medición tenga éxito.

Se recomienda tener en cuenta las siguientes consideraciones:

1. **Los programas de medición que tienen éxito** integran las necesidades de todos aquellos que toman las decisiones. Esta integración contribuye a la simplificación de la recolección de datos y reduce la probabilidad de duplicar información.
2. **El plan de MA debe especificar claramente qué hay que medir y cómo.**

El proceso de MA debe estar alineado con los objetivos de negocio. Por lo que los datos deben proporcionarse a los gerentes en el momento adecuado de cara a agilizar la toma de decisiones. Éstos deben basar su toma de decisiones en datos correctos y teniendo en consideración los riesgos y la información del contexto.

3. **Implementar un juego reducido de métricas.**

Esto reduce el nivel de cambios, los recursos necesarios y el impacto de la carga de trabajo a realizar en las actividades de MA.

Una vez que el sistema de MA vaya demostrando su utilidad y estabilidad se pueden ir añadiendo progresivamente otras métricas, que claramente estén relacionadas directa o indirectamente con la consecución de los objetivos de negocio.

Una buena manera de identificar y definir necesidades de información o de medición es la realización de talleres de trabajo con el objetivo claro de establecer qué medidas que ahora no se toman ayudarían a conseguir determinados objetivos de negocio.

4. Definir claramente las métricas.

Se debe proporcionar una base de métricas bien definidas y consistentes. Es preciso presentar la información en un formato claro y sencillo para que la persona que tome la decisión pueda entenderla. Esto se debe a que generalmente quien hace la medición no es la persona que toma las decisiones.

5. Evolución de las métricas.

Es importante ser flexible en cuanto a las definiciones de las métricas ya que puede ser preciso modificarlas si las necesidades de información varían. En el caso de modificar las definiciones originales de las métricas, se ha de especificar cómo se van a agregar los datos y los resultados del análisis.

6. Asegurar que los participantes en el proceso de MA entienden el proceso y los beneficios que propone él para sus proyectos y su organización. Algunas de las técnicas relacionadas en este sentido pueden ser:

- La formación adecuada debe ayudar a los responsables a definir sus necesidades de información.
- Los talleres de planificación deben involucrar a participantes de todos los niveles de la compañía.

7. Automatizar el procedimiento de recogida de datos siempre que sea posible.

El tiempo requerido para el establecimiento de medidas suele requerir entre 6 y 9 meses. En un primer momento es preciso identificar qué datos recoger y de dónde. Posteriormente, establecer cómo almacenar los datos y cómo utilizarlos. Por último, es preciso ir mejorando el rendimiento tanto de los datos como de su captura.

8. Establecer claramente los mecanismos de información y comunicación tanto a nivel de proyecto como a nivel organizativo.

10. ENFOQUE MODELOS

En apartados anteriores ya hicimos referencia a un proceso de medición de referencia como es el proceso *PSM (Practical Software Measurement)* basado en los estándares ISO/IEC 15939, ISO 9000 y CMMI®. En este apartado vamos a ver el enfoque que dos importantes modelos, como son CMMI® y SPICE, dan al proceso de medición y análisis.

10.1. CMMI®

CMMI® es un modelo de madurez de mejora de procesos para el desarrollo y mantenimiento de productos y servicios. Consiste en una serie de mejores prácticas que dirigen las actividades de desarrollo y mantenimiento que cubren el ciclo de vida del producto.

El área de proceso Medición y Análisis (MA) es de la categoría Soporte, es decir, las prácticas de MA se utilizan para satisfacer las necesidades de información de los proyectos, de los productos, de la organización y las relacionadas con el desempeño de los procesos.

El objetivo principal del área de proceso MA en el modelo CMMI es desarrollar y poner en marcha el sistema de medición de la organización de manera que pueda satisfacer sus propias necesidades de información. Esto implica:

- La especificación de los objetivos de MA de forma que estén alineados a las metas y necesidades de información.
- La especificación de métricas, técnicas de análisis y mecanismos para la recolección, almacenaje, reporte y retroalimentación de la información para su posterior implementación.
- Dar resultados objetivos que puedan ser utilizados para tomar decisiones informadas y tomar acciones correctivas.

El área de proceso tiene dos metas específicas y prácticas específicas asociadas a cada una de las metas:

Tabla 11 Área de proceso MA

Meta específica	Prácticas específicas
SG1 Alinear las actividades de Medición y Análisis <i>Los objetivos y actividades de medición están alineados con las metas y necesidades de información identificadas.</i>	SP1.1 Establecer objetivos de medición
	SP1.2 Especificar mediciones
	SP1.3 Especificar procedimientos de recogida y almacenamiento de datos
	SP1.4 Especificar procedimientos de análisis
SG2 Proporcionar resultados de la medición <i>Los resultados de las mediciones, las cuales soportan las metas y necesidades de información, son proporcionados.</i>	SP2.1 Recolectar datos para las mediciones
	SP2.2 Analizar datos de las mediciones
	SP2.3 Almacenar datos y resultados
	SP2.4 Comunicar resultados

Las relaciones entre las prácticas específicas están demostradas en el siguiente gráfico:

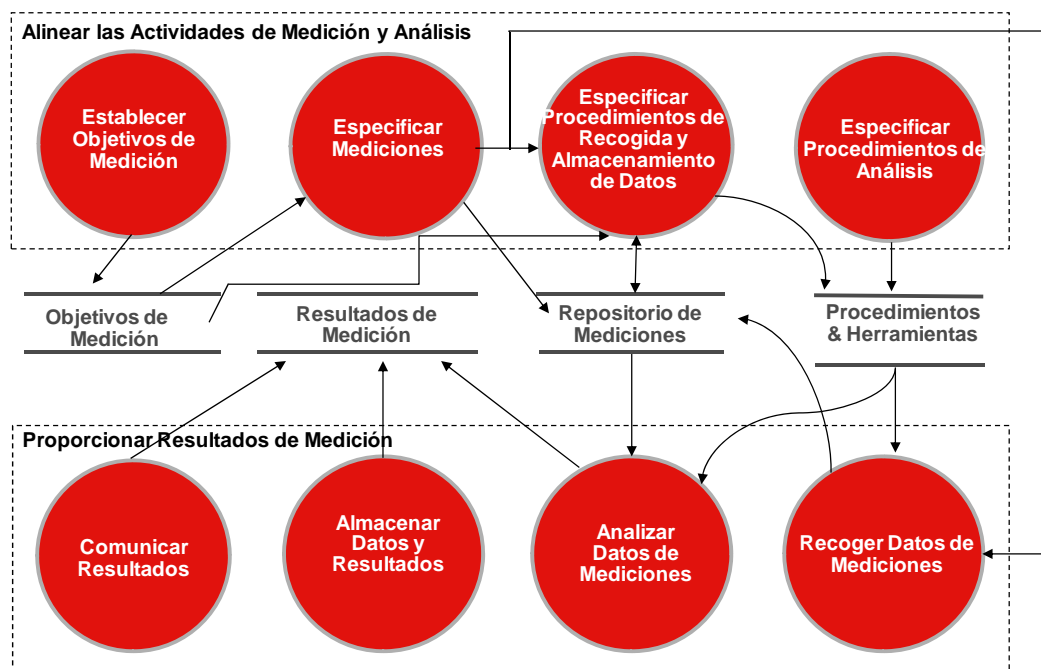


Figura 18 Diagrama de contexto de MA

Además de las metas y prácticas específicas, el área de proceso MA contiene también metas y prácticas genéricas que representan diferentes niveles de institucionalización del proceso en una organización.

10.2. ISO/IEC 15504

El estándar internacional **ISO/IEC 15504 (SPICE)** describe los procesos que una organización debe ejecutar para adquirir, proveer, desarrollar, operar, evolucionar y dar soporte al software, y las prácticas genéricas que caracterizan la capacidad de esos procesos.

Cada proceso del modelo se describe en términos de prácticas básicas, que son las actividades esenciales de un proceso específico. El modelo clasifica los procesos en cinco categorías que son:

- Cliente-Proveedor
- Ingeniería
- Gestión de Proyectos
- Soporte
- Organización

Dentro de la categoría **Gestión de Proyectos** entra el proceso relacionado con la medición. El propósito de este proceso, que se representa como **MAN.6**, es recoger y analizar datos relacionados con los productos desarrollados y los procesos implementados en la organización y sus proyectos. Su objetivo principal es gestionar de forma efectiva los procesos y demostrar de forma objetiva la calidad de los productos.

Las prácticas del proceso se presentan a continuación:

- **BP1:** Establecer el compromiso de la organización para la medición.
El compromiso de la Dirección y del personal se establece, mantiene y comunica a la organización.
- **BP2:** Desarrollar una estrategia de medición.
Definir una estrategia apropiada para identificar, realizar y evaluar las actividades y los resultados de medición, basándose en las necesidades de información de la organización.
- **BP3:** Identificar las necesidades de información a medir.
Identificar las necesidades de información de los procesos organizacionales a medir.
- **BP4:** Especificar medidas.
Identificar y desarrollar un conjunto de métricas que correspondan a las necesidades de información.
- **BP5:** Recoger y almacenar datos de medición.
Identificar, recoger y almacenar datos, incluyendo información contextual necesaria para verificar, entender o evaluar los datos.
- **BP6:** Analizar los datos medidos.
Analizar e interpretar los datos medidos y desarrollar productos informativos.

- **BP7:** Utilizar los productos de información medida para tomar decisiones.

Hacer los productos informativos accesibles para todos los procesos de toma de decisiones donde la información sea relevante.

- **BP8:** Comunicar los resultados de medición.

Difundir los productos informativos a todas las personas que los van a utilizar y recoger comentarios para evaluar si son adecuados para su uso previsto.

- **BP9:** Evaluar y comunicar los productos informativos y las actividades de medición a los propietarios de los procesos.

Evaluar los productos informativos y las actividades de medición contra las necesidades de información y la estrategia de medición, identificar mejoras potenciales en las mediciones y comunicar cualquier mejora potencial de los procesos a sus propietarios.

La medición es una actividad que está presente en las **prácticas genéricas** que determinan el nivel de capacidad de un proceso. Así, tenemos que:

- En el **nivel 2** de capacidad: Planificado y Controlado, dentro de la característica común 2.4: Control de la ejecución, se encuentra la práctica genérica:
 - **Control con mediciones**, cuyo propósito es controlar el estado del progreso contra el plan usando mediciones. El uso de mediciones implica que éstas han sido definidas y seleccionadas y que los datos han sido recogidos.
- En el **nivel 4** de capacidad: Controlado Cuantitativamente, dentro de la característica común 4.1: Establecer unas metas de calidad medibles, se encuentra la práctica genérica:
 - **Establecer metas de calidad**, cuyo propósito es establecer unas metas de calidad medibles para los productos/entregables generados por los procesos estándar de la organización. Estas metas de calidad deben estar alineadas con las metas estratégicas de calidad de la organización, las necesidades particulares y prioridades del cliente y las necesidades tácticas del proyecto.
- En el **nivel 5** de capacidad: Mejora Continua, dentro de la característica común 5.1: Mejorar la capacidad de la organización, se encuentra la práctica genérica:
 - **Establecer metas de efectividad de proceso**, cuyo propósito es establecer metas cuantitativas para mejorar la efectividad de los procesos estándar de la organización, basadas en las metas de negocio de la organización y la capacidad actual de los procesos.

11. ARTEFACTOS RELACIONADOS CON LA MEDICIÓN Y ANÁLISIS

En esta sección se van a proponer una serie de artefactos que pueden ser de utilidad a la hora de implementar el proceso de medición y análisis en la organización. Con el término artefactos nos referimos a procesos/procedimientos, plantillas, herramientas, listas de control (checklist), etc., es decir, todo tipo de elementos que ayudarán a llevar a cabo las prácticas del proceso de medición y análisis.

Algunos de los documentos que pueden servir de apoyo a una organización en las tareas de medición y análisis son:

- **Proceso de medición y análisis:** Documento donde se describe el proceso de medición y análisis especificando las distintas partes del mismo, así como determinando cuáles son los criterios de entrada y de salida, los elementos de entrada y de salida para el proceso, cómo validar el proceso y métricas a recoger sobre la ejecución del proceso.
- **Guía para medición del esfuerzo:** Documento que recoge buenas prácticas y recomendaciones para medir el esfuerzo necesario para estimar, planificar, monitorizar y controlar el proyecto.
- **Guía para medición de defectos:** Documento que recoge buenas prácticas y recomendaciones para medir el número de defectos encontrados durante las revisiones y pruebas de los productos.
- **Guía de métricas por nivel de CMMI:** Documento que muestra algunas métricas que pueden tomarse sobre las áreas de proceso del nivel 2 de madurez de CMMI®.
- **Definición de métricas:** Plantilla para definir una serie de métricas en el proyecto.
- **Validación de datos:** Plantilla para realizar un análisis sobre la validez de los datos recogidos.
- **Cuestionario satisfacción de cliente:** Plantilla para realizar un cuestionario de satisfacción del cliente sobre distintos aspectos del proyecto.
- **Informe satisfacción de cliente:** Plantilla para generar informes periódicos sobre la satisfacción del cliente.
- **Consolidación de métricas:** Plantilla para la consolidación de las métricas recogidas durante todas las fases del proyecto.
- **Checklist de medición y análisis:** Lista de comprobación para asegurar que se está siguiendo adecuadamente el proceso definido para la medición y análisis en el proyecto, según las prácticas del modelo CMMI®.

Algunos ejemplos de estos artefactos pueden encontrarse en el servicio online de 'Repositorio documental de procesos'. Así mismo, en el servicio online 'Directorio de herramientas' pueden encontrarse herramientas de soporte para la implementación del

proceso de MA. Ambos servicios están disponibles de forma gratuita en el portal de INTECO (www.inteco.es), en su sección de 'Calidad de software'.

12. GLOSARIO

- **Atributo:** propiedad mensurable, física o abstracta, que comparten todas las entidades de una categoría de entidad.
- **Checklist:** formulario con elementos que deben ser comprobados. Su propósito principal es ayudar en la recogida y organización de datos para facilitar su consulta en el futuro.
- **Concepto medible:** relación abstracta entre atributos y necesidades de información.
- **Cuadros de mando:** herramientas de gestión del rendimiento que presentan al usuario una visualización de los indicadores empresariales. Una clasificación de cuadros de mando en función de su contenido es: *dashboards*, *scoreboards*.
- **Dashboards:** cuadros de mando que muestran información sin compararla con los objetivos propuestos. Es decir, con ellos únicamente se pueden visualizar los *KPI* (*Key Performance Indicator*).
- **Diagrama de dispersión:** herramienta útil para identificar una relación potencial entre dos variables. En una relación causa-efecto, el eje X se utiliza para la variable independiente y el eje Y para la variable dependiente.
- **Diagrama de Pareto:** gráfico de barras de frecuencia en orden descendente. Las barras de frecuencia se asocian normalmente con tipos de problemas.
- **Entidad:** objeto que va a ser caracterizado mediante una medición de sus atributos [ISO-15939].
- **Escalas de medición:** niveles que categorizan a las métricas de acuerdo a la rigurosidad con que han sido construidas y al propio comportamiento de las variables que miden. Normalmente se clasifican en cuatro tipos: nominal, ordinal, intervalo o razón/ratio.
- **Gráfico de control:** puede considerarse como un tipo avanzado de gráfico de ejecución en situaciones donde se puede definir la capacidad de un proceso.
- **Gráfico de ejecución:** registra el rendimiento de un parámetro de interés a lo largo del tiempo. El eje X es el tiempo y el eje Y es el valor del parámetro.
- **Herramientas de Ishikawa:** herramientas estadísticas promovidas por Ishikawa para el control de la calidad que se han convertido en una parte importante dentro del control de la calidad.
- **Histograma:** representación gráfica de frecuencias dentro de una muestra o población.
- **Indicador:** una métrica cuya forma de medir es un modelo de análisis, es decir, las mediciones de dicha métrica utilizan las medidas obtenidas en las mediciones de otras métricas junto con criterios de decisión.

- **Indicador clave de rendimiento (KPI):** métrica de alto nivel de la efectividad y/o eficiencia que se usa como guía y control progresivo del rendimiento.
- **Instrumento de medición:** herramienta que automatiza parcial o totalmente a un método de medición o cálculo.
- **Medición:** acción que permite obtener el valor de una medida para un atributo de una entidad, usando una forma de medir. Es decir, es el proceso de asignar un número o categoría a una entidad para describir un atributo de dicha entidad.
- **Medida:** número o categoría que se le asigna a un atributo de una entidad cuando se hace una medición. Es el resultado de una medición.
- **Mejores prácticas:** método superior o práctica innovadora que contribuye a la mejora de rendimiento en una organización bajo un contexto dado, normalmente reconocido como el mejor por otras organizaciones similares.
- **Métrica:** una forma de medir (método de medición, función de cálculo o modelo de análisis) y una escala, definidas para realizar mediciones de uno o varios atributos.
- **Métricas de líneas de código (LOC):** métrica básica que sirve como indicador del tamaño del software.
- **Métricas de proceso:** se pueden utilizar para mejorar el desarrollo del software y el mantenimiento. Ejemplos de este tipo son la efectividad de la eliminación de defectos durante el desarrollo, el patrón de la llegada de defectos en las pruebas, y el tiempo de respuesta del proceso de corrección.
- **Métricas de producto:** describen las características del producto como tamaño, complejidad, características de diseño, rendimiento y nivel de calidad.
- **Métricas de proyecto:** describen las características del proyecto y su ejecución. Algún ejemplo puede ser el número de desarrolladores de software, la dotación de recursos a lo largo del ciclo de vida del software, el coste, calendario y productividad.
- **Métricas directas:** métrica de un atributo que no depende de ninguna métrica de otro atributo.
- **Métricas indirectas:** métrica de un atributo que se deriva de una o más métricas de otros atributos.
- **Necesidad de información:** información necesaria para gestionar un proyecto.
- **Plan de medición:** documento que describe el propósito, alcance, objetivos, roles, métodos de selección de métricas, métricas del proyecto... de las actividades de medición.
- **Puntos función:** método para medir el tamaño del software entregado al usuario que es independiente de la tecnología utilizada para la construcción y despliegue del software, y que es útil en cualquiera de las fases del ciclo de vida del software, desde el diseño inicial hasta el despliegue y mantenimiento del mismo.

- **Satisfacción del cliente:** grado de satisfacción del cliente, normalmente medido a través de encuestas realizadas al cliente. Algunos de los parámetros que se evalúan en estas encuestas son: capacidad, funcionalidad, usabilidad, rendimiento, fiabilidad, mantenibilidad, documentación/información y servicio.
- **Scorecards:** cuadros de mando que muestran información estratégica y están orientados a mostrar objetivos.

13. REFERENCIAS

Chrissis M, M. Konrad, S. Shrum, *CMMI® Second Edition. Guidelines for Process Integration and Product Improvement*, Addison-Wesley, 2007

Fenton, Norman; Pfleeger, E. and Shari, Lawrence. *Software Metrics: A Practical And Rigorous Approach*. London, UK: Thomson, 1996

Florac William, Anita D. Carleton, *Measuring the Software Process: Statistical Process Control for Software Process Improvement*, The SEI Series in Software Engineering

Florac W., R. Park, A Carleton, *Practical Software Measurement: Measuring for Process Management and Improvement*, MU/SEI-97-HB-003, 1997

Grady, R.B., and D.L. Caswell, *Software Metrics: Establising a Company-Wide Program*, 1986

Heitzel, B., *Making Software Measurement Work: Bulding an Effective measurement Program*, 1993

John McGarry, David Card, Cheryl Jones, Beth Layman, Elizabeth Clark, Joseph Dean, Fred Hall, *“Practical Software Measurement: Objective Information for Decision Makers”*

Jones, C., *Applied Software Measurement, Assuring Productivity and Quality*, 1997

McGarry, J., D.Card, C.Jones, B.Layman, e.Clark, J.Dean, y F.Hall, *Practical Software Measurement, Objective Information for Decision Makers*, 2001

Stephen H. Kan, *Metrics and models in Software Quality Engineering*, 2003

Enlaces

What is the Balanced scorecard? -

<http://www.balancedscorecard.org/BSCResources/AbouttheBalancedScorecard/tabid/55/Default.aspx>

Cuadros de mando (Dashboards y Scorecards) - <http://bi.abast.es/dashboards.shtml>

Balanced scorecard - http://www.coursework4you.co.uk/balanced_scorecard.htm

Best Practices for Building Digital Dashboards -

<http://www.dundas.com/Dashboards/BestPractices3.aspx>

ISO/IEC JTC1/SC7 /N3302, 2005-07-08 - <http://www.jtc1-sc7.org>

Practical Software and Systems Measurement - <http://www.psmc.com/>

Implementing a Successful Measurement Program: Tried and True Practices and Tools -

http://www.psmc.com/Downloads/Other/Implementing_a_SuccessfulMeasurementProgram.pdf

Software Metrics - <http://www.softwaremetrics.com/>

Total Metrics - <http://www.totalmetrics.com/>

The Accounting Blog – Key Performance Indicators - <http://www.vanilla-accounting.com/blog/archives/000056.php>

Balanced scorecard, Indicator, Balanced Scorecard, Performance Management -
<http://www.visitask.com/balanced-scorecard.asp>

Developing key performance indicators in projects - <http://www.visitask.com/Developing-key-performance-indicators.asp>

Using key performance indicators (KPI) for effective project management -
<http://www.visitask.com/key-performance-indicators.asp>

14. ANEXO: EJEMPLOS DE INDICADORES

14.1. PRODUCTO FÍSICO

14.1.1. Tamaño

Tabla 12 *Plantilla indicador tamaño*

PLANTILLA FICHA IDENTIFICADOR		
*Nombre Indicador:	*Identificador:	Tipo identificador:
Puntos función	FP	Tamaño del producto
*Objetivo del indicador:		*Ámbito del desempeño:
Medir el tamaño del software entregado al usuario independientemente de la tecnología utilizada para la construcción y despliegue del software, útil en cualquiera de las fases del ciclo de vida del software, desde el diseño inicial hasta el despliegue y mantenimiento del mismo.		Código fuente
*Descripción del indicador		
<p>Los puntos función son independientes de la tecnología porque miden los sistemas desde una perspectiva funcional. El número de puntos función de un sistema permanecerá constante independientemente del lenguaje de programación, método de desarrollo o plataforma utilizada. La única variable es la cantidad de esfuerzo necesario para entregar un conjunto de puntos función, por tanto, el análisis de puntos función se puede utilizar para comparar si una herramienta, un entorno o un lenguaje es más productivo que otros dentro de una organización o entre distintas organizaciones. Esto supone uno de los valores más importantes del análisis de puntos función.</p> <p>El análisis de puntos función, además, proporciona un mecanismo para realizar un seguimiento y control de cambios de alcance. Se puede comparar el número de puntos función al final de las fases de requisitos, análisis, diseño, codificación y pruebas.</p>		
Fuente de información:	Datos de entrada:	Definiciones y abreviaturas:
Requisitos funcionales del software	Número de funciones disponibles para el usuario	-
*Fórmula de cálculo:		*Escala o unidad:
El proceso para calcular el número de puntos función sería: identificar las funciones disponibles para el usuario, categorizar estas funciones en componentes (entradas, salidas, consultas, interfaces y ficheros lógicos internos), asignar una complejidad a cada función (alta, media, baja). En función del tipo de componente y la complejidad se le asignará un número de FP sin ajustar. Este número debe calibrarse con 14 elementos que dependen del entorno, puntuando cada uno de los factores según el grado de influencia en una escala del 0 al 5. Una vez puntuados, se utiliza la siguiente ecuación para calcular el factor		Puntos función

de ajuste: $(65+TDI)/100$, donde TDI es la suma de las puntuaciones de todos los elementos.

***Criterios de análisis:**

Al comparar el número de puntos función al final de la especificación de requisitos y/o diseño con el número de puntos función entregados, si el número de puntos función ha crecido indica que ha habido cambios de alcance. El volumen de este crecimiento indicará también lo bien que se han tomado los requisitos y/o se han comunicado al equipo de proyecto. Si el volumen de cambios de alcance disminuye en el tiempo significará que la comunicación con el usuario ha mejorado.

14.1.2. Complejidad

Tabla 13 Plantilla indicador complejidad

PLANTILLA FICHA IDENTIFICADOR		
*Nombre Indicador:	*Identificador:	Tipo identificador:
Complejidad ciclomática	CC	Complejidad
*Objetivo del indicador:		*Ámbito del desempeño:
Conocer el número de caminos independientes que componen un programa. Se puede utilizar para indicar el esfuerzo requerido para probar un programa.		Planificación de pruebas, análisis de riesgo en desarrollo de código o análisis de riesgo de camino durante la fase de mantenimiento.
*Descripción del indicador		
Es una métrica independiente del lenguaje que se basa en el diagrama de flujo determinado por las estructuras de control de un determinado código. De dicho análisis se puede obtener una medida cuantitativa de la dificultad de crear pruebas automáticas del código y también es una medición orientativa de la fiabilidad del mismo.		
Fuente de información:	Datos de entrada:	Definiciones y abreviaturas:
Grafo que represente los caminos posibles en la ejecución de un programa o módulo.	Número de aristas del grafo, número de nodos del grafo, número de partes desconectadas del grafo.	
*Fórmula de cálculo:		*Escala o unidad:
Número de aristas del grafo - Número de nodos + (2 * Número de partes desconectadas del grafo)		
*Criterios de análisis:		
Para que el programa pueda ser fácilmente probado y mantenible, se recomienda que ningún módulo tenga una complejidad ciclomática superior a 10.		

14.1.3. Calidad

Tabla 14 Plantilla indicador calidad

PLANTILLA FICHA IDENTIFICADOR		
*Nombre Indicador:	*Identificador:	Tipo identificador:
Densidad de defectos entregada	DDefE	Calidad
*Objetivo del indicador:		*Ámbito del desempeño:
Obtener visibilidad sobre la calidad de la solución entregada al cliente. Tiene una implicación directa en la satisfacción del cliente.		Producto/solución entregada
*Descripción del indicador		
Esta métrica permite mejorar la calidad de los productos entregados al cliente ampliando el conocimiento acerca del número de defectos encontrados respecto al tamaño del producto que se entrega al cliente.		
Fuente de información:	Datos de entrada:	Definiciones y abreviaturas:
Repositorio donde se registran las incidencias del cliente	Número de defectos encontrados durante revisiones del cliente y pruebas de aceptación; Tamaño del producto.	FP: Puntos función KLOC: Miles de líneas de código
*Fórmula de cálculo:		*Escala o unidad:
Número de defectos encontrados durante revisiones del cliente y pruebas de aceptación / Tamaño del producto		Defectos/KLOC, Defectos/FP...
*Criterios de análisis:		
Si el valor es mayor del previsto, puede ser debido a que los requisitos/escenarios de uso no han sido bien entendidos, las revisiones han sido ineficientes o la cobertura de pruebas inadecuada. Si el valor es menor del previsto, indicará que ha mejorado el proceso de desarrollo, o la utilización de herramientas o reutilización de componentes ha sido exitosa.		

14.2. GESTIÓN DE PROYECTOS (INCL. PROYECTOS DE MEJORA)

14.2.1. Planificación de proyecto

Tabla 15 Plantilla indicador planificación del proyecto

PLANTILLA FICHA IDENTIFICADOR		
*Nombre Indicador:	*Identificador:	Tipo identificador:
Distribución de esfuerzo	DEsf	Gestión de proyectos
*Objetivo del indicador:		*Ámbito del desempeño:
Disponer de una línea base que permita a los gerentes estimar el esfuerzo para cada fase individual de un proyecto o propuesta nueva, mejorando la efectividad de costes.		Planificación del proyecto
*Descripción del indicador		
Permite conocer el esfuerzo dedicado a cada fase del proyecto con respecto al esfuerzo total del proyecto.		
Fuente de información:	Datos de entrada:	Definiciones y abreviaturas:
Parte de horas	Esfuerzo real en la fase, esfuerzo total real en el proyecto, nombre de la fase	
*Fórmula de cálculo:		*Escala o unidad:
Esfuerzo real en la fase * 100/Esfuerzo total real en el proyecto		%
*Criterios de análisis:		
Se puede utilizar un gráfico circular para analizar la distribución del esfuerzo en cada fase del proyecto. Un gráfico de control permitirá monitorizar y comparar esta distribución por fases entre distintos proyectos de una organización.		

14.2.2. Ejecución de proyecto

Tabla 16 Plantilla indicador ejecución del proyecto

PLANTILLA FICHA IDENTIFICADOR		
*Nombre Indicador:	*Identificador:	Tipo identificador:
Índice de trabajo acumulado	ITA	Gestión del proyecto
*Objetivo del indicador:		*Ámbito del desempeño:
Conocer las incidencias abiertas que superan el SLA		Ejecución del proyecto
*Descripción del indicador		
Relaciona el índice de llegada de incidencias con el ritmo al que éstas se van corrigiendo		
Fuente de información:	Datos de entrada:	Definiciones y abreviaturas:
Herramienta para registro de incidencias	Número de incidencias abiertas que superan el SLA, número de incidencias abiertas	SLA: Acuerdo del nivel de servicio
*Fórmula de cálculo:		*Escala o unidad:
(Número de incidencias abiertas que superan el SLA/Número de incidencias abiertas) * 100		%
*Criterios de análisis:		
Se puede analizar mediante un gráfico de tendencias. Si el índice de trabajo acumulado disminuye, mejorará la productividad. El objetivo es que sea cercano a 0.		

14.2.3. Costes

Tabla 17 Plantilla indicador costes

PLANTILLA FICHA IDENTIFICADOR		
*Nombre Indicador:	*Identificador:	Tipo identificador:
Coste de la calidad	CQ	Gestión del proyecto
*Objetivo del indicador:		*Ámbito del desempeño:
Minimizar el re trabajo y mejorar la efectividad de costes.		Costes
*Descripción del indicador		

Permite conocer el coste asociado al nivel de calidad presente en un proyecto o en la organización

Fuente de información:	Datos de entrada:	Definiciones y abreviaturas:
Parte de horas	Coste de evaluación (para asegurar que se alcanza un nivel de calidad aceptable), Coste de prevención (para evitar la mala calidad), coste de fallos (gestión de no conformidades originadas por la falta de calidad), coste total.	
*Fórmula de cálculo:		*Escala o unidad:
$((\text{Esfuerzo en evaluación} + \text{Esfuerzo en prevención} + \text{Esfuerzo en fallos}) / \text{Esfuerzo total del proyecto}) * 100$		%
*Criterios de análisis:		
Se podrían usar gráficos de barras para comparar el % de coste de calidad con la meta establecida a nivel de proyecto.		
Usar gráficos de control para monitorizar el % de costes de calidad en los distintos proyectos de la organización.		
Se puede analizar la tendencia de la métrica como:		
Índice de 'Esfuerzo de evaluación + Esfuerzo prevención' con respecto a 'Esfuerzo fallos'. Debe aproximarse a 1.		
Índice de 'CQ' con respecto a 'Esfuerzo total'. Debe ser menor del 50%.		

14.2.4. Productividad

Tabla 18 Plantilla indicador productividad

PLANTILLA FICHA IDENTIFICADOR		
*Nombre Indicador:	*Identificador:	Tipo identificador:
Productividad de la codificación y pruebas unitarias	PrCPU	Gestión del proyecto
*Objetivo del indicador:		*Ámbito del desempeño:
Conocer la productividad del equipo durante las fases de codificación y pruebas unitarias para tomar medidas adecuadas para mejorar la productividad. También proporciona visibilidad sobre el progreso del trabajo.		Productividad
*Descripción del indicador		
Proporciona información acerca de la productividad en la codificación y pruebas unitarias, en términos de		

tamaño de código desarrollado o probado por el esfuerzo dedicado.

Fuente de información:	Datos de entrada:	Definiciones y abreviaturas:
El tamaño del código se puede obtener a través de una herramienta de inspección de código. El esfuerzo dedicado se obtendrá del parte de horas de los recursos involucrados.	Tamaño del producto, esfuerzo real para la fase de codificación y para la fase de pruebas unitarias.	FP: Puntos función LOC: Líneas de código
*Fórmula de cálculo:		*Escala o unidad:
Tamaño código / Esfuerzo invertido en la fase de codificación y pruebas unitarias		FP/Persona-hora LOC/Persona-hora
*Criterios de análisis:		
Utilizar un gráfico de barras para comparar la productividad conseguida con respecto al objetivo fijado para el proyecto. Mediante un gráfico de control se pueden comparar los resultados de distintos proyectos. Una productividad relativamente baja en una fase inicial puede atribuirse a la curva de aprendizaje.		

14.2.5. Seguimiento

Tabla 19 Plantilla indicador seguimiento

PLANTILLA FICHA IDENTIFICADOR		
*Nombre Indicador:	*Identificador:	Tipo identificador:
Desviación del esfuerzo	DesvC	Gestión del proyecto
*Objetivo del indicador:		*Ámbito del desempeño:
Tener visibilidad sobre varios aspectos de la gestión de proyecto: precisión de las estimaciones, capacidad de alcanzar los compromisos, estabilidad de los requisitos, rentabilidad del proyecto.		Seguimiento
*Descripción del indicador		
Muestra la diferencia, en porcentaje, entre el esfuerzo planificado y el dedicado realmente en una tarea/fase/proyecto.		
Fuente de información:	Datos de entrada:	Definiciones y abreviaturas:
Planificación aprobada, partes de horas	Esfuerzo estimado, esfuerzo real	
*Fórmula de cálculo:		*Escala o unidad:
$((\text{Esfuerzo real} - \text{Esfuerzo estimado}) / \text{Esfuerzo estimado}) * 100$		%

*Criterios de análisis:

Se puede utilizar en todos los niveles de la gestión de proyecto para medir los valores actuales frente a los planificados. Se recomienda utilizar para el análisis un gráfico de tendencias desde el inicio hasta el fin del proyecto.

14.3. SATISFACCIÓN DEL CLIENTE

14.3.1. Problemas en explotación

Tabla 20 Plantilla indicador problemas en explotación

PLANTILLA FICHA IDENTIFICADOR		
*Nombre Indicador:	*Identificador:	Tipo identificador:
Tiempo de respuesta	TResp	Satisfacción del cliente
*Objetivo del indicador:		*Ámbito del desempeño:
Asegurar la conformidad con el nivel de servicio acordado (SLA). Útil para estimar SLA's en proyectos nuevos		Problemas en explotación
*Descripción del indicador		
Permite conocer el porcentaje de peticiones, con respecto al total recibidas, que han sido respondidas dentro del nivel de servicio acordado (SLA)		
Fuente de información:	Datos de entrada:	Definiciones y abreviaturas:
Herramienta para registro de peticiones/incidencias del cliente	Nº peticiones recibidas; Nº peticiones respondidas dentro del SLA	SLA
*Fórmula de cálculo:		*Escala o unidad:
(Nº peticiones respondidas dentro del SLA/Nº total peticiones recibidas)*100		%
*Criterios de análisis:		
Se puede analizar mediante un gráfico de tendencias. El tiempo de respuesta debería reducirse a medida que el proyecto avanza y el equipo tiene mejor conocimiento de la aplicación.		

14.3.2. Percepción del cliente

Tabla 21 Plantilla indicador percepción del cliente

PLANTILLA FICHA IDENTIFICADOR		
*Nombre Indicador:	*Identificador:	Tipo identificador:
Índice de satisfacción del cliente	ISC	Satisfacción del cliente
*Objetivo del indicador:		*Ámbito del desempeño:
Conocer la perspectiva del cliente sobre el proyecto		Percepción del cliente
*Descripción del indicador		
Permite conocer la puntuación media que otorga el cliente en una encuesta de satisfacción sobre el proyecto/producto.		
Fuente de información:	Datos de entrada:	Definiciones y abreviaturas:
Encuesta de satisfacción	Puntuación de cada pregunta contestada en la encuesta de satisfacción sobre el proyecto, número de preguntas contestadas como N/A, número total de preguntas de la encuesta	
*Fórmula de cálculo:		*Escala o unidad:
Sum(Puntuación dada en cada pregunta de la encuesta)/(Nº total de preguntas de la encuesta - Nº preguntas contestadas como N/A)		Escala de puntuación (1-5 por ejemplo)
*Criterios de análisis:		
En proyectos largos, analizar la tendencia en las valoraciones de los clientes para identificar áreas de mejora. Analizar tanto las puntuaciones cuantitativas como las cualitativas.		

14.4. PROCESO

Tabla 22 Plantilla indicador a nivel de proceso

PLANTILLA FICHA IDENTIFICADOR		
*Nombre Indicador:	*Identificador:	Tipo identificador:
Efectividad en la eliminación de defectos	EEDef	Proceso

*Objetivo del indicador:		*Ámbito del desempeño:
Obtener visibilidad sobre la efectividad de los mecanismos de verificación y validación		Revisiones y pruebas
*Descripción del indicador		
Permite conocer el porcentaje de defectos totales del producto que se detectaron durante las pruebas y revisiones, antes de la entrega del producto.		
Fuente de información:	Datos de entrada:	Definiciones y abreviaturas:
Repositorio donde se registran los defectos encontrados durante pruebas y revisiones; Repositorio donde se registran las incidencias del cliente	Número de defectos detectados en pruebas y revisiones, número de defectos detectados después de la entrega	
*Fórmula de cálculo:		*Escala o unidad:
$\frac{(\text{N}^{\circ} \text{ defectos detectados en pruebas y revisiones})}{(\text{N}^{\circ} \text{ defectos detectados en pruebas y revisiones} + \text{N}^{\circ} \text{ defectos detectados después de la entrega})} \times 100$		%
*Criterios de análisis:		
Cuanto más alto sea el valor, más efectivo será el proceso de desarrollo y menos defectos se transmitirán a producción.		