

# **Análisis sistemático de información de la Norma ISO 25010 como base para la implementación en un laboratorio de Testing de software en la Universidad Cooperativa de Colombia Sede Popayán**

## **Systematic analysis of ISO 25010 information as a basis for implementation in a software testing laboratory at the Cooperative University of Colombia Headquarters Popayán**

Julián Andrés Mera Paz<sup>1</sup>, Mari Yicel Miranda Gómez<sup>2</sup>, Sammy Cuaran Rosas<sup>3</sup>

<sup>1</sup>Docente Facultad de Ingenierías, Universidad Cooperativa de Colombia, <sup>2,3</sup>Auxiliar de Investigación, Facultad de Ingenierías, Universidad Cooperativa de Colombia

<sup>1</sup>julian.mera@campusucc.edu.co, <sup>2</sup>mari.mirandag@campusucc.edu.co, <sup>3</sup>sammy.cuaranr@campusucc.edu.co

**Resumen**— Este artículo se centra en el análisis sistemático de información de la Norma ISO/IEC 25010, con el cual se pretende establecer los parámetros para la calidad del producto software y que ellos sean el punto base o de referencia para implementar un laboratorio de testing de software en la Universidad Cooperativa de Colombia sede Popayán, con el fin de ofrecer productos software con altos estándares y aseguramiento de la calidad. Se centra en el proceso investigativo correlacional e indagación con trabajo de campo, el propósito es medir el grado de correlación y viabilidad de adoptar como base para implementar un laboratorio de testing y de aseguramiento de la calidad la norma ISO/IEC 25010. Como resultado se establece una discusión sobre la temática al interior del grupo de investigación, la cual se basa en los soportes teóricos, diario de trabajo de campo y análisis sistemático de la información sobre la norma ISO/IEC 25010. Se tendrán unas conclusiones y recomendaciones para futuros investigadores o estudiantes que quieran tomar esta línea de investigación y contribuir al aseguramiento de la calidad y testeado de productos software.

**Palabras claves**— calidad, ISO 25010, producto, testing, software, usabilidad.

**Abstract**— This article focuses on the systematic analysis of information in ISO / IEC 25010, which aims to establish the parameters for the quality of the software product and that they are the base or reference point to implement a software testing laboratory in the Cooperative University of Colombia headquarters Popayán, to offer software products with high standards and quality assurance. The article focuses on the correlational research process and investigation with fieldwork, the purpose is to measure the degree of correlation and feasibility to adopt as a basis to implement a testing laboratory and quality assurance ISO / IEC 25010. As a result, a discussion is established on the subject within the research group, which is based on the theoretical supports, field work diary and systematic analysis of information on ISO / IEC 25010. Conclusions and recommendations for future researchers or students who want to take this line of research and contribute to the quality assurance and testing of software products.

**Keywords**— Quality, ISO 25010, product, testing, software, usability.

### **1. Introducción**

En los últimos años el auge de los dispositivos móviles (teléfonos tipo Smartphone, reproductores de audio portátil, asistentes personales digitales, navegadores GPS, tabletas, entre muchos más), ha aumentado, a su vez el desarrollo de diferentes aplicaciones para los dispositivos mencionados, por

consecuente es necesario e importante estudiar herramientas y establecer metodologías que permitan realizar pruebas a las aplicaciones y su comportamiento en los dispositivos, teniendo en cuenta que en desarrollo de aplicativos hay diferentes restricciones (capacidad de almacenamiento, conexión a internet, el sistema operativo, etc.).

La Facultad de Ingeniería de Sistemas se renovó con registro calificado donde el programa se denomina Ingeniería de sistemas por competencias y en su plan curricular hay asignaturas que se enfocan al desarrollo de aplicaciones para dispositivos móviles, se presenta un problema crítico ya que en la actualidad en la universidad cooperativa de Colombia sede Popayán no se cuenta con un laboratorio o espacio enfocado a las pruebas de calidad de desarrollo de software de aplicativos móviles, en el cual se puedan realizar prácticas y así desarrollar adecuadamente un proceso de enseñanza-aprendizaje, donde los estudiantes puedan completar integralmente las competencias del ser, saber y hacer.

A raíz del problema, se propone un análisis sistemático de la información de la norma ISO/IEC 25010 para identificar las herramientas, técnicas y metodologías que puedan ser base para la implementación de un laboratorio de testing para los desarrollos de aplicativos móviles en Popayán de la Universidad Cooperativa de Colombia.

Con el objeto de medir la calidad de un producto software, teniendo en cuenta los procesos y factores que impactaron en el mismo. Priorizando las pruebas que se le pueden realizar al producto software, con el análisis de cada uno de los procesos del ciclo de vida, realizando una investigación de cada una de las pruebas que tiene el desarrollo de software de aplicativos móviles como: funcionalidad, compatibilidad, interoperabilidad, localización, usabilidad y teniendo en cuenta los atributos que llevan a tener un aplicativo con estándares de calidad y basados en normatividad.

## 2. Contexto

En la actualidad los productos de software son parte importante e integral en nuestras actividades diarias; ejemplo de ello son los cajeros automáticos, los vehículos, los Smart phone, las tablets, los portátiles, relojes, televisores y muchos otros conectados a través del internet de las cosas. Se debe tener en cuenta que los productos software son creados, desarrollados e implementados por seres humanos y por ende en cualquiera de sus etapas de creación se puede presentar una equivocación, al generarse esa equivocación se puede llevar a un defecto, como la mala digitación, distracción al codificar, entre otras. “Si no se ha identificado ese defecto y el producto software se ejecuta, hay un alto riesgo de que no haga lo que debería hacer o el objeto para lo cual fue creado, es decir se genera un fallo o desperfecto”.

También los fallos se pueden presentar por situaciones del entorno, como la radiación, descarga eléctrica, contaminación, inundaciones, humedad, etc. Por tanto realizar los procesos, métodos, técnicas que ayuden a reducir los riesgos en los productos software, y lograr de esta manera que se identifiquen los defectos antes de que se ejecuten, con esto se previenen los fallos y se puede brindar un adecuado aseguramiento de la calidad.

Con el fin de garantizar el aseguramiento de calidad se han creado una serie de pruebas, como se observa en la Tabla 1.

**Tabla 1.** Enfoques de las pruebas.

| Nombre de la prueba   | Descripción   |
|-----------------------|---|
| Pruebas Unitarias     | Se focaliza en ejecutar cada módulo, lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores.  |
| Prueba de Integración | Identificar errores introducidos por la combinación de programas probados unitariamente.  |
| Prueba de Regresión   | Determinar si los cambios recientes en una parte de la aplicación tienen efecto adverso en otras partes   |
| Pruebas de Humo       | Detectar los errores en releases tempranos y de manera fácil probando el sistema constantemente. Con lo que se aseguran los resultados de las pruebas unitarias y se reducen los riesgos. |
| Pruebas del Sistema   | Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación.  |
| Pruebas de Desempeño  | Las pruebas de desempeño miden tiempos de respuesta, índices de procesamiento de transacciones y otros requisitos sensibles al tiempo   |
| Pruebas de Carga      | Verificar el tiempo de respuesta del sistema para transacciones o casos de uso de negocios, bajo diferentes condiciones de carga.   |

|  |  |
|--|--|
| Pruebas de Stress                              | Verificar que el sistema funciona apropiadamente y sin errores, con memoria baja o no disponible en el servidor o con un número máximo número de clientes conectados. También con múltiples usuarios desempeñando la misma transacción con los mismos datos. |
| Pruebas de Volumen                             | Verificar que el sistema trabaja bien con un máximo número de clientes conectados y todos ejecutando la misma función por un período extendido. También verificar con un tamaño máximo de base de datos y múltiples consultas ejecutadas simultáneamente     |
| Pruebas de tolerancia                          | Verificar que los procesos de recuperación restauran apropiadamente la Base de datos, aplicaciones y sistemas, y los llevan a un estado conocido o deseado.  |
| Prueba de Múltiples Sitios                     | Detectar fallas en configuraciones y comunicaciones de datos entre múltiples sitios.   |
| Pruebas de Integridad de Datos y Base de Datos | Asegurar que los métodos de acceso y procesos funcionan adecuadamente y sin ocasionar corrupción de datos.   |
| Pruebas de Seguridad y Control de Acceso       | Nivel de seguridad de la aplicación: Verifica que un actor solo pueda acceder a las funciones y datos que su usuario tiene permitido.  |
| Pruebas del Ciclo del Negocio                  | Verificar la navegación a través de los objetos de la prueba reflejando las funcionalidades del negocio y requisitos, se realiza una navegación ventana por ventana, usando los modos de acceso (tabuladores, movimientos del mouse, teclas rápidas, etc).   |
| Pruebas de Configuración                       | Validar y verificar que el cliente del sistema funciona apropiadamente en las estaciones de trabajo recomendadas.  |

|                       |  |
|-----------------------|--|
| Prueba de Estilo      | Comprobar que la aplicación sigue los estándares de estilo propios del cliente.  |
| Prueba de Instalación | Verificar y validar que el sistema se instala apropiadamente en cada cliente con instalaciones nuevas y actualizaciones    |
| Prueba de Campo       | Correr el sistema en el ambiente real para encontrar errores y validar el producto contra sus especificaciones originales. |
| Pruebas Beta          | Realizar la validación del sistema por parte del usuario   |

Fuente:

<http://repositorio.unan.edu.ni/2371/1/pruebas%2520de%2520software%2520para%2520dispositivos%2520m%25c3%2593viles%2520andro id.pdf>

Se han establecido unos principios que permiten establecer unas pautas comunes para que las empresas de desarrollo de software y personas dedicadas al desarrollo de productos software (ISTQB 2.011)

**Principio 1. Las pruebas demuestran la presencia de defectos.**

Las pruebas son unas herramientas que permiten identificar la presencia de defectos; sin embargo, no garantizan que no haya defectos ocultos en el software.

**Principio 2. Las pruebas exhaustivas no existen.**

“Probar todo un aplicativo de extremo a extremo con todas las entradas de datos y condiciones es algo imposible”.

**Principio 3. Pruebas tempranas.**

Identificar los defectos en etapas tempranas, cuanto más rápido se identifiquen los defectos, más se ahorrará la empresa en todo tipo de recursos.

**Principio 4. Agrupación de defectos.**

Las pruebas deben concentrarse de manera proporcional. Por lo general, la mayoría de los fallos operativos se concentran en un número reducido de módulos.

**Principio 5. Paradoja del pesticida.**

Si se repite la misma prueba una y otra vez, la misma serie de casos de prueba dejará de encontrar nuevos defectos.

**Principio 6. Las pruebas dependen del contexto.**

Las pruebas dependerán del contexto en el cual se ejecuten.

**Principio 7. Falacia de ausencia de errores.**

La detección y corrección de los defectos no sirven de nada si el sistema no cumple con los requerimientos o necesidades del usuario.

Estos parámetros y/o postulados permiten en marcar la importancia de brindarle un aseguramiento a la calidad de software y por ello el equipo de investigación ha decidido analizar sistemáticamente la información sobre la Norma ISO/IEC 25010 como eje central para orientar el estudio sobre la implementación de un laboratorio de software, sin desmeritar otros estándares normativos como ISTQB, ISO/IEC 29119, ISO/IEC 9126, ISO/IEC 9241 los cuales se utilizaran para fortalecer y estructurar de una mejor manera una guía con métricas, técnicas y métodos que permitan garantizar la calidad de un producto software.

**3. Metodología**

Para el proyecto de investigación Para el desarrollo de la investigación se va a realizar investigación documental, audiovisual y presencial donde se realizara salidas de campo y videoconferencias con actores de distintas universidades, gobierno, salud y sector de tecnologías de Información y comunicaciones, donde se pretende debatir, centrar y consolidar la importancia de la implementación de un laboratorio de Testing de software para desarrollo de Aplicativos Móviles en la Universidad Cooperativa de Colombia Sede Popayán, a través de ello se generará nuevo conocimiento, debido a que se generará relaciones no conocidas entre el problema y las diferentes variables y/o soluciones que se hayan implementado en las empresas.

Para el proceso investigativo se ha implementado la observación redactada en el diario de campo con el contexto internacional, nacional y local, se ha realizado entrevistas de indagación a Docentes de la facultad de ingeniería de la Universidad Cooperativa de Colombia sede Popayán, como grupo focal se trabajó en el debate “la importancia del proceso de pruebas” con algunos estudiantes del programa Ingeniería de sistemas por

competencias, que pertenecen al semillero de investigación Synap.

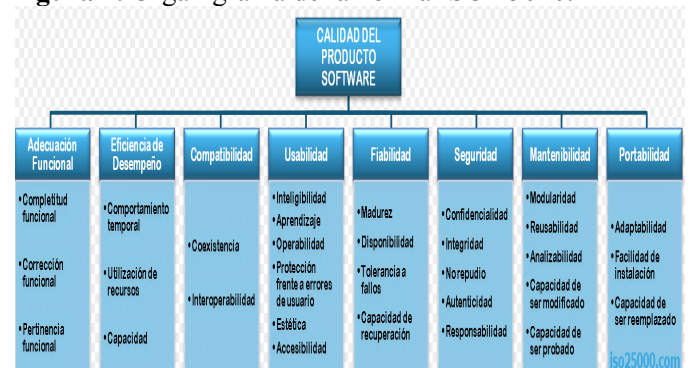
**4. ISO/IEC 25010**

La norma ISO/IEC 25010 hace parte de la familia de normas ISO 25000. Es una norma que está centrada hacia la usabilidad, en el cual se determinan las características de calidad que se deben tener en cuenta en el momento de evaluar las propiedades de un producto software terminado.

Se define que la calidad del producto software se puede tomar como el grado en que satisface los requisitos de sus usuarios, aportando de esta forma valor. Se trata de medir la calidad del producto software. Se puede observar el organigrama en la figura 1.

- *Adecuación Funcional*: referente a completitud, corrección y pertinencia funcional
- *Eficiencia de desempeño*: referente a medir comportamiento temporal, utilización de recursos, capacidad
- *Compatibilidad*: referente a medir la coexistencia, interoperabilidad
- *Usabilidad*: referente a medir capacidad para reconocer su adecuación, capacidad de aprendizaje, capacidad para ser usado, protección contra errores de usuario, estética de la interfaz de usuario, accesibilidad
- *Fiabilidad*: referente a medir madurez, disponibilidad, tolerancia a fallos, capacidad de recuperación.
- *Seguridad*: referente a medir confidencialidad, integridad, no repudio, responsabilidad y autenticidad.

**Figura 1.** Organigrama de la norma ISO 25010.



Fuente: <http://www.iso25000.com/index.php/normas-iso-25000/iso-25010>.

- *Mantenibilidad*: referente a medir modularidad, reusabilidad, analizabilidad, capacidad para ser modificado, capacidad para ser probado.
- *Portabilidad*: referente a medir adaptabilidad, capacidad para ser instalado y capacidad para ser reemplazado.

## 5. Resultados

Con el análisis sistemático de la información se obtiene una guía para la utilización de métricas, técnicas, métodos y buenas prácticas para implementar un laboratorio de testing.

Esto dará como resultado que cada uno de los productos software sometidos a pruebas en el laboratorio de testing de la Universidad cooperativa de Colombia sede Popayán, tengan unos estándares altos de aseguramiento de la calidad.

Este proyecto genera también un informe de investigación, artículo y hace parte de un capítulo de monografía de trabajo de pregrado que será un material de consulta, transferencia de conocimiento e invitación a la investigación para la comunidad universitaria.

## 6. Conclusiones

La ISO/IEC 25010 enmarca la calidad en los productos software, garantizando también la articulación de los procesos para obtener los productos, por tanto es una referencia óptima para la base de implementar un laboratorio de testing.

El tener la percepción de investigadores a nivel local, nacional e internacional permite explorar, analizar y profundizar en las características, componentes, técnicas y métodos que orientan el aseguramiento de calidad de software.

## 7. Recomendaciones

Se sugiere a futuros investigadores o estudiantes interesados en el tema de aseguramiento de la calidad, estar actualizándose en los estándares reconocidos por las entidades internacionalmente reconocidas como la ISO.

Investigar en profundidad el tema de la ISO 25010 y potenciar sus características para difundirlas entre la comunidad universitaria y empresas de desarrollo de software.

## 8. Bibliografía

- [1] Paz, J. M. (2016). Análisis del proceso de pruebas de calidad de software. *Ingeniería solidaria*, 12(20).
- [2] Masri, W., & Zaraket, F. A. (2016). Coverage-Based Software Testing: Beyond Basic Test Requirements. *Advances in*

- Computers.*, pagina. 1-30, 2016 [en línea]. Disponible en: doi: 10.1016 /bs.adcom.2016.04.003
- [3] Deak, A., Stálhane, T., & Sindre, G. (2016). Challenges and strategies for motivating software testing personnel. *Information and Software Technology*, vol.73 paginas. 1-15, 2016 [en línea]. Disponible en: <http://dx.doi.org/10.1016/j.infsof.2016.01.002>
- [4] Ceballos Guerrero, R. (2011). Técnicas automáticas para la diagnosis de errores en software diseñado por contrato. Tesis doctoral en lenguajes y sistemas informáticos, Universidad de Sevilla, pagina 4-54 Sevilla – España 2.011 Disponible en: <https://dialnet.unirioja.es/servlet/tesis?codigo=24794>
- [5] J. O. Navarro, “Estado del arte de métodos, tipos de testing y herramientas para aplicar pruebas de rendimiento”, tesis de grado, Fundación Universitaria Tecnológica de Comfenalco Cartagena, Colombia, 2010
- [6] D. (2013). Disponible en: <http://documents.mx/documents/estado-del-arte-de-metodos-tiposde-testing-y-herramientas-para-aplicar-pruebasde-rendimiento.html>
- [7] Oviedo Vargas, S., & Puello Marrugo, P. D. (2013). Diagnóstico para la implementación de hojas de rutas en la certificación de la industria desarrolladora de software en Cartagena De Indias (Doctoral dissertation, Universidad de Cartagena). Colombia, 2013 [en línea]. Disponible en: <http://190.242.62.234:8080/jspui/handle/11227/391>
- [8] Pardo, C., Hurtado, J. A., & Collazos, C. A. (2010). , “Mejora de procesos de software ágil con Agile spi Process”, *Revista dyna*. Disponible en: <http://www.revistas.unal.edu.co/index.php/dyna/article/view/25595>
- [9] Muñoz, C. C., Velthuis, M. G. P., & de la Rubia, M. Á. M. (2010). *Calidad del producto y proceso software*. Editorial Ra-Ma.
- [10] MIT (2.016), Massachusetts Institute of Technology, “Programa en Ciencia, Tecnología y Sociedad, junio 2016, Estados Unidos [en línea]. Disponible en: <http://web.mit.edu/sts/>
- [11] Enriquez, J. G., & Casas, S. I. (2014). Usabilidad en aplicaciones móviles. *Informes Científicos-Técnicos UNPA*, 5(2), 25-47.
- [12] Navarro, J. M., & Garzás, J. (2010). Experiencia en la implantación de CMMI-DEV v1. 2 en una micropyme con metodologías ágiles y software libre. *Revistas española de innovación, calidad e ingeniería del software*, 6(1), 6-15.
- [13] León Perdomo, Y., Enrique Góngora Rodríguez, A., & Febles Estrada, A. (2013). Aplicando métricas de calidad a proyectos y procesos durante las pruebas exploratorias. *Revista Cubana de Ciencias Informáticas*, 7(2), 193-205.
- [14] Mejía, R., Mauricio, G., Londoño, L., & Felipe, C. (2011). Diseño de juegos para el cambio social. *Kepes*, 8(7).
- [15] Estayno, M. G., Dapozo, G. N., Cuenca Pletsch, L. R., & Greiner, C. L. (2009). Modelos y Métricas para evaluar Calidad de Software. In *XI Workshop de Investigadores en Ciencias de la Computación*.
- [16] Mascheroni, M. A., Greiner, C. L., Petris, R. H., Dapozo, G. N., & Estayno, M. G. (2012). Calidad de software e ingeniería de

- usabilidad. In XIV Workshop de Investigadores en Ciencias de la Computación.
- [17] Ruiz, G. A., Palacio, J. A. P., Castro, C. A. C., Alaguna, A., Areiza, L. M., & Rincón, R. D. (2006). Modelo de Evaluación de Calidad de Software Basado en Lógica Difusa, Aplicada a Métricas de Usabilidad de Acuerdo con la Norma ISO/IEC 9126. *RASI*, 3(2), 25-29.
- [18] Lompfrey, G., & Hernandez, S. (2008). La importancia de la Calidad en el desarrollo de productos de software, 7(2),20
- [19] Lamancha, B. P., & Polo, M. (2009). Generación automática de casos de prueba para Líneas de Producto de Software. *Innovación, Calidad e Ingeniería del Software*, 5(2), 17.
- [20] Puello, O. (2013). Modelo de verificación y Validación basado en CMMI. *Investigación e Innovación en Ingenierías*, 1(1).
- [21] ATOS-ORIGIN, S. A. (2008). TestPAI: Un área de proceso de pruebas integrada con CMMI. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, 4(4), 6-20.
- [22] GÓNGORA, A. (2011). Catálogo automatizado de métricas de calidad para evaluar los productos en las pruebas. Máster en Calidad de Software, Universidad de Ciencias Informáticas.
- [23] ISO 25000 Calidad del producto de software: <http://iso25000.com/>
- [24] Díaz & Hilterscheid, Probador nivel básico de acuerdo al programa de estudios de ISTQB, Versión V.1.2a V ES.1.0.
- [25] Suárez, F.; Garzas, J. I Jornadas sobre Calidad del Producto Software e ISO 25000, Santiagode Compostela, 10 de junio de 2014.
- [26] Scalone, F. (2006). Estudios comparativos de los modelos y estándares de calidad del software. Buenos Aires: Universidad Tecnológica Nacional.
- [27] International Software Testing Qualifications Board [istqb], "Certified Tester Foundation Level Syllabus. Released version 2011", 2011 [en línea]. Disponible en: <http://www.istqb.org/downloads/send/2-foundation-level-documents/3-foundationlevel-syllabus-2011.html4>