

MODELO DE CALIDAD PARA EVALUAR EL SOFTWARE DESARROLLADO EN EL CENTRO DE INVESTIGACIÓN APLICADA Y DESARROLLO EN TECNOLOGÍAS DE INFORMACIÓN CIADTI

A QUALITY MODEL TO EVALUATE THE SOFTWARE DEVELOPED IN THE CENTER OF APPLIED RESEARCH AND DEVELOPMENT IN INFORMATION TECHNOLOGIES CIADTI

Y.S. Estévez, L.A. Esteban
Universidad de Pamplona, Pamplona, Colombia

RESUMEN

El presente artículo presenta un modelo de calidad de producto software para el Centro de Investigación Aplicada y Desarrollo en Tecnologías de Información, CIADTI, de la Universidad de Pamplona, que a diferencia de los modelos generales existentes pretende medir la calidad general de un producto software mediante la valoración de factores y características aplicadas a productos parciales dentro del proceso de desarrollo de software, es decir, a artefactos que se consideran relevantes dentro de la organización y a los cuales se les puede valorar mediante factores que fueron inicialmente pensados para productos finales del proceso de desarrollo. Se referencian antecedentes investigativos sobre modelos de calidad en el desarrollo de software, se realizó un diagnóstico al actual proceso de desarrollo del producto del CIADTI a partir de la norma ISO 9126. También se aplicaron instrumentos para la recolección de información a los desarrolladores y coordinadores del proceso. A partir de allí se analizaron las categorías y atributos para la formulación del modelo, se formula el modelo final y se aplica a los artefactos existentes. Finalmente, se analizan los resultados y el modelo se valida frente a un grupo de expertos para una mayor objetividad.

PALABRAS CLAVE: Software, Modelo Calidad, Artefactos.

ABSTRACT

It made a bibliographical revision about the shoulder's tendinopathy (tendinitis) focused in the general service sector (cleaning), in relationship between the causes that affect to the worker and their activities in the work developed, with the purpose of to establish the main causes that produce this illness to settle down and to provide technical and treatment that are in the literature and the application doesn't require expenses and additional time.

KEYWORDS: efforts, prevention, postures, repetitive movements, shoulder pain.

I. INTRODUCCIÓN

Durante las últimas décadas, desde el surgimiento de las tecnologías de software, se ha pretendido evaluar la calidad de los productos propios teniendo por finalidad mejorar cada una de sus cualidades; lo anterior con el propósito de desarrollar productos software cada vez más especializados que satisfagan las necesidades de usuarios exigentes y conocedores del tema.

Es así como las empresas productoras de software, en la actualidad, tienen trazada por meta establecer un conjunto de características medibles y específicas a los productos que comercializan a fin de ser más profesionales y competitivos en el área, y de evitar errores en los software, que repre-

senten pérdidas económicas o que exijan mayor trabajo de soporte por alguna deficiencia.

Ahora bien, de acuerdo a (Incera G, 2005) respecto de las mediciones de calidad del software “[estas] permiten mejorar sus procesos; ayudar en la planificación, darle seguimiento a los mismos y controlar determinados proyectos”, por lo que se ha estimado que las medidas de calidad en el desarrollo del software debe ser “confiable, mantenible y flexible” (Solarte et al., 2009) con el fin de reducir los costos de mantenimiento y perfeccionamiento durante el tiempo de uso y el ciclo de vida del mismo.

La Universidad de Pamplona tiene por política y objetivos de calidad el desarrollo sistemático de productos y servicios de mejor nivel y el cumplimiento de las necesidades y expectativas de los clientes (Unipamplona, 2007), para esto insta a los profesionales, estudiantes y miembros del Centro de Investigación Aplicada y Desarrollo en Tecnologías de Información (CIADTI) a propiciar escenarios y herramientas útiles para el mejoramiento de los productos que son desarrollados por el Alma Mater.

Durante los últimos once años la Universidad de Pamplona ha consolidado en el mercado de las Entidades Educativas tecnología de software o aplicaciones misionales, que si bien han alcanzado un alto nivel de competitividad y liderazgo debido su cultura organizacional, aún carecen de elementos fundamentales en el proceso de control de calidad de sus productos; la inexistencia de un “modelo de calidad para el desarrollo del software” no permite, como se mencionó previamente, favorecer la constitución de un producto de alto nivel, sostenible y rentable.

Establecer un modelo de calidad del software para los productos de la Universidad de Pamplona desarrollados por el CIADTI resulta favorable desde diversos puntos de vista, así lo consideran (López A., Cabrera C., Valencia L., 2008) quienes afirman que “los productos o servicios que ostentan certificados de calidad son preferidos por los compradores porque transmiten seguridad y confianza”. De la mano de la afirmación anterior (Consuegra D., Shelton R., García A., 2005) señalan como ventaja del proceso que “al utilizarse la misma métrica en todos los proyectos se pueden realizar comparaciones entre ellos y, mejor aún, si es una métrica estándar en la industria podría compararse esta contra otras”.

Es así como las empresas productoras de software, en la actualidad, tienen trazada por meta establecer un conjunto de características medibles y específicas a los productos que comercializan a fin de ser más profesionales y competitivos en el área, y de evitar errores en los software, que representen pérdidas económicas o que exijan mayor trabajo de soporte por alguna deficiencia.

(Monsalve, 2002, p.4) es posible administrar la calidad “asegurando minimizar las diferencias entre los recursos presupuestados y los recursos realmente utilizados en las distintas etapas” lo cual se transforma en un verdadero beneficio tanto para la institución educativa como para sus clientes.

El modelo aquí descrito surge como resultado de la elaboración del estado del arte sobre los modelos de calidad del producto existentes, y bajo la hipótesis de que los factores definidos por estos son aplicables no solo al producto final sino a artefactos puntuales generados por dicho proceso de desarrollo.

II. ANTECEDENTES INVESTIGATIVOS

Para el desarrollo del presente estudio se han indagado y puesto a consideración una serie de documentos desarrollados en la misma línea investigativa del actual. Una vez observados y analizados se han seleccionado, bajo el criterio del autor, algunos de los que representan una influencia importante, un aporte académico o una guía que dé luces a la investigación que se lleva a cabo, y que por lo tanto se constituyen en el Estado del Arte de la evaluación de la calidad del software. Los temas seleccionados se subdividieron en tres grandes tópicos a saber: Calidad del software, Evaluación de la calidad del software y Modelos para la evaluación de la calidad del software.

A. CALIDAD DEL SOFTWARE

Ardila et. al. (2013) abordan el tema de las métricas de gestión con elementos cuantitativos que consideran óptimas para el mejoramiento de los procesos de desarrollo de software, ya que permiten disminuir la variabilidad del producto, alejándose en su enfoque de las grandes compañías y centrándose en su aplicación en las pequeñas organizaciones independientes dedicadas a dicho desarrollo.

Para ello se revisan los modelos: Integración de modelos de madurez de capacidades o Capability maturity model integration (CMMI-DEV) (e ISO/IEC 15504(también conocido como Software Process Improvement Capability Determination, abreviado SPICE, en español, «Determinación de la Capacidad de Mejora del Proceso de Software» es un modelo para la mejora, evaluación de los procesos de desarrollo, mantenimiento de sistemas de información y productos de software), que incorporan elementos de evaluación cuantitativos, pero se determina que están optimizados para grandes organizaciones, y por lo tanto no resultan aplicables en pequeñas compañías.

Se revisan igualmente modelos enfocados en estas últimas, como MOPROSOFT (Modelo de Procesos para la Industria del Software. Modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software) y COMPETISOFT (Es un proyecto financiado por CYTED, programa internacional de cooperación científica y tecnológica multilateral, de ámbito iberoamericano que tiene como propósito incrementar el nivel de competitividad de las PyME iberoamericanas productoras de software mediante la creación y difusión de un marco metodológico común que, ajustado a sus necesidades específicas, llegue a ser la base sobre la que se pueda establecer un mecanismo de evaluación y certificación de la industria del software reconocido en toda Iberoamérica.), pero estos no incluyen elementos cuantitativos (Aguirre et al., 2010).

Los investigadores realizan una revisión sistemática del material y de las metodologías disponibles, buscando metodologías de gestión cuantitativa de procesos dedicadas a pequeñas empresas desarrolladoras de software que indiquen cuáles prácticas efectuar y cómo ejecutarlas. Sin embargo, llegan a la conclusión de que no existe una propuesta especialmente diseñada para pequeñas organizaciones que incluya elementos cuantitativos.

Otálora et. al. (2011) reconociendo la importancia de la aplicación de las herramientas de gestión de calidad en los procesos de desarrollo de software, realizan un sondeo entre las diversas compañías dedicadas a dicho campo dentro del departamento de Boyacá (Colombia) llegando a dos conclusiones iniciales: la casi totalidad de dichas organizaciones entran dentro de la categoría de pequeñas y medianas empresas, y ninguna de las mismas está implementando una metodología de gestión de calidad certificada.

Por lo anterior, los autores deciden colaborar con dichas compañías en la búsqueda de una herramienta de control de calidad accesible, tomando como base la Norma ISO/IEC 15504, pero enfocándose en la pequeña y mediana industria. Estableciendo como parámetros para el desarrollo de dicha metodología los siguientes: acceso libre, bajo costo, adecuados estándares de usabilidad y que pueda ser implementada a través de la red de internet (Cloud Computing. La computación en la nube, concepto conocido también bajo los términos servicios en la nube, informática en la nube, nube de cómputo o nube de conceptos, permite ofrecer servicios de computación a través de Internet) (Torres, 2012).

El fin último es ayudar a las compañías a mejorar la calidad de sus procesos de producción, y por consiguiente de los productos de software elaborados, con el fin de aumentar su competitividad y su participación en el mercado de software.

Mendoza et. al. (2002) del laboratorio de Investigación en Sistemas de Información de la Universidad Simón Bolívar de Caracas (Venezuela) es el gestor del Modelo Sistemático de Calidad MOSCA (2002) creado para ayudar a garantizar la calidad de los procesos de desarrollo de software en la región, haciéndolos más competitivos a nivel internacional.

Buscando facilitar y expandir la aplicación de dicho modelo, los investigadores han buscado desarrollar de igual forma un algoritmo, un conjunto sencillo de pasos pre-establecidos que permita su implementación. Dicho algoritmo tiene como fin guiar el proceso de evaluación implementado por el MOSCA, buscando garantizar la calidad sistémica al igual que una adecuada complementación entre las características propias de la organización desarrolladora de software y las necesidades del usuario.

Como parte del proceso de presentación del algoritmo se describe su aplicación dentro de dos empresas venezolanas de desarrollo de software, en la búsqueda de demostrar cómo el dueto formado por el MOSCA y el algoritmo de implementación desarrollado contribuye a mejorar la calidad sistémica, analizando aspectos relativos tanto al proceso como al producto.

Gutiérrez et.al. (2009) conscientes de la necesidad de implementar metodologías de control de calidad en las diversas compañías nacionales dedicadas al software, para hacerlas más competitivas a nivel internacional, hicieron un sondeo que abarcó 114 diferentes compañías mexicanas del ramo, buscando evaluar el grado de la implementación dentro de las mismas metodologías.

A partir de los resultados obtenidos los investigadores establecieron tres postulados: primero, que las compañías de desarrollo de software dentro del territorio mexicano tienen muy poco conocimiento, y menor implementación, de los modelos de calidad de software. Segundo, dichas empresas desconocen los métodos específicos óptimos para evaluar sus productos. Y tercero, se evidencia la necesidad de un modelo integral que permita evaluar, de manera simultánea, la calidad del proceso y la del producto.

Chávez et. al. (2009) proponen observar la relación de costos/beneficios que se da a partir de la implementación de un modelo que mida la calidad de un software, considerando que para el desarrollo de un producto de software deben de intervenir dos equipos: el primero, el que generará el producto final basado en un estándar de calidad y el segundo, el que corroborará que ese estándar se encuentre bien aplicado.

La autora se cuestiona sobre la eficacia y rentabilidad de dicho proceso y señala que los productos de software generados mediante esta vía son productos caros y, de acuerdo a sus investigaciones, la calidad cuesta, tanto para quien la adquiere como para quien la aplica. La buena calidad cuesta mucho llevarla a cabo, se invierte tiempo, dinero y mucho esfuerzo, pero es mayor el costo de la mala calidad, del mal servicio y de no cumplir con los requerimientos especificados por el cliente.

Es por lo anterior que se consolida un análisis de diversos modelos identificando cuál resulta más favorable, situación que orienta la actual investigación por cuanto el modelo inicial para la evaluación que se llevará a cabo sobre la calidad de los productos desarrollados actualmente en el CIADTI se tomará como guía modelo ISO 9126.

B. EVALUACIÓN DE LA CALIDAD DEL SOFTWARE

Estayno et al. (2009) buscando incrementar la competitividad de las Pequeñas y Medianas empresa (PYMES) dedicadas a la creación, mantenimiento y diseño de productos de software dentro de la región del Nordeste argentino, investigadores de la Universidad Nacional del Nordeste, en conjunto con otras universidades de la región se dedicaron a evaluar la situación actual de la industria del software dentro de la mencionada región administrativa.

Sin embargo, el proyecto tiene como fin no sólo la simple evaluación del estado de las cosas, sino que además busca impulsar en dichas compañías la implementación de herramientas y metodologías de calidad de software, a la vez que busca crear un plan de acción sistematizado y de orden regional, que incluya tanto a las mencionadas empresas como a las universidades y a las autoridades públicas de la región, plan que a su vez tiene como fin el desarrollo de recursos humanos especializados dentro del área de gestión de calidad de software y la implementación de unas políticas públicas orientadas al impulso de la industria del software local.

Consuegra et al. (2005) acopian el estudio de la problemática de la Gestión de la Calidad concerniente a la definición de métricas de tamaño en el ámbito de la Ingeniería del Software, mostrando la necesidad de contar con métodos y sistemas de medición y evaluación, teniendo en cuenta ante todo la satisfacción del cliente y planteando además un cambio en el proceso de recolección y análisis de indicadores de productividad y calidad de una empresa de software. A través de una investigación cuantitativa, los autores tienen por finalidad establecer una métrica ajustada al tamaño, que tenga por principio orientador la satisfacción del cliente; aunque a la fecha el estudio no ha culminado, los autores han consolidado un marco teórico que se adecua como precedente de la actual investigación, por cuanto tiene su misma finalidad.

Dicha investigación se lleva a cabo en el Centro de Estudios de Informática, Universidad Central de Las Villas, y en la medida de su avance sus resultados se han puesto a prueba en equipos de desarrollo de manera que puedan lograr su correcta validación. En sí el trabajo propuesto por los autores es mejorar el proceso de desarrollo de software, representando así un mayor poder de abarcamiento de pedidos y acometimiento en fecha y con calidad, lo cual derivará en el aceleramiento de la informatización y el mejoramiento del resto de las esferas económica, social, política y académica de la institución objeto de estudio.

C. MODELOS PARA LA EVALUACIÓN DE LA CALIDAD DEL SOFTWARE

Ruiz et al. (2006) presentan un modelo computacional basado en la lógica borrosa para el cálculo de la usabilidad del software, de acuerdo con una serie de criterios de calidad

basados en la Norma ISO/IECE9126-1:2.001. El modelo propuesto por los autores se origina como respuesta a la necesidad de establecer criterios sólidos de evaluación del software educativo en virtud del incremento en la generación del mismo. Si bien la necesidad surge del campo específico del software educativo, la propuesta tiene lugar para la evaluación del software en general. Para ello se parte de la identificación y evaluación de cinco atributos básicos por parte del usuario final como son: Comprensibilidad, Facilidad de Aprendizaje, Atractividad, Operabilidad y Conformidad.

El modelo desarrollado partió de la subjetividad de varios atributos de la norma señalada y se utilizaron conceptos propios de la lógica borrosa, y en especial el modelo borroso del tipo Takagi Sugeno Kang (1985). El modelo Takagi-Sugeno-Kang (TSK), (Celikyilmaz et al., 2009) fue presentado por primera vez por T. Takagi y Prof. M. Sugeno en 1985. Otro estudiante de Sugeno, K. T. Kang continuó trabajando en las aplicaciones e identificación del modelo. El modelo TSK fue más utilizado en los 90's por la comunidad de investigadores y la industria. Una de las principales ventajas del modelo TSK es que puede aproximar una función utilizando pocas reglas, el cual a partir de una serie de valores de pertenencia permite el cálculo de un valor para una variable de salida o variable a controlar de manera analítica, a partir de la aplicación de criterios estandarizados en la evaluación de la calidad del software. Una vez obtenido el modelo se hicieron una serie de encuestas a cada uno de los usuarios finales de una determinada aplicación y de acuerdo con un valor de usabilidad preestablecido se estimaron una serie de parámetros que permitieron identificar, por parte de ese usuario, los atributos en los cuales la empresa desarrolladora se debería ajustar a las normas de la ingeniería del software y de esta manera mejorar la aplicación de usuario final, disminuyendo así el tiempo de desarrollo por parte de la empresa encargada del mismo.

Solarte et al. (2009) reconociendo la importancia de la calidad de software en las empresas actualmente, realizan un análisis sobre los productos a partir de la Norma ISO/IEC, la integración del modelo de maduración de la capacidad CMMI y (IT MARK 2004) es un servicio de certificación en procesos técnicos y de negocio creado en el 2004 por el European Software Institute (SEI), enfocado hacia la mejora de procesos en PYMES del sector de tecnologías de información. IT Mark provee una evaluación de calidad y madurez desde tres puntos de vista complementarios, modelo de calidad para PYMES para determinar los beneficios e inconvenientes que presenta para el desarrollo de Software con calidad.

Los investigadores analizan cada una de los modelos de calidad, estableciendo algunas conclusiones sobre sus fun-

ciones en torno a la calidad; del primero de ellos, CMMI, los autores señalan que necesita fundamentarse alrededor de su negocio y de sus objetivos, y que este modelo no es una certificación pero ayuda a encontrar la mejor manera de trabajar. Adicionalmente, consideran que el modelo no detalla procesos, define el qué pero no el cómo, y que por lo tanto no resulta adecuado si no se refuerza y usa apropiadamente, afirmando finalmente que tendrá éxito si es propiedad de los grupos que lo utilizan.

Sobre el IT MARK señalan los investigadores que es el primer modelo de calidad internacional diseñado específicamente para las pequeñas y medianas empresas del sector TIC, y mencionan que, generalmente, las empresas cuentan e incorporan a la organización una diversidad de herramientas que se encuentran disponibles en el mercado para cumplir con las diferentes actividades. Este conjunto de herramientas, de acuerdo a lo que concluyen los autores, habitualmente permanecen desvinculadas entre sí generando dificultades para gestionarlas, mantenerlas y controlarlas.

Finalmente, respecto del modelo ISO/ IEC 15504 señalan que es un estándar SPICE que trabaja con procesos de ingeniería, cliente-servidor, gestión de la organización y del soporte, y que fue diseñado por la alta competitividad del mercado de desarrollo de software, la difícil tarea de identificar los riesgos, cumplir con el calendario, controlar los costos y mejorar la eficiencia y calidad; además realiza una evaluación de los procesos de software relacionados con el conocimiento y la experiencia aportados por otros modelos, situación que permite creer que podría ser un modelo adecuado por el proceso de que lleva a cabo en el CIADTI.

Incera (2005) se plantea una lista de medidas que permitan organizar el desarrollo de software configurando las mediciones que permiten mejorar sus procesos, ayudar en la planificación, darle seguimiento a los mismos y controlar determinados proyectos en aras de lograr una mejor calidad, siendo esta una base importante para una gestión efectiva por parte del equipo de desarrollo.

De acuerdo a la investigadora, las medidas de procesos pueden ser usadas para la mejora del desarrollo y mantenimiento del software. Tal es el caso de la efectividad de la eliminación de defectos durante el desarrollo, el patrón de aparición de defectos de prueba y el tiempo de respuesta de los procesos mejorados. Por otra parte, las medidas de proyectos describen las características y ejecución de este. Se incluyen dentro de esta categoría el número de desarrolladores de software, la estructura del personal a través del ciclo de vida del proyecto, el costo, la planificación y la productividad. Proponer estas herramientas tuvo por finalidad por parte de la autora identificar un método que midiese la calidad del

software en las empresa (situación similar a la desarrollada en la actual investigación), tomando como medidor principal el SPC (Control Estadístico de Procesos), sobre el cual concluyó que no encaja completamente en el dominio de la ingeniería de software. Sin embargo, se afirma en (Incera, 2005), si se implementa de forma adecuada, una gráfica de control puede actuar como auditor demostrando no conformidades existentes en procesos/productos. Por lo tanto, proporciona la oportunidad de detectar problemas y mejorar los procesos de software.

Rolón et. al. (2010) desarrollan de un conjunto de métricas para la evaluación de modelos conceptuales de procesos de negocio. La propuesta supone la adaptación y extensión del marco FMESP (Framework for the Modeling and Evaluation of Software Processes). De acuerdo a los autores, esta adaptación se puede realizar gracias a las similitudes existentes entre ambos tipos de procesos (software y de negocio), de manera que el conjunto de métricas incluidas en FMESP se ha podido adoptar como punto de partida. Parte del proceso de adaptación consiste en tener en cuenta que los lenguajes de modelado de procesos y los meta modelos subyacentes son diferentes en ambos casos. Mientras que en FMESP los procesos software se modelan utilizando el metamodelo Software Process Engineering Metamodel (SPEM) propuesto por la Object Management Group (OMG), en la nueva propuesta para procesos de negocio se utiliza la notación Business Process Modeling Notation (BPMN) en español Notación para el Modelado de Procesos de Negocio (Earls, 2011), que básicamente consiste en utilizar una representación gráfica de tipo diagramas de flujo. Adicionalmente, el conjunto de métricas de FMESP ha debido ser extendido porque los modelos de procesos de negocio representados en BPMN incluyen bastantes aspectos de interés en este dominio que no son considerados en el caso de los procesos software modelados con SPEM.

Dávila et. al. (2005) desarrollan una metodología para la evaluación y el análisis de los atributos de calidad en los productos de software para Internet. A partir de la teoría de modelación estadística se llevó a cabo un proceso que permitiese la evaluación de los atributos de la calidad de los productos. Para el cumplimiento de los objetivos propuestos en el estudio los autores se ajustaron al orden de cinco fases a saber: (i) evaluación del atributo de calidad en el sistema ideal, (ii) evaluación del atributo de calidad en el sistema real, (iii) implementación del sistema de mejora, (iv) evaluación del sistema final y (v) análisis de los resultados y conclusiones. Los primeros dos pasos de dicha metodología se desarrollaron de forma iterativa hasta que los atributos de la calidad estuviesen cercanos al ideal como se requería.

La evaluación del comportamiento de los atributos se realizaron con métricas de software y sus resultados fueron descritos mediante histogramas. Y aunque de la metodología propuesta los estudiosos ejecutaron solo la primera fase, con esta lograron establecer que la modelación estadística pueden predecir los atributos de la calidad del software. Los autores recomiendan incluir un proceso de mejora a través de Personal Software Process (PSP) PSP es una alternativa dirigida a los ingenieros de sistemas, que les permite mejorar la forma en la que construyen software. Considera aspectos como la planeación, calidad, estimación de costos y productividad a fin de mejorar los objetivos de la calidad.

Díaz et. al. (2008) se propusieron como objetivo del estudio presentar el desarrollo de un modelo de calidad del software educativo con sus respectivas métricas siguiendo un enfoque sistémico. Para lo anterior partieron del Modelo Sistémico de Calidad de Software (MOSCA) elaborado por Laboratorio de Investigación en Sistemas de Información (LISI-USB), ampliándolo de acuerdo a los requerimientos particulares de calidad del software educativo, tomando en cuenta no solo los aspectos técnicos del producto sino el diseño pedagógico y los materiales de soporte didáctico.

Finalmente, a través de un caso de estudio real, los investigadores desarrollaron la aplicación del modelo ampliado MOSCA analizando su desempeño y ofreciendo un conjunto de métricas que permiten evaluaciones posteriores. El resultado principal fue un modelo en versión prototipo que permite medir la calidad para software educativo.

III. REFERENTES TEÓRICOS

Dada la casi ilimitada variedad de campos de aplicación de software, y las muy diferentes necesidades y requerimientos que deben ser satisfechos en cada uno de ellos, no existe una herramienta o procedimiento unificado que permita evaluar la calidad de software, sino que existen una gran variedad de modelos, de métricas de calidad, cada una con un enfoque diferenciado y cuyo criterio de aplicación corresponde al grupo de parámetros que se desee sea evaluado. Es por lo tanto más que conveniente revisar las métricas de calidad de software más representativas y el campo óptimo de aplicación de cada una de ellas.

A. MODELO MCCALL

Este modelo, desarrollado originalmente por James “Jim” A. McCall para la Fuerza Aérea de los Estados Unidos en el año de 1977, fue una de las primeras métricas de calidad implantadas con éxito, al punto de mantenerse plenamente vigente al día de hoy.

La intención de McCall era “cerrar la brecha entre los usuarios y los desarrolladores” (McCall, 1978), y para ello intentó plasmar en su esquema la visión del usuario al mismo tiempo que las prioridades del desarrollador. Para ello identificó tres perspectivas principales que a su juicio caracterizaban la calidad de un producto de software. Dichas perspectivas son: Revisión del producto (capacidad de cambio), Transición del producto (capacidad de adaptación) y Operaciones del producto (características básicas del producto).

Cada una de estas perspectivas está dividida en una serie de factores de calidad que la configuran, de esta forma los factores de Revisión son: facilidad de prueba, flexibilidad y facilidad de mantenimiento. Los factores de Transición son: portabilidad, reusabilidad e interoperabilidad. Y finalmente los factores de Operación: exactitud, confiabilidad, eficiencia, integridad y facilidad de uso. Cada uno de los factores, a su vez, y para facilitar su evaluación, abarca un grupo de criterios que lo determinan. Así, por ejemplo, el factor de facilidad de uso, dentro de la perspectiva de Operaciones del producto, está conformado por los criterios de facilidad de operación, facilidad de comunicación, facilidad de aprendizaje y formación. El modelo McCall tiene una connotación de tipo fuertemente industrial, centrándose en el producto y sus características, sobre todo las externas aunque sin olvidar del todo las internas, y prestando muy poca atención al proceso.

B. MODELO FURPS+

El modelo FURPS+ (Hewlett Packard, 1987) fue propuesto por el Ingeniero Robert Grady durante su trabajo en Hewlett Packard. Su nombre es un acrónimo de las cinco características, una funcional y las otras no-funcionales, según las cuales se evalúa la calidad de software en esta métrica: Functionality (funcionalidad), Usability (facilidad de uso), Reliability (confiabilidad), Performance (rendimiento) y Supportability (facilidad de soporte).

El + por su parte hace referencia un grupo de requerimientos adicionales, no presentes en el esquema original, que se han ido agregando con el tiempo y que usualmente son restricciones de tipo externo, como las restricciones físicas (hardware) o legales. Al igual que con el modelo anterior, cada una de estas características engloba un conjunto de factores que deben ser considerados. Así, por ejemplo, la facilidad de soporte abarca la adaptabilidad, la extensibilidad, la facilidad de mantenimiento, la compatibilidad y la facilidad de configuración.



Figura 1. Modelo de Métrica FURPS. Fuente: Al-Qutaish, R. (2010).

C. MODELO DROMEY

Este modelo fue presentado en el año de 1995 por R. Geoff Dromey (Dromey, 1995). Al igual que el modelo de McCall (McCall, 1977) se encuentra centrado en el producto, pero a diferencia de este considera que los criterios de evaluación deben ser dinámicos para adaptarse a la diferente naturaleza de cada producto (Al-Qutaish, 2010). Así, diseña un esquema basado en atributos y sub-atributos, dando múltiples ejemplos/opciones pero sin priorizarlas, y corresponde a cada diseñador, a la hora de evaluar su producto de software, determinar cuáles son los atributos (y los correspondientes sub atributos) que tendrá en cuenta a la hora de realizar la evaluación de calidad del software.

Se podría decir que la principal fortaleza de este modelo es al mismo tiempo su principal debilidad: la excesiva flexibilidad. Si bien facilita la evaluación de muy diferentes tipos de productos de software exige a su vez un mayor compromiso de parte del programador a la hora de evaluar, ya que debe ser él quien determine los atributos que serán tomados en cuenta.



Figura 2. Modelo de Métrica Dromey Fuente: Al-Qutaish, (2010)

D. ISO 9000

Bajo la denominación ISO 9000, la International Organization for Standardization (ISO), ha agrupado una serie de normas que versan sobre calidad y gestión de calidad, y que pueden ser aplicadas a cualquier tipo de empresa, organización o de actividad que se encuentre orientada a la producción de bienes y servicios. En este sentido, es necesario decir que más que una métrica de calidad de software se refiere a una normatividad estandarizada para garantizar la calidad de los procesos de desarrollo de cualquier tipo de producto. (ISO, 1994)

En este sentido, si bien son normas enfocadas a garantizar principalmente la calidad de los procesos de producción, su aplicación al campo del software no sólo resulta posible sino conveniente, ya que permite garantizar unos adecuados niveles de calidad en la etapa del proceso de producción, a la vez que permite monitorear -y mejorar- dichos procesos.



Figura 3. Proceso de Calidad ISO 9000. Fuente: (ISO, 2000)

E. ISO/IEC 9126

En el año 1991, la ISO publicó su primer compendio de términos y normas referentes, de manera específica, a la calidad de software. Dicho compendio, entre los años de 2001 y 2004, fue revisado y expandido de tal forma que en la actualidad es la norma ISO 9126 este estándar proviene desde el modelo establecido en 1977 por McCall y sus colegas, los cuales propusieron un modelo para especificar la calidad del software. Está conformada por un estándar internacional y por tres reportes técnicos. Dicho estándar establece un modelo de calidad dividido en dos partes (que algunos autores consideran que son realmente cuatro, subdividiendo la primera etapa) donde la primera hace referencia a la calidad interna y externa del producto, y la segunda a la calidad del producto sometido a condiciones reales de uso (Al-Qutaish, 2010).

F. MODELO SISTEMÁTICO DE CALIDAD - MOSCA

Se mencionará, en último lugar, el Modelo Sistemático de Calidad (MOSCA)(=), creado en el año 2005 en la Universidad Simón Bolívar de Caracas, Venezuela. Este modelo tiene como intención integrar en un solo sistema a un grupo variado de modelos de calidad de software, entre los que se incluye a los mencionados Dromey e ISO 9126, además de otros, considerándolos como sub-modelos al interior de MOSCA (Mendoza, 2002).

A juicio de sus autores, es imposible desligar la calidad del producto de la calidad de proceso, y viceversa, por lo cual resulta ineficiente tener modelos separados para evaluar la calidad de cada uno de ellos y se hace necesaria su valoración en un modelo sistemático y unificado, en el cual exista una interacción tal que ambas cosas, proceso y producto, sean vistos como un todo.

Cada una de estas etapas, al igual que sucede en casos anteriores, comprende un conjunto de características/atributos a evaluar. Sin embargo, como es común en las normas ISO a diferencia de los tres primeros modelos revisados, este tiene un interés mucho mayor sobre el proceso de producción y tiende a centrarse más sobre este que sobre el producto de software.

G. ISO/IEC 25000

Basada en los protocolos ISO/IEC 9126 (Aenor, 2004) y ISO/IEC 14598 (Aenor, 2006), normativas previas igualmente diseñadas para evaluar la calidad del software, la norma ISO/IEC 25000 busca establecer un estándar más completo y evolucionado para evaluar dicha calidad, tratando para ello de llenar los vacíos presentados por sus antecesoras, a la vez que utiliza sus puntos fuertes. (Carvalho, 2007)

Para ello establece un modelo denominado SQuaRE (Software product Quality Requirements and Evaluation) (Del Rey, 2003), requisitos para la calidad de productos de software y su evaluación) que, a diferencia de sus antecesores, no se centra de manera tan marcada y exclusiva en el proceso de producción (punto especialmente criticado de las normas inicialmente mencionadas), sino que trata de buscar un enfoque de carácter más amplio y completo que, sin descuidar en ningún momento dicho proceso, preste mucha mayor atención a las características internas y externas del producto terminado (ISO, 2005). De igual forma, el protocolo permite establecer de antemano los requerimientos de calidad esperados del producto.

La norma ISO/IEC 25000 (ISO, 2005), para efectos de claridad y facilidad de aplicación, está compuesta por cinco divisiones internas, a saber: ISO 2500n (Dave, 2004) enfocada

en la gestión de calidad y que incluye todos los estándares básicos de la norma. ISO 2501n (Dave, 2004) que establece un modelo de calidad detallado que abarca tanto la calidad externa e interna del producto como su calidad de uso. ISO 2502n (Dave, 2004) que habla de las mediciones de calidad y propone un modelo de referencia de calidad de producto de software. ISO 2503n (Dave, 2004) que establece los requisitos de la calidad. Y finalmente la división ISO 2504 (Dave, 2004) dirigida a la evaluación de calidad misma, incluyendo para ello requisitos, recomendaciones y guías para evaluación.

IV. MODELO PROPUESTO

Se ha estipulado a través de la revisión de la literatura en el presente documento que el interés por la calidad crece de forma continua, a medida que los clientes se vuelven más selectivos y comienzan a rechazar los productos poco fiables o que realmente no dan respuesta a sus necesidades.

Por lo tanto, las metas que se establezcan para la calidad del producto van a determinar las metas a establecer para la evaluación de la calidad del proceso de desarrollo, ya que la calidad del producto obedece entre otros factores a dicho proceso evaluativo; quiere decir lo anterior que sin un buen proceso de desarrollo es casi imposible obtener un buen producto y por lo tanto la evaluación de dicho proceso aparece y cobra importancia.

Pues bien, importante destacar que la calidad de un producto software debe ser considerada en todos sus estados de evolución a medida que avanza el desarrollo de acuerdo al ciclo de vida seleccionado para su construcción (especificaciones, diseño, código, etc.). Los principales problemas a los que se enfrenta el desarrollo de software a la hora de tratar la calidad de un producto software son la definición de calidad y su comprobación, así mismo y como se ha mencionado previamente, la calidad es un concepto que se deriva de un conjunto de sub-conceptos, por lo tanto el modelo que se ha creado para evaluar dicho proceso se debe a los criterios que presentan las normas estipuladas pero también se debe observar el contexto en el cual se desarrolla.

Es decir, no todas las categorías expuestas en un modelo dado y que se adecuan a determinada institución se adaptan a las condiciones que le son propias a otras entidades. Es allí donde se debe considerar unificar los conceptos que resulten relevantes para evaluar la calidad del software en el contexto que está siendo observado. Así las cosas, el modelo de calidad para la evaluación del software del CIADTI se sujeta a las cuatro fases en la cuales se ejecuta el producto: levantamiento de información, análisis y diseño, programación y pruebas.

Para esta etapa es necesario hacer claridad que la evaluación de la calidad se realizó sobre los artefactos que son propios del proceso de desarrollo del producto del CIADTI. Estos artefactos son unos documentos que se han elaborado en el marco del Sistema de Gestión de Calidad de la Organización para soportar cada una de las actividades realizadas por el equipo de trabajo y los cuales constituyen un elemento fundamental para el estudio llevado a cabo en la presente investigación.

Cada una de estas etapas debe estar acorde a unas categorías que ya fueron previamente analizadas en el diagnóstico y de las cuales, gracias a los criterios obtenidos por la revisión de la literatura, se seleccionaron cinco: fiabilidad, usabilidad, adaptabilidad, reusabilidad y simplicidad. Estas categorías están definidas como se muestra en la fig. 4.

Para la aplicación del modelo se deben tener presentes los siguientes pasos o protocolos; se debe identificar la etapa del

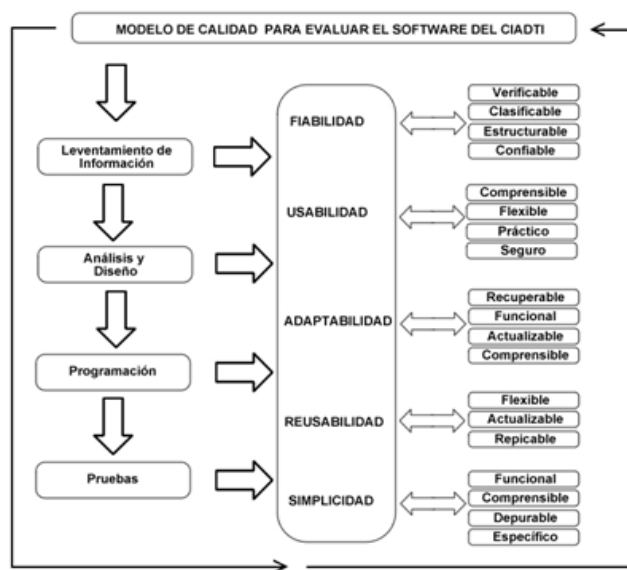


Figura 4. Modelo de Calidad CIADTI para Evaluar los Artefactos del Proceso. Fuente: Autor

proceso y el artefacto, se evalúa si es viable la aplicación del modelo y una vez aprobado se asignan porcentajes de valor a cada una de las categorías. Por último se aplica el instrumento, se tabulan y analizan los resultados por el equipo de expertos, se dan las conclusiones y recomendación al coordinador o encargado del proceso. Es importante resaltar que el modelo se puede aplicar a cualquier documento o artefacto y es flexible en la asignación de atributos a evaluar.

V. VALIDACIÓN DEL MODELO

El proceso de validación consistió en la aplicación de un instrumento al personal de desarrollo en sus áreas específicas

de trabajo, identificadas de la siguiente manera: levantamiento de Información, Análisis y Diseño y Programación. El siguiente paso fue el análisis de la información, donde se identificaron los valores más bajos y más altos en cada uno de los instrumentos y, para finalizar, se tabularon en un cuadro de resultados. La segunda tarea para validar el instrumento fue aplicarlo a un panel de expertos en el CIADTI, entre coordinadores y consultores con más de ocho años de experiencia en la construcción de software; a este instrumento se le aplicaron los mismos pasos que al equipo de desarrolladores y la validación surgió del análisis comparativo entre las dos tabulaciones y las observaciones generales de los expertos.

VI. CONCLUSIONES Y RECOMENDACIONES

Se realizó un estado del arte sobre la calidad del producto, teniendo en cuenta estudios internacionales y modelos de evaluación, logrando establecer que durante las últimas décadas muchas investigaciones se han realizado con el fin de crear herramientas y procesos estandarizados que permitan tanto desarrollar software de calidad, en primera instancia, así como evaluar dicho software una vez se encuentra terminado; en la misma línea fue posible establecer que no existe una herramienta o procedimiento unificado que permita evaluar la calidad de software, sino que existen una gran variedad de modelos, cada uno con un enfoque diferenciado, y cuyo criterio de aplicación corresponde al grupo de parámetros que se desee sea evaluado, por lo tanto se definió un modelo de evaluación para medir la calidad del producto del CIADTI a partir de la ejecución de tres etapas: diagnóstico, diseño y aplicación.

El modelo fue evaluado, configurado a través de medidores internacionales como la norma ISO 9126, el modelo FURPS+, el modelo MCCALL y el modelo DROMEY, entre otros.

Se utilizó una grupo muestral de 15 personas del CIADTI, quienes participaron y retroalimentaron el proceso de diagnóstico y la siguiente etapa de autoevaluación del modelo para que este finalmente fuese consolidado. El modelo de evaluación final se sujeta a las cuatro fases en las cuales se ejecuta el producto: levantamiento de información, análisis y diseño, programación y pruebas. Cada una de estas etapas estuvo acorde a unas categorías que fueron previamente analizadas en el diagnóstico, y de las cuales gracias a los criterios obtenidos por la revisión de la literatura se seleccionaron cinco: fiabilidad, usabilidad, adaptabilidad, reusabilidad y simplicidad.

El modelo producto de esta investigación es innovador ya que se manejaron los conceptos y parámetros para el desarrollo de software, los cuales se caracterizaron para luego

aplicarlos en los artefactos que hacen parte de las etapas del desarrollo de Software. El resultado de este trabajo puede ser aplicado a cualquier artefacto o documento que maneje algún proceso dentro de un sistema.

Se propuso la creación de un comité evaluador, conformado por cuatro personas de mayor experiencia en la unidad de trabajo, que calificaron en conjunto todos los componentes del modelo desde una óptica objetiva; esto dio lugar a la comparación de resultados entre desarrolladores y comité a fin de proponer un consolidado de mejoras para el desarrollo del producto en el CIADTI.

Se dejan ensamblados en formato Excel los instrumentos, calificaciones y valoraciones respectivas del modelo de evaluación para que, en adelante y de acuerdo al Sistema de Gestión de Calidad del CIADTI, se lleve el proceso de evaluación cada periodo de tiempo conveniente, propiciando con ello el fortalecimiento de cada fase de desarrollo a partir del análisis de las respectivas categorías correspondientes a la calidad del software.

AGRADECIMIENTOS

Al grupo de investigación en Ciencias Computacionales de la Universidad de Pamplona que apoya el programa de Maestría en Gestión de proyectos Informáticos, dentro de la cual son estudiadas y discutidas diversas áreas de conocimiento de la gestión de proyectos y la ingeniería del software, junto con docentes y estudiantes del programa.

REFERENCIAS

- AENOR. Asociación Española de Normalización y Certificación. ISO/IEC 14598: Tecnología de la Información. Evaluación del Producto Software. Parte 5: procedimiento para evaluadores (ISO/IEC 14598-5:1998), Parte 5. 2004.
- AGUIRRE, ANDRÉS FELIPE., PARDO, CÉSAR PANTOJA, WILSON LIBARDO., PINO, FRANCISCO J - Reporte de Experiencias de La Aplicación de Competisoft en Cinco Mipymes Colombianas. Consultado en línea en el URL <http://hdl.handle.net/11190/186>. 2010.
- AL-QUTAISH, R: Quality. Models in Software Engineering Literature: An Analytical and Comparative Study. Journal of American Science 6 (3). <http://publications.rafa-elayyan.ca/33.pdf>. 2010.
- ARDILA, CARLOS., PINO, FRANCISCO. panorama de Gestión Cuantitativa de Procesos de Desarrollo de Software en Pequeñas Organizaciones. Universidad del Cauca, Popayán, Colombia, 2013.
- CARVALLO VEGA, JUAN PABLO Towards a Unified Catalogue of Bon-Technical Quality Attributes to Support COTS-Based Systems Lifecycle Activities. Sixth International Conference on Commercial-off-the-Shelf (COTS)-Based Software. System.IEEE Computer Society, 2007.
- CHÁVEZ ROJAS, ALMA DELI. Calidad del Software. "El Camino al Éxito." Instituto Tecnológico de Colima, Av. Tecnológico no. 1, Villa de Álvarez, Colima, México. 2009.
- CONSUEGRA RODRÍGUEZ., DIALYS ALIASKA., SHELTON NADAL, RONALD., GARCÍA PÉREZ, ANA MARÍA. Automatización de La Gestión de La Calidad de Una Organización de Software Partiendo de La Medición del Tamaño y Tomando Como Principio La Satisfacción Del Cliente. Centro de estudios de informática. Universidad central de las Villas "Martha Abreu" Uclv. Santa Clara. 2005.
- CONSUEGRA, RODRÍGUEZ., DIALYS ALIASKA., SHELTON NADAL, RONALD., GARCÍA PÉREZ, ANA MARÍA., Automatización de la gestión de la calidad de una organización de software partiendo de la medición del tamaño y tomando como principio la satisfacción del cliente. Centro de estudios de informática. Universidad central de las Villas "Martha Abreu" Uclv. Santa Clara. 2005.
- DAVE ZUBROW. Measuring Software Product Quality: The ISO 2500 Series and CMMI, European SEP - June 14, 2004
- DÁVILA NICANOR, LETICIA; MEJÍA ÁLVAREZ, PEDRO. Evaluación de la Calidad de Software en Sistemas de Información en Internet. Cinvestav-ipn. Sección de Computación Zacatenco. México. 2005.
- DÁVILA, LETICIA. Evaluación de la Calidad de Software en Sistemas de Información en Internet. Recuperado de: <http://delta.cs.cinvestav.mx/~pmejia/davila-mejia.pdf>. 2005.
- DÍAZ-ANTÓN, G., PÉREZ, M., GRIMÁN, A., MENDOZA, L. Instrumento de Evaluación de Software Educativo Bajo un Enfoque Sistémico. 2008. Consultado en línea en el URL http://www.labf.usb.ve/?page_id=393. 2008.
- ESTAYNO, M., DAPOZO, G., CUENCA PLETCH, L., GREINER, C., MEDINA, Evaluación de Calidad de Software, Formación de Recursos Humanos y Políticas Públicas para la Promoción de la Industria del Software en la Región del Noreste Argentino. Universidad Nacional del Nordeste, Argentina. 2009.
- EARLS, A., Modeling, Artículo publicado en el portal ebizq.net 2011. Fuente.Consultada: http://www.ebizq.net/topics/bpm_process_modeling/features/13202.html. 2011.
- GUTIÉRREZ, EDNA., GUTIÉRREZ, AGUSTÍN., PÉREZ, AURORA., MÁRQUEZ, LUIS. Acerca de la implementación de los modelos de calidad de la construcción de software en México. Universidad Autónoma de México, México. 2009. Consultado en línea en el URL: <http://www.redalyc.org/pdf/2570/257020617011.pdf> 2009.
- INCERA, G., Medidas de Calidad en Proceso, Producto y Mantenimiento, Aplicadas al Control estadístico de Procesos. Escuela Superior de Informática de la Universidad de Castilla, la Mancha España. 2005.
- IEEE, 1990. Institute of Electrical and Electronics Engineers IEEE, Standard Glossary of Software Engineering Terminology. STD 610-1990.
- ISO, 1994. International Organization for Standardization. ISO 8042:1994. Sistemas de Gestión de Calidad, Conceptos y Vocabulario.
- ISO, 1998. International Organization for Standardization. Software Process Improvement Capability Determination, SPICE, En español Determinación de la Capacidad de Mejora del Proceso de Software. 1998.
- ISO, 1999 International Organization for Standardization. Quality management systems - Requirements. ISO/IEC 9001. 1999.
- ISO, 2000 International Organization for Standardization. Sistemas de Gestión de Calidad, Conceptos y Vocabulario (traducción), Norma ISO 9000:2000.
- ISO, 2000. International Organization for Standardization, ISO/IEC 9126: Ingeniería del software: Calidad del Producto Software: Modelo de Calidad. 2005.
- ISO, 2005. International Organization for Standardization. ISO/IEC 25000:2005. Software product Quality Requirements

- and Evaluation (SQuaRE) - Guide to SQuaRE.
- LÓPEZ, A., CABRERA C., VALENCIA L., Calidad de Sistemas de Información. 2ª edición, 2008.
- MENDOZA, LUIS., PÉREZ, MARÍA., GRIMÁN, ANNA., ROJAS, TERESA. Algoritmo para la Evaluación de la Calidad Sistemática del Software. Laboratorio de Investigación en Sistemas de Información, Universidad Simón Bolívar, Caracas, Venezuela. 2002.
- MONSALVE, LUIS., Calidad del Software. Consultado en línea en el URL: <http://prof.usb.ve/lmendoza/Documentos/PS-6117%20%28Teor%EDa%29/PS6117%20Calidad%20del%20Software.pdf> 2002.
- McCALL, J., & CAVANO, J. A framework for the measurement of software quality. ACM. 1978
- OTÁLORA, JORGE., GUTIÉRREZ ELIANA. Herramienta de Gestión de Calidad para el Proceso de Software. Orientada a Mipymes Basado en la Norma ISO/IEC 15504. Fundación Universitaria Católica del Norte, Colombia. 2011.
- ROLÓN ELVIRA; RUIZ, FRANCISCO; GARCÍA, FÉLIX; PIATTINI MARIO. Aplicación de métricas Software en la Evaluación de Modelos de Procesos de Negocio. Universidad Autónoma de Tamaulipas-Fians; Universidad de Castilla-la Mancha. 2010.
- RUIZ, GUSTAVO ALBERTO., PEÑA, ALEJANDRO., CASTRO , CARLOS., ARTURO., ALAGUNA, LUZ MERY; ARAIZA ÁNGELA & RINCÓN, RAFAEL DAVID. Modelo de Evaluación de Calidad de Software Basado en Lógica Difusa, Aplicada a Métricas de Usabilidad de Acuerdo con la Norma ISO/IEC 9126. Universidad de San Buenaventura, Medellín. 2006.
- SOLARTE, G., MUÑOZ LUIS., ARIAS, BIBIANA. Modelos de Calidad para Procesos de Software. Scientia El Tedmica, vol. XV, núm. 42, Agosto, 2009, pp. 375-379 - Universidad Tecnológica de Pereira, Colombia. 2009.
- TAKAGI, ASLI, CELIKYILMAZ, I., BURHAN TÜRKSEN Modeling Uncertainty with Fuzzy Logic, With Recent Theory and Applications. Editorial: Springer 2009.
- TORRES, JORDI., Del Cloud Computing Al Big Data: Visión introductoria para jóvenes emprendedores - Editorial UOC - PID_00194204 Primera edición: septiembre 2012.
- UNIVERSIDAD DE PAMPLONA, Sistema Integrado de Gestión: Mapa de Procesos, Concepción, Elaboración, Construcción y Transición de Soluciones de Aplicativos Empresariales y de Misión Crítica Específica. 2007.

Informáticos (lesteban@unipamplona.edu.co).

Recibido en mayo 14 de 2014. Recibido con correcciones en junio 21 de 2014. Aceptado en junio 25 de 2014. Publicado en junio 30 de 2014.

Citar este artículo como:

ESTÉVEZ, Y. S., ESTEBAN, L. A. (2014). Modelo de calidad para evaluar el software desarrollado en el centro de investigación aplicada y desarrollo en tecnologías de información CIADTI. Revista TECKNE, vol. 12, n. 1, p. 18

AUTORES

YIMMY S. ESTÉVEZ CARVAJAL es *Administrador Comercial y de Sistemas, Especialista en Gestión de Proyectos Informáticos y (C) Magister en Gestión de Proyectos Informáticos, Analista del Centro de Gestión del Conocimiento del CIADTI, Universidad de Pamplona, es* **DOCENTE HORA CATEDRA ADSCRITO A LA Facultad de Educación de la Universidad de Pamplona.** (vestevez@unipamplona.edu.co)

LUIS ALBERTO ESTEBAN VILLAMIZAR es *Licenciado en Matemáticas y Computación, Máster en Informática. Es docente Tiempo Completo de la Universidad de Pamplona, Colombia. Está adscrito a la Facultad de Ingenierías y Arquitectura. Director de la Maestría Gestión de Proyectos*