

ARTÍCULO ORIGINAL

Buenas prácticas de ingeniería de software: pruebas de software

Good Practices of Software Engineering: Software Tests

Anays Gómez García

anaysgg@citmatel.inf.cu • <https://orcid.org/0000-0002-7515-4958>

Mercedes Sosa Hernández

mechuco@citmatel.inf.cu • <https://orcid.org/0000-0003-1322-812X>

EMPRESA DE TECNOLOGÍAS DE LA INFORMACIÓN Y SERVICIOS TELEMÁTICOS AVANZADOS, CITMATEL, CUBA

Sandra Verona Marcos

sverona@ceis.cujae.edu.cu • <https://orcid.org/0000-0002-2794-092X>

Martha Dunia Delgado Dapena

marta@ceis.cujae.edu.cu • <https://orcid.org/0000-0002-2601-3462>

UNIVERSIDAD TECNOLÓGICA DE LA HABANA "JOSÉ ANTONIO ECHEVERRÍA", CUJAE, CUBA

Recibido: 2023-01-28 • Aceptado: 2023-03-30

RESUMEN

Este trabajo aborda aspectos relacionados con las pruebas de *software*, desde la mirada de la normalización y las buenas prácticas empleadas en la actualidad nacional e internacional. Se plantean técnicas y herramientas de apoyo al proceso de pruebas. De igual manera, se introduce el estudio al modelo Mtest. Search, propuesto por el Grupo de Calidad y Pruebas de *Software* (CyPSoft), de la Facultad de Ingeniería Informática de la Universidad Tecnológica de La Habana José Antonio Echeverría (Cujae). Teniendo como premisa implantar mejoras al proceso de desarrollo de *software* de la empresa CITMATEL, se plantea una nueva forma de hacer, vinculando las normativas existentes en la organización con los aportes prácticos del Mtest.Search. Finalmente se propone una estrategia de abordaje de la automatización de pruebas de *software* en el proceso de producción de *software* CITMATEL, que cuenta con tres etapas fundamentales centradas en la adopción de buenas prácticas, en toda la estructura organizativa vinculada a este proceso. El aporte de la investigación va destinado a trazar un camino inicial en la vinculación entre el qué hacer y

el cómo hacer, durante el proceso de evaluación de la calidad del producto de *software*, a partir de la implementación de los resultados de la academia en el sector empresarial.

PALABRAS CLAVE: aseguramiento de la calidad, pruebas de *software*, automatización de pruebas de *software*, Mtest.Search, GeCodP.

ABSTRACT

This paper addresses aspects related to software testing from the perspective of standardization and good practices currently used nationally and internationally. Techniques and tools to support the testing process are proposed. Similarly, the study is presented to the Mtest.Search model proposed by the Software Quality and Testing Group (CyPSoft) of the Faculty of Computer Engineering of the Cujae. Having as a premise to implement improvements to the software development process of the CITMATEL company, a new way of doing is proposed, linking the existing regulations in the organization with the practical contributions of Mtest.Search. Finally, a strategy to approach the automation of software tests in the CITMATEL software production process is proposed, which has three fundamental stages focused on the adoption of good practices throughout the organizational structure linked to this process. The contribution of the research is intended to trace an initial path in the link between what to do and how, during the process of evaluating the quality of the software product, from the implementation of the results of the academy in the sector.

KEYWORDS: *quality assurance, software testing, software test automation, Mtest.Search, GeCodP.*

INTRODUCCIÓN

La complejidad de los sistemas de *software* ha aumentado a un nivel sin precedentes. Esto ha llevado a nuevas oportunidades y mayores desafíos para las organizaciones que crean y utilizan sistemas. Estos desafíos existen a lo largo del ciclo de vida de un sistema y en todos los niveles del detalle arquitectónico (González *et al.*, 2015). La automatización de pruebas en las organizaciones es cada vez más frecuente por diversas razones, entre ellas, mayor reusabilidad, mejor cobertura de prueba y, por consiguiente, mayor calidad en el producto desarrollado.

La Empresa de Tecnologías de la Información y Servicios Telemáticos Avanzados (en lo adelante CITMATEL), tiene entre sus actividades principales la producción de *software*. En

el proceso de pruebas realizadas a los productos de *software*, se han manifestado deficiencias que atentan contra su calidad y, por supuesto, la del producto final.

Como una alternativa de solución, en este trabajo se propone una estrategia de abordaje de la automatización de pruebas de *software* en el proceso de producción de *software* de la empresa, que cuenta con tres etapas fundamentales centradas en la adopción de buenas prácticas, en toda la estructura organizativa vinculada a este proceso. Se muestran además los resultados de la ejecución de las tareas comprendidas en la primera etapa.

METODOLOGÍA

Para el desarrollo de esta investigación fueron empleados métodos y técnicas (Alonso, 2017) de análisis y síntesis, con el objetivo de consultar la bibliografía relacionada con el proceso de pruebas de *software* en la industria cubana del *software*; el análisis documental y el método histórico-lógico, para indagar sobre el avance alcanzado en el tema.

UNA MIRADA DESDE LA NORMALIZACIÓN

Para garantizar la calidad de *software* (Callejas, 2017; Pressman, 2010) es importante implementar algún modelo o estándar de calidad, que permita la gestión de atributos en el proceso de construcción de *software*, teniendo en cuenta que la concordancia de los requisitos y su construcción son la base de las medidas de calidad establecidas (Callejas, 2017).

A lo largo del tiempo se han desarrollado diferentes modelos para evaluar la calidad del *software*, que intentan descomponer la calidad en una categoría de características más sencillas. En Callejas (2017) se plantea un estudio del estado del arte de los modelos de calidad de *software*, quedando reflejados los principales modelos según su clasificación:

- A nivel de procesos ITIL, ISO/IEC 15504 (INGERTEC, 2021), Dromey, Cobit 4.0, CMMI (Team, 2010), Bootstrap, PSP, TPS, ISO 90003, ISO IEC IEEE 12207 (ISO/IEC/IEEE, 2017);
- A nivel de producto McCall, GQM, Boehm, FURPS, Gilb, ISO/IEC 9126-1, SQAe o familia de normas SQuaRE (Normalización, 2011, 2016a, 2016b, 2017), WebQEM.

Otros modelos de calidad de *software* han sido desarrollados con el aumento de las crecientes necesidades de la industria de *software* iberoamericana, de ser más competitiva a nivel global, tal es el caso de los modelos:

- MoProSoft (Oktaba *et al.*, 2005), COMPETISOFT (COMPETISOFT, 2006), MPS.BR (SOFTEX, 2012), Si.MPS.Cu (Trujillo *et al.*, 2014), CMMI-DEV® (Team, 2010), ISO IEC IEEE 15288, entre otros.

En el ámbito nacional se dispone de una serie de normas internacionales adoptadas como cubanas y con un Modelo de Calidad para el Desarrollo de Aplicaciones Informáticas (MCDAI) (COMUNICACIONES, 2021a, 2021b, 2021c), resultado de proyectos de investigación (Alvarado *et al.*, 2018; Lazo *et al.*, 2016; Oro *et al.*, 2019) ejecutados por las entidades CALISOFT, UCI, SEISA, Cujae y XETID; de revisiones realizadas por el Subcomité 7 Ingeniería de *software* y Sistemas (SC7), en los períodos 2014-2016 y 2016-2020, respectivamente, y del quehacer de la industria cubana del *software*.

En 2020 fue aprobado como norma ramal, NRCM 3-1:2020 Requisitos de la Calidad para los Sistemas y las Aplicaciones Informáticas, que especifica los requisitos para el desarrollo de *software*, cuando una organización pretende: mejorar su proceso productivo, validar el nivel de madurez (básico, intermedio o avanzado) de los procesos o aumentar la satisfacción de sus clientes.

Las pruebas de *software* tienen presencia en todos los niveles de madurez definidos por el MCDAI. En COMUNICACIONES (2021a) pueden consultarse los resultados esperados para cada nivel, así como la descripción de cada uno de los requisitos.

A partir del diagnóstico realizado por CALISOFT a la empresa CITMATEL para determinar la conformidad con el nivel intermedio de este modelo, se considera que su implementación no supone modificaciones estresantes al proceso de pruebas, teniendo en cuenta que las actividades derivadas de los requisitos para este nivel se encuentran previamente institucionalizadas en la empresa.

ENFOQUE ÁGIL EN LA ACTIVIDAD DE PRUEBAS DE SOFTWARE

Las pruebas de *software* son seguramente la actividad más común de control de calidad realizada en los proyectos de desarrollo o mantenimiento de aplicaciones y sistemas, es por ello que en dependencia de lo que se quiera probar se traza la estrategia, y se escoge el nivel, el tipo y el método de prueba que se va a utilizar (Marin *et al.*, 2020; Myers *et al.*, 2011).

Las guías tradicionales de desarrollo de *software* plantean una fase del ciclo de vida destinada a las actividades de pruebas de *software*, donde se debe garantizar mejor calidad del producto; no siempre sucede el resultado esperado, teniendo en cuenta el plan inicial y el real. Por su parte, los métodos ágiles valoran la obtención de *software* funcional, afirmando que esta es la medida principal de progreso. Para ello plantean desde el Manifiesto Ágil una serie de principios y valores reflejados en todas las metodologías ágiles, y a su vez están presentes en las buenas prácticas de ingeniería de *software* que estas proponen. La figura 1 muestra cuáles son las técnicas y prácticas ágiles más empleadas según los resultados del 15th Annual State of Agile Report (Digital.ai, 2022).

Aunque en esta edición del Annual State of Agile Report no se incluyen explícitamente técnicas y prácticas ágiles de pruebas de *software*. Se considera que están relacionadas, teniendo en cuenta el principio de pruebas tempranas (Benavides, 2020), apoyándose en las pruebas unitarias (Bandara & Perera, 2020; Rojas *et al.*, 2019; Uribazco *et al.*, 2019), el desarrollo guiado por pruebas (Abushama *et al.*, 2021), Desarrollo Guiado por Comportamiento (Lawrence & Rayner, 2019; Wynne *et al.*, 2017), entre otras variantes (Gärtner, 2013).

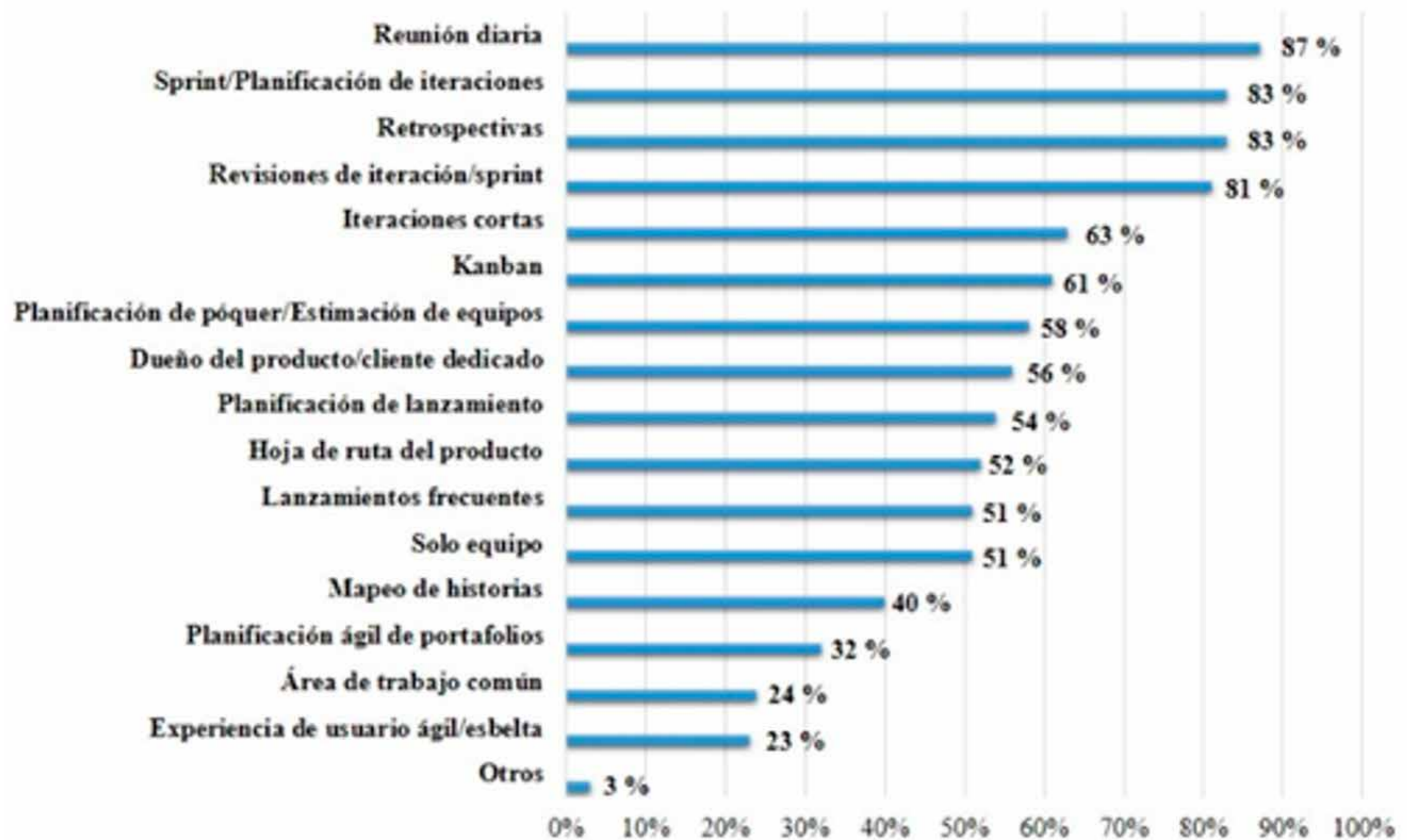


Fig. 1 Técnicas y prácticas ágiles más empleadas según (Digital.ai, 2022).

PRUEBAS DE SOFTWARE EN ENTORNOS PRODUCTIVOS

Los entornos productivos de desarrollo de *software* se caracterizan, entre otros aspectos, por la integración de sus actividades como un proceso, en donde las entradas corresponden a las necesidades del cliente y la salida representa el producto final correctamente implementado, de acuerdo con las entradas recibidas. A lo interno del proceso se ejecutan entre otras actividades, las pruebas de *software* que pueden ser automatizadas o no.

Según lo conceptualizado en Colorado & Plaza (2013), Godoy *et al.* (2020), Normalización (2021), Quintana *et al.* (2014) y Serna *et al.* (2017), en términos generales la automatización de pruebas consiste en utilizar *software* para ejecutar o apoyar las actividades del ciclo de vida de las pruebas de *software* (STLC por sus siglas en inglés), tales como el análisis de requisitos, la planificación de pruebas, el desarrollo de los casos de prueba, la configuración del entorno, la ejecución de pruebas y el cierre del ciclo de pruebas.

En Díaz *et al.* (2017) se presentan una serie de herramientas OpenSource, de apoyo a la etapa de pruebas al *software*, según el tipo de pruebas que se van a realizar. Este estudio se considera muy acertado a nuestra realidad —a pesar de ser resultado de investigaciones realizadas en la Universidad Nacional de La Plata, Argentina—, teniendo en cuenta el Decreto Ley N° 370, sobre la informatización de la sociedad cubana.

La investigación propuesta en Delgado *et al.* (2017) incluye las actividades para la ejecución de pruebas en un entorno productivo, así como el modelo de optimización que reduce la cantidad de casos de prueba funcionales, considerando criterios de cobertura de escenario y utilizando algoritmos de búsqueda heurística. Su fortaleza radica en los beneficios proporcionados por la relación universidad-empresa, a través de la interacción con el grupo de

investigación CyPSoft, que garantiza la novedad de los resultados y su actualización permanente con las mejores prácticas en la gestión de calidad y pruebas de *software* a nivel nacional e internacional (Delgado, 2019). El grupo de investigación CyPSoft afirma en Delgado (2019), que los principales beneficios que aporta el modelo MTest.Search a las empresas desarrolladoras de *software* y a los laboratorios de calidad, están orientados a elevar la calidad del procesos de pruebas y por ende la calidad del producto final, de forma tal que la confianza en las actividades de aseguramiento de la calidad para los desarrolladores, gestores de calidad y clientes de la empresa aumenta de forma significativa. Adicionalmente, el modelo contiene un conjunto de herramientas basadas en búsquedas que dan soporte a la generación de casos de pruebas funcionales. En Larrosa (2019) se plantea una herramienta informática que da soporte desde la práctica al modelo antes mencionado.

AUTOMATIZACIÓN DE PRUEBAS DE SOFTWARE EN EL PROCESO DE PRODUCCIÓN DE SOFTWARE DE CITMATEL

La Empresa de Tecnologías de la Información y Servicios Telemáticos Avanzados (CITMATEL), devenida Empresa de Alta Tecnología desde marzo de 2021, trabaja con una alta vocación innovadora e introduce sistemáticamente nuevas tecnologías con el objetivo de obtener resultados de mayor impacto en la sociedad y la economía del país.

Dispone desde 2013 de un Sistema Integrado de Gestión (SIG) con alcance para los sistemas Calidad, Medio ambiente y Seguridad y Salud en el trabajo, certificado por la Oficina Nacional de Normalización, según las normas cubanas NC ISO 9001:2015, 14001:2015 y 45001:2018, y sus homólogos certificables. La experiencia de trabajo bajo las premisas de un SIG son garantía de sus productos y servicios de cara a los clientes de la organización. Su matriz de procesos tiene definidos quince procesos divididos en estratégicos, de apoyo y operacionales. En este último grupo se encuentra el proceso P03 Producción de *software*, que tiene como objetivo: «garantizar con la implementación del proceso la calidad de los desarrollos, la preservación del medio ambiente y la seguridad y salud de los trabajadores».

En el proceso P03 intervienen varias áreas organizativas de la empresa, entre ellas, el Departamento de Gestión de la Calidad y Auditoría (DGCA), principal responsable de la evaluación y liberación de los productos de *software* resultado de la actividad productiva de las Unidades Empresariales de Base Proyectos Multimedia y Desarrollo de *Software*, respectivamente. Durante las pruebas realizadas a los productos de *software*, el DGCA evalúa algunas de las características del modelo de calidad del producto planteado en la norma NC ISO/IEC 25010:2016 (Normalización, 2016a): adecuación funcional, seguridad, usabilidad y eficiencia de desempeño; otras características son evaluadas cuando el formato del producto o los requisitos planteados por el equipo de desarrollo lo requieren.

Para la identificación de las oportunidades de mejora al proceso P03 Producción de *software* se toma como base la revisión por la dirección realizada de forma semestral, requisito de la norma NC ISO 9001:2015 Sistemas de Gestión de la Calidad-Requisitos; y el análisis —sin fines evaluativos— realizado en el marco de la transferencia científico-tecnológica del

Modelo de Generación Automática de Pruebas Basado en Búsquedas, MTest.Search, desarrollado en la Universidad Tecnológica de La Habana José Antonio Echeverría (Cujae), que tuvo como principal objetivo conocer el nivel de automatización de las pruebas de *software* en la entidad.

Fueron identificadas un conjunto de buenas prácticas de pruebas de *software* que se realizan a nivel organizacional y otras que pudieran ser implementadas a través de la adopción del modelo MTest.Search:

- Sistematizar el diseño de pruebas y el uso de indicadores para la evaluación de la calidad de los productos.
- Aplicar pruebas específicas para las diferentes clasificaciones de soluciones que se desarrollan: sistemas de gestión, sitios web, aplicaciones Android y desktop, entre otras.
- Definir una estrategia de pruebas a nivel organizacional.
- Elaborar procedimientos para el desarrollo de la actividad de pruebas, incluyendo el uso de técnicas de diseño, la estimación de coberturas de la prueba y el reúso de los componentes de pruebas.
- Seleccionar e introducir herramientas para el diseño, la generación, la ejecución y el control de incidencias en el proceso de pruebas.

A partir de los análisis realizados al proceso P03 Producción de *software*, para la introducción de buenas prácticas de pruebas de *software* en la organización, se proponen tres etapas de abordaje de la automatización (figura 2): una primera para cubrir la actividad del DGCA en los tipos de pruebas que ya hoy se realizan; una segunda para dar soporte automatizado a otros tipos de pruebas que no se alcanzan a realizar en los momentos actuales y que responden a la compatibilidad, fiabilidad, mantenibilidad y portabilidad; y una tercera para llegar hasta la actividad del propio equipo de desarrollo.

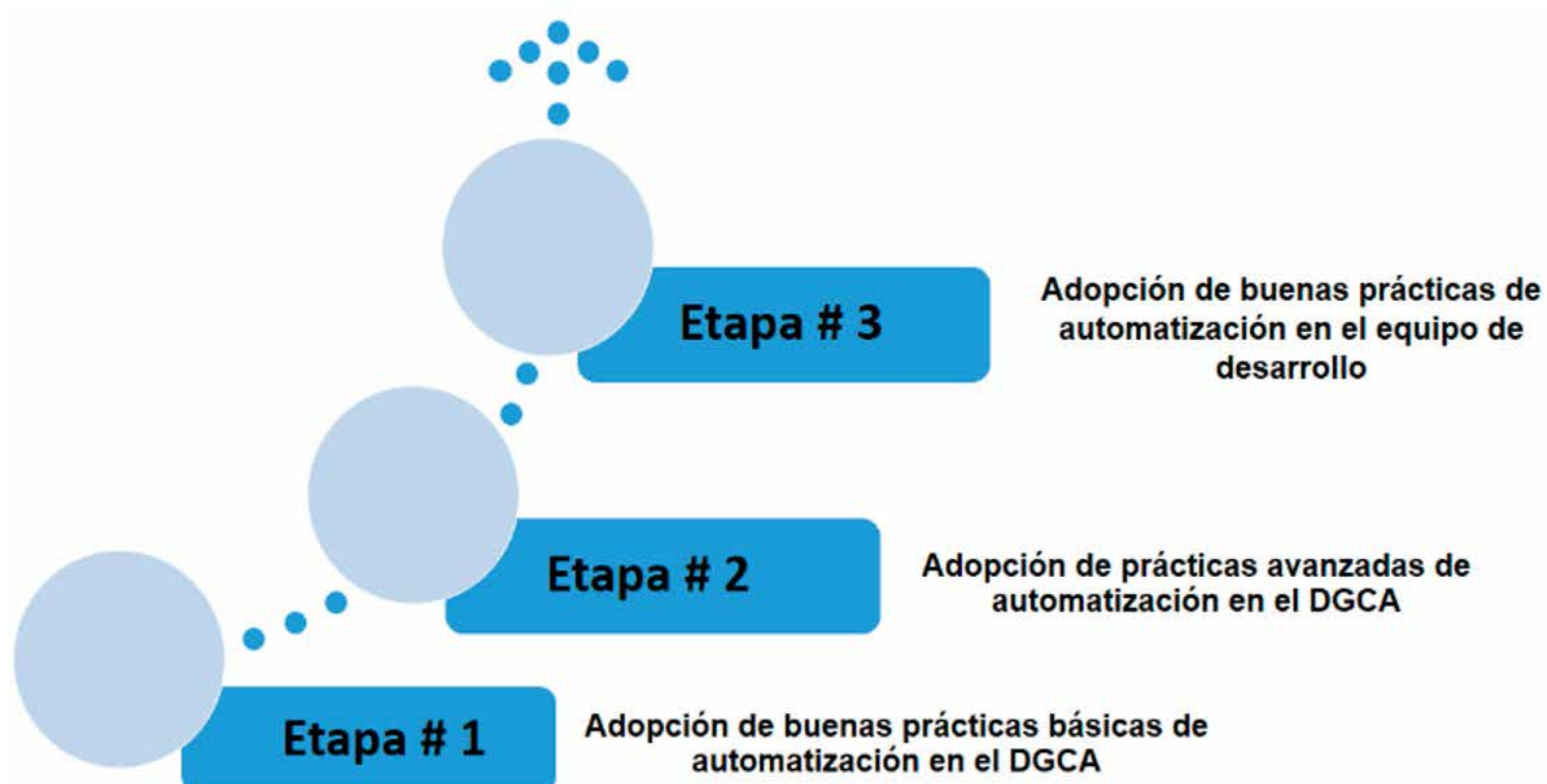


Fig. 2 Etapas de abordaje de la automatización (Fuente: elaboración propia).

RESULTADOS Y DISCUSIÓN

De la ejecución de las tareas comprendidas en la primera etapa de abordaje de la automatización, se obtienen como resultado las siguientes mejoras:

- El esquema normativo para la evaluación del producto de *software* es modificado (figura 3), incluyendo no solo normas y procedimientos, sino también buenas prácticas importadas de las metodologías ágiles y herramientas automatizadas de apoyo al proceso, específicamente GeCodP en su versión 1.0. El subesquema normativo se relaciona con el subesquema práctico, ya que el primero sienta las bases por las que pueden ser empleadas las buenas prácticas ágiles de ingeniería de *software* y proveen al modelo de la información necesaria para la ejecución de pruebas tempranas en el entorno de producción.

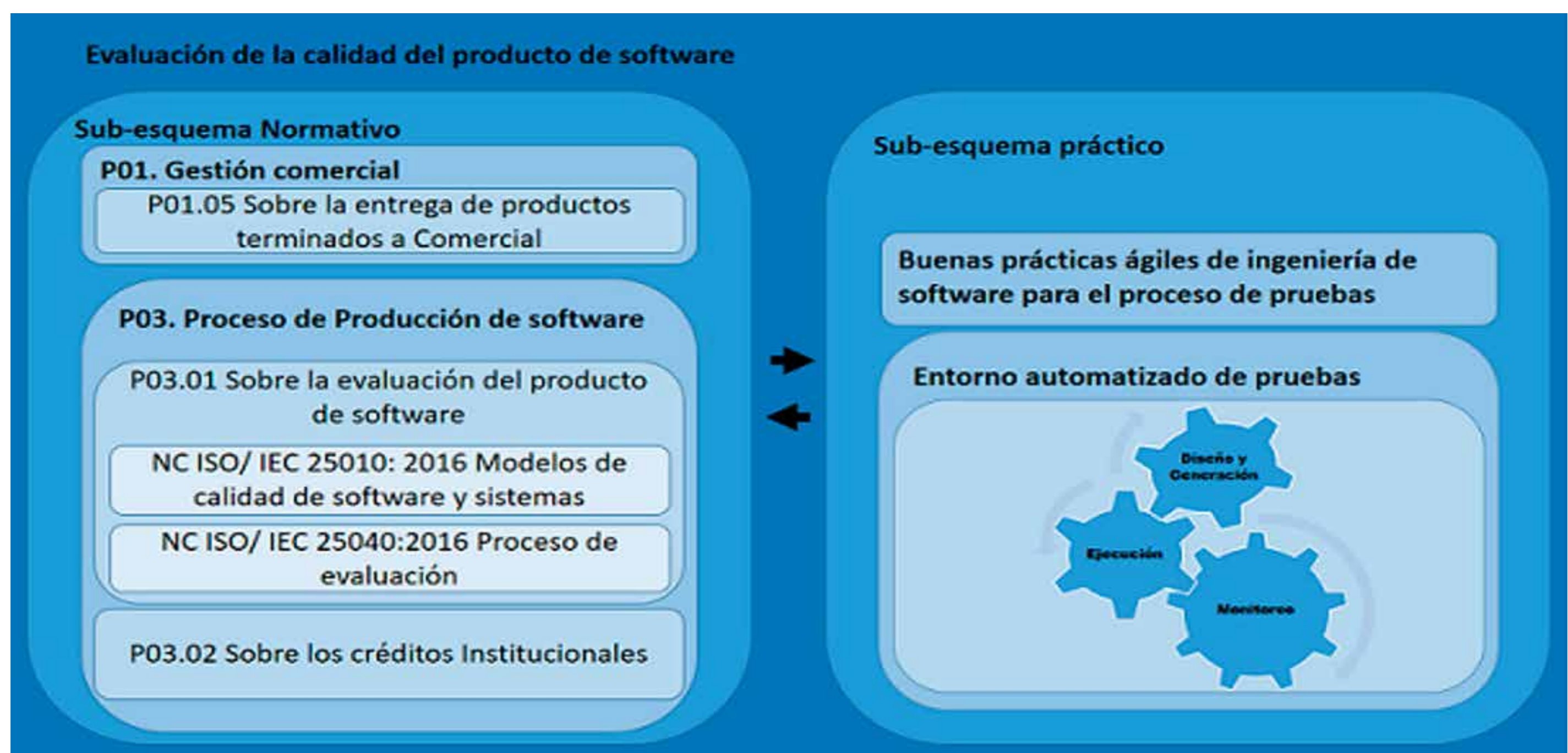


Fig. 3 Propuesta de esquema de evaluación de la calidad del producto de *software* en CITMATEL (Fuente: elaboración propia).

- El flujo de trabajo actual para la evaluación del producto de *software* en el DGCA, es modificado incluyendo actividades asociadas al diseño, la generación y la ejecución de los casos de prueba, de forma tal que responda a las buenas prácticas derivadas del Mtest. Search y mencionadas previamente.
- Propuestos un total de seis indicadores de eficacia para la evaluación del producto y el proceso, la tabla 1 muestra esta información. Se considera de suma importancia que estos puedan ser incluidos en la ficha del proceso P03 Producción de *software*, como parte de la mejora continua a este y respuesta al requisito genérico G10 Gestionar indicadores, planteado en el MCDAI, con vista a su certificación en la empresa. Se propone además que su evaluación sea de forma trimestral, incluyendo su resultado en el Informe de Revisión por la Dirección, y sea presentada a la alta dirección de la empresa para la toma de decisiones al respecto.

Teniendo en cuenta que en los momentos actuales los indicadores para la medición de la eficacia del proceso P03 Producción de *software* responden al cumplimiento de planes empresariales y de resguardo de información, el valor agregado de los propuestos por este colectivo de autores radica en la posibilidad de medir dentro del ciclo de desarrollo del producto de *software* : la eficacia del proceso pruebas de *software*, la conformidad con el producto desarrollado y la cobertura de las pruebas realizadas. Destacar que todos estos aspectos son de nueva y necesaria gestión para la organización, a partir de la categorización como Empresa de Alta Tecnología.

Tabla 1. Indicadores para medir la mejora del proceso de pruebas de *software* y el producto

Indicadores para medir		Calificación
Proceso		
1. Promedio de dictámenes técnicos realizados por proyecto.		
2. Estadía del desarrollo en calidad interna.		
3. Cumplimiento del ciclo de vida del producto.		
4. Estadía del desarrollo en el DGCA.		
Producto		
Conformidad de los proyectos entregados al DGCA para su revisión.		
Cantidad de funcionalidades probadas y las No conformidades asociadas.		
Evaluación total:		
Proceso eficaz:	Sí_____	No_____
Elaborado por:	Cargo:	Firma:

Criterio de Evaluación: se considera que el proceso es eficaz siempre que:

$$Ie \geq 4 \quad (1)$$

El indicador de eficacia (Ie) se calcula a partir de la fórmula:

$$Ie = \frac{\sum I_1 + I_2 + \dots + I_n}{n} \quad (2)$$

Siendo I₁, I₂, ..., I_n las puntuaciones alcanzadas por los indicadores 1, 2 y hasta n, respectivamente, donde n es el número máximo de indicadores definidos.

- Para la implantación de un entorno automatizado de pruebas en el DGCA, fueron analizadas algunas herramientas informáticas (Ç & Demir, 2020; Larrosa, 2019), que dan soporte al STLC. Destacar que con la transferencia de la herramienta GeCodP se cubre el desarrollo de los casos de pruebas.

En el esquema de abordaje de la automatización, las dos primeras etapas son ejecutadas en el marco de la tesis de maestría de un miembro de este colectivo de autores. Una vez finalizadas estas se prevé continuar hacia la tercera y última etapa, abarcando de forma gradual cada uno de los equipos de desarrollo que conforman las UEB que intervienen en el proceso

P03 Producción de *software*. Se considera de suma importancia que la ejecución se realice de forma gradual, para en la medida que lo requiera refinar la propuesta realizada.

CONCLUSIONES

El modelo Mtest.Search se encuentra alineado y contribuye a avanzar en los niveles de madurez del MCDAI, teniendo en cuenta que con su implementación se da cumplimiento a los requisitos básicos y genéricos del proceso base de pruebas de *software* del MCDAI.

Cada una de las tareas incluidas en las etapas de abordaje de la automatización, responden al cumplimiento de los requisitos del proceso base Pruebas de *Software* del MCDAI. Estas tareas serán ejecutadas de forma desescalada y gradual, es decir, desde el DGCA hacia los equipos de desarrollo y con un solo equipo de desarrollo en primera instancia, para trabajar en el refinamiento de la propuesta.

La implantación de la herramienta GeCodP en el entorno productivo empresarial de CIT-MATEL, contribuye a avanzar en los niveles de automatización del proceso de pruebas de la empresa, y facilita el diseño y la generación de los casos de prueba con un enfoque ágil, ya que implementa la técnica Desarrollo Guiado por Comportamiento.

REFERENCIAS

- Abushama, H. M., Alassam, H. A., & Elhaj, F. A. (2021, 26 Feb.-1 March 2021). The effect of Test-Driven Development and Behavior-Driven Development on Project Success Factors: A Systematic Literature Review Based Study. 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE),
- Alonso-Berenguer, I. & G.-S., A. (2017). Orientaciones metodológicas para la estructuración y redacción de un informe de tesis (C. M. R. García, Ed.). Edición cooperada Ediciones UO y GIDMAC.
- Alvarado, Y. L., Oro, L. T., Pérez, O. E., Montalván, D. P., & Osorio, Y. S. (2018). Apuntes sobre el modelo de la calidad para el desarrollo de aplicaciones informáticas (MCDAI) Notes on the quality model for the development of computer applications (MCDAI). VIII Taller Internacional de Calidad en las Tecnologías de la Información y las Comunicaciones (CALIDAD 2018). <http://www.informaticahabana.cu/sites/default/files/ponencias2018/CAL10.pdf>
- Bandara, R., & Perera, I. (2020, 28-30 July 2020). Unit Test Code Generation Tool Support for Lower Level Programming Languages. 2020 Moratuwa Engineering Research Conference (MERCon).
- Benavides, Y. (2020). Pruebas tempranas, una buena práctica en el desarrollo de aplicaciones. Empresa de Soluciones Informáticas DATAZUCAR. Recuperado de: <https://www.datazucar.cu/?p=898>
- Ç, A., & Demir, E. (2020, 7-9 Oct. 2020). Detection of Duplicate Bug Reports in Jira and Bugzilla Tools. 2020 Turkish National Software Engineering Symposium (UYMS),

- Callejas Cuervo, M. A. A., Andrea Catherine, Álvarez Carreño, A. M. (2017). Modelos de calidad del software, un estado del arte. *Entramado*, 13(1): 236-250. <http://dx.doi.org/10.18041/entramado.2017v13n1.25125>
- Colorado, S. M. G., & Plaza, J. E. G. (2013). Sistema de generación automática de scripts de ejecución para pruebas unitarias en aplicaciones web. *Revista Politécnica*, 9(17): 105-112. <https://revistas.elpoli.edu.co/index.php/pol/article/view/345>
- COMPETISOFT. (2006). COMPETISOFT-Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica. Versión 0.2. In (pp. 294): Proyecto COMPETISOFT 506PI287.
- COMUNICACIONES, M. D. (2021a). NRCM 3-3: 2020 Modelo de la calidad para el desarrollo de aplicaciones informáticas (MCDAI)-Guía de implementación. In *Pruebas de Software* (Vol. NRCM 3-3:2020). La Habana, Cuba: Mincom.
- Comunicaciones, M. D. (2021b). NRCM 3-2: 2020 Modelo de la calidad para el desarrollo de aplicaciones informáticas (MCDAI)-Guía general. In. La Habana, Cuba: Mincom.
- Comunicaciones, M. D. (2021c). NRCM 3-4: 2020 Modelo de la calidad para el desarrollo de aplicaciones informáticas (MCDAI)-Guía de evaluación. In. La Habana, Cuba: Mincom.
- Delgado Dapena, M. D., Macías Rojas, A., Larrosa Uribe, D., Verona Marcos, S., & Fernández Oliva, P. B. (2017). Modelo para la generación automática de pruebas tempranas basadas en búsquedas. *Computación y Sistemas*, 21: 503-513. http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462017000300503&nrm=iso
- Delgado Dapena, M. D. F. O., Perla; Verona Marcos, S.; Larrosa Uribe, D. (2019). MTest. search: Modelo de Generación automática de Pruebas basado en búsquedas.
- Díaz, J., Tzancoff, C. B., Rodríguez, A., & Soria, V. (2017). Herramientas open source para testing de aplicaciones Web. Evaluación y usos.
- Digital.ai. (2022). 15th Annual State of Agile Report Digital.ai Software Inc. <https://digital.ai/resource-center/analyst-reports/state-of-agile-report>
- Gärtner, M. (2013). ATDD by Example: A Practical Guide to Acceptance Test-driven Development. Addison-Wesley. <https://books.google.com/cu/books?id=XYU5R9Win1UC>
- Godoy, Y. V., Cintra, A. V., & Rolo, L. C. (2020). Diseño de herramienta para casos de pruebas funcionales en la universidad de las ciencias informáticas. *Serie Científica de la Universidad de las Ciencias Informáticas*, 13(1): 47-61. <https://publicaciones.uci.cu/index.php/serie/article/view/531>
- González Reyes, A., Andre Ampuero, M., & Hernández González, A. (2015). Análisis comparativo de modelos y estándares para evaluar la calidad del producto de software [modelos de calidad; estándares de calidad; características de calidad de los productos de software]. 2015, 6(3), 10. <https://doi.org/10.1234/rci.v6i3.411>
- INGERTEC. (2021, enero 2021). Normas ISO. Recuperado de: 11 de enero de 2021 <https://www.normas-iso.com/iso-iec-15504-spice/>
- ISO/IEC/IEEE. (2017). ISO/IEC/IEEE 12207:2017 Systems and software engineering-Software life cycle processes. In (pp. 158). Switzerland

- Larrosa Danay, D. M. D., and Guemes Alejandro Miguel. (2019). GeCodP: Herramienta para la generación de código de pruebas integradas a entornos productivos. 22nd Ibero-American Conference on Software Engineering, CibSE 2019.
- Lawrence, R., & Rayner, P. (2019). Behavior-Driven Development with Cucumber: Better Collaboration for Better Software. Addison Wesley Professional. <https://books.google.com.cu/books?id=IEhpmwEACAAJ>
- Lazo Alvarado, Y., Gómez Barroso, C., Mariño Zayas, Y., Bony Fernández, M. M., Agramonte Díaz, E., & Batista González, D. (2016). Proceso de aseguramiento de la calidad para un modelo de la calidad en Cuba. *Revista Cubana de Ciencias Informáticas*, 10: 124-137.
- MarinDíaz, A., Trujillo Casañola, Y., & Buedo Hidalgo, D. (2020). Estrategia de pruebas para organizaciones desarrolladoras de software. *Revista Cubanade Ciencias Informáticas*, 14: 83-104. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992020000300083&nr-m=iso
- Myers, G. J., Sandler, C., & Badgett, T. (2011). The art of software testing. John Wiley & Sons.
- Normalización, O. N. d. (2011). Ingeniería de software-Requisitos de calidad y evaluación de productos software (SQuaRE)-Guía para SQuaRE (ISO/IEC 25000: 2005, IDT) In (pp. 44). La Habana.
- Normalización, O. N. d. (2016a). Ingeniería de software y sistemas-Requisitos de la calidad y evaluación de software (SQuaRE)-Modelos de la calidad de software y sistemas (ISO/IEC 25010: 2011, IDT) In (pp. 48). La Habana.
- Normalización, O. N. d. (2016b). NC ISO/IEC 25040:2016 Ingeniería de software- Requisitos de la calidad y evaluación de software y sistemas (SQuaRE)- Proceso de evaluación. In (pp. 60). La Habana: Oficina Nacional de Normalización.
- Normalización, O. N. d. (2017). Ingeniería de software-Requisitos de la calidad y evaluación de productos de software (SQuaRE)-Requisitos de la calidad (ISO/IEC 25030: 2007, IDT) In (pp. 42). La Habana.
- Normalización, O. N. d. (2021). Ingeniería de software y sistemas-Pruebas de software-Parte 1: Conceptos y definiciones (ISO/IEC/IEEE 29119-1:2013, IDT). In (pp. 76). La Habana: Oficina Nacional de Normalización.
- Oktaba, H., Esquivel, C. A., Ramos, A. S., Martínez, A. M., Osorio, G. Q., López, M. R., Ordóñez, Y. F. (2005). Modelo de Procesos para la Industria de Software MoProSoft. Versión 1.3.
- Oro, L. T., Lazo Alvarado, Y., & Ramírez Pérez, J. F. (2019). Proceso de diseño del software para un modelo de la calidad en Cuba. *Revista de I+D Tecnológico*, Vol. 15(1). <https://revistas.utp.ac.pa/index.php/id-tecnologico/article/view/210310.33412/idt.v15.1.2103>
- Pressman, R. S. (2010). Ingeniería del Software. Un enfoque práctico (7ma ed.). McGraw-Hill Interamericana.
- Quintana, G., Solari, M., & Matalonga, S. (2014). Estudio de caso sobre herramientas de generación automática de casos de prueba. 17th Conferencia Iberoamericana en Software Engineering: CibSE 2014,

- Rojas-Robert, D. M., Pérez-Morales, Z., & Delgado-Dapena, M. D. (2019). Generador de valores interesantes para casos de pruebas unitarias. *Ingeniería Industrial*, 40: 183-193. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59362019000200183&nrm=iso
- Serna M., E., Martínez M., R., & Tamayo O., P. A. (2017). Un modelo para determinar la madurez de la automatización de las pruebas del software como área de investigación y desarrollo. *Computación y Sistemas*, 21: 337-352. http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462017000200337&nrm=iso
- SOFTEX. (2012). MPS.BR - Mejora de Proceso del Software Brasileño. Guía de Implementación--Parte 5: Fundamentos para Implementación del Nivel C del MR-MPS. In (pp. 57). Brasil: SOFTEX.
- Team, C. P. (2010). CMMI para Desarrollo, Versión 1.3 (Vol. 1.3). Software Engineering Institute, Carnegie Mellon University. <http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=966110.1184/R1/6572342.v1>
- Trujillo Casañola, M. Y., Febles Estrada, D. C. A., & León-Rodríguez, D. C. G. (2014). Modelo Si.MPS.CU para valorar las organizaciones al iniciar la mejora de proceso de software. *Revista Cubana de Ciencias Informáticas*, 8: 92-103. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992014000500008&nrm=iso
- Uribaz, D. L., Marcos, S. V., Oliva, P. F., & Dapena, M. D. D. (2019). Ejecución automática de pruebas en entornos empresariales de producción de software. *Informática y Sistemas: Revista de Tecnologías de la Informática y las Comunicaciones*, 3(1): 32-44.
- Wynne, M., Hellesøy, A., & Tooke, S. (2017). The Cucumber Book: Behaviour-driven Development for Testers and Developers. Pragmatic Bookshelf. <https://books.google.com.cu/books?id=zzOqnQAACAAJ>

Copyright © 2023 Gómez García, A., Sosa Hernández, M., Verona Marcos, S., Delgado Dapena, M. D.



Este obra está bajo una licencia de Creative Commons Atribución-No Comercial 4.0 Internacional