

Modelos de calidad del software, un estado del arte*

Mauro Callejas-Cuervo

Doctor en Energía y Control de Procesos, Universidad de Oviedo, España. Magíster en Ciencias Computacionales, Instituto Tecnológico y de Estudios Superiores de Monterrey – ITESM, México. Profesor Asociado en la Universidad Pedagógica y Tecnológica de Colombia, Tunja – Colombia.

 <http://orcid.org/0000-0001-9894-8737>

mauro.callejas@uptc.edu.co

Andrea Catherine Alarcón-Aldana

Magíster en Software Libre, Universidad Autónoma de Bucaramanga, UNAB, en Convenio con la Universitat Oberta de Catalunya. Profesora Asistente en la Universidad Pedagógica y Tecnológica de Colombia, Tunja – Colombia.

andrea.alarconaldana@uptc.edu.co

Ana María Álvarez-Carreño

Estudiante Maestría en Tecnología Informática, Universidad Pedagógica y Tecnológica de Colombia, Tunja – Colombia.

ana.anita8905@gmail.com

RESUMEN

Existen diferentes modelos de calidad del software que permiten un proceso de mejora continua con su implementación. En el presente escrito se exponen características de los modelos investigados a través de una revisión de información, se realiza una clasificación de los mismos de acuerdo con el nivel de proceso o de producto; cada modelo cuenta con unas características y una estructura la cual se enuncia brevemente en el escrito. Las diferentes empresas, con el fin de verse inmersas en una mejora continua de sus procesos, han identificado la importancia de hacerse a la implementación de modelos que permitan certificar y garantizar la calidad de sus productos y procesos, se encontraron algunos casos de éxito de la implementación de los modelos estudiados, los cuales permitieron a las empresas posicionarse o destacarse por mejorar los procesos de su negocio.

PALABRAS CLAVE

Calidad, modelo, estándar, software, evaluación, calidad de software

Software quality models, a state of the art

ABSTRACT

There are many software quality models that allows a process of continuous improvement with its implementation. In this paper some characteristics of the investigated models are exposed, through an information revision, a classification of these models is performed according to the level of the process or product; each model count with some characteristics and a structure that is briefly stated in the written. The different enterprises in order to be immersed on a continuous improvement of their processes have identified the importance of getting the implementation of models that allows to certify an guarantee the quality of their processes and products, some success cases were found for the implementation of the studied models, which allowed the enterprises to position or stand by the improvement of their business process.

KEYWORDS

Quality, model, standard, software, evaluation, software quality

Recibido: 22/06/2016 Aceptado: 08/10/2016

* Este artículo hace parte de los resultados obtenidos en desarrollo del proyecto de investigación denominado: Implementación de un ambiente virtual de aprendizaje como soporte al proceso de enseñanza-aprendizaje en las áreas de ingeniería del software y bases de datos de la Escuela de Ingeniería de Sistemas y Computación de la UPTC.

<http://dx.doi.org/10.18041/entramado.2017v13n1.25125> Este es un artículo Open Access bajo la licencia BY-NC-SA (<http://creativecommons.org/licenses/by-nc-sa/4.0/>)

Cómo citar este artículo: CALLEJAS-CUERVO, Mauro ; ALARCÓN-ALDANA, Andrea Catherine; ÁLVAREZ-CARREÑO, Ana María. Modelos de calidad del software, un estado del arte. *En:* Entramado. Enero - Junio, 2017. vol. 13, no. 1, p. 236-250, <http://dx.doi.org/10.18041/entramado.2017v13n1.25125>



Modelos de qualidade de software, um estado da arte

R E S U M O

Existem diferentes modelos de qualidade de software que permitem processo de melhoria contínua com a sua implementação. Neste escrita características dos modelos investigados através de uma revisão das informações forem definidos, a mesma classificados de acordo com o nível de processo ou produto é feito; cada modelo tem características e uma estrutura que brevemente indicado na carta. Diferentes empresas, a fim de ser imerso em um processo contínuo de melhoria identificaram a importância de começar a implementação de modelos para certificar e garantir a qualidade de seus produtos e processos, alguns casos de sucesso foram encontrados implementação dos modelos estudados, o que permitiu que as empresas a posicionar ou ficar para melhorar seus processos de negócios.

PALAVRAS-CHAVE

Qualidade, modelo, padrão, software, teste, qualidade de software

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 organiza-

ción 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.

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 de 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 minimi-

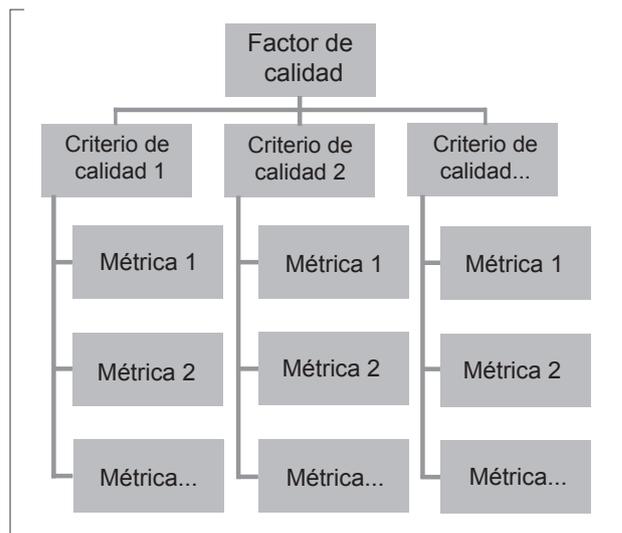


Figura 1. Estructura de la calidad de software.
Fuente: Los autores

zar 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 de 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.

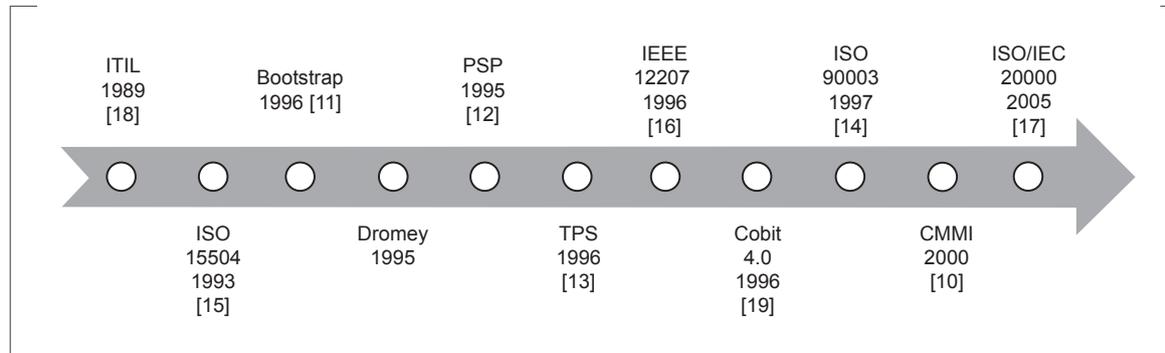


Figura 2. Línea de tiempo de modelos a nivel de proceso
Fuente: Los autores

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 subcaracterísticas 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 autodirigidos, 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 20000-1:2005, 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.

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 subatributos, 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, fiabilidad, 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).

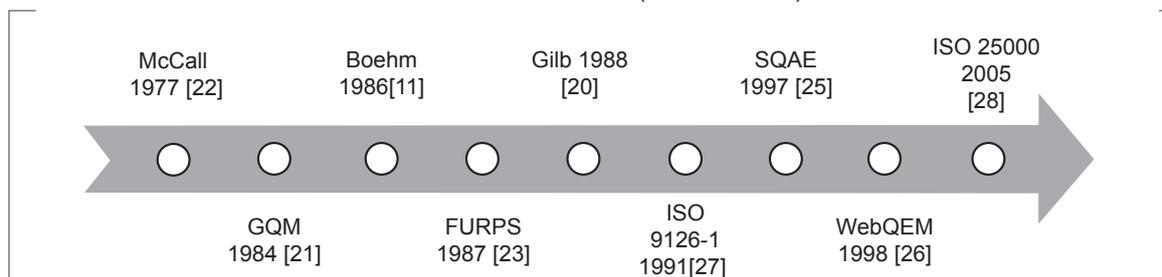


Figura 3. Modelos de calidad a nivel de producto
Fuente: Los autores

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, fiabilidad, 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

Fuente: Los autores

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

Fuente: Los autores

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 los 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

Fuente: Los autores

ISO 9003

Brainstorm Software SRL, es una empresa desarrolladora de software enfocada en el sector asegurador, obtuvo las certificaciones ISO 9001:2008 e ISO 9003:2004, 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 9003

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

Fuente: Los autores

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
Agrupo 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

Fuente: Los autores

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

Fuente: Los autores

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

Fuente: Los autores

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), (Garzás, 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

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

Fuente: Los autores

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 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

Fuente: Los autores

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), (Matković, 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

Fuente: Los autores

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), (Tahhán, 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 (Tahhán, 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

Fuente: Los autores

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 25000

Empresa	Área	País
GC Buzz	Académico	México
Instituto Nacional Sena	Académico	Colombia
Intedya	Consultoría	España
Reservado	Evaluación de software	España
Aplies	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

Fuente: Los autores

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 sociotécnicas 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), (Olsina, 2002), (Alfonzo, 2012), (Molina, 2007).

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

Fuente: Los autores

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.

5. Conclusiones

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. ☰

Conflicto de intereses

Los autores declaran no tener ningún conflicto de intereses.

Referencias bibliográficas

1. ACOSTA, Julio. *et al.* "Evaluación de mantenibilidad de un gestor de contenidos open source utilizando métricas de orientación a objetos." Anales de las 42 JALIO - Jornadas Argentinas de Informática. 10^o Jornadas Argentinas de Software Libre, JSL 2013. 2015. p. 15-29
2. ADVE, Sarita and BOEHM, Hans-J. Memory models: a case for re-thinking parallel languages and hardware. *In: Communications of the ACM.* 2010, vol. 53, no 8, p. 90-101.
3. AL-QUTAISH, Rafa E. Quality models in software engineering literature: an analytical and comparative study. *In: Journal of American Science.* 2010. vol. 6, no.3, p.166-175
4. ALARCÓN ALDANA, Andrea C.; GONZÁLEZ SANABRIA, Juan Sebastián, y RODRÍGUEZ TORRES, Sandra Lucia. "Guía para pymes desarrolladoras de software, basada en la norma ISO/IEC 15504." *En: Revista Virtual Universidad Católica del Norte.* Sept.-Dic., 2011. no. 34, p. 285-313.
5. ALFONSO, Pedro L. "Revisión de modelos para evaluar la calidad de productos web. Experimentación en portales bancarios del NEA", M.S. tesis, Dep: facultad de Informática, Universidad Nacional de La Plata, Argentina, 2012.
6. ALVAREZ CHIRIBOGA, Daniel Alejandro, y ORTEGA NAVARRETE, Fernando Enrique. Plan de implantación de nivel de madurez CMMI 3 para una empresa de desarrollo de software ecuatoriana. Facultad De Ingeniería, Escuela De Sistemas, Escuela Politécnica Nacional, Ecuador, 2009, p.156.
7. AMP Group, Available: <http://www.isoiec20000certification.com/home/ISOCertifiedOrganizations/ISOCountryListings.aspx?view2page=27>
8. AMPUDIA LARCO, Andrés Enrique. Uso del PSP (Personal Software Process) en el desarrollo de software. Quito, Ecuador: Escuela de Ingeniería de sistemas, 2007, 1.168 p.
9. ANGO HERRERA, Luis Fernando. "Evaluación de Sistemas", Tesis, Ibarra, Pontificia Universidad Católica del Ecuador, 2014, 19p.
10. ANTILANK, Available: <http://www.alnitak.es/index-1.html>, 2016
11. APPLIES, Soluciones de Gestión, Procesos, Tecnología y Minera. Available: <http://www.applies.cl/empresa.htm>.
12. ASENCIO MERA, José Luis *et al.* Implementación de un sistema integrado utilizando procesos de software en equipo (tsp). 2009, p.1-12, 2009.
13. B. L. Roció Gabriela. Bootstrap, Estándares y modelos de calidad aplicados al software, Conferencia, Tepic, Nayarit, 02 de Septiembre de 2015.
14. BAPTISTA, P.T. "MODELOS DE CALIDAD MC CALL, ISO/IEC 9126, ISO 25000". *In Gestión Operativa de la Calidad en el Software, Ingeniería de Sistemas de la Universidad Autónoma Gabriel Rene Moreno,* 2012.
15. BARRERA HOLGUÍN, Jhon Alexander. Integración de marcos de trabajo para desarrollo de software: Scrum, PSP e ISO 25000 Integrating software development frameworks: Scrum, PSP and ISO25000.. Ven-tana Informática, Facultad de Ciencias e Ingeniería, Universidad de Manizales, Colombia Vol: 32, pp: 151-164. 2015.
16. BAUTISTA, Leydi, *et al.* "Modelos De Calidad De Software", Tesis: Control de Calidad de software, ICA PERU, 2012, pp: 1-46
17. BENITEZ, Jackeline y FLORES AMAO, Alodia. Sistema Web basado en la Norma ISO/IEC 25010: 2010 (SQuaRE). Facultad de Ingeniería Mecánica Eléctrica, Electrónica y Sistemas Escuela Profesional de Ingeniería de Sistemas, CIENCIA DE LA INGENIERIA DE SISTEMAS. 2014. vol.1, no.1.
18. BERNARDO, Miguel *et al.* "El modelo de McCall como aplicación de la calidad a la revisión del software de gestión empresarial." 2000.
19. BEVAN, Nigel. "Los nuevos modelos de ISO para la calidad y la calidad en uso del software." *Calidad del producto y proceso software*, Cap: 2, España: Editorial Ra-Ma, 2010, pp: 5-75.
20. BOEHM, Barry W. "Software risk management: principles and practices." *In: IEEE software.* 1991. vol 8, no.1, p. 32-41.
21. Brainstorm Software logra la Certificación en las Normas de Calidad ISO 9001-2008 e ISO 9000, 2016.
22. BUSTOS Recalde, Geovanna Patricia *et al.* Uso del TSP (Team software process) en el desarrollo de software. Escuela de Ingeniería de sistemas, Quito, Ecuador, pp: 1-203, 2007.
23. CARDONA TORRES, Sergio Augusto. Diseño de una estrategia de aprendizaje para implementar prácticas de PSP y TSP en cursos básicos de programación: caso programa de ingeniería de sistemas y computación Universidad del Quindío. Tesis de Maestría. Colombia: Universidad Eafit, Escuela de Ingeniería Departamento de Informática y Sistemas. 2012. 179 p.
24. CARVALLO, Juan Pablo, FRANCH, Xavier y QUER, Carmen, "Calidad de componentes software." *En: Calidad del Producto y Proceso Software.* 2010, p. 287-316.
25. CASTELLANOS, P. M. V D, GEXRENOF: Herramienta para la gestión de pruebas no funcionales basada en el estándar ISO/IEC 25000, Universidad de las Ciencias Informáticas (UCI), Facultad 5, Carretera a San Antonio de los Baños, Cuba, II Congreso Multidisciplinario de ciencias aplicadas, en Latinoamérica, 2014.
26. CASTILLO, Noel Humberto. Sistema de Soporte de Decisiones (DSS), en el Área de comercialización de productos, en el Supermercado La Matagalpa, Matagalpa 2013. Diss. Universidad Nacional Autónoma de Nicaragua, Managua, 2015.
27. CATOTA TOCA, Ximena Alexandra. "Análisis, diseño e implementación de un sistema informático para gestionar los trabajos de Help Desk del área de tecnología de la Cooperativa Codesarrollo basado en Itil y Silverlight." Tesis de Ingeniería de Sistemas. Quito, 2015.
28. CENDEJAS Valdés, José Luis *et al.* "Diseño del modelo integral colaborativo para el desarrollo ágil de software en las empresas de la zona centro-occidente en México." *En: Nova scientia.* 2015 vol. 7, no. 13, p. 133-148.
29. CIFUENTES, Diego. Modelos de la calidad de software, Transcripción de Modelos de la calidad de software, 2015.
30. CMMI-DEV, V1.3, "Mejora de los procesos para el desarrollo de mejores productos y servicios" Software Engineering Institute, 2010, CMU/SEI-2010-TR-033, pp: 1-562
31. COBIT: Caso de Estudio—Banco Supervielle S.A., Argentina, Available: <http://www.isaca.org/Knowledge-Center/cobit/Pages/COBIT-Caso-de-Estudio-Banco-Supervielle-SA-Argentina.aspx>
32. CORDOBA Bayas, Marco Raúl *et al.* "Estandar IEEE-I2207", Facultad de Ingeniería de sistemas, Escuela Politecnica Nacional, Quito Ecuador, 2012.

33. CÔTÉ, Marc Alexis *et al.* "Evolving a Corporate Software Quality Assessment Exercise: A Migration Path to ISO/IEC-9126." *Software Quality Professional*, 16.3, 2004, pp:4-17.
34. COVELLA, Guillermo Juan. *Medición y evaluación de calidad en uso de aplicaciones web.* Tesis Doctoral. Argentina: Facultad de Informática. Universidad Nacional de La Plata. 2005
35. CRUZ FIGUEROA, Enrique. *Aplicación de un modelo para la mejora y evaluación, de la empresa Santos CMI, basada en el modelo CMMI (capability maturity model integration).* Facultad De Ingeniería, Escuela De Sistemas, Pontificia Universidad Católica Del Ecuador, 2011. 94 p.
36. DROMEY, Geoff. *Cornering the chimera. [software quality]* In: *IEEE Software*. Jan, 1996. vol. 13, no 1, p. 33- 43
37. EELES, Peter. "Capturing architectural requirements." *IBM Rational developer works*. 2005.
38. ELMINIR, Hamdy K *et al.* *Application and Evaluation of the Personal Software Process.* In: *International Journal of Basic & Applied Sciences IJBAS-IJENS*. 2009. vol. 9 no. 10, p. 33-54
39. ESTAYNO, Marcelo *et al.* *Modelos y Métricas para evaluar Calidad de Software.* XI Workshop de Investigadores en Ciencias de la Computación. 2009.
40. FUERTES CASTRO, José Luis. "Modelo de calidad para el software orientado a objetos." Tesis de grado. Madrid - España: Dep: Facultad De Informática, Universidad Politécnica de Madrid, 2002.
41. gA certifica ISO por tercer año consecutivo y suma nueva norma. 2016 Available: http://www.grupoassa.com/es/press_releases/grupo-assa-earns-iso-certification-for-the-third-consecutive-year-adding-an-additional-norm/
42. GARZAS, Javier *et al.*, "Una aplicación de la Norma ISO/IEC 15504 para la evaluación por nivel de madurez de Pymes y pequeños equipos de desarrollo". In: *Kybele Consulting S.L.*, 2009.
43. GARZÁS, Javier *Cómo realizamos nuestra primera implantación real de la ISO 25000 (año 2007)*, Available: <http://www.javiergarzas.com/2014/04/iso-25000-implantacion.html>
44. GÓMEZ, Javier *et al.* "Enfoque metodológico para el diseño de interfaces durante el ciclo de vida de desarrollo de software." *En: Revista Gti*. 2014. vol. 12, no. 34.
45. GRAY, Lewis, "A Comparison of IEEE/EIA 12207, ISO/IEC 12207, J-STD-016, and MIL-STD-498 for Acquirers and Developers." *Abelia Corporation, Fairfax, Virginia, USA*, 1999.
46. GUERRERO, G. "Cobit 4.0", in *IT GOVERNANCE INSTITUTE*, ISBN 1-933284-37-4 COBIT 4.0, 2006, 207 p.
47. HERRERA H, Andres. "Bootstrap", *Procesos Software*, 2012.
48. HERRERA HERNÁNDEZ, Francisco y VARGAS ROLDÁN, Manuel. *Implementación de COBIT 4.0 en Scotiabank, Costa Rica*, Comments regarding the editorial content may be directed to Jennifer Hajigeorgiou, senior editorial manager, at, Costa Rica, 2012, Available: <http://www.isaca.org/Knowledge-Center/cobit/cobit-focus/Pages/Implementacion-de-COBIT-4-0-en-Scotiabank-Costa-Rica.aspx>
49. HERRERA, Jhon Fredy. *Informe de Asesoría para La Implementación De ITIL, Outsourcing S.A, Medellín Antioquia – Colombia*, 2010
50. HOLGUÍN BARRERA, Jhon Alexander. *Definición de un proceso de desarrollo de software con control de calidad del producto en una empresa pyme de la región.* Tesis de Maestría en Gestión y Desarrollo de Proyectos de Software. Manizales - Colombia: Universidad Autónoma de Manizales. 2015. 185 p.
51. HUMPHREY, Watts S. *Managing the software process.* Boston - Estados Unidos: Addison-Wesley Professional, 1989. 494 p. ISBN-10: 0201180952
52. HUMPHREY, Watts S. *Introduction to the personal software process.* Boston - Estados Unidos: Addison-Wesley Professional, 1996. ISBN-10: 0201548097
53. INFT certificó bajo la norma ISO 90003:2004, 2016 Available: <http://www.cessi.org.ar/ver-noticias-inft-certifico-bajo-la-norma-iso-90003-2004-249>.
54. INTEDYA, Aviable: <http://www.intedya.com/internacional/96/consultoria-calidad-en-el-software-iso-25000.html#submenhome>
55. ISO 9000-3 1997 Standard in Plain English, Guidelines for Applying ISO 9001 1994 to Computer Software.
56. ISO 9000-3, UNE-EN ISO 9000-3:1998
57. ISO/IEC 12207, UNE-EN ISO 12207:2008: *Systems and software engineering -- Software life cycle processes*
58. ISO/IEC FDIS 9126-1, *Information technology — Software product quality*
59. KHOSRAVI, Khashayar and GUÉHÉNEUC, Yann-Gaël. *A quality model for design patterns.* In *German Industry Standard*, 2004.
60. KYBELE CONSULTING. *Evaluación y Mejora de Procesos Software.* s.f. Available: <http://www.kybeleconsulting.com/servicios/evaluacion-y-mejora-de-procesos-software/>
61. MATKOVIC, Predrag and TUMBAS, Pere. *A Comparative Overview of the Evolution of Software Development Models.* In: *International Journal of Industrial Engineering and Management (IJEM)*. 2010. vol. 1, Issue 4, p. 163-172. ISSN: 2217-2661.
62. MCMURTREY, Mark. *A case study of the application of the systems development life cycle (sdic) in 21st century health care: Something old, something new?* In: *Journal of the Southern Association for Information Systems*. Jan., 2013. vol. 1, no. 1. DOI: 10.3998/jais.11880084.0001.103
63. MITRE HERNÁNDEZ, Hugo Armando. *Alineación de la gestión estratégica con la medición de productos y procesos para organizaciones de ingeniería del software.* Tesis M.S. Madrid: Departamento De Informática, Diss. Universidad Carlos III de 2010.
64. MOLINA, Hernán *et al.* *Sistema Colaborativo de Revisión para el soporte de información de contexto en el marco C-INCAMI.* Departamento de Computación, Facultad de Ingeniería, UNLPam, General Pico, La Pampa XIII Congreso Argentino de Ciencias de la Computación. 2007.
65. MONDRAGÓN CAMPOS, Oscar. *Integrando TSP y CMMI: Lo mejor de dos mundos.* In: *Software Guru*. 2011. vol. 50.
66. MONIRUZZAMAN, A B M and AKHTER HOSSAI, Syed. *Comparative Study on Agile software development methodologies.* In: *Global Journal of Computer Science and Technology*. 2013. vol. 13 no. 7. Version 1.
67. MORENO GARCÍA, María N. "Modelos de proceso del software" in *Departamento de Informática y Automática, Universidad de Salamanca Universidad de Salamanca*, 2010
68. MORENO, P.A. *Estándares Y Modelos De Calidad Del Software.* Escuela De Ciencias Básicas, Tecnología E Ingeniería, Programa Ingeniería De Sistemas, Colombia, 2012.
69. MOSZKOWITZ, A. E., *Modelos de excelencia en la gestión.* En: *Revista de Antiguos Alumnos del IEEM 2.3*, Uruguay, 1999-2010, pp: 26-30, ISSN 1510-4214.

70. NAKAMA Arakai, Erick Daniel. Implementación de 4 áreas de proceso del modelo de calidad CMMI: gestión y desarrollo de requerimientos, validación, y verificación. Facultad De Ingeniería, Carrera De Ingeniería De Sistemas De Información, Universidad Peruana De Ciencias Aplicadas, Lima, Perú, 2013.
71. NAVARRO, José Manuel y GARZÁS Javier. Experiencia en la implantación de CMMI-DEV v1. 2 en una micropyme con metodologías ágiles y software libre. REICIS. *En*: Revista Española de Innovación, Calidad e Ingeniería del Software. 2010. vol. 6, no. 1, p. 6-15.
72. OKTABA, Hanna. SQUARE: Modelo actualizado de las características de calidad, SG Buzz, Conocimiento Para crear Software Grandioso. UNAM, IPRC, COMPETISOFT. 2012
73. OLSINA SANTOS, Luis Antonio. Metodología cuantitativa para la evaluación y comparación de la calidad de sitios web. Tesis Ph.D Ciencias Exactas de la UNLP, Diss. Argentina: Facultad de Ciencias Exactas, Universidad Nacional de La Plata – Argentina, 1999, 255 p.
74. OLSINA, Luis and ROSSI, Gustavo. Measuring Web application quality with WebQEM. *In*: IEEE Multimedia. 2002. vol.9, no.4, p.20-29.
75. Organizaciones Certificadas en ISO 15504, Available: <http://www.iso15504.es/>
76. ORTEGA, M, et al. T.A model for software Product Quality with a Systemic Focus. *En* 4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2000 and The 6th International Conference on Information Systems, Analysis and Synthesis ISAS 2000. International Institute of Informatics and Systemics, Orlando, 2000. p. 395-401.
77. PEÑALVA, Mirta del Carmen. Un modelo de evaluación de la calidad de aplicaciones Web en e-government. Diss. Facultad de Informática, 2014.
78. PEREIRA, Betzabeth. Métricas de Calidad de Software. *En*: Calidad Del Software. 2012.
79. PETRIE, María Mercedes. et al. Modelo de Registro y Acreditación de Instituciones de Educación Superior basado en el Modelo CMMI., *In* Seventh LACCEI Latin American and Caribbean Conference for Engineering and Technology, 2009.
80. PRESSMAN, Roger. Ingeniería del Software. Un enfoque práctico. 7ta edición. España: Ed: McGraw-Hill Interamericana. 2010.
81. QUEVEDO Valencia, Ana. Implementación de una metodología de procesos para la mejora de TI en una empresa. 2009.
82. QUINTERO NAVARRO, Pilar Katherin y LOZANO PERDOMO, Yina. Guía de implementación ISO/IEC 9126: 2005 para la metodología V. Universidad Militar Nueva Granada, Facultad De Ingeniería Especialización En Gerencia De La Calidad, Colombia 2012.
83. RAMÍREZ AGUIRRE, Paola Andrea y RAMÍREZ ARIAS, Carolina. Estudio de las prácticas de calidad del software implementadas en las MIPYMES desarrolladoras de software de Pereira. 2010.
84. RAMOS, Fernando et al. "Implantación de CMMi nivel de madurez 2 en una PYME. *En*: Innovación, Calidad e Ingeniería del Software. 2010. vol.: 6, no. 3, p. 35-46.
85. RAWASHDEH, Adnan and MATALKAH, Bassem. A new software quality model for evaluating COTS components. *In*: Journal of Computer Science. 2006. vol. 2. Num. 4, pp: 373-381, 2006.
86. RODRÍGUEZ MONJE, Moisés. Calidad de procesos y productos de software", Calidad de Productos de Software- ISO/IEC 25000, 2016.
87. RODRÍGUEZ, Guillermo. Desarrollo e implementación de métricas para el análisis de las interacciones del Dispositivo Hipermedial Dinámico. Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas, CIFASIS (CONICET-UNR-UPCAM), Jornadas Argentinas de Informática, Rosario, Argentina, 2010
88. RODRÍGUEZ, Moisés, et al. Evaluation of Software Product Functional Suitability: A Case Study. *In*: Software Quality Professional Magazine. 2016. vol. 18, no. 3.
89. ROJAS CÓRSICO, Ivana Soledad. Normas COBIT. Ingeniería en Sistemas, Auditoría de Sistemas de Información, 2010.
90. ROMERA, Alvaro, et al. Improving the McCall herbage growth model. *In*: New Zealand Journal of Agricultural Research. 2009. vol. 52, no 4, p. 477-494.
91. ROMERO, Felipe y BLANCO, Mónica. Mejoramiento de procesos de software en pequeñas empresas: Algunas experiencias en el caso colombiano. *En*: Paradigma en construcción de software 2. 2008 p. 1-6.
92. SÁNCHEZ SECO, José Manuel. Métricas de gestión: Caso práctico: Métricas para un proyecto de implantación de un core bancario. Ingeniería Técnica Informática de Gestión, universidad Abierta de Cataluña, España, 2015.
93. SÁNCHEZ, Luis et al. "Proceso de Reificación de las Competencias Generales y Específicas para el Grado en Ingeniería Informática y Definición de un Plan de Métricas de Evaluación de dichas Competencias." XVII Jornadas de Enseñanza Universitaria de la Informática (JENUII) 2011. p. 51-58.
94. SANDIA SALDIVIA, Beatriz Elena. Modelo Organizacional y de Gestión de Formación Flexible Basada en Entornos Tecnológicos para la Universidad de Los Andes. Trabajo de investigación conducente al Diploma de Estudios Avanzados (DEA). España: Universidad de Las Islas Baleares. 2007.
95. SCALONE, Fernanda. Estudio comparativo de los modelos y estándares de calidad del software. Tesis Ingeniería de Calidad. Buenos Aires: Universidad Tecnológica Nacional Regional de Buenos Aires. 2006. 488 p.
96. STANDAR COORDINATING COMMITTEE OF THE COMPUTER SOCIETY OF THE IEEE. IEEE Standard Glossary of Software Engineering Terminology. Usa: The Institute of Electrical and Electronics Engineers. 1990. 84 p.
97. SOLEDISPA, CHÁVEZ, et al. Aplicación de psp (personal software process) para el desarrollo de un sistema administrador de códigos de barra a partir de la evaluación de procesos de reingeniería. Ecuador: Escuela Superior Politécnica del Litoral. 2010.
98. SOLEMON, Badariah. Evaluation Criteria for Requirements Engineering Process Improvement Models and Frameworks. Computational Methods in Science and Engineering, Department of Software Engineering Universiti Tenaga Nasional Malaysia, 2013.
99. SOTO DURAN, Darío E, et al. INTRODUCIENDO PSP (PROCESOS PERSONAL DE SOFTWARE) EN EL AULA. *In*: Revista Colombiana de Tecnologías de Avanzada. 2010. vol. 2, no. 16. p. 1-5.
100. SOTO PEÑA, José Rosario, et al. Actividad 2.2: Cuadro Comparativo de Modelos para evaluar la calidad del Software (módulo: evaluación de la calidad de la tecnología educativa)". *En*: ISO 69. 2015
101. STASINOS, Dimitris. Applying Gilb's method of inspections into telecommunications software development, European Software Process Improvement, 2012
102. STRUB, Ana María, et al. Métricas estadísticas asociadas al proceso de desarrollo y mejora continua de software. *En*: Congreso Virtual sobre Tecnología, Educación y Sociedad. (2: 22-25, enero: México) Memorias. México: Centro de Estudios e Investigaciones para el Desarrollo Docente. CENID A.C, 2015. p. 1-26.

103. TAHHÁNN, Mirla *et al.* Modelo para estimar la calidad en los Sistemas de Información Geográfica. AMCIS 2006 Proceedings, 2006, p. 512.
104. TAHHÁNN, Mirla *et al.* Modelo para estimar la calidad en los Sistemas de Información Geográfica. AMCIS 2006 Proceedings. 2006. 512 p.
105. TOPALOVIC, Drago. "ITIL and ISO/IEC 20000 History: Parallel Worlds", in 20000 academy. May 2013.
106. VALVERDE, Francisco. ECOPETROL S.A. Un modelo de éxito en la implementación de COBIT, Escuela Politécnica Nacional, Facultad de Sistemas, Maestría en Gestión de las Comunicaciones y TI. Colombia, 2014.
107. VAN BON, Jean *et al.*, "ISO/IEC 20000 Una Introducción", in ITSM Library, ed VHP, España. 2008, 261 p.
108. VARGAS, Fabio y SOTO DURAN, Dario. "INTRODUCIENDO PSP (PROCESOS PERSONAL DE SOFTWARE) EN EL AULA. *En:* Revista Colombiana de Tecnologías de Avanzada. 2010. vol. 2 , no. 16.
109. VEGA HERNÁNDEZ, José Ramón *et al.* Modelando con UML el proceso de evaluación de productos de software utilizando el enfoque GQM. *En:* Revista Cubana de Ciencias Informáticas. 2015. vol. 9 p. 76-84.
110. VELANDÍA BLANCO, Jorge A. MODELO DE CALIDAD ISO 25000 (SQuaRE), SERVICIO NACIONAL DE APRENDIZAJE SENA, Colombia, 2011
111. VELAZCO, Alvaro. Modelo en espiral Introducción Boehm", Junio 2016. [Online]. Available: https://www.academia.edu/7972897/Modelo_Espiral_INTRODUCCION_Boehm
112. VILLALBA LLAMOSAS, Ricardo y ESTRADA DÍAZ, Lilia Yarley. Aprendizaje y Aplicación del Modelo Cmmi-Dev en Pymes de Software Colombianas. La experiencia RCCS. *En:* Revista GTI. 2011. vol. 9, no. 24, p. 57-76.
113. VILLARROEL, Rodolfo y VISCONTI, Marcello. Un modelo de madurez para el proceso de gestión de configuración de software. in VII Encuentro Chileno de Computación. Talca, Chile, 1999.
114. WEBB, David, *et al.* Using the TSP on the TaskView Project. Cross-Talk. 1999. vol. 12, no. 2, p. 3-1.
115. WECKMAN, Jerry; COLVIN, Jedd T.; GASKINS, Robert; MACKULAK, Gerald T. Application of simulation and the Boehm spiral model to 300-mm logistics system risk reduction. *In:* Simulation Conference Proceedings. (5-8, december: Phoenix, AZ, USA) Memories. USA: IEEE *Xplore*, 1999.
116. YARCE, David y STEEVEN, Roy. Uso de tecnologías y metodologías de desarrollo manejados en Pragma SA, para la construcción de portales web. Tesis Doctoral. Antioquia: Corporación Universitaria Lasallista. Facultad De Ingenierías Ingeniería Informática, 2012. 46 p.