



# Estándares para la calidad de *software*

## Standards for Software Quality

Néstor Javier Acosta<sup>1</sup> Luis Alfonso Espinel<sup>2</sup> Jaime Leonardo García<sup>3</sup>

**Para citar este artículo:** Acosta, N. J.; Espinel, L. A.; García, J. L. (2017). Estándares para la calidad de *software*. *TIA*, 5(1), pp. 75-84.

### Resumen

Hoy en día los productos de *software* se han convertido en herramientas estratégicas para el cumplimiento de los objetivos en las organizaciones; por lo tanto, el interés por la calidad del *software* crece en la medida que los usuarios son más exigentes y requieren productos que cumplan con sus necesidades. Los estándares o metodologías definen criterios de desarrollo que tienen como objetivo principal producir *software* confiable de alta calidad. El presente documento describe diferentes modelos de evaluación de calidad de *software*, los cuales pueden ser utilizados como medio para auditar la calidad de los productos de *software* ya implementados en entornos productivos, así como de productos que se encuentran en proceso de desarrollo.

**Palabras clave:** calidad de *software*, estándares de calidad, metodología métrica, normas ISO (International Organization for Standardization), CMMI, SPICE, SQuARE.

### Abstract

Nowadays software products have become strategic tools to get organization aims. Therefore, interest in software quality is increasing as users are more demanding and require products that meet their needs. Standards or methodologies defined development criteria whose main aim is to get high quality reliable software. This paper describes different models of software quality evaluation, which can be used to audit the software quality products already deployed in production environments, as well as products that are in development.

**Keywords:** software quality, quality standards, metrics methodology, ISO rules, Capability Maturity Model Integration, Software Process Improvement and Capability Determination, Software Quality Requirements and Evaluation.

### ARTÍCULO DE INVESTIGACIÓN

Fecha de recepción:  
08-04-2015

Fecha de aceptación:  
03-06-2017

ISSN: 2344-8288

Vol. 5 No. 1

Enero - junio 2017

Bogotá-Colombia

<sup>1</sup> Ingeniero de Sistemas, Universidad Nacional Abierta y a Distancia. Especialista en Proyectos Informático, Universidad Distrital Francisco José de Caldas. Correo electrónico: ing.nestoracosta@gmail.com

<sup>2</sup> Ingeniero, Universidad Distrital Francisco José de Caldas. Especialista en Proyectos Informático, Universidad Distrital Francisco José de Caldas. Correo electrónico: lespinel@gmail.com

<sup>3</sup> Ingeniero Electrónico, Universidad Santo Tomás. Especialista en Proyectos Informático, Universidad Distrital Francisco José de Caldas. Correo electrónico: ing.leotovar89@gmail.com

## INTRODUCCIÓN

La calidad del *software* hoy en día juega un papel importante dentro de las empresas, por esto es necesario tener un punto de comparación y mirar si el producto entregado cumple con los requerimientos y necesidades que tienen los usuarios o clientes; para esto se parte de diferentes estándares que se enfocan en la calidad del *software*.

Así, se puede decir que los requisitos del *software* son la base para la medida de la calidad y que la carencia de los requisitos es una falta de calidad; los estándares definen un conjunto de criterios que sirven como guía para la entrega de un *software*, si no se tiene un estándar lo más probable es que un entregable de *software* tenga muchas deficiencias.

## CONTENIDO

### Estándares para la calidad del *software*

Un estándar es una serie de recomendaciones a seguir para la entrega de un producto, dentro de la calidad del *software* es una metodología para que los entregables cumplan con las expectativas del negocio. Dentro de los estándares más aplicados a la calidad del software se encuentran los siguientes:

ISO/IEC 9126, ISO/IEC 14598, ISO/IEC 25000, IEEE 1061, CMMI, SPICE, los cuales serán descritos a continuación.

#### *Norma ISO/IEC 9126*

La norma ISO/IEC 9126, tiene como fin cuantificar los productos de *software*, esta norma nos indica las características de la calidad del *software* y fue elaborado para cubrir las necesidades de error que genera. La norma fue diseñada en los siguientes factores: calidad de proceso, calidad del producto, calidad del *software* y calidad de uso. Por otro lado, presenta dos modelos de calidad, el primero

referido a la calidad interna y externa y el segundo modelo referido a la calidad en uso; a continuación se describe cada uno de ellos.

ISO/IEC 9126-1: hace referencia al modelo de calidad de un producto de *software*, en primera instancia especifica las características del modelo que están divididas en seis tanto en calidad interna como externa, además dentro de estas se dividen en otras categorías. La calidad externa es aquella que es medible con el uso del producto, y la calidad interna realiza una revisión relacionada con los atributos del *software* (Figura 1). Permite especificar y calificar la calidad de un *software* desde diferentes puntos de vista, los cuales se relacionan con los requerimientos, uso, adquisición, desarrollo, auditoria de *software* y mantenimiento.

#### Funcionalidad:

- Adecuación.
- Exactitud.
- Interoperabilidad.
- Seguridad de acceso.
- Cumplimiento funcional.

#### Fiabilidad:

- Madurez.
- Tolerancia a fallos.
- Capacidad de recuperación.
- Cumplimiento de la fiabilidad.

#### Usabilidad:

- Capacidad para ser entendido.
- Capacidad para ser aprendido.
- Capacidad para ser operado.
- Capacidad de atracción.
- Cumplimiento de la usabilidad.

#### Eficiencia:

- Comportamiento temporal.
- Utilización de recursos.
- Cumplimiento de la eficiencia.

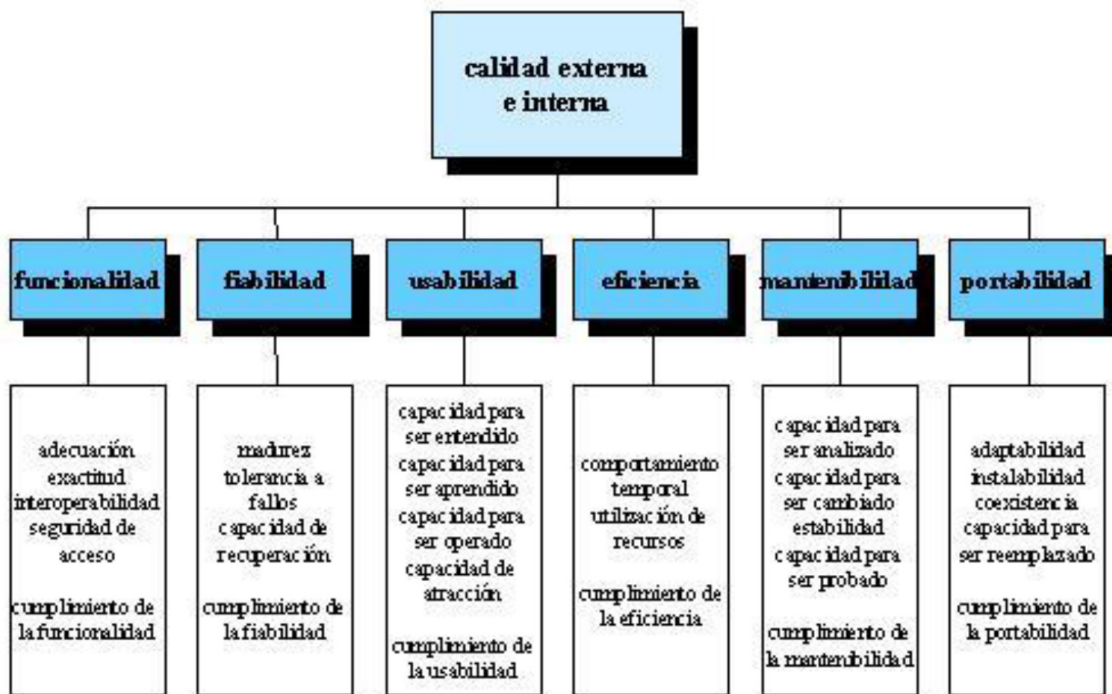
**Mantenibilidad:**

- Capacidad para ser analizado.
- Capacidad para ser cambiado.
- Estabilidad.
- Capacidad para ser probado.
- Cumplimiento de la mantenibilidad.

**Portabilidad:**

- Adaptabilidad.
- Instalabilidad.
- Coexistencia.
- Capacidad para reemplazar.

ISO/IEC TR 9126-2:2003 (métricas externas): suministra unos parámetros que hacen posible la medición a través de los seis atributos que tiene la norma; los parámetros externos hacen posible cuantificar el comportamiento de un sistema basada en un PC a través de *software*. Estos parámetros proporcionan a los verificadores, usuarios, desarrolladores y evaluadores la capacidad para evaluar la calidad del *software* durante su funcionamiento. Los parámetros los pueden definir los evaluadores o el área usuaria, con el fin de obtener unos parámetros básicos y así aplicarlos a los productos, basados en el modelo estándar de calidad.



**Figura 1.** Modelo de calidad

Fuente: [1]

ISO/IEC TR 9126-3:2003 (métricas internas): el fundamento de esta métrica es lograr que la calidad externa y la calidad de uso pretendida, esta métrica hace posible evaluar la calidad del *software* relacionado con los inconvenientes presentados antes de su puesta en producción. Este tipo de métrica utiliza número de elementos de construcciones de *software*, donde se visualizan las sentencias de código fuente, gráficos, flujo de datos, estado de procesos, entre otros.

ISO/IEC TR 9126-4:2004 (calidad en uso): las métricas de calidad en uso calculan los efectos de uso del *software* en un determinado campo de uso; estas métricas evalúan si el producto cumple con las necesidades del usuario. La calidad de uso se ve reflejada en un escenario real. A continuación se nombran algunos atributos que se deben tener en cuenta para la calidad en uso:

- Efectividad.
- Productividad.
- Seguridad física.
- Satisfacción.

### ISO/IEC 14598

La ISO/IEC 14598 es utilizada actualmente como una metodología para evaluar el productor de *software*. La norma establece una serie de etapas e involucra el marco de trabajo donde se manipula el *software* y se evalúa la calidad del producto, definiendo dentro de esto las siguientes características primordiales en los procesos de evaluación:

- Repetitividad.
- Reproducibilidad.
- Imparcialidad.
- Objetividad.

Sin embargo, es necesario hacer énfasis en lo siguiente:

- Análisis de los requisitos de evaluación.
- Evaluación de las especificaciones.

- Evaluación del diseño y definición del plan de evaluación.
- Ejecución del plan de evaluación.
- Evaluación de la conclusión.

La Norma ISO/IEC 14598 define el proceso para evaluar un producto de *software*, el mismo consta de seis partes:

- ISO/IEC 14598-1 (visión general): suministra una visión general de las otras cinco partes a continuación mencionadas y explica la relación entre la evaluación del producto *software* y el modelo de calidad definido en la norma ISO/IEC 9126.
- ISO/IEC 14598-2 (planeamiento y gestión): está compuesto por guías y requisitos para determinar las funciones de soporte como lo es la planificación y gestión de la evaluación del producto.
- ISO/IEC 14598-3 (proceso para desarrolladores): determina los requisitos y guías para la evaluación del producto *software* cuando esta es llevada a cabo en paralelo con el desarrollo por parte del desarrollador.
- ISO/IEC 14598-4 (proceso para adquirientes): suministra las guías y requisitos para llevar a cabo la evaluación del producto de *software* mostrándole a los compradores que desean adquirir o utilizar el producto existente.
- ISO/IEC 14598-5 (proceso para avaladores): determina las guías y requisitos para la evaluación del producto *software* cuando la evaluación es realizada por evaluadores independientes.
- ISO/IEC 14598-6 (documentación de módulos): suministra las guías para la documentación del módulo de evaluación.

La evaluación de los productos de *software* va relacionada con las necesidades que tengan los usuarios finales o proveedor; así, normalmente una evaluación de *software* está compuesta por lo siguiente:

- Definición de perfiles de calidad de referencia de *software*.
- Evaluación de acuerdo con los modelos de calidad predefinidos.
- Certificación de la calidad del *software* de acuerdo a los modelos de calidad y normas.
- Las comparaciones entre productos.
- La reingeniería del *software*.
- Servicio de monitoreo de calidad del producto.

### Norma ISO/IEC 25000 (SQuaRE)

Algunas características que se espera de todo *software* desarrollado son:

- Golpe estratégico; oportunidad de ventaja competitiva.
- Los requisitos del *software* son la base de las medidas de calidad.
- Afecta la organización en general (directivos, trabajadores, clientes).
- Planificación, diseño de objetivos, coordinación, formación, adaptación de toda la organización.

- Si el *software* se ajusta a sus requisitos explícitos, pero falla en obtener los implícitos, la calidad del *software* queda en entredicho.
- Los estándares específicos definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del *software*.

Estas son características que se deben tener en cuenta para desarrollar un *software*, para ello es importante guiarse por un estándar, donde se mantendrán unos parámetros de métricas que el *software* deberá alcanzar al momento de evaluarlo, en esta oportunidad hablaremos de la norma ISO 25000.

ISO 25000:2005 (SQuaRE, del inglés Software Quality Requirements and Evaluation): son una serie de normas que se basan en ISO 9126 y en ISO 14598 (evaluación del *software*); cuenta con los siguientes pasos: recopilar los datos, preparación de los datos y análisis de los datos. Algunos aspectos importantes de la ISO 25000 (Figura 2) son:

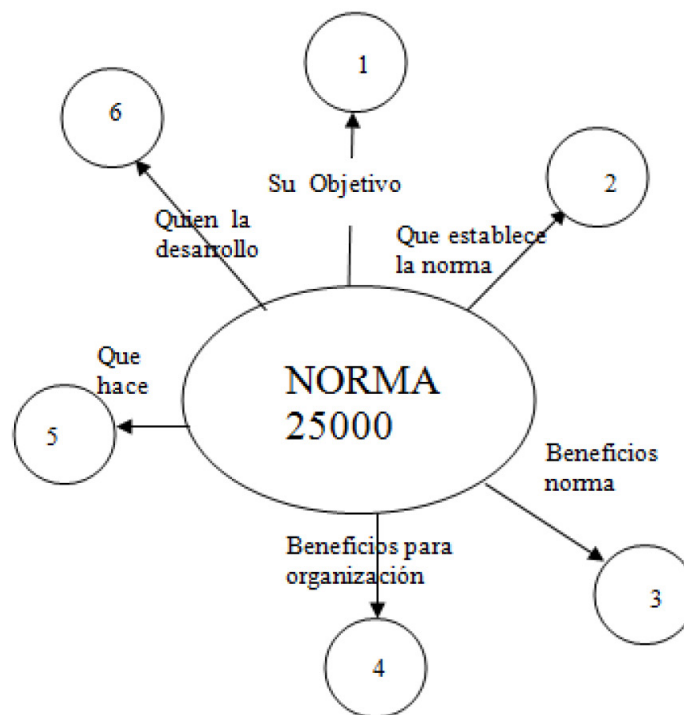


Figura 2. Aspectos norma ISO 25000

Fuente: elaboración propia.

### Su objetivo:

Guiar el desarrollo de los productos de *software* con la especificación y evaluación de requisitos de calidad. Establece criterios para la especificación de requisitos de calidad de productos *software*, sus métricas y su evaluación.

### Qué establece la norma:

- La calidad del *software* está compuesta por características de calidad, las cuales a su vez se componen de subcaracterísticas.
- Las medidas de calidad *software* indican las características y sus subcaracterísticas de calidad del producto.

### Beneficios de la norma:

- Mayor eficiencia en los requerimientos del *software*.
- Mejora la calidad del producto.
- Propone una calidad final a través de las evaluaciones internas.

### Beneficios para la organización:

- Alinea los objetivos del *software* con las necesidades reales que se le demandan.
- Evita ineficiencias y maximiza la rentabilidad y calidad del producto de *software*; por otro lado, certificar el *software* aumenta la satisfacción del cliente y mejora la imagen de la empresa.
- Cumplir los requisitos contractuales y demostrar a los clientes que la calidad del *software* es primordial.
- El proceso de evaluaciones periódicas ayuda a supervisar continuamente el rendimiento y la mejora.

### Qué hace:

Refiere a calidad del producto, proporciona una guía para el uso de las nuevas series de estándares

internacionales, llamados Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE, por sus siglas en inglés).

### Quién la desarrolló:

La norma ISO 25000 ha sido desarrollada por el subcomité SC 7 (ingeniería de *software* y sistemas) del comité técnico conjunto ISO/IEC J. El estándar de calidad ISO 25000 maneja divisiones llamadas modelos N, cada división se encarga de aspectos diferentes en calidad del *software* (Tabla 1).

Las definiciones de cada uno de los modelos N se presentan a continuación:

- Los modelos que forman esta división se definen como los modelos comunes, términos y referencias a los que se refieren en las demás divisiones de SQuaRE. El modelo también posee unos requerimientos y guías para un grupo de evaluación, el cual es responsable de la gestión de las especificaciones de los requerimientos de calidad del producto de *software* y la evaluación de la calidad del mismo.
- Esta segmentación N presenta un modelo detallado de calidad incluyendo características de la calidad de *software* interno, externo y de la calidad de *software* en uso; además, las características de calidad de *software* internas y externas están descompuestas en subcaracterísticas.
- Incluyen un modelo de referencia de medición de la calidad del producto de *software*, definiciones matemáticas de las medidas de calidad.
- Los estándares de este modelo ayudan a especificar los requisitos de calidad. Estos requisitos pueden ser usados en el proceso de especificación de requisitos de calidad para un producto *software* que va a ser desarrollado o como entrada para un proceso de evaluación.
- Estos estándares proporcionan requisitos, recomendaciones y guías para la evaluación de un producto *software*, tanto si la llevan a cabo evaluadores, como clientes o desarrolladores [2].

**Tabla 1.** Divisiones de ISO 25000

ISO/IEC 25000 (SQuaRE)		
Modelos N	Modelos	Nombres
A: Administración de calidad ISO 2500n	ISO 25000 ISO 25001	Guía de SQUARE Planeación y gestión
B: Modelo de calidad 2501n	ISO 25010	Calidad del modelo
C: Medidas de Calidad 2502n	ISO 25020 ISO 25021 ISO 25022	Calidad de las métricas, Modelo de referencia de las métricas Primitivas de medición Medidas de calidad interna Medidas de calidad externa
D: Requerimientos de calidad 2503n	ISO 25023 ISO 25024 ISO 25030	Medidas de calidad de uso Requerimientos de cali- dad
E: Evaluación de calidad 2504n	ISO 25040 ISO 25041 ISO 25042 ISO 25043 ISO 25044	Visión General de evaluación de calidad Módulo de evaluación Proceso para desarrolladores Proceso para adquirientes Proceso de evaluadores

**Fuente:** elaboración propia.

### EEE

El estándar IEEE 1061-1992 (Tabla 2): es un estándar para una metodología de métricas y calidad del *software*, este estándar cubre tres grupos (medidas, procesos y objetivos), suministra una metodología para establecer requerimientos de calidad e identificar, implementar, analizar y validar métricas de calidad de productos y procesos *software*. La metodología es adaptable a todo el *software* en todas las fases de cualquier estructura de ciclo de vida, el objetivo de este estándar es:

- Alcanzar las metas de calidad.
- Establecer requisitos de calidad.
- Establecer criterios.
- Evaluar nivel de calidad.
- Detectar anomalías.
- Supervisar cambios.

• Validar el control del sistema.  
Este estándar se divide por niveles jerárquicos los cuales se describen a continuación:

- El primer nivel de la jerarquía del marco métricas de calidad de *software* comienza con el establecimiento de la calidad las necesidades por la asignación de diferentes atributos de calidad, los cuales se utilizan para describir la calidad del sistema entidad X. Todos los atributos que definen los requisitos de calidad acordados entre el equipo del proyecto.
- En el segundo nivel de la jerarquía son los subfactores de calidad que representan atributos de *software* orientados a indicar la calidad; estos pueden ser obtenidos por la descomposición de cada factor de calidad en el *software* medible atributos. Sus factores

de calidad son tributos independientes de *software* y, por lo tanto, pueden corresponder a más de un factor de calidad. Los subfactores de calidad son atributos concretos de *software* que son más significativas de los factores de calidad al personal técnico, como analistas, diseñadores, programadores y mantenedores.

En el tercer nivel de la jerarquía los subfactores de calidad se descomponen en las métricas utilizadas para medir el sistema de productos y procesos durante el ciclo de vida de desarrollo. La carencia de los requisitos es una falta de calidad, los estándares definen un conjunto de criterios que sirven como guía para la entrega de un *software* [4].

### Norma ISO/IEC 15504 (SPICE)

En 1993 la ISO aprobó un proyecto para el desarrollo de un modelo, base de un futuro estándar internacional, que permitiera evaluar la capacidad y madurez de los procesos de *software* de las organizaciones. Este trabajo recibió el nombre de SPICE (por el inglés *Software Process Improvement and Capability Determination*). En 1998 tras las primeras evaluaciones, el trabajo pasó a la fase de informe técnico ISO/IEC TR (*Technical Report*) 15504.

SPICE “es un estándar internacional que es aplicable a cualquier organización que quiera conocer y mejorar la capacidad de sus procesos”. Además, SPICE “No pretende fijar la manera de realizar los procesos dentro de la organización, sino que valora su capacidad y ayuda a proponer mejoras que aumenten esta capacidad” (European Software Institute ESI, 2011).

**Tabla 2.** Estándares IEEE

Estándar IEEE de calidad del software	Definición
IEEE 1012-1998, Standard for Software Verification and Validation	Software de verificación y validación de procesos determina si los productos de desarrollo de una determinada actividad se ajustan a los requisitos de dicha actividad y si el software satisface sus usos y necesidades de los usuarios previstos.
IEEE 1028-1997, Standard for Software Reviews	La norma proporciona las definiciones y requisitos uniformes para la revisión y auditoría procesos. No establece la necesidad de realizar revisiones o auditorías específicas
IEEE 1061-1998, Standard for a Software Quality Metrics Methodology	Una metodología para establecer los requisitos de calidad y definición, ejecución, análisis y validación de los procesos y de software del producto, se definen métricas de calidad. La metodología abarca todo el ciclo de vida del software. [3]
IEEE 1465-1998, Standard - Adoption of International Standard ISO/IEC 12119: 1994(E)	Se establecen los requisitos de calidad para los paquetes de software y las instrucciones sobre cómo probar un paquete de software en contra de estos requisitos.
IEEE 730-2002, Standard for Software Quality Assurance Plans	Este estándar ayuda a determinar el contenido y la elaboración de los planes de aseguramiento de calidad de software y proporciona un estándar contra el cual estos planes pueden ser preparados y evaluados. Se dirige hacia el desarrollo y mantenimiento de software.

**Fuente:** elaboración propia.



## Componentes

SPICE inicialmente fue estructurado en nueve apartados (Figura 3). El contenido de estos documentos permite implementar paso a paso el modelo con su correspondiente evaluación [5].

Sin embargo, con el transcurso del tiempo, luego de diferentes debates y votaciones con el objetivo de reducir el tamaño del estándar y consolidarlo, se decide dividir en cinco partes tal como muestra la Tabla 3.

## Dimensiones

Para la evaluación de la capacidad de los procesos, SPICE desarrolla un modelo de dos dimensiones. Dimensión del proceso y dimensión de la capacidad, a esto se denomina arquitectura del modelo de referencia (Figura 4). En el eje X

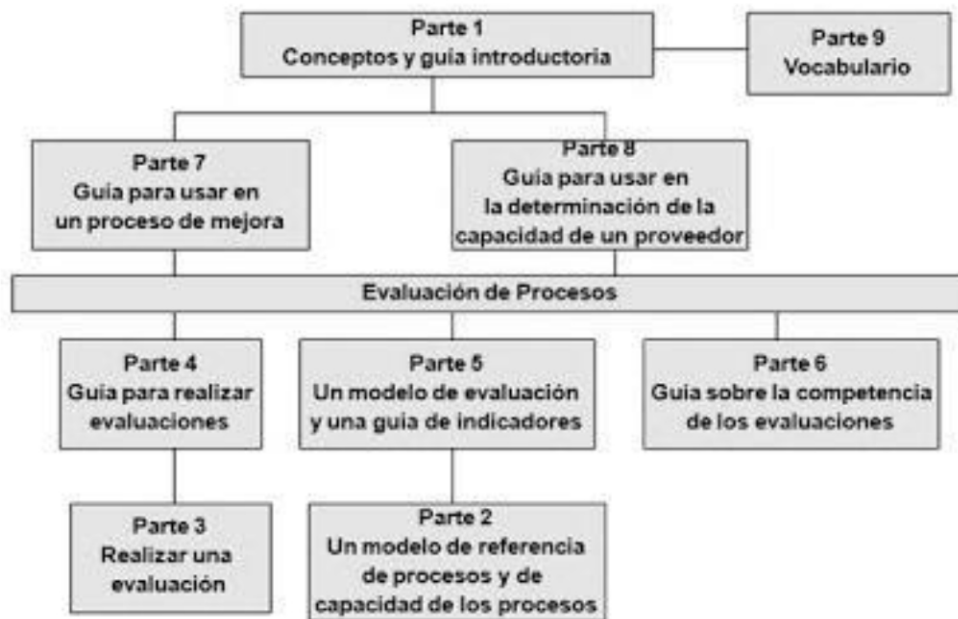
(dimensión del proceso) se ubica el proceso a evaluar junto con las actividades que se ejecutan en el proceso, en el eje Y (dimensión de la capacidad) el nivel de capacidad o madurez de los procesos junto con los atributos de cada nivel [6].

### Dimensión procesos

Los procesos de desarrollo *software* evaluados con la ISO 15504 se encuentran en la norma ISO/IEC 12207; esta norma contiene tres categorías de procesos (principales, soporte, organización), los cuales contienen una serie de procesos que soportan el ciclo completo de un proyecto *software*, desde la definición de un proyecto hasta la entrega y cierre del mismo.

Procesos principales:

Principales promotores que ayudan a mejorar las funciones dentro del ciclo de vida de un proyecto de *software*.



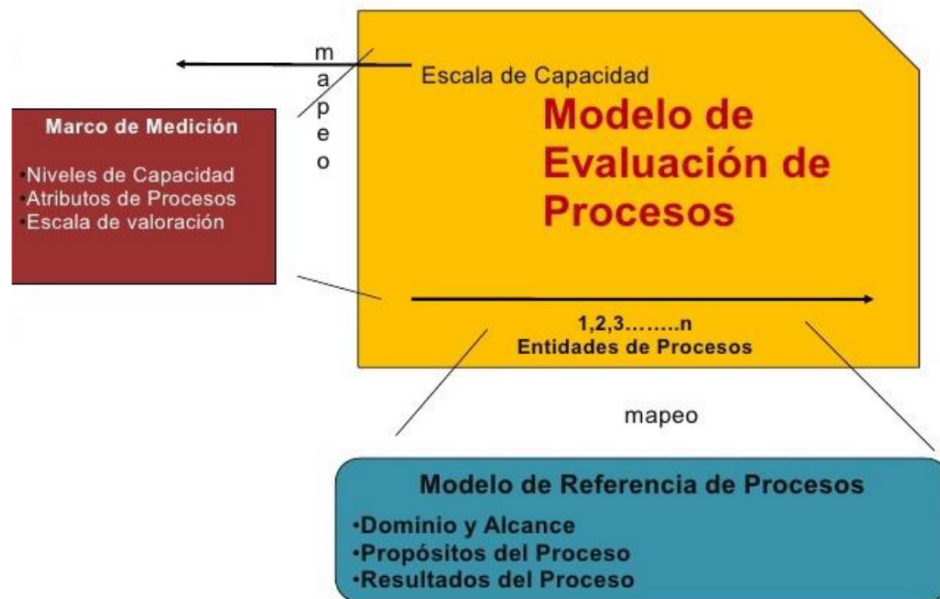
**Figura 3.** Estructura inicial SPICE ISO/IEC 15504

Fuente: elaboración propia.

**Tabla 3.** Estructura actual SPICE ISO/IEC 15504

Componentes		Estándar Actual (2015)
2015	2003	
1	1	ISO/IEC 33001:2015 Concepts and terminology
	9	
2	2	ISO/IEC 33002:2015 Requirements for performing process assessment
	3	
3	4	ISO/IEC 15504-3:2004 Guidance on performing an assessment
	6	
4	7	ISO/IEC 15504-4:2004 Guidance on use for process improvement and process capability determination
	8	
5	5	ISO/IEC 15504-5:2012 An exemplar software life cycle process assessment model

**Fuente:** elaboración propia.



**Figura 4.** Arquitectura de modelo de referencia

**Fuente:** elaboración propia

- Adquisición, que define las actividades del contratante. La empresa que adquiere un producto o servicio.
- Suministro, que define las actividades del proveedor. La empresa que ofrece el producto o servicio.
- Desarrollo, que define las actividades de los desarrolladores. La organización que define y desarrolla el *software*.
- Operación, que define las actividades de los operadores. La organización que presta el servicio de operar los sistemas de información en el ambiente de los usuarios.
- Mantenimiento, que define las actividades de la organización que presta el servicio de mantener el *software*, conservándolo actualizado y utilizable.

#### Procesos de soporte:

Estas son las actividades que soportan y coordinan el desarrollo y el ciclo de vida de las actividades primarias. Un proceso de soporte apoya otros procesos que llevan a cabo una función especializada.

- Documentación, que define las actividades requeridas para consignar toda la información producida por los procesos del ciclo de vida.
- Gestión de la configuración, que incluye actividades de identificación, control, estadística y evaluación de las configuraciones, así como las de la administración de versiones.
- Aseguramiento de calidad, que define las actividades para asegurar de forma objetiva que los productos de *software* cumplen los requerimientos especificados y si están alineados a los planes establecidos. Las revisiones conjuntas, auditorías, verificación y validación.
- Verificación, que define las actividades (para el cliente o proveedor) para verificar los productos y servicios de *software*.
- Validación, que define las actividades (para el cliente o proveedor) para validar los productos de *software* del proyecto de *software*.
- Revisión conjunta. Proceso para revisiones tanto técnicas como administrativas, para que de esta

forma una de las partes evalúe el estado de los productos producidos y las actividades realizadas por la otra parte.

- Auditoría, que define las actividades para verificar/validar el cumplimiento de los requerimientos, planes y contratos. Este proceso puede ser usado por cualquiera de las dos partes, donde una parte (auditora) audita los productos o actividades de la otra parte (auditada).
- Resolución de problemas, que define un proceso para analizar y resolver los problemas (incluyendo inconformidades), sin importar su naturaleza u origen, que sean descubiertos durante la ejecución de alguno de los procesos.

#### Procesos de la organización:

Se refiere a la gestión y al apoyo en general para todo el ambiente de desarrollo. Se conforma de cuatro procesos.

- Gestión. El objetivo de este proceso es la de proporcionar administración a todos los demás procesos del proyecto, incluyendo administración del producto y administración del proyecto.
- Infraestructura. Establece y mantiene el *hardware*, *software*, herramientas, técnicas y estándares; necesaria para la ejecución de los otros procesos. Cualquier estándar que sea utilizado en un proyecto o por una organización se vuelve parte de la infraestructura.
- Mejora. Define las actividades que una organización realiza para establecer, medir, controlar y mejorar los procesos de su ciclo de vida.
- Recursos humanos. Define las actividades para proveer e personal capacitado adecuado.

#### Dimensión de capacidad

Niveles de capacidad o madurez de los procesos (Figura 5):

Desde la dimensión de capacidad del modelo, se define una escala de seis niveles para determinar la capacidad o nivel de madurez de cualquier proceso.



**Figura 5.** Niveles de capacidad o madurez

Fuente: [7]

- Nivel 0: incompleto. La organización no tiene una implementación efectiva de los procesos.
- Nivel 1: realizado (se realizan). Se alcanza el propósito de los procesos en términos generales. Los procesos se realizan cuando es necesario, pero no se hace de una forma planificada ni se realiza ningún seguimiento.
- Nivel 2: gestionado (planificado). Se obtienen los productos del proceso, pero esta vez, de acuerdo con una planificación y realizándose un seguimiento. Estos productos se ajustan a unos estándares y a unas especificaciones prefijadas. También se tiene definidos plazos y recursos.
- Nivel 3: establecido (definidos desplegados). Los procesos se realizan y se gestionan utilizando procedimientos definidos. Cada implementación de un proceso se hace utilizando procedimientos creados según un estándar y documentados.
- Nivel 4: predecible (medidos analizados). Se recogen medidas detalladas del nivel de realización de los procesos y se analizan; esto permite mantener los procesos dentro de unos límites predefinidos, así como disponer de una mejor posición para poder cuantificar la capacidad del proceso y predecir su comportamiento.
- Nivel 5: en optimización (mejora continua innovación, experiencia). La realización de los procesos se optimizan de forma continuada, de cara a su contribución a alcanzar los objetivos del negocio de la organización. Se establecen objetivos cuantitativos de eficacia y eficiencia en la realización de los procesos, basados en los objetivos de negocio de la organización. Se lleva a cabo un monitoreo continuo de los procesos y se analizan los datos obtenidos, esto permite que los procesos estándar definidos dentro de la organización cambien dinámicamente, para adaptarse de forma efectiva a los actuales y futuros objetivos de la empresa.

Atributos de los procesos [8]:

Un atributo del proceso representa una característica medible de cualquier proceso. Los atributos de capacidad del proceso son los elementos básicos del esquema de evaluación. Estos atributos se evalúan entre un rango de cuatro puntos que se presentan a continuación.

N (*Not*): no conseguido. No se evidencia que se consiga el atributo definido.

P (*Partially*): conseguido parcialmente. Se ha conseguido algo del atributo definido

L (*Largely*): bastante conseguido. Se ha conseguido significativamente el atributo definido.

F (*Fully*): conseguido completamente. Se ha conseguido completamente el atributo definido.

## Ejemplo evaluación del proceso

El siguiente es un ejemplo de la evaluación del proceso de “desarrollo” utilizando una arquitectura de modelo de referencia. Este proceso contiene siete actividades a realizar las cuales se describen a continuación.

- DES.1 Requerimientos y diseño del sistema.
- DES.2 Requerimientos de *software*.
- DES.3 Diseño del *software*.
- DES.4 Implementación del diseño.
- DES.5 Integración y pruebas de *software*.
- DES.6 Integración y pruebas del sistema.
- DES.7 Mantenimiento del *software* y el sistema.

En el eje X del modelo de referencia se ubican los procesos con sus actividades relacionadas (DES.1

a DES.7). En el eje Y del modelo de referencia se ubican los cinco niveles de capacidad (1 a 5) junto con los atributos asociados a cada nivel (1.1 a 5.2), como se muestra en la Tabla 4.

En la intersección de estos dos ejes se ubica el atributo del proceso (F, L, P, N), el cual permite evaluar la madurez de cada una de las actividades del proceso.

## CMMI (*Capability maturity model integration*)

Es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de *software*; la versión actual de CMMI es la versión 1.3, que fue liberada el 1 de noviembre de 2010.

Para comprender que es el CMMI es necesario revisar el concepto de CMM. Las siglas CMM vienen del inglés *Capability Maturity Model*, que en español significa modelo de capacidad de madurez del *software*, se trata de un modelo para la evaluación y mejora de los procesos de una organización. Fue desarrollado inicialmente para los procesos relacionados al *software* por la Universidad Carnegie Mellon para el SEI (del inglés Software Engineering Institute).

**Tabla 4.** Modelo de referencia (evaluación del proceso de desarrollo)

5	5.2	N	N	N	N	N	N	P
	5.1	N	N	N	N	N	N	P
4	4.2	N	N	L	N	N	P	L
	4.1	N	P	P	N	P	P	L
3	3.2	P	L	P	N	L	L	F
	3.1	L	F	L	P	F	L	F
2	2.2	L	F	F	L	F	F	F
	2.1	F	F	F	L	F	F	F
1	1.1	F	F	F	F	F	F	F
		DES.1	DES.2	DES.3	DES.4	DES.5	DES.6	DES.7

**Fuente:** elaboración propia.

El modelo CMM indica que deben existir áreas o procesos clave dentro de la organización que deberán realizar alguna función específica, estas áreas son denominadas áreas clave de proceso o KPA (del inglés *Key Process Area*).

El modelo define un conjunto de buenas prácticas para cada una de las áreas, se puede identificar el nivel de madurez de una organización dependiendo de que tanto se ajusten estas buenas prácticas a cada una de las áreas.

La diferencia entre el CMM y el modelo CMMI es básicamente que el primero se enfoca en las organizaciones o áreas de tecnologías de información, mientras que el modelo CMMI, como su nombre lo indica, es un modelo integrado y mejorado que se puede aplicar a un amplio número de organizaciones de diferentes sectores.

El modelo CMM define cinco niveles de madurez de los que se habla a continuación.

**Inicial:** no es necesario hacer ningún esfuerzo para llegar al nivel inicial, en este nivel las organizaciones no disponen de un ambiente adecuado para el desarrollo de *software*, se denota la falta de planificación. Los procesos varían según el recurso humano, el éxito de los proyectos en un alto grado tienen fundamento en el esfuerzo personal, es común que se produzcan fracasos, retrasos y sobrecostos. El resultado de los proyectos es impredecible y no hay un control adecuado.

Como lo muestra la Figura 6, el resultado de los proyectos es impredecible y no hay un control adecuado.

No hay una definición formal de funciones, así como hay carencia de control en los procesos;

no se tiene una visión clara del proyecto, solo se puede tener una visión clara de las cosas cuando se empieza el proyecto y cuando se logra acabar, pero no es posible conocer de manera adecuada el estado del proyecto en sus procesos intermedios. Debido a lo anterior, en algunas ocasiones aparecen individuos que se responsabilizan del proyecto y lo logran sacar adelante, aunque en general este tipo de proyectos sufrirá demoras y probablemente no se culminará.

**Repetible:** en este nivel se ubican las organizaciones en las que hay planificación y seguimiento de proyectos, además de estar implementada la gestión de los mismos; sin embargo, existe alta probabilidad de no cumplir con los objetivos del proyecto.

Se logra en la Figura 7 que ya hay diferencias entre el nivel inicial y el nivel repetible, en el segundo nivel se puede observar que se definen puntos de control en cada etapa principal del proyecto, esto permite tener un mejor control sobre el proyecto. Hay que tener en cuenta que cada etapa es aún una caja negra, no se sabe con precisión cómo se está implementando el proyecto dentro de cada etapa.

Los procesos que hay que implantar para alcanzar este nivel son:

- Gestión de requisitos.
- Planificación de proyectos.
- Seguimiento y control de proyectos.
- Gestión de proveedores.
- Aseguramiento de la calidad.
- Gestión de la configuración.



**Figura 6.** Nivel de madurez 1 CMMI

Fuente: [9]

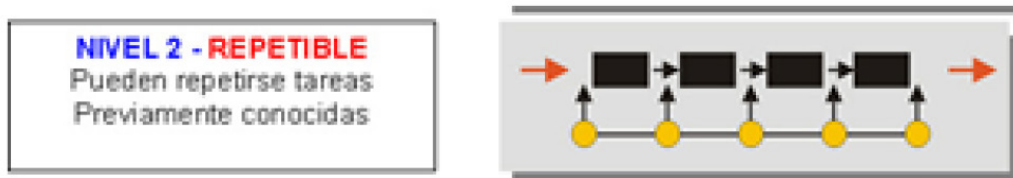


Figura 7. Nivel de madurez 2 CMMI

Fuente: [9]

Definido: en este nivel están definidos y establecidos un conjunto de procesos estándar con objetivos bien definidos. Existe un sistema de gestión de los proyectos. Una diferencia crítica entre los niveles 2 y 3 de madurez es el alcance de los estándares, descripciones de los procesos y procedimientos. En el nivel 2 pueden variar entre las distintas instancias de los procesos, entre diferentes proyectos; a nivel 3 los procesos son globales dentro de la organización (Figura 8).

Los procesos comunes para desarrollo y mantenimiento del *software* están documentados y almacenados en medios accesibles a los equipos de desarrollo. El recurso humano se ha capacitado para comprender los procesos; cada uno de los procesos es transparente para todos.

Los procesos que hay que implantar para alcanzar este nivel son:

- Desarrollo de requisitos.
- Solución técnica.
- Integración del producto.
- Verificación.
- Validación.
- Desarrollo y mejora de los procesos de la organización.
- Definición de los procesos de la organización.
- Planificación de la formación.
- Gestión de riesgos.
- Análisis y resolución de toma de decisiones.

Gestionado: en este nivel las organizaciones tienen un conjunto de métricas significativas de calidad y productividad, estas se utilizan de modo sistemático para la toma de decisiones y la gestión de riesgos. El *software* resultante es de alta calidad (Figura 9).

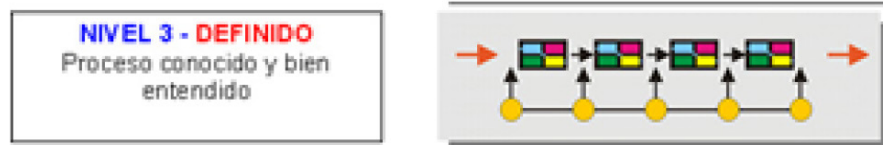


Figura 8. Nivel de madurez 3 CMMI

Fuente: [9]

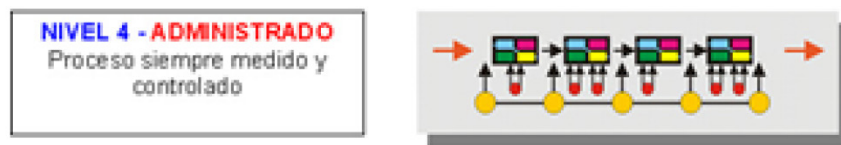


Figura 9. Nivel de madurez 4 CMMI

Fuente: [9]

La principal diferencia con el nivel “definido” es la medición y control de los procesos por medio de métricas que se han establecido con criterios cuantitativos formalmente definidos. A medida que transcurre el proyecto estos controles facilitan mejor información sobre la calidad y estado del proyecto, permitiendo compararlo con otros proyectos similares e identificar cualquier desviación oportunamente para poder corregirlo.

Los procesos que hay que implantar para alcanzar este nivel son:

- Gestión cuantitativa de proyectos.
- Mejora de los procesos de la organización.

Optimizado: la organización está enfocada en la mejora continua de los procesos. Se utilizan las métricas para tal objetivo y se gestiona el proceso de innovación (Figura 10).

En este nivel cada proceso se analiza y controla permanentemente con el objetivo de ser mejorado en todo momento, los controles permiten un mejoramiento continuo y se tienen implementadas todas las áreas clave de proceso recomendadas por el modelo.

Los procesos a implantar para alcanzar este nivel son:

- Innovación organizacional.
- Análisis y resolución de las causas.

Como se muestra en el texto a partir del segundo nivel del modelo CMM se debe contar con áreas específicas que permitirán tener un mayor control del proyecto de *software*.

## CONCLUSIONES

Para generar un producto de *software* con calidad, es necesario definir un estándar como marco de referencia, con el fin de evaluar el estado del producto o proceso, realizar los ajustes necesarios, para luego obtener los resultados esperados partiendo de las necesidades del usuario.

Hoy en día existen diferentes normativas de calidad de *software*, algunas enfocadas a calidad en los procesos como ISO 15504 (SPICE) y CMMI (*Capability Maturity Model Integration*), y otros enfocados a la calidad del producto como ISO 9126 (*software engineering-product quality*), ISO 14598 (*software product evaluation*), IEEE 1061 e ISO 25000 (*product quality and evaluation*). La evaluación de estos estándares en las organizaciones permite dar un punto de referencia del estado actual, teniendo la posibilidad de verificar las fortalezas y debilidades para luego realizar un plan de acción que permita la generación de productos de *software* calidad.

El estándar ISO/IEC 9126 permite definir un modelo de calidad, para cualquier organización, partiendo de las seis características que indican la calidad en el ciclo de vida del *software*. El modelo de calidad que se define dará como resultado el grado de calidad de cada producto de *software* que se tenga o adquiera.



Figura 10. Nivel de madurez 5 CMMI

Fuente: [9]



La segmentación de la norma ISO/IEC 25000, es la evolución de la norma para darle importancia a cada fase de calidad de *software*, teniendo en cuenta modelos, requisitos, métricas, gestión y evaluación de la calidad del *software*, esto conlleva a que las herramientas desarrolladas bajo la norma sean más potentes y eficientes en la parte productiva.

El estándar SPICE (ISO 15504) presenta un modelo que permite evaluar la madurez de cada uno de los procesos del ciclo de vida del *software* indicados en la norma ISO/IEC 12207. De esta manera una organización que desarrolla e implementa *software* puede ser auditada tomando como base esta norma para certificar la madurez de los procesos de *software* y por lo tanto medir la calidad de la ejecución del trabajo que se está realizando.

## REFERENCIAS

- [1] Mginformatica.com.ar. (s.f.). *Modelo de calidad*. Recuperado de: <http://www.mginformatica.com.ar/modelo-de-calidad.htm>
- [2] ISO 25000. (s.f.). *La familia de normas ISO/IEC 25000*. Recuperado de: <http://iso25000.com/index.php/normas-iso-25000>
- [3] IEEE-SA. (s.f.). *1061-1998 - IEEE Standard for a Software Quality Metrics Methodology*. Recuperado de: <https://standards.ieee.org/findstds/standard/1061-1998.html>
- [4] Acceso público. (s.f.). *Estándares IEEE 1061-1992*. Recuperado de: <https://prezi.com/prieimhzzmz/estandares-ieee-1061-1992/>
- [5] Ingeniería del software. (S.f.). *Modelo de calidad SPICE*. Recuperado de: <http://software-ufps.blogspot.com/search/label/CLASE%2014%3A%20Modelo%20de%20Calidad%20SPICE>
- [6] Kobayashi, A. (s.f.). *ISO/IEC 15504 - Introducción a la norma de evaluación de procesos de software*. Recuperado de: <http://es.slideshare.net/quasarprocess/isoiec-15504-introduccion-a-la-norma-de-evaluacion-de-procesos-de-software>
- [7] Grajales, A. (2014). *MoProSoft con ISO 15504, ISO 9001:2000*. Recuperado de: <http://abnergrajales.blogspot.com/2014/02/moprosoft-con-iso-15504-iso-90012000.html>
- [8] Bejarano, J. (2013). *Modelo SPICE*. Recuperado de: <http://es.slideshare.net/jsx2060/modelo-spice-34978417>
- [9] Calidadysoftware.com (s.f.). *Introducción al CMMI-CMM*. Recuperado de: [http://www.calidadysoftware.com/otros/introduccion\\_cmml.php](http://www.calidadysoftware.com/otros/introduccion_cmml.php)