

El rol de las normas en la acreditación y certificación de pruebas de software

The role of standards on accreditation and certification of software testing

Sandra VELÁSQUEZ Restrepo [1](#); Juan David VAHOS Montoya [2](#); Carlos LÓPEZ Romero [3](#); Marta Ester GÓMEZ Adasme [4](#); Jennifer LONDOÑO Gallego [5](#); Santiago PATIÑO Hurtado [6](#)

Recibido: 29/06/2018 • Aprobado: 15/09/2018 • Publicado 15/12/2018

Contenido

- [1. Introducción](#)
 - [2. Metodología](#)
 - [3. Reportes de caso sobre certificación y acreditación de laboratorios de pruebas de software](#)
 - [4. Normas para pruebas de software relevantes para validar la calidad de procesos y productos](#)
 - [5. Conclusiones](#)
- [Agradecimientos](#)
[Referencias bibliográficas](#)

RESUMEN:

Este trabajo analiza el rol de las normas empleadas en los procesos de certificación y/o acreditación de pruebas de software. Una revisión bibliográfica permitió identificar el aporte de las normas más utilizadas y la pertinencia de seguir lineamientos de normas internacionales, pero también la relevancia de considerar el contexto de cada país para alcanzar resultados exitosos. Finalmente, se discuten los retos que enfrenta una entidad de formación e investigación que busca certificar un laboratorio de pruebas de software.

Palabras clave: Pruebas de software, normas de certificación, normas de acreditación, laboratorios para pruebas.

ABSTRACT:

This paper analyzes the role of the standards used in the processes of certification and accreditation of software tests. A bibliographic review made it possible to identify the contribution of the most commonly used standards and the relevance of following guidelines of international standards, but also the relevance of considering the context of each country in order to achieve successful results. Finally, the challenges faced by a training and research institution that seeks to certify a testing laboratory are discussed.

Keywords: Software testing, certification standards, accreditation standards, testing laboratories.

1. Introducción

La demanda por los productos y servicios de la industria del software tiene una de las tasas de crecimiento mundiales más altas de la actualidad (Otálora, Bermeo y González, 2009). Una de las estrategias clave para el éxito de esta industria es certificar la calidad de procesos y productos (Martínez, Arango y Robledo, 2015), un factor clave que el direccionamiento estratégico de las organizaciones debe considerar en su plan de gestión del conocimiento (Valdez, Pérez y Maldonado, 2017).

Es sumamente relevante procurar la calidad del software empleado en todo tipo de tecnologías; las transacciones electrónicas, negocios de la bolsa de valores, la telemedicina o el transporte aéreo son algunos ejemplos representativos de entornos donde la confiabilidad del software es un aspecto crucial (Mera-Paz, 2016). Salazar, Lizano y Lara (2011) definen las pruebas de software como la "verificación dinámica del comportamiento de un determinado elemento de software, utilizando un conjunto finito de casos de prueba, seleccionados a partir de un conjunto usualmente infinito de posibilidades de ejecución de ese elemento de software, para verificar el comportamiento esperado buscando minimizar la cantidad de errores". El uso de software validado permite disminuir errores y bajar los costos de desarrollo, aumentando la confianza y seguridad en la aplicación del software (Lara, 2013). Se estima, incluso, que entre el 50% y el 60% del esfuerzo total de producir software se invierte en tareas de aseguramiento de calidad (Kollanus y Koskinen, 2009), teniendo en cuenta que las pruebas deben realizarse bajo normatividad aceptada internacionalmente (Brito, Góngora y Capote, 2014) pero también considerando que la elección de las normas debe realizarse según las necesidades particulares de cada organización (García, Dávila y Pessoa, 2014).

Garousi y Mäntylä (2016) realizaron un detallado estudio terciario sobre pruebas de software, en el cual se plantea la importancia de desarrollar revisiones de la literatura en el área de ingeniería de software si se desea atender la necesidad de formación e investigación en la temática. Khan y Malik (2017) realizaron una revisión sobre el aporte que las normas del IEEE (Institute of Electrical and Electronics Engineers) para ingeniería de software pueden hacer en diversos contextos situacionales que se enfrentan durante el desarrollo de productos, sin incluir la formación como uno de los campos estudiados. Es de resaltar que la página web Evidence-Based Software Engineering (EBSE, s.f.) documenta estudios secundarios sobre diferentes tópicos en ingeniería de software definidos por la guía para el Software Engineering Body of Knowledge (Bourque y Fairley, 2014), pero a pesar de ser dos fuentes referentes para dicha industria, no reportan estudios como el que estamos presentando en este documento. A la fecha no se reportan revisiones bibliográficas sobre el rol de las normas en los procesos de certificación o acreditación de las pruebas a software, y la temática tampoco se ha abordado en el caso de instituciones de formación.

En este trabajo se analiza el rol que tienen las normas en los procesos de certificación y acreditación de pruebas de software, realizando una revisión de la literatura y una discusión sobre los aspectos más relevantes que deben considerar instituciones de formación, investigación y desarrollo tecnológico, como es el caso del Centro de Servicios y Gestión Empresarial del Servicio Nacional de Aprendizaje SENA ubicado en Medellín, Colombia. Se espera que el estudio aquí presentado proporcione criterios adecuadamente sustentados para elegir e implementar las normas a emplear en laboratorios de pruebas de software.

2. Metodología

Para escoger la metodología a emplear se tuvo como referencia el estudio terciario de Garousi y Mäntylä (2016) que analiza 101 artículos publicados entre 1994 y 2015 sobre pruebas de software, encontrando que nueve de los diez artículos más citados son revisiones de la literatura, superando de manera estadísticamente significativa a las revisiones sistemáticas y los mapeos sistemáticos. Realizar una revisión de la literatura sirve, además, como un punto de referencia para que cuando a futuro se realicen suficientes estudios primarios y secundarios sobre la temática abordada en este trabajo, sea entonces posible realizar estudios sistemáticos de la literatura o mapeos sistemáticos.

Este estudio se orientó teniendo como referencia dos preguntas de investigación. La primera fue: ¿qué normas se aplican para la realización de pruebas de software y cómo se han empleado en los estudios reportados en la literatura? Responder esta pregunta permitió determinar las normas específicas de prueba de software incluidas en los estudios consultados, así como identificar las normas más empleadas y las razones para elegir las. La segunda pregunta de investigación fue ¿qué tipos de procesos (misionales, apoyo, estratégicos) abarcan estas normas para el desarrollo de pruebas de software? Responder esta pregunta permitió identificar cuáles son las normas más completas para la certificación

y/o acreditación de las pruebas teniendo en cuenta la planeación estratégica de la organización.

La búsqueda de la información se realizó en las bases de datos bibliográficas Redalyc, Scielo, ScienceDirect, IEEEExplore, ACM Digital Library, Inspec, Compendex y Scopus. Se prestó especial atención a esta última pues según Dieste, Grimán y Juristo (2009) es la base de datos que presenta menos debilidades, cubre una amplia gama de publicaciones en el campo de la ciencia de la computación y mantiene una base de datos completa y consistente.

En la primera etapa de este trabajo, la búsqueda se enfocó en estudios que abordaran la certificación y acreditación de pruebas de software en laboratorios o empresas, con especial atención a las normas que se reportaban que fueran relevantes para las preguntas de investigación planteadas. Se evaluaron las tendencias identificadas en esa primera etapa del estudio, procediendo luego con una consulta en las bases de datos empleando como términos de consulta normas para pruebas a software; una vez se analizaron los títulos, resúmenes y conclusiones de los artículos identificados, se seleccionaron aquéllos que se centraban en las temáticas consideradas en las dos preguntas de investigación de este estudio. Si durante el análisis detallado de los artículos elegidos surgían referencias a otras fuentes de información relevantes, se consultaban para identificar si aportaban información relevante para los objetivos de este trabajo. Es pertinente resaltar que la metodología empleada permitió encontrar no solo artículos científicos en las bases de datos consultadas, sino también libros y trabajos de grado.

3. Reportes de caso sobre certificación y acreditación de laboratorios de pruebas de software

En esta fase del estudio se encontraron solo algunos reportes de caso sobre la temática de interés, todos ellos publicados en los últimos cinco años lo cual indica que, si bien es un campo de investigación de interés vigente, aún no existe una masa crítica de publicaciones que permitan comparar resultados obtenidos en diversos contextos.

Se destacan las diversas publicaciones que se han realizado sobre la acreditación del Laboratorio Industrial de Pruebas de Software (LIPS), el cual está adscrito al Centro Nacional de Calidad de Software (CALISOFT) de la Universidad de las Ciencias Informáticas de Cuba. El trabajo de Brito y Capote (2013) se expone el proceso para diseñar e implementar un Sistema de Gestión de Calidad (SGC) a partir de las normativas, tendencias y especificidades de un laboratorio como el LIPS que apoya a la industria cubana del software. En un trabajo posterior publicado por Brito, Góngora y Capote (2014) se expone el marco de trabajo para el LIPS que integra diferentes normas y modelos para alcanzar niveles de madurez en la prestación de servicios de pruebas al software, procurando la certificación de ese laboratorio, mientras que en el estudio de Capote y otros (2014) se expone la estrategia para desarrollar la perspectiva procesos internos en un laboratorio de pruebas de software que brinda servicios de outsourcing, facilitando su acreditación y certificación así como un aumento en la eficiencia de las pruebas de software. Finalmente, Capote y otros (2015) expusieron el uso de un cuadro de mando integral con perspectivas personalizadas para laboratorios de pruebas de software, siendo las normas internacionales una de las perspectivas empleadas.

Si bien una parte de la información presentada en los cuatro trabajos sobre el LIPS es repetitiva, lo cual debe evitarse en publicaciones secuenciales como las mencionadas, sí es de resaltar el espíritu académico que llevó a los investigadores a divulgar los aprendizajes que se han tenido en ese laboratorio. Un planteamiento que deben considerar quienes afrontan el proceso de acreditar laboratorios de pruebas de software, es que si bien la acreditación del LIPS se realizó bajo la norma NC-ISO/IEC17025:2006, una adaptación para Cuba de la norma ISO/IEC17025 (ISO/IEC, 2017), los requisitos descritos en esa norma están expresados en términos sumamente generales (Brito y Capote, 2013). Aunque esta norma es aplicable a todo tipo de laboratorios de ensayo, es necesario efectuar aclaraciones, modificaciones y definiciones para aquellas áreas que realizan ensayos en temas más específicos, como es el caso de las pruebas de software (Brito, Góngora y Capote, 2014). El

LIPS optó entonces por definir un conjunto de requisitos específicos denominado guía de requisitos complementarios, documento basado en la información recopilada de la norma cubana NC-ISO/IEC 17011:2005 (ISO/IEC, 2005). Con base en estos criterios, se definieron en el LIPS los procesos estratégicos y de apoyo que soportan el SGC del laboratorio (Capote y otros, 2014).

Como modelo de calidad el LIPS adoptó las recomendaciones de la parte 1 de la norma ISO/IEC 9126 (ISO-IEC, 2005), pues garantiza la satisfacción de requisitos de un sistema de gestión de la calidad durante el desarrollo, evaluación y certificación del software, estableciendo también que con la acreditación se alcanza la demostración de la competencia técnica (Capote y otros, 2014). Sin embargo, dado que la ISO/IEC 9126 especifica un modelo de calidad general, en el LIPS se decidió aplicar complementariamente la NC-ISO/IEC 14598 (ISO/IEC, 2001) para establecer los procesos a realizar durante la evaluación de la calidad del producto software. La relación entre las normas ISO/IEC 9126 e ISO/IEC 14598 se esquematiza en la Figura 1. No obstante, los autores reconocen que idealmente debe emplearse la norma ISO/IEC 25000 (ISO/IEC, 2013) conocida como SQuaRE (requerimientos y evaluación de la calidad de productos de software, por sus siglas en inglés), pero no la aplicaron pues para la fecha no estaba aún homologada en su país (Brito, Góngora y Capote, 2014).

Figura 1
Relación entre las normas ISO/IEC 9126 e ISO/IEC 14598.

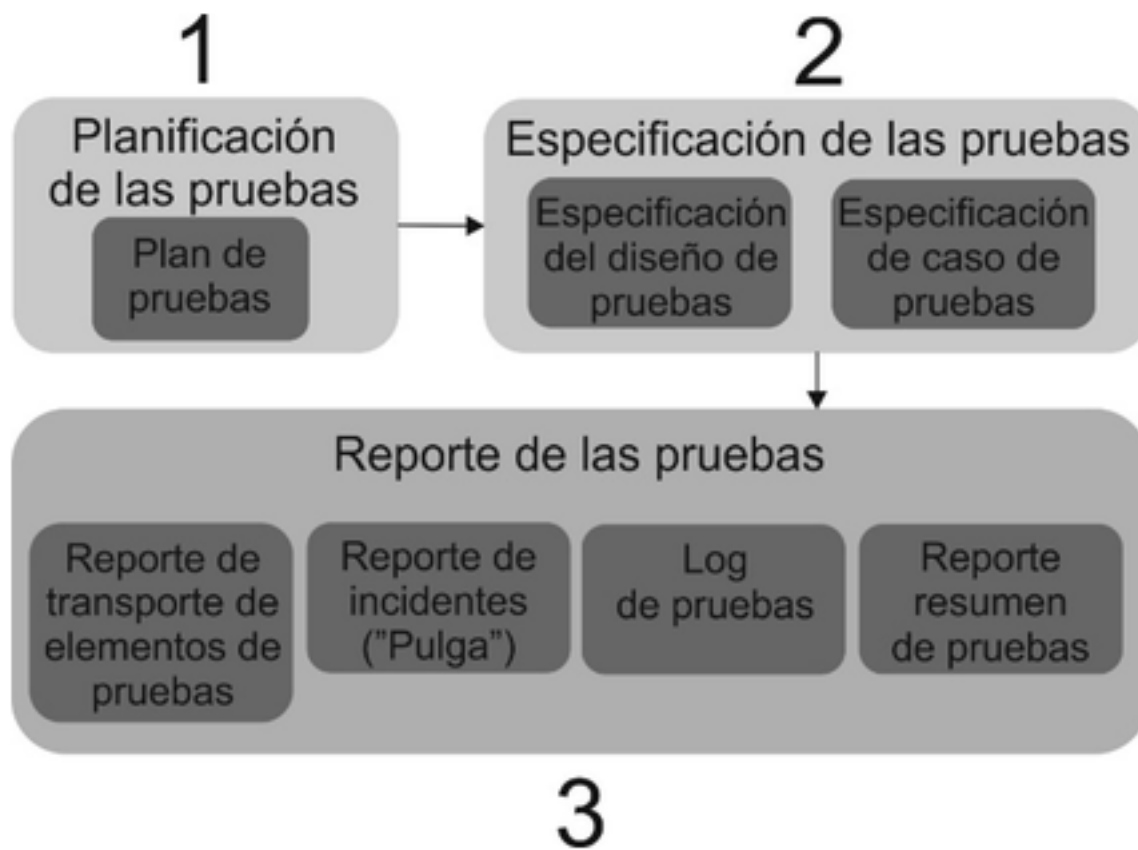


Adaptado de (Carvallo et al. 2010).

Salazar, Lizano y Lara (2011) reportaron los procedimientos requeridos para implementar un sistema de gestión de calidad (SGC) que considerara la correlación entre componentes de la norma ISO/IEC 17025 y de la norma ISO/IEC 9001, para lograr que el Laboratorio de Aseguramiento de Calidad del Software (LACSOFT) de la Universidad de Costa Rica fuese acreditado bajo la norma ISO/IEC 17025. En el trabajo mencionado los autores identificaron los procedimientos requeridos para implementar un sistema de gestión de calidad (SGC) que considerara la correlación entre componentes de la norma ISO/IEC 17025 y de la norma ISO/IEC 9001, para lograr que el LACSOFT sea acreditado bajo la norma ISO/IEC 17025. LACSOFT implementó los requisitos técnicos imprescindibles, usando un proceso de prueba y documentación que inició por pruebas de planificación o plan de pruebas, en el cual se especifican qué elementos o componentes se van a probar para que el grupo de trabajo pueda realizar el proceso de validación y verificación de los requerimientos funcionales y no funcionales (Loaiza y Zorro, 2010). En el plan de pruebas también se identifican las técnicas de pruebas que verifican los diferentes criterios del software. Los casos de prueba están dados por la información o los datos que se emplean como entrada en el software para ejecutar la prueba, y los reportes de las pruebas son el resultado final de la ejecución de la prueba, donde se demuestra que casos de pruebas pasaron o cuáles no. En el reporte se

debe entregar un informe el cual especifiquen los elementos utilizados, incidentes presentados y registro de prueba, como se esquematiza en la Figura 2.

Figura 2
Proceso de pruebas y documentos para el Laboratorio de Aseguramiento de Calidad del Software (LACSOFT) de la Universidad de Costa Rica.



Adaptado de (Salazar, Lizano y Lara, 2011).

Es pertinente resaltar que, al igual que en el LACSOFT en Costa Rica, en el LIPS de Cuba fue necesario identificar los aspectos comunes entre las normas ISO/IEC 9001 e ISO/IEC 17025, con miras a establecer los procedimientos más convenientes para la organización, lo cual sugiere que es una práctica recomendable para los laboratorios de pruebas de software.

La baja cantidad de estudios de caso reportados en las bases de datos sobre procesos de certificación y acreditación de laboratorios para pruebas de software demuestran la necesidad de investigar sobre la temática, dados los retos que deben afrontar las organizaciones interesadas en realizar ese tipo de procesos. Los estudios publicados sobre el tema coinciden en la ventaja de articular los SGC con la norma ISO 9001, basando la acreditación en la norma ISO/IEC 17025, con la gran ventaja de que abarca procesos estratégicos, misionales y de apoyo para pruebas de laboratorio. Sin embargo, por tratarse de normas genéricas deben complementarse con una adecuada elección del modelo de calidad para pruebas de software, para lo cual las tendencias actuales recomiendan guiarse por las normas ISO/IEC 25000. Los estudios de caso consultados no reportan la implementación de otros modelos de calidad diferentes a los ya presentados, pero con el fin de elaborar un marco de referencia normativo para pruebas de software certificadas o acreditadas, en la siguiente sección se presenta un análisis de normas reportadas en la literatura para pruebas de software.

4. Normas para pruebas de software relevantes para validar la calidad de procesos y productos

Curcio y otros (2016) realizaron una consulta entre 175 empresas brasileñas de desarrollo de software, indagando sobre los factores que inciden en la calidad del software usando como criterio de calidad las normas ISO/IEC 9126 (ISO/IEC, 2005) e ISO/IEC 25000 (2017). El estudio demostró que la calidad del software es marcadamente influenciada por nueve factores asociados al perfil de los usuarios, el tipo de organización y la capacidad tecnológica de las organizaciones, sin que el tipo de SGC sea considerado como uno de los factores significativos. Sin embargo, las normas ISO/IEC 9126 e ISO/IEC 25000 fueron reconocidas

por las organizaciones como los modelos de calidad de software más representativos en la industria.

Acosta, Nieto y Barahona (2015) propusieron una metodología para evaluar la calidad del software desarrollado en la Universidad de Cundinamarca, cuya cobertura comprende todo el Departamento de Cundinamarca, Colombia. Los investigadores prefirieron acogerse a la norma ISO/IEC/IEEE 29119 (ISO/IEC, 2013), un conjunto de procedimientos acordados internacionalmente que pueden ser utilizados por cualquier organización al realizar cualquier tipo de prueba de software (ISO/IEC/IEEE, 2014). Los autores eligieron esta norma pues consideraron que incluía de forma más integral los criterios de calidad estipulados en la norma ISO 9126 en lo concerniente a la funcionalidad y la satisfacción del cliente. Como objeto de estudio, usaron el modelo para evaluar una aplicación para la autoevaluación de los programas de la Universidad. En dicho trabajo no se presenta una discusión sobre la manera como el modelo de calidad de la norma ISO/IEC/IEEE 29119 se integra con el SGC de la institución, ni su aporte a la certificación del laboratorio de pruebas donde se realizó el estudio, factores que deben ser considerados en futuros estudios que se desarrollen sobre la temática. Esta misma norma fue abordada en el trabajo de Dávila, García y Cóndor (2017), quienes realizaron un estudio exploratorio sobre la adopción de prácticas de pruebas de software bajo la norma ISO/IEC 29119-2 en pequeñas, medianas y grandes empresas de Lima, Perú. Se encontró que las empresas que cuentan con fábricas de software presentan una mayor adopción de las prácticas establecidas en la norma que las empresas que prestan el servicio de pruebas de software, y que la existencia de factores que ejercen presión, como los contratos, tienen un efecto significativo sobre el grado de adopción de las prácticas estipuladas en la norma.

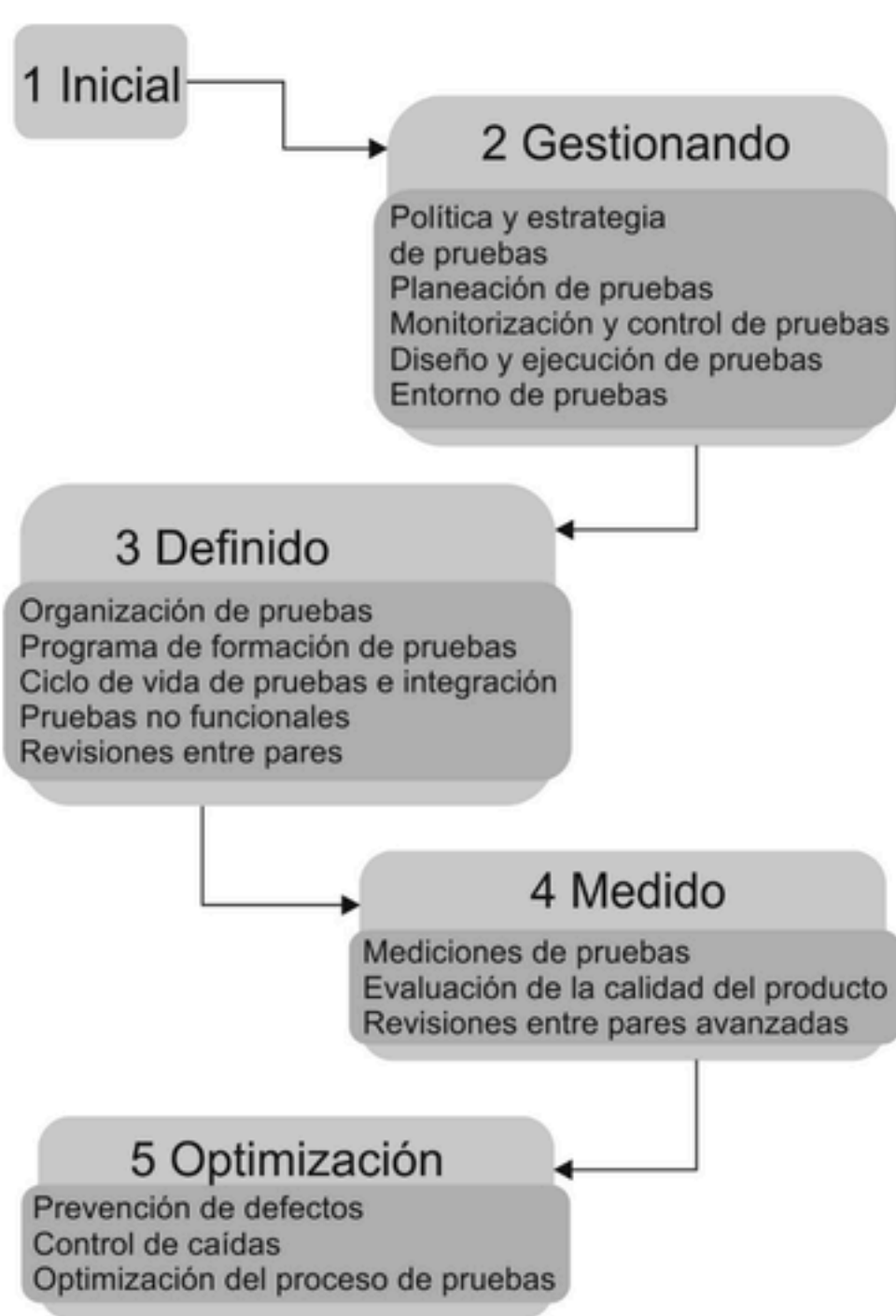
Se ha reportado que la norma ISO/IEC 29119 proporciona a las pequeñas empresas de desarrollo de software un camino claro para implementar procesos sistemáticos cubriendo los procesos estratégicos, misionales y de apoyo, de acuerdo con los resultados obtenidos con cuatro compañías españolas y polacas (Eito-Brun y Sicilia, 2017). Sin embargo, Monteiro, Machado y Kazman (2009) reportaron mejores resultados en una compañía portuguesa de desarrollo de software al emplear la norma ISO/IEC 29119 para asegurar que todo el ciclo de vida del producto satisfaga los criterios de calidad, pero empleando de manera articulada el enfoque CMMI (Capability Maturity Model Integration for Development) para las pruebas de software.

Los enfoques de integración de modelos como el CMMI han sido objeto de estudios secundarios detallados como el realizado por Afzal y otros (2016), reportando que el más empleado es el TMMi ((Testing Maturity Model Integration) que utiliza el concepto de niveles de madurez para la evaluación y mejora del proceso. Se ha reportado que la aplicación de los criterios de madurez de la TMMi impacta de forma positiva en la calidad del producto, la productividad de la organización de prueba y el esfuerzo de tiempo de ciclo o plazo de entrega; esto quiere decir que aborda las cuatro piedras angulares para las pruebas estructuradas: ciclo de vida, infraestructura, organización y técnicas (Veenendaal, 2012), abarcando también todos los procesos estratégicos, misionales y de apoyo. La guía TMMi es una arquitectura de 5 niveles como se observa en la Figura 3, determinando el grado de madurez y el camino evolutivo para probar la mejora del proceso.

Cada nivel tiene un conjunto de áreas de proceso que una organización debe implementar, y cada área de proceso a su vez cuenta con varios objetivos específicos y prácticas específicas (Garousi, Felderer, y Hacaloğlu, 2017). Alcanzar cada uno de los niveles asegura que una mejora ha sido consolidada como base para el siguiente nivel (Veenendaal y Cannegieter, 2011). Los niveles están dispuestos de la siguiente manera:

Figura 3

Niveles de madurez y áreas de proceso de TMMi



Adaptado de (Veenendaal y Cannegieter, 2011)

Nivel 1 - Inicial: tiene como objetivo demostrar que el software es funcional, por lo cual normalmente se realiza de un modo rápido. A menudo las organizaciones que no proporcionan un entorno estable y que dependa de la competencia del personal utilizan este proceso para encontrar rápidamente las fallas mayores.

Nivel 2 - Gestionado: ayuda a salvaguardar el mantenimiento de las prácticas en momentos de estrés. Su objetivo es verificar que el software satisface la especificación de requisitos. El tipo de pruebas que se realiza en este nivel son de componente, integración, sistema y aceptación.

Nivel 3 - Definido: en este nivel las pruebas son definidas desde el plan maestro para evitar que se retrase el inicio de los procesos de pruebas en el ciclo de desarrollo del software. Las organizaciones comprenden la importancia de realizar un control en la calidad del producto e implementan el tipo de pruebas no funcionales como usabilidad y/o fiabilidad, dependiendo de los criterios del negocio (Zamora, 2011).

Nivel 4 - Medido: alcanza los objetivos de los niveles 2 y 3, obteniendo como provecho el diseño de una infraestructura técnica, convirtiendo las pruebas en un proceso medido y consistente en todo el ciclo de desarrollo del producto, detallando y documentando el proceso para soportar el mejoramiento de las pruebas. Las mediciones al programa de métricas de la organización pueden dar apoyo a la toma de decisiones, ya que ayuda a predecir el rendimiento y los costos de las pruebas.

Nivel 5 - Optimización: es el logro de todos los objetivos de los niveles del 1 hasta el 4 de la TMMi. Las organizaciones que están en este nivel han creado infraestructura que soporta un proceso definido y medido, mejorando de forma continua sus procesos. A lo largo del ciclo

de vida del desarrollo del software se identifican y analizan los defectos para prevenir fallas similares en el futuro, gestionando estadísticamente los procesos de prueba por el área de control de calidad teniendo como pilar las mediciones de los niveles de bondad, fiabilidad, muestreo estadístico y confianza basándose en muestras. El nivel 5 tiene dentro de su infraestructura tres áreas claves: control de calidad, prevención de defectos y optimización de pruebas; es sumamente importante tener en cuenta que estas áreas están estrechamente relacionadas entre sí (Burnstein, 2003).

Para hacer la evaluación de una organización y obtener su nivel de madurez, la TMMi usa el enfoque de evaluación de cuestionario-entrevista, en donde se definen un conjunto de preguntas relacionadas a los objetivos de madurez con la intención de evaluar el proceso de pruebas llevado por la organización (García y Amarillo, 2009).

Sin embargo, según van Veenendaal, Grooff y Hendriks (2007), este modelo es fácil de entender pero bastante difícil de implementar en una organización para mejorar el proceso de pruebas, pues en promedio toma aproximadamente dos años lograr el nivel 2 de TMMi.

Como se mencionó en la sección introductoria de este artículo, es necesario definir el marco normativo para diseñar e implementar un laboratorio de pruebas de software en el Centro de Servicios y Gestión Empresarial (CESGE) del Sena Regional Antioquia, con sede en Medellín, Colombia. Una vez analizada la información recopilada de los estudios reportados en la literatura, este trabajo resume en la Tabla 1 las ventajas de los modelos de calidad de software reportados en la literatura empleando como criterio el aporte que realizan a los procesos de certificación, así como el tipo de proceso que abarca cada normatividad.

Tabla 1
Cuadro comparativo sobre los modelos de calidad que pueden emplearse para certificar pruebas de software

Norma	Ventajas	Procesos que abarca
ISO/IEC 9126 Calidad del producto de software (ISO/IEC, 2001). Se ha revisado por las ISO/IEC 25010 (2011), ISO/IEC 25022 (2016) e ISO/IEC 25023 (2016).	<ul style="list-style-type: none"> • Representa la calidad de un producto de software respecto a las características y subcaracterísticas (Hurtado, 2012). • Valida el cumplimiento del software respecto a los requisitos de calidad interna (Jung, H., Jung, W. y Yang, 2006). • Predice el nivel de calidad de uso del producto (Capote y otros, 2014) 	<ul style="list-style-type: none"> • Misionales (Jung, H., Jung, W. y Yang, 2006).
ISO/IEC 25000 Ingeniería de sistemas y de software, requisitos y evaluación de la calidad de sistemas y de software (SQuaRE) (ISO/IEC, 2014).	<ul style="list-style-type: none"> • Diferenciación de los competidores, asegurando tiempos de entrega y reducción de fallos en el producto tras su implantación en producción (Ali y Yue, 2015). • Detecta los defectos en el producto software y procede a su eliminación antes de la entrega, lo que supone un ahorro de costes en la fase de mantenimiento posterior (Dávila, García y Córdor, 2017). 	<ul style="list-style-type: none"> • Estratégicos, misionales y de apoyo (Dávila, García y Córdor, 2017; Jung, H., Jung, W. y Yang, 2006)
ISO/IEC /IEEE 29119 Ingeniería de software – pruebas a software (ISO/IEC, 2013)	<ul style="list-style-type: none"> • Optimiza recursos (Dávila, García y Córdor, 2017). • Mayor control de procesos (Ali y Yue, 2015). • Administración de procesos implementando las mejores prácticas internacionales (Ali y Yue, 2015). • Optimizar el desarrollo de software con mayor valor agregado (Jung et al. 2006). • Es una de las más actuales, haciendo una recopilación de los criterios de calidad presentes en la ISO/IEC 9126 que está orientada a la funcionalidad y la satisfacción del cliente (Dávila, García y Córdor, 2017). 	<ul style="list-style-type: none"> • Estratégicos, misionales y de apoyo (Jung, H., Jung, W. y Yang, 2006).
TMMI Modelo de calidad	<ul style="list-style-type: none"> • Mejora los procesos de pruebas constantemente, a través de 	<ul style="list-style-type: none"> • Estratégicos,

especializado en pruebas de software (Van Veenendaal, 2012).

una ruta específica (Capote y otros, 2015).
• Aumento de la satisfacción del cliente (Van Veenendaal, 2012).

misionales y de apoyo (Van Veenendaal, 2012).

Fuente: elaboración propia

En el caso del laboratorio de pruebas de software que se desea certificar y/o acreditar en el CESGE del Sena, el equipo de trabajo tendrá en cuenta las tendencias identificadas considerando que la misión del Centro es apoyar procesos de formación de aprendices en áreas afines a la ingeniería de software, validar de forma independiente los desarrollos de la fábrica de software del CESGE y de los centros Sena del país, y apoyar a la industria nacional del software para que tengan disponible el laboratorio para sus desarrollos. Bajo este panorama, es de resaltar que en la literatura especializada existe abundante información sobre normas aplicables en pruebas de software, pero son pocos los estudios que reportan criterios para elegir el método a emplear para certificar y acreditar laboratorios de software.

Se aconseja que cada organización interesada en realizar pruebas de software elija los aspectos a emplear de la normatividad según sus necesidades particulares, teniendo presente los aportes de cada norma según se relaciona en este artículo, seleccionando los modelos de calidad más acordes a la necesidad específica de cada organización según los criterios presentados en la Tabla 1.

Es pertinente en este punto mencionar que la revisión bibliográfica en el caso particular de Colombia no reportó ningún estudio sobre acreditación y/o certificación de laboratorios para pruebas de software. Los autores entonces procedieron a contactar al Organismo Nacional de Acreditación de Colombia (ONAC), entidad responsable por la temática en el país (ONAC, s.f), así como al Instituto Colombiano de Normas Técnicas (ICONTEC) en calidad de Organismo Nacional de Normalización de Colombia (ICONTEC, s.f.), quienes informaron que hasta el año 2017 no existía ningún laboratorio acreditado en el país bajo la norma ISO/IEC 17025. Ambas entidades recomendaron que antes de empezar el proceso de acreditación se debería certificar e implementar como modelo de calidad la norma ISO/IEC 25000, aconsejando que debía tenerse en cuenta que la tendencia mundial es complementarla con la ISO/IEC/IEEE 29119. Este panorama confirma la recomendación de considerar las circunstancias propias del entorno de cada organización, prestando especial atención al papel de los organismos de acreditación y certificación que rigen los procesos en cada país.

5. Conclusiones

Las normas a emplear en pruebas de software son un área activa en la ingeniería de software, como lo demuestra la dinámica de los estudios publicados sobre esta temática. En la búsqueda realizada, se encontró que el estudio terciario presentado por Garousi y Mäntylä (2016) es el más completo y actualizado sobre pruebas de software, pero con muy poca atención a las normas aplicables. En este trabajo se llevó a cabo una revisión que proporciona una visión general sobre las normas que son más reportadas para realizar pruebas de software buscando que sean acreditadas y/o certificadas, identificando que la ISO/IEC 17025 es la empleada de manera unánime para la acreditación y que la norma ISO/IEC 25000 es el modelo de calidad de mayor atención en los estudios reportados en la literatura.

Se encontraron reportes sobre la manera como dos laboratorios diseñaron su SGC comenzando por identificar los aspectos comunes entre las normas ISO/IEC 9001 e ISO/IEC 17025, adaptarlos a los procedimientos gubernamentales de cada país y considerar la opinión de ingenieros de pruebas de la industria, con miras a establecer los procedimientos más idóneos para realizar pruebas de software con calidad internacional. Con base en los resultados del estudio presentado en este documento, se recomienda a los interesados en realizar pruebas de software bajo estándares internacionales considerar estas experiencias reportadas en la literatura, para identificar estrategias útiles para cada contexto particular.

Algunos estudios consultados afirman que la calidad de los productos software está

fuertemente influenciada por la calidad del proceso que los genera, y que particularmente las pruebas contribuyen en gran medida a este propósito. Además de las normas presentadas, también se reporta información sobre modelos de procesos de prueba que han sido definidos, adaptados o extendidos en la industria del software, muchos de ellos adaptados o extendidos desde la Test Maturity Model Integration (TMMi) y desde la norma ISO/IEC/IEEE 29119 como enfoque arquitectónico alineado con otros modelos.

Es de resaltar el aporte que puede efectuar la norma ISO/IEC17025 a las pruebas de software, porque establece criterios generales para la acreditación de laboratorios, y de acuerdo con casos de aplicación reportados en algunos trabajos ayuda al mejoramiento continuo y a la efectividad. Es relevante tener en cuenta que, para poder implementar la norma ISO/IEC17025 con miras a acreditar un laboratorio de pruebas de software, se debe implementar primero la norma ISO/IEC 29119. Se espera que la información reportada facilite la valoración de las distintas normas a aplicar, de tal manera que estudiantes, desarrolladores, arquitectos de software y directores de proyectos se enfoquen en generar criterios enfocados a la calidad de software durante todo el ciclo de vida del producto tecnológico. De igual manera, al identificar para las principales normas empleadas en pruebas de software si abarcan procesos misionales, estratégicos y/o de apoyo, se espera establecer lineamientos que puedan ser considerados por los responsables por el diseño, implementación y/o administración de pruebas de software certificadas y/o acreditadas.

Agradecimientos

Los autores agradecen al Sistema de Investigación, Desarrollo Tecnológico e Innovación del SENA (SENNOVA) y el Centro de Servicios y Gestión Empresarial de Medellín, Colombia, por el financiamiento del proyecto "Fortalecimiento de la Oferta de Servicios Tecnológicos a Empresas y al Sector Académico, por medio de un Laboratorio Acreditado y/o Certificado en las Normas ISO 29119 y/o ISO 17025 y/o ISO 20000 y/o TMMi de pruebas de calidad a Software del Centro de Servicios y Gestión Empresarial" con número de radicado SGPS-1621-2017.

Referencias bibliográficas

- Acosta, I., Nieto, E., y Barahona, C. (2015). Metodología para la evaluación de calidad de los productos software de la Universidad de Cundinamarca. *ENGI Revista Electrónica de la Facultad de Ingeniería*, 3(2).
- Afzal, W., Alone, S., Glocksien, K., y Torkar, R. (2016). Software test process improvement approaches: a systematic literature review and an industrial case study. *Journal of Systems and Software*, 111(C), 1-33. doi: 10.1016/j.jss.2015.08.048
- Ali, S., y Yue, T. (2015). Formalizing the ISO/IEC/IEEE 29119 Software Testing Standard. *ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, (pp. 396-405). IEEE, Ottawa, Canada.
- Bourque, P., y Fairley, R. E. (2014). Guide to the software engineering body of knowledge, version 3.0. IEEE Computer Society.
- Brito, Y., y Capote, T. (2013). Sistema de Gestión de la Calidad en un laboratorio de evaluación de software que opta por la acreditación. *Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software*, 3(2), 13-18.
- Brito, Y., Góngora, A., y Capote, T. (2014). Marco de trabajo integrado para asegurar calidad en los servicios de testing del LIPS. *Memorias del Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology - LACCEI 2014*, 1-10.
- Burnstein, I. (2003). Practical software testing: a process-oriented approach. New York, USA: Springer-Verlag.
- Carvallo, J., Franch, X., y Quer, C. (2010). Calidad de componentes software. *Calidad del Producto y Proceso Software*, 30. España: RA-MA Editorial
- Capote, T., Brito, Y., Yzquierdo, R., y Febles, A. (2014). Estrategia para desarrollar la

perspectiva procesos internos en un laboratorio de pruebas de software. *Revista Cubana de Ciencias Informáticas*, 8, 145-156. Recuperado de: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992014000400009&lng=es&nrm=iso

Capote, T., Pérez, Y., Yzquierdo, R., Febles, A., y Estrada, V. (2015). Perspectivas del cuadro de mando integral personalizadas para laboratorios de pruebas de software. *Ingeniería Industrial*, XXXVI, 138-150. Recuperado de: <http://scielo.sld.cu/pdf/rii/v36n2/rii04215.pdf>

Curcio, K., Malucelli, A., Reinehr, S., y Paludo, M. A. (2016). An analysis of the factors determining software product quality: a comparative study. *Computer Standards & Interfaces*, 48, 10-18. <https://doi.org/10.1016/j.csi.2016.04.002>.

Dávila, A., García, C., y Córdor, S. (2017). Análisis exploratorio en la adopción de prácticas de pruebas de software de la ISO/IEC 29119-2 en organizaciones de Lima, Perú. *RISTI-Revista Ibérica de Sistemas e Tecnologias de Informação*, 21, 1-17. doi: <http://dx.doi.org/10.17013/risti.21.1-17>

Dieste, O., Grimán, A., y Juristo, N. (2009). Developing search strategies for detecting relevant experiments. *Empirical Software Engineering*, 14, 513-539. doi: 10.1007/s10664-008-9091-7

EBSE - Evidence-Based Software Engineering. (s.f.). *Software engineering evidence map*. Recuperado de: <http://www.dur.ac.uk/ebse/evidence.php>.

Eito-Brun, R., y Sicilia, M. A. (2017). An innovation activity model for very small entities in the software sector: an empirical study. *R&D Management*, 47(5), E13-E25. doi: <http://dx.doi.org/10.1111/radm.12226>

García, A., y Amarillo, P. (2009). Marco de evaluación del proceso de pruebas funcionales utilizado en el CES con extensión para pruebas automatizadas. Tesis de grado, Facultad de Ingeniería, Instituto de Computación, Universidad de la República, Montevideo, Uruguay.

García, C., Dávila, A., y Pessoa, M. (2014). Test process models: systematic literature review. *Communications in Computer and Information Science*, 477, 84-93. doi: 10.1007/978-3-319-13036-1_8.

Garousi, V., Felderer, M., y Hacaloğlu, T. (2017). Software test maturity assessment and test process improvement: a multivocal literature review. *Information and Software Technology*, 85, 16-42. doi: <https://doi.org/10.1016/j.infsof.2017.01.001>

Garousi, V., y Mäntylä, M. V. (2016). A systematic literature review of literature reviews in software testing. *Information and Software Technology*, 80(C), 195-216. doi: 10.1016/j.infsof.2016.09.002

Hurtado, D. K. (2012). Diseño de un sistema de calidad de pruebas de software para una empresa de servicios basados en el estándar ISO 9126. Tesis de grado, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador.

ICONTEC (s.f.). *Instituto Colombiano de Normas Técnicas*. Recuperado de: <http://www.icontec.org/Paginas/Home.aspx>

ISO/IEC. (2001a). *ISO/IEC 9126 Software engineering - Product quality*. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC (2001b). *ISO/IEC 14598-6:2001 Software engineering — Product evaluation — Part 6: Documentation of evaluation modules*. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2005). *ISO/IEC 17011 Requisitos generales para la competencia de los laboratorios de ensayo y de calibración*. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2011). *ISO/IEC 25010 Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC (2013). *ISO/IEC/IEEE 29119-2:2013 Software and systems engineering - Software testing - Part 2: Test processes*. Geneva, Switzerland: International Organization for Standardization.

- ISO/IEC (2014). *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*. Geneva, Switzerland: International Organization for Standardization.
- ISO/IEC. (2016a). *ISO/IEC 25022 Systems and software engineering - Systems and software quality requirements and evaluation (SQuaRE) - Measurement of quality in use*. Geneva, Switzerland: International Organization for Standardization.
- ISO/IEC. (2016b). *ISO/IEC 25023 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of system and software product quality*. Geneva, Switzerland: International Organization for Standardization.
- ISO/IEC. (2017). *ISO/IEC 17025 Requisitos generales para la competencia de los laboratorios de ensayo y de calibración*. Geneva, Switzerland: International Organization for Standardization.
- Jung, H. J., Jung, W. T., y Yang, H. S. (2006). A study on the standard of software quality testing. *Computational Science and Its Applications - ICCSA 2006*, 1052-1059. doi: https://doi.org/10.1007/11751632_113
- Khan, H. H. y Malik, N. M. (2017). Software standards and software failures: a review with the perspective of varying situational contexts. *IEEE Access*, 5, 17501-17513.
- Kollanus, S., y Koskinen, J. (2009). Survey of software inspection research. *Open Software Engineering Journal*, 3, 15-34.
- Lara, M., (2013). Pruebas de software: capacitación y guía para el desarrollo de software. Recuperado de: <http://materias.fi.uba.ar/7548/Pruebas-Intro.pdf>
- Loaiza, V. C. y Zorro, L. C. (2010). Herramienta para la administración de requerimientos de los proyectos de las asignaturas de Ingeniería de Software y Arquitectura de Software de la Pontificia Universidad Javeriana. Tesis de pregrado, Pontificia Universidad Javeriana. Bogotá, Colombia.
- Martínez, S., Arango, S., y Robledo J. (2015). El crecimiento de la industria del software en Colombia: un análisis sistémico. *Revista EIA*, 12(23), 95-106.
- Mera-Paz, J. A. (2016). Análisis del proceso de pruebas de calidad de software. *Ingeniería Solidaria*, 12(20), 163-176. doi: <https://doi.org/10.16925/in.v12i20.1482>
- Monteiro, P., Machado, R. J., y Kazman, R. (2009). Inception of software validation and verification practices within CMMI Level 2. *ICSEA'09 Fourth International Conference on Software Engineering Advances*. IEEE, Porto, Portugal.
- ONAC. (s.f). *Organismo Nacional de Acreditación de Colombia*. Recuperado de: <https://onac.org.co/>
- Otálora, J., Bermeo, H., y González, D. (2009). Caracterización de la innovación tecnológica y el capital relacional en la industria del software de Ibagué, Colombia. *Puente*, 3(2), 29-37. doi: <http://dx.doi.org/10.18566/puente.v3n2.a03>
- Salazar, G., Lizano, F. y Lara, V. (2011). Propuesta de un sistema de gestión de la calidad para un laboratorio de pruebas. *Memorias de Informática*, La Habana, Cuba.
- Valdez, L. E., Pérez, D. G., Maldonado, G. (2017). TIC y la gestión del conocimiento como elementos determinantes del crecimiento de la PyME. *Investigación y Ciencia de la Universidad Autónoma de Aguascalientes*, 70, 50-62.
- Van Veenendaal, E. (2012). *Test Maturity Model Integration (TMMi) Release 1.0*. TMMi Foundation. Recuperado de: <https://www.tmmi.org/wp-content/uploads/2016/09/TMMi.Framework.pdf>
- Van Veenendaal, E., Grooff, R., y Hendriks, R. (2007). Test process improvement using TMM(i). *STAR Tester*, 19, 1-7. Recuperado de [http://www.erikvanveenendaal.nl/UK/files/Test Process Improvement using TMM\(i\).pdf](http://www.erikvanveenendaal.nl/UK/files/Test Process Improvement using TMM(i).pdf)
- Van Veenendaal, E., y Cannegieter, J. J. (2011). *The little TMMi: Objective-Driven Test Process Improvement*. Hertogenbosch, Países Bajos: UTN Publishers.

Verner, J. M., Brereton, O. P., Kitchenham, B. A., Turner, M., y Niazi, M. (2014). Risks and risk mitigation in global software development: a tertiary study. *Information and Software Technology*, 56(1), 54-78. doi: doi>10.1016/j.infsof.2013.06.005

Zamora, J. (2011). Análisis de los procesos de verificación y validación en las organizaciones software. Proyecto de fin de carrera, Universidad Carlos III de Madrid, Leganés, España.

1. Bioingeniera, Magíster en Ingeniería, Grupo GIGAT del Centro de Servicio y Gestión Empresarial del Servicio Nacional de Aprendizaje SENA, Medellín, Colombia. Correo electrónico: smvelasquez@sena.edu.co
 2. Ingeniero, Tecnólogo en Análisis y Sistemas de Información, Grupo GIGAT del Centro de Servicio y Gestión Empresarial del Servicio Nacional de Aprendizaje SENA, Medellín, Colombia. Correo electrónico: jdvahos@misena.edu.co
 3. Tecnólogo en Análisis de Desarrollo y Sistemas de Información, Grupo GIGAT del Centro de Servicio y Gestión Empresarial del Servicio Nacional de Aprendizaje SENA, Medellín, Colombia. Correo electrónico: clopez030@misena.edu.co
 4. Ingeniera de Sistemas, Grupo GIGAT del Centro de Servicio y Gestión Empresarial del Servicio Nacional de Aprendizaje SENA, Medellín, Colombia. Correo electrónico: mega2808@misena.edu.co
 5. Tecnóloga en Formulación de Proyectos, Grupo GIGAT del Centro de Servicio y Gestión Empresarial del Servicio Nacional de Aprendizaje SENA, Medellín, Colombia. Correo electrónico: jlondonog@sena.edu.co
 6. Tecnólogo en Formulación de Proyectos, Grupo GIGAT del Centro de Servicio y Gestión Empresarial del Servicio Nacional de Aprendizaje SENA, Medellín, Colombia. Correo electrónico: spatinoh@sena.edu.co
-

Revista ESPACIOS. ISSN 0798 1015
Vol. 39 (Nº 50) Año 2018

[Índice]

[En caso de encontrar algún error en este website favor enviar email a webmaster]