



DESARROLLO DE SOFTWARE

Subtítulo Consulta sobre calidad

► **ASIGNATURA** | Calidad del software

► **PROFESOR** | Solís Santiago

► **APELLIDOS Y NOMBRE** | David Alba

► **EMAIL** | _____ **TELÉFONO** | 0979908514

► **TRABAJO GRUPAL N°** | _____ **FECHA DE ENTREGA** | _____

Fundamentos de calidad del software

Calidad de software

ISO en la norma 8402:1994, la define como la “Totalidad de propiedades y características de un producto, proceso o servicio que le confiere su aptitud para satisfacer unas necesidades expresadas o implícitas.”

En la ISO 9000:2000 se la define como “Grado en el que un conjunto de características inherentes cumple con los requisitos”. En esta última se hace especial énfasis en los requerimientos del cliente.

Glass (1998), argumenta que es conveniente generar una relación más intuitiva. Satisfacción del usuario = producto manejable + buena calidad + entrega dentro del presupuesto y tiempo.

Control de calidad.

Esto involucra la serie de inspecciones, revisiones y pruebas empleadas a lo largo del proceso del software para garantizar que cada producto del trabajo satisfaga los requisitos que se le han asignado. El control de calidad incluye un bucle de retroalimentación con el proceso que creó el producto del trabajo. La combinación de mediciones retroalimentación permite afinar el proceso cuando los productos de trabajo creados fracasan en cuanto a satisfacer sus especificaciones. Un concepto clave del control de calidad es que todos los productos de trabajo tienen especificaciones definidas mensurables con las cuales se puede comparar la salida de cada proceso. Dicho bucle es esencial para minimizar los efectos producidos.

Garantía de la calidad

La garantía de la calidad consiste en un conjunto de funciones de auditoría e información que evalúan la efectividad y qué tan completa son las actividades de control de calidad. La meta del aseguramiento de la calidad es presentarle al gestor los datos necesarios para que esté informado acerca de la calidad del producto y por consiguiente que comprenda y confíe en que la calidad del producto está satisfaciendo las metas y objetivos. Si se identifican problemas en el proceso de aseguramiento de calidad, es responsabilidad del gestor aportarlos y aplicar los recursos necesarios para resolver los conflictos de calidad.

Una organización de software obtendrá el registro ISO 9001:2000 si establece políticas y procedimientos para abordar cada uno de los requisitos anotados además, ser capaz de demostrar que se siguen dichas políticas y procedimientos. Entre las políticas y procedimientos que se deben de demostrar en una auditoría están las siguientes:

1. Establecer los elementos de un sistema de gestión de calidad

- Desarrollar, implementar y mejorar el sistema.
- Definir una política enfatice la importancia del sistema.

2. Documentar el sistema de calidad

- Describir el proceso.
- Producir un manual operativo.
- Desarrollar métodos para controlar los documentos
- Establecer métodos para la conservación de registros.

3. Soporte del control y la garantía de calidad

- Promover la importancia de la calidad entre todos los participantes.

- Enfocarse en la satisfacción del cliente.
 - Definir un plan de calidad que aborde objetivos, responsabilidades de autoridad.
 - Definir mecanismos de comunicación entre los participantes.
4. **Establecer mecanismos de revisión para el sistema de gestión de calidad**
- Identificar métodos de revisión y mecanismos de retroalimentación.
 - Definir procedimientos de seguimiento.
 - Identificar recursos de calidad que incluyan personal, entrenamiento, elementos de infraestructura.
5. **Establecer mecanismos de control**
- Para planeación.
 - Para requisitos del cliente.
 - Para actividades técnicas, por ejemplo análisis diseño y pruebas.
 - Para supervisión y gestión del proyecto.
6. **Definir métodos para corrección**
- Valorar los datos y métricas de calidad.
 - Definir enfoques para procesos continuos y de mejora de la calidad.

Buenas prácticas de calidad software a nivel de diseño y codificación

A nivel de código o diseño, no olvides que:

Duplicar, o copy pegar, código no es una buena idea.

- Un case o switch con muchas clausulas, o muchos ifs anidados, tampoco es una buena idea.
- Que las estructuras de datos deben estar ocultas en un sistema software (2/2) y (1/2).
- Y que hay incluso listas de malos olores que te pueden dar pistas de que problemas de calidad software te puedes encontrar.

CALIDAD DE SOFTWARE

Definición

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. La calidad del software es medible y varía de un sistema o programa a otro. Un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad mientras que un software para ser explotado durante un largo necesita ser confiable, mantenible y flexible para disminuir los costos. La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas deriva dos de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.

Conclusión

La calidad es primordial en la elaboración, diseño, comercialización, etc. De cualquier producto, bien o servicio. Para hacer énfasis en calidad referente a software se puede decir que los requisitos son la base de la calidad, esto se puede interpretar de manera muy lógica. Cuando se hace un software el cliente es quien califica la estructura

y diseño del programa, también es el cliente quien hace referencia de los requisitos que necesita su software. Por otro lado, existen normas y protocolos de calidad regidas por diferentes instituciones, dejando de lado la opinión del cliente, para que un software puede ser catalogado como producto de calidad es necesario cumplir con las normas regidas y el gusto del usuario. Un producto como el software será de calidad cuando siga la metodología propuestas por organizaciones dedicadas a la calidad de software y el programa sea útil en todos los aspectos de requerimientos para el cliente

La medición de los costos de la mala calidad afectan a:

Los costos internos y externos que resultan de no cumplir con los requisitos.

Los costos internos son los costos causados por los productos o servicios que no cumplan con los requisitos o necesidades del cliente o usuario y se encuentran antes de la entrega de productos o servicios a clientes externos. Que de otro modo hubieran conducido a que el cliente no estuviera satisfecho. Las deficiencias son causadas tanto por errores en los productos o las ineficiencias en los procesos.

Ejemplo de costos internos por fallas son:

- Re trabajos
- Los retrasos
- Re-diseño
- Faltantes
- Análisis de fallas
- Volver hacer pruebas validaciones inspecciones
- Retrocesos de versiones de desarrollo
- Tiempo de inactividad
- La falta de flexibilidad y adaptabilidad

Los costos externos por fallas son los costos causados por deficiencias encontradas después de la entrega de productos y servicios a los clientes externos que conducen a la insatisfacción del cliente

Ejemplo de costos externos incluyen costos de:

- Quejas
- Reparación de bienes y rehacer servicios
- Garantías
- Mala voluntad de los clientes
- Las pérdidas debidas a la reducción de ventas
- Los costos ambientales por impacto

La calidad, un concepto en permanente evolución

La norma ISO 9001 es el punto de encuentro de numerosos esfuerzos por establecer un marco común para la gestión de calidad. Vigente desde los años 80, ha incorporado mejoras y avances en función de las necesidades y, sobre todo, de la propia evolución del concepto de calidad empresarial.

Autores de la calidad

Al hablar de Calidad, tenemos que hacer referencia a grandes personajes que con sus ideas han construido las bases de los sistemas de gestión de calidad. Algunas de estas ideas han trascendido en el tiempo, de tal forma que sus creadores permanecen hoy como grandes pensadores y transformadores de la calidad en las organizaciones a nivel mundial.



Figura1. Factores de liderazgo.

1. Edward Deming



Figura2.

(USA, 14 de octubre de 1900 – 20 de diciembre de 1993)

Es considerado por muchos como el **padre de la calidad**. Compartió sus famosos 14 puntos y 7 pecados mortales. Su definición de calidad:

“la producción de bienes y servicios competitivos requiere de un sistema basado en el control estadístico de procesos, esto genera la calidad. El sistema debe enfocarse en prevenir el error y no en detectarlo o corregirlo. La calidad debe estar definida en términos de satisfacción al cliente”

REPORT THIS AD

El círculo de Deming es una metodología recomendada para la realización de cualquier actividad, permite lograr los resultados esperados. Tiene cuatro fases:

1. Planear
2. Hacer
3. Verificar
4. Actuar

2. Joseph M. Juran

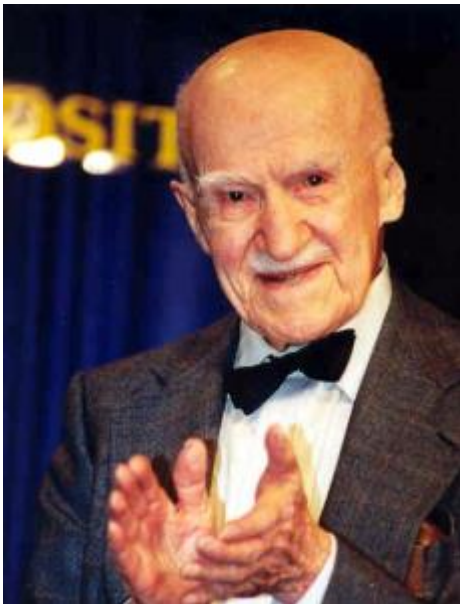


Figura 3

(Braila, Rumania, 24 de diciembre de 1904 – New York, 28 de febrero de 2008)

Es uno de los pioneros en la promoción de los **enfoques de calidad**. Su **definición de calidad**:

“Resalta la adecuación al uso, lo cual implica que los productos y servicios cuenten con las características que el usuario ha definido como útiles, o bien le generan un beneficio. La adecuación siempre será determinada por el comprador y nunca por el vendedor. “

Distingue dos tipos de calidad:

1. De diseño: el producto satisfaga las necesidades del usuario
2. De conformidad: el grado en que los productos se apegan a las calidades definidas.

Además resalta las siguientes **ventajas competitivas**:

1. Disponibilidad: tiene que ver con su desempeño y vida útil.
2. Servicio técnico

Propuso la “**Trilogía de Jurán**” la cual consta de:

1. Planificación de la calidad
2. Control de la calidad
3. Mejora de la calidad

3. Kaouru Ishikawa



(Japón, 13 de julio de 1915 – 16 de abril de 1989)

REPORT THIS AD

Resaltó que **control de calidad** debe aplicarse no sólo en las actividades de producción, sino también en todas las actividades de la empresa, tales como ventas, abastecimiento, y administración en general.

Predicaba que **la calidad** debe ser llevada más allá del trabajo, a la vida diaria.

El control de la calidad tiene los siguientes objetivos:

1. Mejorar la productividad
2. Mejorar la calidad de los productos
3. Aplicar la calidad a todas las actividades de la empresa
4. Dividir los beneficios obtenidos entre consumidores, empleados y accionistas
5. Mejorar el nivel de vida de las personas

4. Armand Feigenbaum

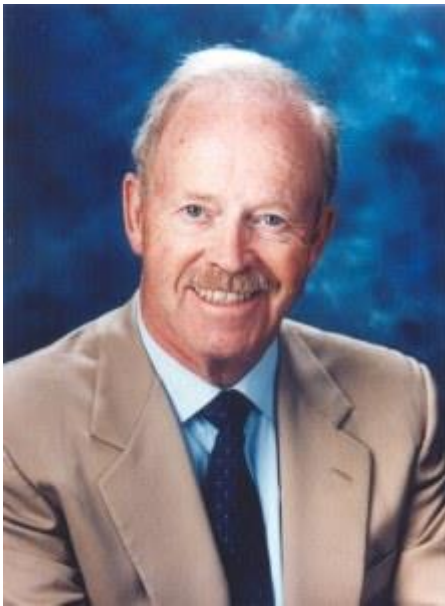


(Nueva York, Nueva York, 6 de abril de 1922 – Pittsfield, Massachusetts, 13 de noviembre de 2014)

En 1956, en su libro **Total Quality Control**, propone por primera vez el concepto de “**control total de calidad**”. Su definición de calidad:

“La calidad es trabajo de todos y de cada uno de los que intervienen en cada etapa del proceso.”

5. Philip Crosby



(* Wheeling, 1926 – Winter Park, 2001)

Es un importante consultor de calidad en USA. Hizo famosa la frase **“calidad es hacer las cosas bien desde la primera vez”**. Su filosofía está orientada hacia: calidad no cuesta. No es un regalo, pero es gratuita. Lo que cuesta dinero son las cosas que no tienen calidad. Su **definición de calidad**:

1. Calidad es cumplir los requisitos
2. El sistema para asegurar calidad es la prevención
3. El estándar de desempeño es cero defectos
4. Lo que cuesta es el incumplimiento de los requisitos

Introducción

El software es una de las herramientas de mayor utilidad en la optimización de procesos en las organizaciones, con el propósito de contar y ofrecer optimización, eficiencia y satisfacción de necesidades, razón por la cual el software debe contar con criterios que garanticen su calidad. De acuerdo con esta necesidad, diferentes entidades o investigadores han propuesto estrategias modelos, metodologías, guías, incluso normas y estándares de calidad que brindan apoyo al desarrollo y/o uso de un producto software y permiten evaluar si efectivamente tiene un nivel de calidad durante su ciclo de vida, y de esta manera fomentar un ambiente de calidad, con base en la adecuada administración de la información.

En este documento se contextualiza inicialmente en cuanto a términos propios de calidad de software, posterior a esto se realiza una clasificación de los modelos de acuerdo con el enfoque presentado (proceso, producto y uso) y al tiempo de aparición; esto con el fin de dar a conocer aquellos modelos que se consideran pioneros o base del desarrollo de otros recientes, de igual manera se realiza una descripción de las características más

relevantes de algunos modelos, su estructura y objetivo, finalmente se presentan casos de aplicación de algunos modelos en el sector empresarial.

1. Contextualización de calidad de software

Es importante conocer los conceptos y características acerca de lo que es la calidad de software, y en cuanto a los modelos de calidad de software, su estructura y enfoque.

Calidad de software

El término calidad de software se refiere al grado de desempeño de las principales características con las que debe cumplir un sistema computacional durante su ciclo de vida, dichas características de cierta manera garantizan que el cliente cuente con un sistema confiable, lo cual aumenta su satisfacción frente a la funcionalidad y eficiencia del sistema construido.

El concepto de calidad de software, según Pressman (2010) se asocia a la "concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo plenamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente", con base en los requisitos funcionales y no funcionales identificados en la etapa de análisis del sistema, insumo principal para implementar dichos requisitos con los atributos mínimos de calidad, fomentando la aplicación de procesos estandarizados y criterios necesarios en cada una de sus etapas, así se fomenta que el avance en el ciclo de vida del software minimice el riesgo de fracaso del proyecto. Por su parte, el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, 1990) define calidad de software como "el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario", denotando que el

énfasis radica en los requisitos específicos del sistema y en la búsqueda de la satisfacción del cliente.

Para garantizar la calidad de software es importante implementar algún modelo o estándar de calidad que permita la gestión de atributos en el proceso de construcción de software, teniendo en cuenta que la concordancia de los requisitos y su construcción son la base de las medidas de calidad establecidas.

2. Modelos de calidad de software

Aunque modelo y metodología distan en su definición, se rescata la cita dada por Moszkowitz (2010) en la que presenta una metodología que permite a cualquier organización realizar una autoevaluación o autodiagnóstico, por medio de una revisión sistemática de sus estrategias y prácticas de gestión.

En el caso de la calidad de software el modelo debe ir enfocado a hacer seguimiento y evaluación a cada etapa de construcción del producto software. Por otro lado se menciona (Scalone, 2006) que

los modelos de calidad son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos clave y permiten medir los avances en calidad.

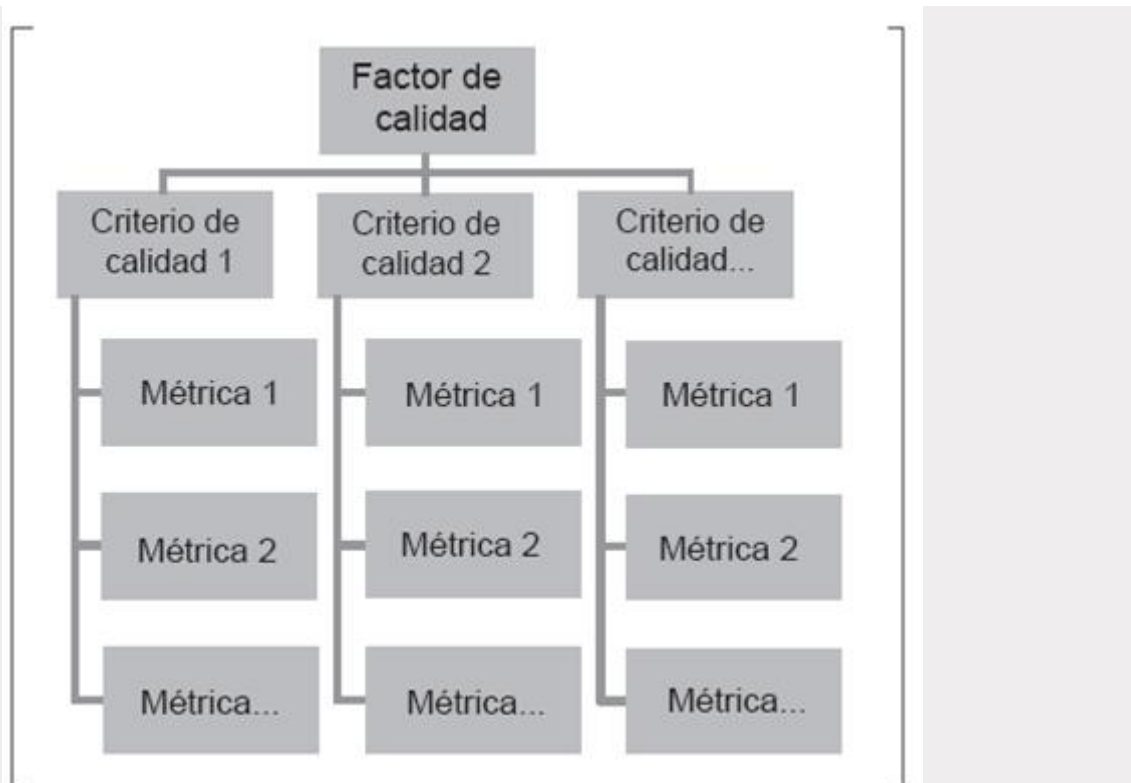
Esta definición, enfocada a la calidad del software, identifica que la organización debe contar con un proceso que como soporte al mismo lleve una documentación, y se valga de distintas prácticas definidas en el modelo, dando apoyo a la organización para tener

una mejora continua y ser más competentes, para así poder medir la calidad y brindar productor o servicios de alto nivel.

En el ámbito de la construcción de software, el modelo de calidad debe permitir evaluar el sistema, bien sea cualitativa o cuantitativamente, y de acuerdo con esta evaluación la organización podrá proponer e implementar estrategias que permitan la mejora del proceso dentro de las etapas de análisis, diseño, desarrollo y pruebas del software.

3. Estructura y enfoque de los modelos de calidad de software

Los modelos de calidad de software generalmente están estructurados como se muestra en la Figura 1 (Scalone, 2006) y (Bautista, 2012), donde se pueden tener diversos factores de calidad que a su vez se componen de criterios que son evaluados por métricas, con el propósito de abordar la evaluación desde lo general a lo particular, y permitir la reducción de la subjetividad en la asignación de un valor, ya sea cuantitativo o cualitativo.



Figura

Estructura de la calidad de software.

Así mismo, los modelos de calidad de software se clasifican de acuerdo con el enfoque de evaluación, ya sea a nivel de proceso, producto o calidad en uso.

Calidad a nivel del proceso

La calidad de un sistema software debe ser programada desde el inicio del proyecto, y posteriormente en cada etapa del proceso de desarrollo se debe llevar a cabo el control y seguimiento de los aspectos de calidad, para minimizar los riesgos y ofrecer soporte continuo, se garantiza así un óptimo nivel de cumplimiento de los factores de calidad, teniendo en cuenta que si en alguna de las etapas se deja de lado la verificación de los factores y criterios es posible que se presente deficiencia en alguno de éstos y disminuirá el nivel de calidad no solo del proceso, sino también del producto en desarrollo.

Calidad a nivel del producto

La principal finalidad del modelo de calidad de producto es especificar y evaluar el cumplimiento de criterios del producto, para lo cual se aplican medidas internas y/o medidas externas (Bevan, 2010). Por esta razón, algunas normas y estándares han definido la calidad a nivel de producto en tres tipos: interna, externa y en uso (Rodríguez, 2016). Este enfoque está orientado a verificar el cumplimiento de las características que permitan alcanzar la satisfacción del cliente en cuanto a los requisitos definidos en las etapas iniciales del proceso de desarrollo.

Calidad en uso

Es importante resaltar que aunque en diferentes escenarios se utilizan los términos usabilidad y calidad en uso, con el mismo propósito y de forma intercambiable tienen significados distintos, principalmente porque el concepto de calidad en uso es más amplio y abarca más elementos que la usabilidad (Covella, 2005), y esta última es una de las características de calidad de un producto software. La calidad en uso se define como el "conjunto de atributos relacionados con la aceptación por parte del usuario final y seguridad", y está basada en la eficacia, productividad, seguridad y satisfacción, según ISO/IEC 9126.

Modelos a nivel de proceso

Con base en la información recopilada se presenta la Figura 2, en la que se muestra la línea de tiempo de algunos modelos a nivel de proceso discutidos en este documento.

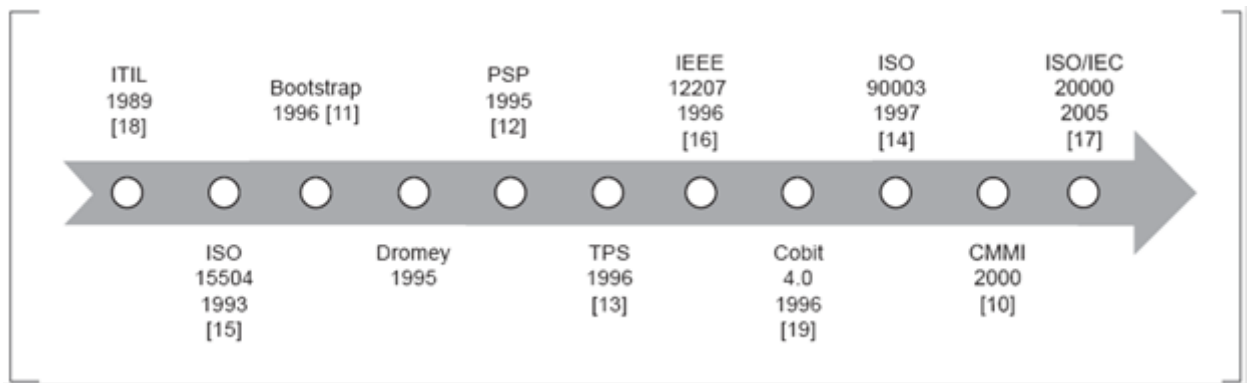


Figura.2

Línea de tiempo de modelos a nivel de proceso

ITIL: Desarrollado en el Reino Unido, con el fin de fortalecer la gestión gubernamental, a partir de cinco elementos fundamentales: la perspectiva del negocio, entrega del servicio, soporte del servicio, manejo de la infraestructura y manejo de aplicaciones, con el propósito de ofrecer una estructura integral para prestar a la organización un servicio completo, cubriendo necesidades de apoyo de instalación, adecuación de redes, comunicaciones, hardware, servidores, sistema operativo, y software necesarios.

ISO/IEC 15504: Permite adaptar la evaluación para procesos en pequeñas y medianas empresas (pymes) y grupos de desarrollo pequeños, mediante la estructuración en seis niveles de madurez: Nivel 0- Organización inmadura, Nivel 1- Organización básica, Nivel 2- Organización gestionada, Nivel 3- Organización establecida, Nivel 4- Organización predecible y Nivel 5- Organización optimizando. Su objetivo es llegar a que la organización logre ser madura, lo cual conlleva que la organización tenga procesos definidos, responsabilidades definidas, predicción de resultados, productos entregados con calidad, que las entregas se den en los tiempos pactados, incrementar la productividad, clientes satisfechos, y empleados felices (Córdoba, 2012).

Bootstrap: Metodología de evaluación que permite la mejora de procesos a partir de seis actividades básicas: Examinar la necesidad, Iniciar proceso de mejora, preparación y dirección de la evaluación, análisis de resultados, implantación y finalización de mejoras (Herrera, 2012).

Dromey: Es un modelo adaptable a evaluar varias etapas del proceso de desarrollo como levantamiento de requisitos, diseño e implementación. Se estructura con características y sub características de calidad; propone tres modelos distintos para cada etapa de construcción del producto: modelo de requerimientos, modelo de diseño y modelo de calidad de la implementación, a partir de la evaluación establecida en cinco etapas, para características como: eficiencia, confiabilidad, mantenibilidad, portabilidad, facilidad de uso y funcionalidad (Scalone, 2006).

Personal Software Process (PSP): Este modelo está enfocado al desarrollo profesional del ingeniero, fomentando una adecuada administración de calidad de los proyectos de desarrollo, reducción de defectos del producto, estimación y planeación del trabajo (Vargas, 2010).

Team Software Process (TSP): TSP es la fase posterior de PSP, está diseñado para el trabajo de equipos de desarrollo de software auto dirigidos, que se orienta al desarrollo de productos con el mínimo de defectos en tiempo y costos estimados. Cuenta con planes detallados y procesos como revisiones personales, inspecciones e índices de desempeño de calidad, y el fomento de la integración del equipo (Mondragón, 2011).

IEEE / EIA 12207: Este estándar establece un marco de trabajo común para el ciclo de vida del desarrollo de software, a partir del planteamiento de procesos, actividades y

tareas que pueden ser aplicadas durante la adquisición, suministro, desarrollo, operación, mantenimiento y/o despliegue de un producto software (ISO/IEC, 2008), (Córdoba, 2012)

Cobit 4.0: Se caracteriza por ser orientado a negocios y proceso, además de ser basado en controles, trabaja con siete criterios de información que son definidos como requerimientos de control del negocio: efectividad, eficiencia, confidencialidad, integridad, disponibilidad, cumplimiento y confiabilidad (Guerrero, 2006).

ISO 90003: Conjunto de estándares utilizados para el desarrollo, suministro y soporte del software, cuyo propósito es ofrecer una guía de aplicación de la norma 9001 que pretende ser utilizada para demostrar o soportar que la entidad está en capacidad de desarrollar software con criterios de calidad. (ISO, 1998).

CMMI (Capability Maturity Model Integration):

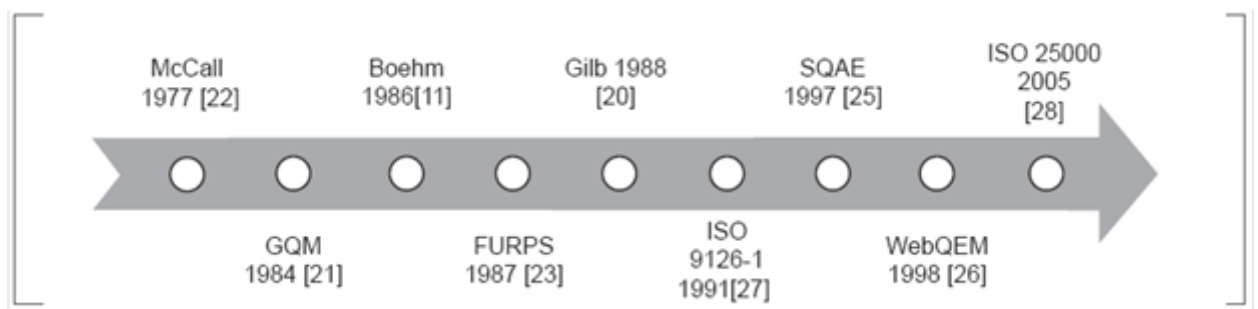
Es de los modelos más utilizados en las empresas de construcción de software, con el propósito de verificar el cumplimiento de estándares de calidad a partir de la medición con niveles de madurez. Este modelo se representa de dos maneras: escalonada y continua, donde el modelo escalonado está dirigido al software y permite clasificar las organizaciones en cinco tipos de nivel establecidos: Inicial, gestionado, definido, gestionado cuantitativamente y en optimización; y por su parte el modelo continuo se enfoca al análisis de la capacidad de cada proceso inmerso en las áreas de la ingeniería de sistemas y lo clasifica en uno de los siguientes seis niveles: Incompleto (0), ejecutado (1), gestionado (2), definido (3), cuantitativamente gestionado (4) y en optimización (5) (Petrie, 2009).

ISO/IEC 20000: El objetivo principal de esta norma es el de avalar que la prestación de servicios gestionados de TI de una empresa cuentan con la calidad necesaria para brindar

dichos servicios a los clientes. Se subdivide en dos partes: "Especificaciones", publicada como ISO 200001:2005, y "Código de buenas prácticas" publicada como ISO 20000-2:2005 (Van Bon, 2008).

Modelos a nivel de producto

La Figura 3 presenta la línea de tiempo de algunos modelos de evaluación a nivel de producto.



Figura

3

Modelos de calidad a nivel de producto

McCall: Uno de los modelos pioneros en la evaluación de la calidad de software, tiene tres etapas definidas: factores, criterios y métricas. Los once criterios base, son: Exactitud, confiabilidad, eficiencia, integridad, usabilidad, mantenibilidad, testeabilidad, flexibilidad, portabilidad, reusabilidad e interoperabilidad (Khosravi, 2004).

GQM o Goal Question Metric: Se enfoca a proporcionar una forma que permita definir métricas para medir el avance como los resultados de algún proyecto, a partir de la aplicación de unas preguntas relacionadas con el proyecto, que permitan alcanzar unas metas previamente planteadas, el modelo trabaja sobre metas, preguntas y métricas (Villarroel, 1999).

Boehm: Es un modelo incremental, dividido en regiones de tareas y estas a su vez en conjuntos de tareas, las cuales se ajustan a la cantidad de iteraciones que el equipo defina, y cada iteración se divide en cuatro sectores: planeación, análisis de riesgo, ingeniería y evaluación (Velazco, 2016).

FURPS: Modelo desarrollado por Hewlett-Packard, cuyo nombre proviene de los criterios que evalúa: Funcionalidad, usabilidad, confiabilidad (reliability), desempeño (performance) y soportabilidad (Soto, 2015).

GILB: Modelo de calidad que orienta la evaluación de software a partir de los atributos: Capacidad de trabajo, adaptabilidad, disponibilidad y utilizabilidad, los cuales se dividen en subtributos, de tal manera que sirva de apoyo a la gestión de proyectos, y proporcione una guía para solucionar problemas y detectar riesgos (Khosravi, 2004).

ISO 9126: Estándar basado en el modelo de McCall, dirigido a desarrolladores, aseguradores de calidad, evaluadores, analistas y cualquier otro involucrado en el proceso de construcción de software. Está dividido en cuatro partes: modelo de calidad, métricas externas, métricas internas y calidad de métricas en uso; elementos en torno a seis características (funcionalidad, Habilidad, usabilidad, eficiencia, mantenibilidad y portabilidad) y subcaracterísticas asociadas (Ango, 2014).

SQAE o Software Quality Assessment Exercise: Este modelo, basado en Boehm, McCall, Dromey e ISO 9126, está orientado principalmente a realizar evaluación por terceros que no están directamente involucrados con el desarrollo, siguiendo tres capas: área, factor y atributo de calidad, que permiten orientar la evaluación jerárquicamente (Moreno, 2010).

WebQEM: es una metodología de evaluación de calidad de sitios Web (Web-site Quality Evaluation method), diseñada para la evaluación siguiendo seis fases: planificación y programación de la evaluación de calidad, definición y especificación de requerimientos de calidad, definición e implementación de la evaluación elemental, definición e implementación de la evaluación global, análisis de resultados, conclusión y documentación, validación de métricas (Olsina, 1999).

ISO 25000: También llamadas como SQuaRE, cuyo propósito es guiar el desarrollo con los requisitos y la evaluación de atributos de calidad, principalmente: la adecuación funcional, eficiencia de desempeño, compatibilidad, capacidad de uso, Habilidad, seguridad, mantenibilidad y portabilidad (Alfonso, 2012).

4. Experiencias de implementación de modelos de calidad de software

A continuación se presentan algunas experiencias de aplicación de modelos y estándares de calidad de software.

CMMI

El modelo CMMI es uno de los modelos de mayor acogida para la evaluación de grandes empresas, como por ejemplo empresas desarrolladoras de software, la cuales necesitan cumplir con cierto nivel de madurez de los que propone el modelo, certificando así que el producto software cumple con criterios de calidad. En la Tabla 1, se presenta un listado de algunas de las empresas que realizaron la implementación de dicho modelo según (Nakama, 2013), (Villalba, 2011), (Navarro et al., 2010), (Sandia, 2007), (Ramos et al., 2010), (Romero et al., 2008), (Cruz, 2011) (Álvarez *et al.*, 2011).

Tabla 1**Implementación del modelo CMMI**

Empresa	Área	País
Sistemas y Software Consultores	Desarrollo de software	Perú
RCCS	Academia y producción	Colombia
UNKASOFT	Desarrollo de software	España
CEIDIS(Coordinación de Estudios Interactivos a Distancia)	Academia	Venezuela
Icosis Grupo Avalon	Desarrollo de software	España
Alianza Sinertic	Tecnología	Colombia
Santos CMI	Desarrollo de software	Ecuador
inSoft Cia. Ltda.	Desarrollo de software	Ecuador

La empresa "Sistemas y Software Consultores" implementó dos de los niveles de CMMI (niveles 2 y 3), con el fin de mejorar y optimizar las actividades de desarrollo en procesos piloto de la empresa, esta organización es prestadora de servicios de programación a los proyectos de software, inicialmente se realizó una medición del estado actual de la empresa con respecto a la evaluación que realiza CMMI. Con base en esta medición se identificaron cuatro áreas críticas en el proceso: gestión de requerimientos, desarrollo de requerimientos, verificación y validación. Se seleccionaron tres proyectos como muestra para la implementación del modelo. La empresa utiliza como metodología RUP (Rational Unified Process) e IBM Rational para el diseño, posterior a un análisis que permitió identificar relaciones entre el modelo CMMI vs RUP se determinó que algunos aspectos del modelo CMMI son cubiertos por actividades y artefactos de RUP, lo cual permitió no realizar un cambio de gran impacto en los procesos de desarrollo de los proyectos (Romero et al., 2008).

La implementación de CMMI en "Sistemas y Software Consultores" se estructuró en cuatro etapas: preparación, inducción y capacitación, implementación y seguimiento, y análisis de preparación.

Bootstrap

Este modelo se ha implementado principalmente en empresas europeas, dentro de la revisión bibliográfica es escasa la documentación encontrada con respecto a su implementación. Según "Servicios en informática y desarrollo de México" (B. L, 2015), cuyo objetivo es el desarrollo de software y "CERTUM" empresa ecuatoriana, dedicada a procesos de certificación (B. L, 2015) son empresas que implementaron el modelo Bootstrap.

PSP Personal Software Process

PSP (Personal Software Process), es un modelo enfocado al personal involucrado en el proceso, este modelo se ha implementado en ámbitos académicos, desarrollo de software y mejora de procesos empresariales, uno de los casos de estudio que se revisaron es el de una organización desarrolladora de productos de software ERP, CRM, Educativos y otros productos especiales donde se encontró una integración de metodologías ágiles (SCRUM) con PSP, identificando que el porcentaje de error cada vez era más bajo para la mayoría de desarrolladores, favoreciendo así el proceso de estimación, y mejorando el proceso de desarrollo (Holguín, 2015). En la Tabla 2 se presenta un listado de algunas de las empresas que realizaron la implementación del modelo PSP para la mejora de sus procesos (Soledispa, 2010), (Soto, 2010), (Cardona, 2012), (Ampudia, 2007), (Holguín, 2015), (Elminir, 2009).

Tabla 2

Implementación del modelo de PSP

Empresa	Área	País
Molemotor S.A	Mejora de procesos	Ecuador
Universidad de pamplona	Académico	España
Universidad Quindío	Académico	Colombia
Escuela Politécnica Nacional	Desarrollo de software	Ecuador
Reservado	Desarrollo de software	Colombia
Reservado	Desarrollo de software	Egipto

TSP Team Software Process

PRAGMA es una empresa colombiana, enfocada en el desarrollo de soluciones de negocio, con una trayectoria de 15 años, que decide implementar la metodología TSP (Team Software Process), para lograr alcanzar un grado de madurez alto en CMMI. Como primera medida, la empresa realiza una capacitación a las directivas de la compañía, a la cual asisten las gerentes en su totalidad. En tanto los desarrolladores se capacitan en PSP (Personal Software Process), para el proceso de TSP se definió un proceso de juntas dirigidas por un coach TSP donde se identifican tres roles: moderador, cronometrista y anotador. Se realizaron durante cuatro días juntas con las siguientes tareas: establecer los objetivos del negocio y producto, asignación de roles y definición de objetivos del equipo, generar una estrategia de desarrollo, elaboración descendente del plan, desarrollo de plan de calidad, construcción de los planes detallados para la siguiente fase, conducción de la evaluación de riesgo, preparar presentación a la dirección y reporte de lanzamiento, revisión con la alta dirección, y post mortem del lanzamiento (Yarce, 2012).

En la Tabla 3 se presenta un listado de algunas de las empresas que han implementado el modelo (Asencio, 2009), (Bustos, 2007), (Webb, 1999).

Tabla 3

Implementación del modelo TSP

Empresa	Área	País
Reservado	Desarrollo de software	Ecuador
Reservado	Desarrollo de software	Ecuador
Hill Air Force Base	Base de la fuerza aérea de los Estados Unidos	Estados unidos

ISO 90003

Brainstorm Software SRL, es una empresa desarrolladora de software enfocada en el sector asegurador, obtuvo las certificaciones ISO 9001:2008 e ISO 90003:2004, las cuáles fueron otorgadas por IQNet e IRAM. Durante este proceso de certificación Brainstorm fue asesorada por GW consulting. Los beneficios de obtener esta certificación para la empresa son: reducción de defectos e incidencias, aumento de productividad, aumento de compromiso con los requisitos del cliente y una mejora continua en los servicios (Brainstorm, 2016). En la Tabla 4 se presenta un listado de algunas de las empresas que realizaron la implementación de dicho modelo (Brainstorm, 2016), (INFT, 2016), (Anti-lank, 2016), (gA, 2016).

Tabla 4**Implementación del modelo ISO 90003**

Empresa	Área	País
Brainstorm software	Desarrollo de software	Argentina
INTF	Desarrollo de software	Argentina
Alnitak Informática	Desarrollo de software	España
gA	Consultoría y Outsourcing	Argentina

ISO 15504

Para la norma ISO 15504 se encuentra un listado de empresas certificadas registradas, a continuación se enuncian las certificadas entre los años 2012 a 2016: Bitware, Agrupo Sistemas S.L, Quental Technologies S.L y Tahbit Software S.A, dentro de dicho listado se logró identificar que las empresas son enfocadas en tecnologías de la información y soluciones tecnológicas. En la Tabla 5 se muestra un listado de algunas de las empresas que realizaron la implementación de la norma ISO 15504 (ISO 15504, 2016).

Tabla 5**Implementación de modelo ISO 15504**

Empresa	Área	País
Bitware	Tecnologías de la información	España
Agrupó Sistemas S.L.	Tecnología e innovación	España
Quental Technologies S.L	Servicios y soluciones tecnológicas	España
Tahbit Software S.A	Servicios y soluciones tecnológicas	España

ISO/IEC 20000

Para la norma ISO/IEC 20000, se encuentra un listado de empresas certificadas registradas, enfocadas en la prestación de servicios de TI en la siguiente dirección web www.isoiec20000certification.com (AMP, 2016).

ITIL

Eurotrans (Holanda), es una empresa dedicada al transporte de mercancía y pasajeros por carretera, encontró la necesidad de implementar ITIL para evolucionar la operativa y alcanzar los objetivos planteados. Para su implementación se realizó un análisis de todos los procesos definidos por ITIL identificando qué procesos se encontraban actualmente implementados en la empresa.

Se realizó un análisis en cada proceso, identificando los puntos fuertes y los débiles del mismo, empezando por realizar un estudio de viabilidad y valoración de la situación actual, posterior a esto se tiene en cuenta PRINCE2 (Entorno controlado de proyectos IN), para analizar la coherencia y aproximación de los proyectos con respecto

a ITIL. En cuanto a los procesos del servicio de soporte, Eurotrans no tenía un servicio de escritorio implementado, por tanto se tomó la decisión de implantar un servicio de escritorio centralizado para clientes, proveedores y la organización. Se definieron tareas y estrategias para el tratamiento de gestión de incidencias, gestión de problemas, gestión del cambio, gestión de la configuración y gestión de software.

Para los procesos de servicio de entrega de igual forma se identificaron posibilidades de mejora y se propusieron estrategias para gestión de niveles de servicio, gestión de disponibilidad, gestión de capacidad, gestión de continuidad y gestión financiera, permitiendo así realizar mejoras u optimizar procesos con la aplicación de procesos definidos en ITIL (Herrera, 2010). Así como el caso de Eurotrans, la empresa Outsourcing S.A, una empresa colombiana prestadora de servicios de TI también logró implementar el modelo ITIL (Quevedo, 2009).

COBIT 4.0

Ecopetrol es una empresa colombiana dedicada al ejercicio de las actividades propias de la industria y el comercio del petróleo y sus afines. Esta empresa comenzó a desarrollar el proceso de implementación de cobit desde el año 2008, este proceso ha permitido la implementación de 28 de los 34 procesos definidos por cobit, teniendo como prioridad los objetivos de control, se realizó el desarrollo, diseño y documentación de los procesos y a continuación su implementación y monitoreo, haciendo los ajustes necesarios.

Hacia el año 2009 se implementaron 14 de los procesos elegidos.

En la Tabla 6 se observan algunas de las empresas del sector bancario que lograron implementar COBIT 4.0. (Herrera, 2012), (COBIT, 2016), (Valverde, 2014).

Tabla 6

Implementación del modelo COBIT 4.0

Empresa	Área	País
Bank of Nova Scotia	Banco	Canadá
Banco Supervielle S.A.	Banco	Argentina
Ecopetrol S. A.	Empresa petrolera	Colombia

GILB

El modelo Gilb fue aplicado en el Process Improvement Experiment funded under the European Systems and Software Initiative (ESSI), siendo implementado por el proyecto GINSENG (Gilb's Inspections for Software Engineering), con el objetivo de mejorar las prácticas de implementación de redes de telecomunicación y desarrollo del software, aumentando así la efectividad y eficiencia de las actividades para identificar los defectos en los procesos de desarrollo e implementación. Cuenta con cuatro fases fundamentales, la primera se denomina introducción, la segunda permite definir los procesos, la tercera describe los procesos en ejecución y finalmente se lleva a cabo la evaluación de procesos (Stasinos, 2012). Intracom S.A. (Grecia), es una empresa cuya área de trabajo son las telecomunicaciones, que implemento el modelo de GILB (Stasinos, 2012).

GQM

En Cuba se aplica el modelo GQM en contextos académicos, en donde se ha identificado una pequeña brecha entre los resultados de la evaluación de software y la comprensión del equipo de desarrollo, de esta manera en la aplicación de las tres etapas del modelo en conjunto con UML se describe la estructura a partir del proceso de pruebas,

teniendo en cuenta la arquitectura y el comportamiento de los datos en el momento de ser probados (Vega, 2015).

Entre las experiencias de aplicación de GQM en el campo empresarial se puede mencionar el caso de la implantación del modelo en un core bancario, a partir de la definición de métricas, alineadas con los objetivos y metas del negocio, se orienta al mejoramiento en el proceso del desarrollo de software, conducente a la obtención de un producto de calidad para la empresa (Sánchez, 2015). En la Tabla 7 se presenta un listado de empresas que realizaron la implementación de GQM (Sánchez, 2015), (Sánchez, 2011), (Acosta, 2015).

Tabla 7

Implementación del modelo GQM

Empresa	Área	País
TERABANK	Desarrollo de software bancario	Georgia
Universidades europeas	Competencias en ingeniería académica	Europa
Reservado	Evaluación de plataforma Joomla	Argentina
Universidad de Ciencias Informáticas	Modelo de desarrollo de software	Cuba

McCall

A través de los años, el modelo McCall ha sido implementado en diferentes contextos, es el caso de la aplicación realizada en Nueva Zelanda, en la cual se centraron en el sector agrícola, definiendo los parámetros que requieren del ajuste del lugar en donde se va a realizar el estudio, esto para poder restringir la portabilidad para otras regiones o situaciones que puedan ser similares. Se basó en definir todas las variables

posibles que puedan afectar el resultado de los sembríos, teniendo presente medidas de tiempo, cantidad y temperatura que influyen en el producto sea del agrado para su venta (Romera, 2009). Así mismo, en México se implementó en un proyecto de aplicación de un modelo colaborativo para el desarrollo ágil de software, con base en los factores del modelo de McCall se evalúa la calidad del software a partir de listas de comprobación de las características operativas, aceptación del cambio, y adaptabilidad a nuevos ambientes (Cedejas *et al.*, 2015). En la implementación de este modelo se encuentra que las empresas que han implementado este modelo en su mayoría se enfocan en el sector del software, en la Tabla 8 se listan algunas empresas encontradas en la revisión realizada (Romera, 2009), (Cedejas et al., 2015), (Ramírez, 2010), (Bernardo, 2000), (Garzías, 2007).

FURPS

El modelo FURPS ha sido utilizado para el diseño y validación de interfaces para usuarios finales, evaluando su funcionalidad, usabilidad, confiabilidad, desempeño y soporte, para tener como salida final un producto que cumpla las reglas del negocio (Eeles, 2005), es así que se ha utilizado como un clasificador de requisitos, ayudando a la asignación correcta de requisitos, implementación, y diseño de interfaces; aunque se ha identificado que implica un amplio número de métricas para su desarrollo, concluyendo de esta manera que se debe estimar el tiempo necesario para su implementación (Cifuentes, 2015). En la Tabla 9 se presentan algunas empresas que han acogido e implementado el modelo de FURPS (Gómez et al., 2014), (Eeles, 2005), (Cifuentes, 2015), (Peñalva, 2014), (Al-Qutaish, 2010).

Tabla 8**Implementación del modelo de McCall**

Empresa	Área	País
DarbySoft	Desarrollo de software	España
Universidad de Pereira	Implementación de software	Colombia
Universidad de México	Desarrollo de software	México
Reservado	Agricultura	Nueva Zelanda
ESD (Electronic System Decision)	Desarrollo de software	Estados Unidos
RADC (Rome Air Developer Center)	Comunicación	Estados Unidos
GE General Electric	Electrodomésticos	Estados Unidos

Tabla 9**Implementación del modelo FURPS**

Empresa	Área	País
Universidad De Santander	Académico y diseño de interfaces	Colombia
Hewlett Packard	Creación y ensamblaje de equipos de computo	Estados unidos
Universidad de Santander	Académico	Colombia
IBM Rational Software Company	Desarrollo de software	Estados unidos
IBM	Desarrollo de software	Estados unidos

BOEHM

La universidad de Malasya se encargó de evaluar los criterios del proceso de la ingeniería de software para la evaluación de costos (Solemon, 2013), identificando cada uno de los objetivos que se deben cumplir para una aplicación acorde, con el diseño del sistema, siendo útil para la evaluación en los costos del software y viendose reflejado en las buenas práctica de estimación.

Otra aplicación de este modelo es la definición de un lenguaje de programación que sea apropiado, reconocido y que permita la evolución del sistema, lo anterior a partir de la evaluación de características como consistencia, integridad, transporte de datos, permitiendo de esta forma que los sistemas evolucionen con calidad (Strub, 2015). La Tabla 10, revela algunas empresas que implementaron el modelo (Solemon, 2013), (Strub, 2015), (Boehm, 1991), (Mcmurtrey, 2013), (Moniruzzaman, 2013), (Matkovic, 2010), (Weckman, Colvin, Gaskins, Mackulak 1999).

Tabla 10

Implementación del modelo BOEHM

Empresa	Área	País
Home Health	Hospital	Argentina
Universidad de Lyon y Londres	Académico en desarrollo de software	Inglaterra – Francia
Universidad de Malasya	Académico en evaluación de criterios para requerimientos	Malasya
Universidad de Illinois	Académico	Estados Unidos
Universidad de Novi Sad	Académico	Servia
SEMATECH's	Riesgos logísticos	Estados Unidos
Universidad de California	Académico, software de gestión de riesgos	Estados Unidos

DROMEY

Gracias a la flexibilidad de las propiedades este modelo se puede utilizar en diferentes contextos, y responder a preguntas asociadas a la identificación de propiedades de calidad, impacto de la medición de atributos, entre otras que conllevan a responder categorizando el sistema con propiedades de exactitud, estructura y descripción (Ortega, 2000); la implementación del modelo Dromey se refleja como una definición de métricas estadísticas asociadas al desarrollo del software y la mejora continua de éste, por lo cual en este caso de estudio se procedió a la selección del conjunto de atributos a evaluar en la aplicación, realizando después una lista de chequeo de los componentes y módulos del sistema, para llegar a identificar cada una de las propiedades de calidad que contienen estos módulos y cómo estas afectan cada atributo de calidad. Los atributos considerados fueron: Atributos de proceso (relacionados con el software), producto (documentos, componentes, entregas de resultados de una actividad) y recurso (describen las medidas de ejecución del proyecto); con estas métricas se tiene una orientación a la calidad de software (Adve, 2010). De igual manera, el modelo Dromey se aplicó a un metamodelo para realizar la especificación de modelos de calidad en Model-Driven Ingeniería, permitiendo que por su flexibilidad defina los atributos de calidad enfocados al producto e identificar los bienes tangibles y propiedades para alcanzar los objetivos trazados. Estas propiedades principalmente son: de exactitud, internas y descriptivas (Adve, 2010). La Universidad de España, así como la Universidad de Simón Bolívar (Venezuela) lograron implementar DROMEY, en los procesos académicos así como de desarrollo de software (Strub, 2015), (Tahhánn, 2006).

ISO9126

Según el caso de estudio realizado en Australia, el modelo ISO 9126 comienza a tener sus primeras implementaciones en industrias de desarrollo de software, con el propósito de mejorar el proceso de calidad de software, e identificar aquellos atributos de calidad que permitieran generar requisitos que sean comprendidos y adaptados a cualquier diseño propuesto, en cumplimiento de un desarrollo y un plan de pruebas efectivo (Dromey, 1996).

Algunas empresas de TI haciendo uso de ISO 9216 han aplicado métricas de calidad del software a partir de la estructura básica de la norma: Modelo de calidad, métricas externas, métricas internas, métricas para la calidad de uso, y así se categorizan los atributos en funcionalidades de fiabilidad, usabilidad, eficiencia, entre otros, permitiendo que sea más robusto las pruebas de calidad a los sistemas de TI (Estayno, 2009). En la Tabla 11 se presenta una recopilación de algunos casos que implementaron el estándar ISO 9126 (Tahhánn, 2006), (Ortega, 2000), (Dromey, 1996), (Estayno, 2009), (Moreno, 2012), (Quintero, 2005), (Estayno, 2009).

Tabla 11**Implementación del modelo ISO 9126**

Empresa	Área	País
Universidad Nacional de los Llanos Occidentales	Desarrollo de software	Estados Unidos
Instituto Australiano de Calidad de Software	Desarrollo de software	Australia
Universidad De Simón Bolívar- Venezuela	Universidad De Simón Bolívar- Venezuela	Venezuela
Universidad Nacional del Nordeste	Académico, estudio de calidad de software	Argentina
Universidad del Valle	Académico, estudio de tecnología de la información	Argentina
Universidad Nacional Abierta a Distancia (UNAD)	Académico, implementación de norma	Colombia
Universidad Militar Nueva Granada	Académico	Colombia

SQAE, ISO25000

En INTEDAYA (International Dynamic Advisors) se realizó la implementación de ISO 25000 (SQuaRE), en la cual, uno de los servicios es la consultoría enfocándose en la definición de requisitos de calidad para productos de software a fin de alinear las necesidades reales de los sistemas de software, evitando ineficiencias, maximizando la rentabilidad para obtener un producto de calidad, e ir cumpliendo con todos requisitos que el cliente propuso (Intedya, 2016). La Tabla 12 muestra una recopilación de los casos de éxito encontrados en la revisión que implementaron el modelo (Oktaba, 2012), (Velandía, 2011), (Rodríguez, 2016), (Aplies, 2016), (Garzás, 2007), (Kybeleconsulting, 2016), (Barrera, 2015), (Castellanos, 2014), (Benitez, 2014), (Castillo, 2015), (Intedya, 2016).

Tabla 12**Implementación del modelo de SQAE, ISO**

Empresa	Área	País
GC Buzz	Académico	México
Instituto Nacional Sena	Académico	Colombia
Intedya	Consultoría	España
Reservado	Evaluación de software	España
Applies	Solución de procesos y gestión tecnológica y minera	Chile
233 grados TI	Desarrollo de software	España
Kybele Consulting	Consultoría de software	España
Universidad de Manizales	Académico	Colombia
GEXRENOF	Pruebas de desarrollo	Cuba
Colegio de ingenieros del Perú	Desarrollo de software	Perú
Supermercados La Matagalpa	Venta de productos naturales	Nicaragua

WEBQEM

Algunas universidades de Argentina, como la Universidad del Rosario y la Universidad de La Plata realizaron el estudio del comportamiento de WebQem en la implementación de software institucional, enfocándose en la evaluación de la calidad en redes socio técnicas para el desarrollo del conocimiento, junto a algunas métricas que fueron utilizadas por el portal bancario a nivel de servicio de sus clientes. En la Tabla

13 hay una recopilación de los casos de éxito encontrados en la revisión que implementaron el modelo (Rodríguez, 2016)

Tabla 13

Implementación del modelo WEBQEM

Empresa	Área	País
Universidad del Rosario	Académico, estudio de redes sociotécnicas (Desarrollo)	Argentina
Universidad de la Plata	Académico	Argentina
Portales bancarios	Evaluación de calidad web	Argentina
INCAMI	Pruebas de desarrollo y calidad de software	Argentina

Una vez realizada esta revisión de estándares y modelos de evaluación de calidad, así como su implementación, cabe resaltar que los países suramericanos tienen una gran acogida e interés en certificar sus procesos para garantizar la calidad de sus productos, así como también es notorio que el estándar CMMI predomina en su implementación en empresas de desarrollo de software.

Tanto los estándares como los modelos garantizan la calidad tanto del producto como del proceso, su implementación depende de las necesidades y enfoques de la empresa interesada en certificarse.

Conclusiones

Los estándares de calidad de software surgen con la necesidad de obtener productos libres de fallas, que satisfagan las necesidades del cliente y así lograr su satisfacción. Si bien los factores de calidad dependen del modelo, existen algunos estándares, los cuales sirven de base a la hora de adaptar o crear nuevos modelos de calidad. Es importante contar con un modelo de calidad propio y adecuado al enfoque utilizado para el desarrollo, que se centre especialmente en los detalles de calidad particulares para la metodología. Es necesario identificar el conjunto de atributos de calidad y métricas para realizar su evaluación y que se ajusten exclusivamente a la metodología a utilizar, ya sea, por una particularización de un modelo general existente como las ISO o la creación de uno nuevo basándose en los ya existentes.

Algunos modelos de calidad clásicos han sido la base para los de calidad más recientes, y han permitido que los modelos actuales se consoliden como los más completos con base en la evolución del software, para así optimizar los procesos de las organizaciones y garantizar que se cumple con criterios o estándares que respaldan la calidad de la gestión de procesos del negocio.

Es importante que las empresas se certifiquen bajo alguna norma o estándar, pues esto permite que la misma tenga una mejor posición, reconocimiento y demanda en el mercado, ya que al estar avalada por alguna entidad competente garantiza un nivel de satisfacción mayor para los clientes.

En su mayoría, la implementación de modelos de calidad de software ha sido adoptada por empresas desarrolladoras de software, sin embargo algunos modelos

permiten adaptarse a contextos empresariales con fines diferentes al del desarrollo o construcción de software.

Referencias bibliográficas

ACOSTA, Julio. *et al* . "Evaluación de mantenibilidad de un gestor de contenidos open source utilizando métricas de orientación a objetos." Anales de las 42 JA11O - Jornadas Argentinas de Informática. 10º Jornadas Argentinas de Software Libre, JSL 2013. 2015. p. 15-29 <https://albertdel.wordpress.com/actividades/fundamentos-de-calidad-de-software/>

ADVE, Sarita and BOEHM, Hans-J. Memory models: a case for rethinking parallel languages and hardware. In: Communications of the ACM. 2010, vol. 53, no 8, p. 90-101. <https://es.slideshare.net/lidizzg/definicion-de-calidad-y-calidad-de-software>