



# **Universidad Autónoma de Ciudad Juárez**

Metricas de software

Actividad: Tarea 01

*214466 – Carlos Adrian Vasquez Rivera*

Fecha: 29 de enero del 2025

## Necesidad de medir

La necesidad de medir la calidad del software se ha vuelto fundamental debido a la creciente complejidad de los sistemas y la importancia que tienen tanto en la vida cotidiana como en las operaciones empresariales. Dávila, Gómez y Gómez (2007) destacan que las métricas de calidad son herramientas clave para evaluar atributos como confiabilidad, usabilidad y eficiencia, permitiendo asegurar que los productos cumplen con los estándares necesarios para reducir errores y fallos, los cuales pueden ocasionar desde inconvenientes menores hasta pérdidas económicas significativas. En un entorno donde el software tiene un impacto cada vez mayor, la evaluación de su calidad no es un lujo, sino una necesidad estratégica.

El proceso de medición del software, según Castañeda y Pardo (2007), es un esfuerzo continuo que abarca todo el ciclo de vida del desarrollo. Este enfoque implica la incorporación de métricas objetivas desde las etapas iniciales de diseño hasta el mantenimiento, evaluando atributos clave que permitan identificar áreas de mejora y garantizar el cumplimiento de los estándares de calidad. Esto, a su vez, contribuye a entregar productos que cumplan con los plazos establecidos, se ajusten al presupuesto y satisfagan las necesidades del negocio.

Sin embargo, García et al. (2020) señalan un desafío importante en este ámbito: la proliferación de métricas ha generado confusión respecto a sus relaciones, aplicaciones y formas de interpretación. La falta de claridad en cómo utilizar estas métricas puede derivar en decisiones inadecuadas o en una percepción incorrecta del rendimiento del software. Esto enfatiza la necesidad de adoptar un enfoque sistemático para la evaluación, que no solo considere las métricas más adecuadas, sino que también garantice su correcta aplicación en cada contexto.

En este sentido, medir la calidad del software no solo es un acto técnico, sino también un ejercicio estratégico que requiere alinearse con los objetivos organizacionales. Las métricas no deben ser vistas únicamente como indicadores cuantitativos, sino como herramientas esenciales para la toma de decisiones, que permiten mejorar continuamente los procesos de desarrollo y responder eficazmente a las demandas de un mercado en constante cambio.

## Concepto medible

Mancha (2004) nos dice que un concepto medible es cualquier característica, atributo o propiedad de un producto o proceso de software que pueda ser cuantificada de manera objetiva, ósea que es algo que podemos expresar numéricamente y sobre lo cual podemos realizar cálculos, formulas o comparaciones. Estos conceptos deben ser claros, precisos y definidos de manera que diferentes personas puedan llegar a la misma medición.

La importancia de estas es que proporcionan datos en concreto que permiten a los equipos de desarrollo tomar decisiones informadas sobre la calidad, el progreso y la eficiencia de

sus proyectos (Strub & Xavi, 2013). Al cuantificar diversos aspectos del software, es posible no solo identificar las áreas de mejora, sino también establecer objetivos claros para optimizar los procesos (Ferreira et al., 2006). Además, las métricas ofrecen un lenguaje común que facilita la comunicación entre los equipos de desarrollo, los gerentes y los clientes, lo que contribuye a una mejor comprensión del estado del proyecto y de los resultados obtenidos (Strub & Savi, 2013). También juegan un papel crucial en la evaluación objetiva de la calidad del software y en la comparación entre diferentes productos o versiones.

### Atributo de medición

Volviendo a Ferreira et al. (2006), estos autores mencionan que “los atributos de calidad de software son características intrínsecas que determinan su valor y utilidad más allá de las funciones básicas que realiza”. Estos atributos definen cómo se comporta el software en diferentes escenarios, evaluando aspectos como su eficiencia, seguridad, escalabilidad, y otros factores cruciales.

Según Mancha (2004), la importancia de estos atributos radica en su influencia directa en la toma de decisiones durante el desarrollo del software, guiando las decisiones arquitectónicas y de diseño. Estos atributos ayudan a establecer expectativas claras con los clientes, asegurando que el producto final cumpla con lo prometido. Además, sirven como base para la mejora continua, ya que permiten identificar áreas de mejora y optimizar el software. Si se evalúan los atributos de calidad, también se pueden reducir riesgos al identificar y mitigar posibles problemas antes de que afecten al producto final.

En cuanto a su clasificación, Acosta et al. (2006) explican que los atributos de calidad se dividen en dos grandes grupos: funcionalidad y estructura. Los de funcionalidad están relacionados con el cumplimiento de los requisitos del negocio y las necesidades del usuario, mientras que los de estructura (también conocidos como atributos no funcionales) se refieren a características como la eficiencia, la seguridad y la escalabilidad del software. Esta clasificación permite un enfoque más estructurado en la evaluación y mejora de la calidad del software.

### Entidad de medición

Berta y Vallecillo (2006) señalan que una entidad en la medición de software es cualquier componente del sistema al que se le pueden asignar atributos medibles, la medición de software implica asignar valores a los atributos de estas entidades. Estos valores pueden ser numéricos, como el número de líneas de código, o categóricos, como las clasificaciones de calidad (alto, medio, bajo). Según estos autores, “estos atributos medibles proporcionan una base para evaluar diferentes aspectos del software, permitiendo una comprensión más detallada de su rendimiento y calidad.”

La relación entre las entidades y los atributos es fundamental para obtener métricas precisas y útiles ya que, al asignar valores a los atributos de las entidades, se pueden

evaluar aspectos específicos del software y tomar decisiones basadas en datos concretos (Ferreira et al., 2006).

Según Mancha (2004), elegir las entidades correctas para medir es crucial, ya que una selección inadecuada puede resultar en métricas que no reflejan adecuadamente el desempeño o la calidad del software.

### Métrica de software

Desde hace ya unas décadas las métricas de software han sido una herramienta fundamental para evaluar y mejorar el software. Sin embargo, su aplicación requiere tener en cuenta varios factores, como el tipo de software, las condiciones en las que se realiza la medición y las restricciones que pueden llegar a presentarse.

En cuanto a la definición, una métrica en software es un estándar de medida que indica el grado en el que un sistema o proceso de software posee una determinada propiedad (Doria, 2001). Aunque se suelen confundir las mediciones con las métricas, estas representan roles o funciones y las mediciones son los valores numéricos obtenidos al aplicar esas métricas. López et al. (2006) mencionan que las mediciones cuantitativas son fundamentales en todas las ciencias y en el desarrollo de software se busca adoptar enfoques similares para garantizar mediciones objetivas, reproducibles y cuantificables. Doria (2001) también señala que las mediciones son valiosas en diversas áreas, como la planificación de calendarios y presupuestos, la gestión presupuestaria, el aseguramiento de calidad, las pruebas, la depuración, la optimización del rendimiento y la asignación eficiente de tareas.

¿Por qué son tan cruciales las métricas de pruebas para el desarrollo de software?

Evaluación imparcial: Estayno et al. (2009) indican que las métricas permiten a los desarrolladores y al equipo de aseguramiento de calidad evaluar si sus herramientas y sistemas de pruebas cumplen con los estándares requeridos o si es necesario modificarlos para mejorar la eficacia y con esto al basarse en datos cuantitativos, se minimizan las opiniones subjetivas.

Mejor toma de decisiones: Según Martínez (1994), las métricas de pruebas ayudan a los equipos a tomar decisiones más informadas sobre el proceso de pruebas. Poniendo como ejemplo las métricas de pruebas automatizadas que pueden revelar si una prueba de este estilo está tomando más tiempo del que debería, permitiendo ajustar las pruebas futuras en función de estos datos.

Visibilidad mejorada: López et al. (2006) afirman que las métricas de pruebas proporcionan al equipo de entrega la información necesaria para realizar ajustes en el control de calidad y garantizar la detección consistente de errores críticos. También con esto se logra identificar áreas que se suelen o puedan cometer errores, con esto se entiende mejor la relación entre las acciones de los usuarios y los fallos en la aplicación.

### Unidad de medición

Muñoz y Velthuis (2010) explican que “una unidad de medida en software es una escala o estándar utilizado para expresar el valor de un atributo específico. Es crucial que esta unidad sea clara, consistente y relevante para el atributo que se está evaluando.” Emplear unidades de medida precisas hace que la comunicación y la interpretación de los datos sean mucho más sencillas, asegurando que todos los involucrados tengan una comprensión común sobre lo que se está midiendo.

Las unidades de medida en software se dividen en dos grandes categorías, según muestran García et al. (2024). Las numéricas pueden ser absolutas, como el número de líneas de código o el tiempo de ejecución en segundos, o relativas, como la complejidad ciclomática o la densidad de defectos. Por otro lado, las categóricas incluyen las nominales, que utilizan etiquetas para clasificar entidades como el tipo de lenguaje de programación o el estado de un defecto, y las ordinales, que establecen un orden entre categorías sin implicar distancias iguales, como la prioridad de un requerimiento o la severidad de un defecto.

Estayno et al. (2009) aclaran que las unidades de medida son fundamentales en el desarrollo de software por varias razones; facilitan la comparabilidad al permitir que se vean los resultados entre proyectos o equipos diferentes y son esenciales para el análisis, ya que permiten realizar análisis estadísticos y correlacionar diversas métricas.

### Indicador de medición

García y Beltrán (1995) definieron los indicadores como “una medida compuesta que se obtiene a partir de una o varias unidades de medida, y que nos permite evaluar el desempeño de un sistema o proceso de software en relación con un objetivo específico”. A diferencia de una unidad de medida, que cuantifica un atributo específico de una entidad, un indicador combina varias unidades de medida para ofrecer una visión general del desempeño. García y Beltrán (1995) ejemplifican que, mientras que el número de defectos y el tiempo de ejecución son unidades de medida, la densidad de defectos (número de defectos por kilolínea de código) actúa como un indicador que relaciona errores con el tamaño del código, ofreciendo una visión más completa de la calidad del software.

Gonzales et al. (2012) señalan que las características que definen un buen indicador incluyen relevancia, precisión, consistencia, comprensibilidad y comparabilidad. Esto quiere decir que un indicador debe medir aspectos importantes para la evaluación, ser exacto y calculado de manera uniforme a lo largo del tiempo, ser fácil de interpretar y permitir comparaciones entre diferentes proyectos o versiones del software. Los indicadores se dividen en varias categorías: indicadores de producto, que miden la calidad del producto final; indicadores de proceso, que evalúan la eficiencia de los procesos de desarrollo; e indicadores de proyecto, que valoran el desempeño general del proyecto (Doria, 2001).

## ISO/IEC 12207

El ISO/IEC 12207 es un estándar internacional que establece un marco integral para los procesos del ciclo de vida del software. Según Singh (1996), este estándar proporciona una estructura sistemática para gestionar y controlar las actividades de desarrollo y mantenimiento del software, garantizando la calidad y eficiencia en los procesos involucrados.

### Medición ISO/IEC 12207

Es un proceso de apoyo esencial que busca proporcionar información cuantitativa y objetiva para facilitar la toma de decisiones y promover la mejora continua de los procesos y productos de software. Pino et al. (2006) destacan la importancia de adaptar y aplicar correctamente este estándar en países en desarrollo para evaluar la madurez de los procesos de software, enfatizando que la medición es clave para identificar áreas de mejora y asegurar el cumplimiento de objetivos estratégicos.

Los objetivos principales del proceso de medición incluyen:

**Planificación de la medición:** Identificar las necesidades de información y definir las métricas adecuadas que serán utilizadas para satisfacer estas necesidades.

**Implementación de la medición:** Recopilar, almacenar y verificar los datos de acuerdo con procedimientos establecidos y estandarizados.

**Análisis de datos:** Interpretar los datos recopilados para obtener información útil y relevante que apoye la gestión del proyecto.

**Reporte de resultados:** Comunicar los hallazgos a las partes interesadas de manera clara y oportuna, facilitando la toma de decisiones informadas.

**Uso de la información:** Aplicar los resultados del análisis para mejorar los procesos y productos, fomentando una cultura de mejora continua.

La importancia de la medición en el ciclo de vida del software radica en su capacidad para:

**Control y seguimiento:** Permitir el monitoreo constante del progreso del proyecto y el rendimiento de los procesos.

**Toma de decisiones informada:** Proporcionar datos objetivos que respaldan las decisiones de gestión y planificación.

**Mejora continua:** Identificar áreas de mejora en procesos y productos, facilitando la implementación de acciones correctivas y preventivas.

**Cumplimiento de objetivos:** Asegurar que se alcancen los objetivos de calidad, tiempo y costo establecidos por la organización.

Para una implementación efectiva de la medición, es crucial:

Definir indicadores clave de rendimiento (KPIs): Seleccionar métricas que estén alineadas con los objetivos del proyecto y de la organización.

Establecer procedimientos claros: Documentar cómo se realizarán las actividades de medición, incluyendo herramientas y métodos a utilizar, asegurando la consistencia y repetibilidad.

Capacitación del equipo: Asegurar que el personal involucrado comprenda la importancia de la medición y esté capacitado para llevarla a cabo de manera efectiva.

Revisión y adaptación: Evaluar regularmente el proceso de medición y ajustar según sea necesario para mejorar su eficacia y relevancia.

En el ámbito educativo, Aydan et al. (2017) proponen enfoques innovadores, como el uso de juegos serios, para enseñar los procesos del ciclo de vida del software según el ISO/IEC 12207. Este enfoque resalta la necesidad de comprender profundamente estos procesos para aplicarlos eficazmente en proyectos reales, promoviendo una mejor adopción y adaptación de las prácticas de medición.

## Planeación

La fase de planeación de la medición implica identificar las necesidades de información y definir las métricas adecuadas para satisfacer estas necesidades. Es esencial establecer objetivos claros y alinearlos con las metas del proyecto y de la organización. Se deben seleccionar indicadores clave de rendimiento (KPIs) que permitan monitorear el progreso y el rendimiento de los procesos. Según Aydan et al. (2017), una comprensión profunda de los procesos del ciclo de vida del software es crucial en esta etapa, y enfoques educativos innovadores, como el uso de juegos serios, pueden facilitar esta comprensión y la correcta implementación de los estándares.

## Ejecución

En la fase de ejecución, se implementa el plan de medición establecido. Esto incluye la recopilación, almacenamiento y verificación de los datos de acuerdo con procedimientos estandarizados. Es fundamental garantizar que el personal involucrado esté capacitado y comprenda la importancia de la medición para llevar a cabo estas actividades de manera efectiva. La consistencia y precisión en la recopilación de datos son esenciales para obtener resultados fiables que puedan apoyar la toma de decisiones informadas.

## Evaluación

La evaluación implica el análisis de los datos recopilados para obtener información útil y relevante. Se interpretan los datos para identificar tendencias, desviaciones y oportunidades de mejora. Los hallazgos se comunican a las partes interesadas de manera clara y oportuna, facilitando la implementación de acciones correctivas y preventivas. Pino

et al. (2006) señalan que esta etapa es crucial para evaluar la madurez de los procesos de software y promover acciones que conduzcan a la mejora continua.

#### ISO/IEC TR 15504

El ISO/IEC TR 15504, también conocido como SPICE (Software Process Improvement and Capability Determination), es un estándar internacional que proporciona un marco para la evaluación y mejora de los procesos de desarrollo de software. Este estándar permite a las organizaciones analizar la capacidad y madurez de sus procesos, identificando áreas de mejora para optimizar la calidad y eficiencia de sus productos. Según Rout (2003), el ISO/IEC TR 15504 ha evolucionado significativamente para convertirse en una referencia esencial en la industria del software, facilitando la adopción de prácticas consistentes y efectivas a nivel global.

#### Medición en el ISO/IEC TR 15504

La medición es un componente fundamental en el ISO/IEC TR 15504, ya que proporciona los datos necesarios para evaluar el desempeño de los procesos y determinar su nivel de madurez. A través de la medición, las organizaciones pueden obtener información cuantitativa y cualitativa que respalda la toma de decisiones y fomenta la mejora continua. Torres Samaniefo (2007) destaca que una adecuada aplicación de las prácticas de medición es crucial para comparar los procesos existentes con los modelos de referencia y para planificar estrategias de mejora efectivas. La importancia de la medición radica en varios aspectos clave:

**Diagnóstico preciso:** Permite identificar con exactitud las fortalezas y debilidades de los procesos actuales.

**Benchmarking:** Facilita la comparación con estándares internacionales y con otras organizaciones del sector.

**Toma de decisiones informada:** Proporciona datos objetivos que respaldan decisiones estratégicas y tácticas.

**Mejora continua:** Establece una base para implementar ciclos de mejora continua, esenciales para mantener la competitividad en un mercado dinámico.

#### Ejecución

La fase de ejecución en el contexto de la medición según el ISO/IEC TR 15504 implica llevar a cabo una serie de actividades diseñadas para recopilar y analizar información detallada sobre los procesos de software. Esta fase es crítica, ya que la calidad de los datos recopilados influye directamente en la precisión de la evaluación y en la efectividad de las acciones de mejora posteriores. Durante esta fase, se desarrollan las siguientes acciones:

**Planificación de la medición:** Se establecen los objetivos específicos de la medición, se seleccionan las métricas adecuadas y se definen los métodos y herramientas que se



utilizarán. Es esencial que esta planificación esté alineada con los objetivos estratégicos de la organización.

**Recopilación de datos:** Se recolecta información relevante sobre los procesos y prácticas actuales. Esto incluye métricas como tiempos de desarrollo, tasas de defectos, cumplimiento de hitos y satisfacción del cliente. Lee et al. (2004) enfatizan la importancia de utilizar herramientas y técnicas adecuadas para garantizar la precisión y confiabilidad de los datos, como sistemas automatizados de seguimiento y bases de datos centralizadas.

**Aplicación de métodos de evaluación:** Se emplean diversas técnicas para obtener una visión completa de cómo se ejecutan los procesos en la práctica. Estas técnicas incluyen:

Entrevistas

Cuestionarios

Observaciones directas

Revisión de documentación

**Verificación y validación de datos:** Se aseguran la consistencia y precisión de los datos recopilados. Esto implica revisar los datos para detectar inconsistencias, errores o anomalías que puedan afectar la interpretación de los resultados. La validación cruzada con múltiples fuentes de información es una práctica recomendada.

**Participación del personal clave:** Se involucra activamente a los responsables de los procesos y a otros stakeholders. Esto no solo enriquece la calidad de la información obtenida, sino que también promueve el compromiso y la aceptación de los resultados y recomendaciones posteriores.

Evaluación

La fase de evaluación se centra en analizar y interpretar los datos recopilados durante la ejecución para determinar el nivel de madurez y capacidad de los procesos de software. Este análisis es esencial para identificar fortalezas, debilidades y oportunidades de mejora, y para establecer planes de acción concretos. Los pasos clave en esta fase incluyen:

**Análisis de datos:** Se examinan los datos utilizando métodos estadísticos y cualitativos para extraer información significativa. Esto puede implicar el uso de software especializado para el análisis de métricas y tendencias.

**Comparación con modelos de referencia:** Los resultados se contrastan con los niveles de capacidad definidos en el ISO/IEC TR 15504, que van desde el Nivel 1 (Proceso Inicial) hasta el Nivel 5 (Proceso Optimizado).

**Identificación de brechas y áreas de mejora:** Se detectan discrepancias entre las prácticas actuales y las recomendadas por el estándar. Esto facilita la identificación de áreas

específicas que requieren mejoras y permite priorizar acciones según su impacto y urgencia.

**Análisis de causas raíz:** Se investigan las causas subyacentes de las deficiencias identificadas. Este análisis es crucial para desarrollar soluciones efectivas y sostenibles, evitando abordar solo síntomas superficiales.

**Formulación de recomendaciones:** Basándose en el análisis, se elaboran propuestas concretas para mejorar los procesos.

**Documentación y reporte:** Se prepara un informe detallado que presenta los hallazgos, conclusiones y recomendaciones. Este documento sirve como base para la toma de decisiones y la planificación de acciones futuras. Es fundamental que el reporte sea claro, objetivo y orientado a las necesidades de los stakeholders.

Torres Samaniefo (2007) señala que la evaluación no es un fin en sí misma, sino un medio para impulsar mejoras continuas. Al establecer metas realistas y medir el progreso a lo largo del tiempo, las organizaciones pueden avanzar sistemáticamente hacia niveles superiores de madurez y excelencia operativa.

## COMPETISOFT

El COMPETISOFT es un modelo de mejora de procesos de software diseñado específicamente para pequeñas y medianas empresas (PYMEs) en Iberoamérica. Su objetivo principal es fortalecer la competitividad y capacidad de las PYMEs desarrolladoras de software mediante la implementación de prácticas de calidad adaptadas a sus necesidades y contextos locales. Cornide-Reyes et al. (2024) destacan que COMPETISOFT ofrece una guía práctica y flexible, permitiendo a las empresas mejorar sus procesos sin incurrir en los altos costos o complejidades asociadas a otros modelos más generalizados.

### Medición en el COMPETISOFT

La medición es un componente esencial en el modelo COMPETISOFT, ya que permite a las organizaciones evaluar y controlar el desempeño de sus procesos y productos de software. A través de prácticas de medición efectivas, las PYMEs pueden identificar áreas de mejora, tomar decisiones basadas en datos y asegurar la calidad de sus productos.

Esponda et al. (2012) señalan que la incorporación de prácticas de medición es fundamental para cumplir con estándares de calidad internacionales, como la ISO 9001, especialmente cuando se utiliza el modelo COMPETISOFT como guía de implementación. La medición en COMPETISOFT se enfoca en tres aspectos clave:

**Planificación de la Medición:** Implica definir claramente los objetivos de medición, seleccionar las métricas adecuadas y establecer procedimientos para la recopilación y

análisis de datos. Es fundamental que las métricas elegidas estén alineadas con los objetivos estratégicos de la empresa y sean relevantes para los procesos y productos específicos.

**Implementación y Ejecución:** Una vez planificada, la medición debe ser implementada de manera consistente en todos los proyectos y procesos relevantes. Esto incluye la recopilación sistemática de datos, el uso de herramientas adecuadas y la formación del personal en prácticas de medición. Pasini et al. (2008) mencionan que, en el contexto de las PYMEs, es importante utilizar métodos y herramientas que sean accesibles y no impongan una carga significativa al equipo de desarrollo. La simplicidad y practicidad son esenciales para asegurar que la medición se integre de forma natural en las actividades diarias.

**Análisis y Mejora Continua:** Los datos recopilados deben ser analizados para extraer información útil que permita mejorar los procesos y productos. Este análisis puede revelar tendencias, identificar problemas recurrentes y evaluar el cumplimiento de los objetivos establecidos. Esponda et al. (2012) enfatizan que el análisis de datos debe conducir a acciones concretas, como ajustes en los procesos, actualizaciones en los planes de proyecto o mejoras en las prácticas de desarrollo. La retroalimentación obtenida a través de la medición es esencial para el ciclo de mejora continua que promueve COMPETISOFT, permitiendo a las PYMEs adaptarse y evolucionar en un entorno competitivo.

## Modelo CMMI

El Modelo de Integración de Capacidad y Madurez (CMMI, por sus siglas en inglés) es un marco de referencia que proporciona prácticas y directrices para mejorar los procesos de una organización, especialmente en el ámbito del desarrollo de software y sistemas. CMMI ayuda a las organizaciones a evaluar y mejorar su capacidad para desarrollar productos y servicios de alta calidad de manera eficiente y predecible. Según Solarte et al. (2009), CMMI es uno de los modelos de calidad más reconocidos y utilizados en la industria del software para optimizar procesos y garantizar la calidad de los productos.

## Medición en el Modelo CMMI

La medición es un componente fundamental en el modelo CMMI, ya que permite a las organizaciones evaluar el desempeño de sus procesos y productos, identificar áreas de mejora y tomar decisiones basadas en datos objetivos. Puella (2013) destaca que CMMI incorpora prácticas de verificación y validación que se apoyan en la medición para asegurar que los productos cumplen con los requisitos y estándares de calidad establecidos. Estas prácticas implican:

**Definición de métricas y objetivos:** Establecer indicadores clave de rendimiento que permitan medir el avance y la efectividad de los procesos.

**Recopilación de datos:** Implementar mecanismos para recolectar información relevante de manera sistemática y consistente.

**Análisis de resultados:** Interpretar los datos recopilados para identificar tendencias, desviaciones y oportunidades de mejora.

**Retroalimentación y mejora continua:** Utilizar los hallazgos para ajustar procesos y prácticas, fomentando un ciclo de mejora continua.

#### Aspectos Clave de la Medición en CMMI

**Establecimiento de Objetivos de Medición:** Definir claramente qué se quiere medir y por qué. Los objetivos deben estar alineados con las metas estratégicas de la organización y las necesidades del proyecto.

**Selección de Métricas Adecuadas:** Elegir métricas que sean relevantes, significativas y accionables. Esto incluye métricas de proceso (como eficiencia y cumplimiento de plazos) y métricas de producto (como defectos y calidad funcional).

**Recopilación y Análisis de Datos:** Implementar procedimientos para recopilar datos de manera consistente y confiable. El análisis de estos datos permite identificar tendencias, variaciones y oportunidades de mejora.

**Uso de los Resultados para la Toma de Decisiones:** Los hallazgos obtenidos de la medición deben informar las decisiones de gestión, permitiendo ajustar planes, asignar recursos y mejorar prácticas.

**Integración en la Cultura Organizacional:** Fomentar una cultura que valore la medición y el uso de datos para mejorar. Esto implica capacitar al personal y promover la transparencia y el aprendizaje continuo.

#### Desafíos en la Implementación de la Medición en CMMI

Palacios y Porcell (2012) señalan que las organizaciones pueden enfrentar obstáculos al implementar el modelo CMMI, especialmente en lo que respecta a la medición. Algunos de estos desafíos incluyen:

**Resistencia al Cambio:** El personal puede mostrar reticencia a adoptar nuevas prácticas de medición, especialmente si perciben que aumentará su carga de trabajo.

**Falta de Recursos:** La implementación efectiva de la medición requiere inversión en herramientas, capacitación y tiempo dedicado a la recopilación y análisis de datos.

**Comprensión Inadecuada del Modelo:** Sin una comprensión profunda de CMMI y su enfoque en la medición, las organizaciones pueden implementar prácticas de manera superficial o incorrecta.

Para superar estos obstáculos, es fundamental contar con el compromiso de la alta dirección, proporcionar capacitación adecuada y comunicar claramente los beneficios de la medición y la mejora de procesos.

## ISO 9001/IEC 90003

El ISO 9001 es un estándar internacional que especifica los requisitos para un sistema de gestión de la calidad (SGC), diseñado para ayudar a las organizaciones a garantizar que cumplen con las necesidades de los clientes y otras partes interesadas, además de cumplir con los requisitos legales y reglamentarios relacionados con un producto o servicio. Por su parte, el ISO/IEC 90003 proporciona una guía específica para la aplicación del ISO 9001 en organizaciones que desarrollan, suministran y mantienen software y servicios relacionados. Según Martínez (2016), estos estándares son fundamentales para mejorar la eficiencia y la satisfacción del cliente en el sector del software.

### Medición en el ISO 9001/IEC 90003

La medición es un componente esencial en la implementación de un SGC según ISO 9001 e ISO/IEC 90003, ya que permite a las organizaciones evaluar el desempeño de sus procesos y productos, asegurando la mejora continua y el cumplimiento de los requisitos establecidos. Yáñez (2008) destaca que la medición proporciona información valiosa para la toma de decisiones y ayuda a identificar áreas de mejora en la calidad del software y los servicios asociados.

#### Planeación

La etapa de planeación es crucial para establecer una base sólida para el proceso de medición. Según el Equipo Vértice (2010), esta fase implica:

**Definir objetivos de medición:** Establecer claramente qué se desea medir y por qué. Los objetivos deben estar alineados con las metas estratégicas de la organización y enfocados en mejorar la calidad y satisfacción del cliente.

**Seleccionar métricas adecuadas:** Identificar indicadores clave de rendimiento (KPIs) que sean relevantes y significativos para los procesos y productos específicos.

**Desarrollar un plan de medición:** Elaborar un plan detallado que describa cómo se recopilarán, analizarán y utilizarán los datos.

**Capacitación del personal:** Asegurar que el equipo esté familiarizado con las prácticas de medición y comprenda la importancia de su participación en el proceso.

#### Ejecución

La fase de ejecución implica llevar a cabo el plan de medición establecido. Martínez (2016) señala que esta etapa incluye:

**Recopilación de datos:** Implementar los procedimientos para recolectar datos de manera sistemática y consistente. Es fundamental garantizar la precisión y confiabilidad de la información obtenida.

Uso de herramientas y técnicas adecuadas: Emplear software y metodologías que faciliten la recopilación y análisis de datos, reduciendo errores y aumentando la eficiencia.

Monitoreo continuo: Realizar un seguimiento regular de los procesos y productos, asegurando que se cumplen los estándares de calidad y se detectan oportunamente posibles desviaciones.

Comunicación efectiva: Mantener una comunicación abierta entre los miembros del equipo y otras partes interesadas, compartiendo avances y hallazgos relevantes.

### Evaluación

La etapa de evaluación se centra en analizar y utilizar los datos recopilados para mejorar los procesos y productos. De acuerdo con Yáñez (2008), esta fase comprende el análisis de datos para identificar tendencias, patrones y áreas que requieren atención, utilizando técnicas estadísticas y herramientas de análisis. Se evalúa el desempeño real frente a los objetivos establecidos y los requisitos de calidad definidos en el ISO 9001/IEC 90003.

La implementación de estas acciones y el monitoreo de su efectividad promueven un ciclo constante de mejora en los procesos y productos. La elaboración de informes detallados que reflejen los resultados de la evaluación facilita la toma de decisiones y la transparencia en la gestión de la calidad.

## Referencias

### Necesidad de medir

López, A. V., Sánchez, A., & Montejano, G. A. (2016, June). Definición de Métricas de Calidad para Productos de Software. In XVIII Workshop de Investigadores En Ciencias de La Computación (WICC 2016, Entre Ríos, Argentina).

Castañeda, S., & Pardo, E. (2007). Métricas de calidad de software. Recuperado de

Dávila, A. L., Gómez, J. G., & Gómez, M. M. (2007). Métricas para medir la calidad del software. Ingeniería Industrial, 28(3), 183–192. Recuperado de

García, M. N. M., Quintales, L. A. M., García-Peñalvo, F. J., & Martín, M. J. P. (2001). Aplicación de Técnicas de Minería de Datos en la Construcción y Validación de Modelos Predictivos y Asociativos a Partir de Especificaciones de Requisitos De Software. In ADIS.

### Concepto medible

Ferreira, M., García, F., Ruiz, F., Bertoa, M. F., Calero, C., Vallecillo, A., ... & Mora, B. (2006). Medición del software ontología y metamodelo. Departamento de Tecnologías y Sistemas de la Información, Castilla La Mancha.

Mancha, C. L. (2004). Una Ontología de la Medición del Software (Doctoral dissertation, Universidad de Castilla-La Mancha).

Strub, A. M., & Savi, C. A. (2013). Métricas estadísticas asociadas al proceso de desarrollo y mejora continua de software. Revista Iberoamericana para la Investigación y el Desarrollo Educativo ISSN: 2007-2619, (10).

### Atributo de medición

Acosta, J. C., Greiner, C. L., Dapozo, G. N., & Estayno, M. G. (2012). Medición de atributos POO en frameworks de desarrollo PHP. In XVIII Congreso Argentino de Ciencias de la Computación.

Ferreira, M., García, F., Ruiz, F., Bertoa, M. F., Calero, C., Vallecillo, A., ... & Mora, B. (2006). Medición del software ontología y metamodelo. Departamento de Tecnologías y Sistemas de la Información, Castilla La Mancha.

Mancha, C. L. (2004). Una Ontología de la Medición del Software (Doctoral dissertation, Universidad de Castilla-La Mancha).

### Entidad de medición

Bertoa, M. F., & Vallecillo, A. (2006). Medidas de Usabilidad de Componentes Software. IEEE Latin America Transactions, 4(2), 136-143.

Ferreira, M., García, F., Ruiz, F., Bertoa, M. F., Calero, C., Vallecillo, A., ... & Mora, B. (2006). Medición del software ontología y metamodelo. Departamento de Tecnologías y Sistemas de la Información, Castilla La Mancha.

Mancha, C. L. (2004). Una Ontología de la Medición del Software (Doctoral dissertation, Universidad de Castilla-La Mancha).

#### Métricas de software

Doria, H. G. (2001). Las Métricas de Software y su Uso en la Región.

Estayno, M. G., Dapozo, G. N., Cuenca Pletsch, L. R., & Greiner, C. L. (2009). Modelos y Métricas para evaluar Calidad de Software. In XI Workshop de Investigadores en Ciencias de la Computación.

López, A. V., Sánchez, A., & Montejano, G. A. (2016, June). Definición de Métricas de Calidad para Productos de Software. In XVIII Workshop de Investigadores En Ciencias de La Computación (WICC 2016, Entre Ríos, Argentina).

Martínez Flores, J. L. (1994). Métricas de software en lenguajes de cuarta generación (Doctoral dissertation, Universidad Autónoma de Nuevo León).

#### Unidad de medición

Estayno, M. G., Dapozo, G. N., Cuenca Pletsch, L. R., & Greiner, C. L. (2009). Modelos y Métricas para evaluar Calidad de Software. In XI Workshop de Investigadores en Ciencias de la Computación.

García-Mireles, G. A., Olivero, N. P., & Avila-George, H. (2024). DevOps y la Medición de la Calidad del Producto de Software: Hallazgos Preliminares. RISTI-Revista Iberica de Sistemas e Tecnologias de Informacao, 2024(53), 37-52.

Muñoz, C. C., & Velthuis, M. G. P. (2010). Calidad del producto y proceso software. Editorial Ra-Ma.

#### Indicador de medición

Doria, H. G. (2001). Las Metricas de Software y su Uso en la Region.

García León, D., & Beltrán Benavides, A. (1995). Un enfoque actual sobre la calidad del software. Acimed, 3(3), 40-42.

González-Sánchez, J. L., Montero-Simarro, F., & Gutiérrez-Vela, F. L. (2012). Evolución del concepto de usabilidad como indicador de calidad del software. Profesional de la información, 21(5), 529-536.



Aydan, U., Yilmaz, M., Clarke, P. M., & O'Connor, R. V. (2017). Teaching ISO/IEC 12207 software lifecycle processes: a serious game approach. *Computer Standards & Interfaces*, 54, 129-138.

Pino, F. J., García, F., Ruiz, F., & Piattini, M. (2006). Adaptación de las normas ISO/IEC 12207: 2002 e ISO/IEC 15504: 2003 para la evaluación de la madurez de procesos software en países en desarrollo. *IEEE Latin America Transactions*, 4(2), 17-24.

Singh, R. (1996). International Standard ISO/IEC 12207 software life cycle processes. *Software Process Improvement and Practice*, 2(1), 35-50.

#### ISO/IEC TR 15504

Estudio comparativo entre los estándares ISO/IEC TR 15504 y CMMI (Bachelor's thesis, QUITO/EPN/2007).

Lee, E. S., Lee, K. W., Kim, T. H., & Jung, I. H. (2004, May). Introduction and evaluation of development system security process of ISO/IEC TR 15504. In *International Conference on Computational Science and Its Applications* (pp. 451-460). Berlin, Heidelberg: Springer Berlin Heidelberg.

Rout, T. P. (2003). ISO/IEC 15504—Evolution to an international standard. *Software Process: Improvement and Practice*, 8(1), 27-40. Torres Samaniefo, M. Z. (2007).

#### COMPETISOFT

Cornide-Reyes, H., Madrigal, G., Muñoz, G., Duran, C., Jorquera, J., & Morales, J. (2024). Análisis del uso de modelos de mejora de procesos de software en el desarrollo ágil. *Ingeniare. Revista chilena de ingeniería*, 32, 0-0.

Esponda, S., Pasini, A. C., Pesado, P. M., & Boracchia, M. (2012). ISO 9001 in software-developing VSEs assisted by the COMPETISOFT Model. In *XVIII Congreso Argentino de Ciencias de la Computación*.

Pasini, A., Esponda, S., Bertone, R., & Pesado, P. (2008). Aseguramiento de Calidad en PYMES que desarrollan software. Una experiencia desde el proyecto COMPETISOFT. *Repos. Inst. la Univ. Nac. la Plata*, 10.

#### Modelo CMMI

Palacios, H., & Porcell, N. (2012). Obstáculos al implantar el modelo CMMI. *Revista EAN*, (72), 110-127.

Puello, O. (2013). Modelo de verificación y validación basado en CMMI. Investigación e Innovación en Ingenierías, 1(1).

Solarte, G., Muñoz, L., & Arias, B. (2009). Modelos de calidad para procesos de software. Scientia Et Technica, 2(42).

ISO 9001/IEC 90003

Equipo Vértice. (2010). Gestión de la calidad (ISO 9001/2008). Editorial Vértice.

Martínez, J. A. G. (2016). Guía para la aplicación de ISO 9001 2015. Alpha Editorial.

Yáñez, C. (2008). Sistema de gestión de calidad en base a la norma ISO 9001. Internacional eventos, 9(1), 1-9.