



Facultad de Ingeniería

ISSN: 0121-1129

revista.ingenieria@uptc.edu.co

Universidad Pedagógica y Tecnológica  
de Colombia  
Colombia

Rojas-Montes, Martha Lucía; Pino-Correa, Francisco José; Martínez, James Mauricio  
Proceso de pruebas para pequeñas organizaciones desarrolladoras de software  
Facultad de Ingeniería, vol. 24, núm. 39, julio-diciembre, 2015, pp. 55-70  
Universidad Pedagógica y Tecnológica de Colombia  
Tunja, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=413940776006>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica  
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

# Proceso de pruebas para pequeñas organizaciones desarrolladoras de software

Testing process for small software development organizations

Processo de provas para pequenas organizações desenvolvedoras de software

Fecha de Recepción: 01 de Febrero de 2015  
Fecha de Aceptación: 10 de Abril de 2015

Martha Lucía Rojas-Montes\*  
Francisco José Pino-Correa\*\*  
James Mauricio Martínez\*\*\*

## Resumen

Definir un proceso de pruebas dentro de una organización puede ser valioso en la medida en que permita: (i) incrementar la calidad del producto, (ii) facilitar la comunicación y la comprensión entre los miembros del equipo, (iii) apoyar la mejora continua del proceso, y (iv) soportar la ejecución automática de ciertas tareas. En este sentido, este artículo propone un proceso liviano definido para guiar y apoyar la realización de las pruebas en pequeñas organizaciones desarrolladoras de software. El proceso propuesto incorpora técnicas de pruebas funcionales con el objetivo de concretar las actividades requeridas para evaluar la funcionalidad de un producto software de manera organizada y sistemática, especialmente aquellas que tienen que ver con el diseño y ejecución de las pruebas. Este proceso ha sido definido a partir del análisis de los procesos de pruebas existentes en algunos modelos de referencia y en diferentes estudios revisados de la literatura. Tras su ejecución exitosa en una pequeña empresa de desarrollo software, se ha observado que este proceso puede ser útil y práctico para llevar a cabo las pruebas de un producto software por organizaciones de este tipo.

**Palabras clave:** Proceso de pruebas, técnicas de pruebas funcionales, pequeñas organizaciones desarrolladoras de software.

---

\* Universidad del Cauca (Popayán - Cauca, Colombia). mrojas@unicauca.edu.co.

\*\* Ph.D. Universidad del Cauca (Popayán - Cauca, Colombia). fjpino@unicauca.edu.co.

\*\*\* M.Sc. Nexura Internacional S.A.S (Cali – Valle del Cauca, Colombia). jmartinez@nexura.com.

## Abstract

Define a testing process in an organization can be valuable to the extent that it allows to: (i) increase product quality, (ii) facilitate the understanding and communication between team members, (iii) support the process improvement in a manner continuous, (iv) and support the tasks' automatic execution. In this sense, this paper proposes a light weight process defined to guide and support the test process in small software development organizations. The proposed process incorporates functional testing techniques in order to setting the activities required to evaluate the software product's functionality from an organized and systematic manner, especially those having to do with the design and execution of tests. This process has been defined from the analysis of: (i) few existing testing process reference models, and (ii) different studies reviewed of the literature. This process has been successful used in a small software development company answer have observed that it can be useful and practical to perform tests of software product in this type of companies.

**Keywords:** Testing Process, Functional Testing Techniques, Small Software Development Organizations.

## Resumo

Definir um processo de provas dentro de uma organização pode ser valioso na medida em que permita: (i) incrementar a qualidade do produto, (ii) facilitar a comunicação e a compreensão entre os membros da equipe, (iii) apoiar a melhora contínua do processo, e (iv) suportar a execução automática de certas tarefas. Neste sentido, este artigo propõe um processo leve definido para guiar e apoiar a realização das provas em pequenas organizações desenvolvedoras de software. O processo proposto incorpora técnicas de provas funcionais com o objetivo de concretar as atividades requeridas para avaliar a funcionalidade de um produto software de maneira organizada e sistemática, especialmente aquelas que têm que ver com o desenho e execução das provas. Este processo tem sido definido a partir da análise dos processos de provas existentes em alguns modelos de referência e em diferentes estudos revisados da literatura. Após sua execução exitosa em uma pequena empresa de desenvolvimento software, se tem observado que este processo pode ser útil e prático para levar a cabo as provas de um produto software por organizações deste tipo.

**Palavras chave:** Processo de provas, técnicas de provas funcionais, pequenas organizações desenvolvedoras de software.

Cómo citar este artículo:

[1] M.L. Rojas-Montes, F.J. Pino-Correa & J.M. Martínez, "Proceso de pruebas para pequeñas organizaciones desarrolladoras de software", Fac. Ing., vol. 24 (39), pp. 55–70, Mayo-Ago. 2015.

## I. INTRODUCCIÓN

La industria del software reconoce hoy la importancia de llevar a cabo pruebas de software, como instrumento para asegurar la calidad de los productos desarrollados [1]; así, las empresas que forman esta industria intentan protegerse, en cierta medida, de los riesgos de litigios, sobre costos y pérdida de reputación ante los clientes [2]. La industria del software en Colombia está conformada, en su mayoría, por pequeñas organizaciones [3] (el término “pequeñas organizaciones” en este artículo equivale al de “very small entities” –VSEs–, que denota a una empresa, organización, departamento, grupo o proyecto que cuenta con menos de 25 personas [4]); entonces, es importante abordar trabajos que fortalezcan el quehacer de este tipo de empresas, con el fin de que puedan tener mayor competitividad [5, 6].

Las pequeñas organizaciones del software necesitan llevar a cabo sus pruebas de manera planeada y sistemática, con el fin de promover el desarrollo de software de calidad y, así, obtener ventajas competitivas [7, 8]. El mercado valora, cada día más, la calidad, por lo tanto, las compañías exigen la disminución de errores, y penalizan los retrasos en entregas y las cancelaciones de proyectos [9]. Esta situación conduce a que las pequeñas organizaciones cambien su orientación hacia las pruebas de software, y las consideren un proceso necesario para apoyar la calidad de sus productos [1]. La ingeniería del software cuenta con la premisa: “la calidad del producto depende en gran medida de la calidad del proceso” [10, 11], que destaca la importancia de los procesos para proporcionar software de alta calidad. Estos procesos (incluyendo el de pruebas de software), que involucran personas, aspectos organizacionales y procedimientos utilizados para definirlos, ejecutarlos y mantenerlos, forman la base sobre la cual se sustenta el desarrollo exitoso de productos software de alta calidad [12], por lo que contribuir a su implantación adecuada por pequeñas organizaciones debe ser un propósito fundamental.

Este artículo propone un proceso liviano definido para guiar y apoyar la realización de las pruebas en pequeñas organizaciones desarrolladoras de software. La incorporación de técnicas de pruebas funcionales en el proceso propuesto ofrece una manera organizada y sistemática de llevar a cabo actividades requeridas

para evaluar un producto software, especialmente aquellas labores que tienen que ver con el diseño y la ejecución de las pruebas. Este proceso ha sido definido a partir del análisis de algunos procesos de pruebas existentes en modelos de referencia y en la literatura, y ha sido aplicado exitosamente en una pequeña empresa de desarrollo software.

En la sección II del artículo se describen algunas investigaciones realizadas sobre pruebas de software; en la III se define y desarrolla el proceso de pruebas propuesto y las técnicas de pruebas funcionales que lo apoyan; en la IV se presenta la aplicación y el ajuste del proceso de pruebas en la práctica, mediante la descripción de un caso de estudio real en una pequeña organización desarrolladora de software; en la sección V se presenta la discusión, y, finalmente, en la sección VI se esbozan las conclusiones del trabajo y se proponen varias ideas de trabajo futuro.

## II. TRABAJOS RELACIONADOS

Para desarrollar el proceso de pruebas propuesto en este artículo se analizaron las propuestas clasificadas e indicadas a continuación:

- *Modelos de Referencia de Procesos para la Industria del Software*: ISO/IEC 12207 Procesos del ciclo de vida del Software [13], CMMI [14] y Competisoft [15].
- *Estándares Relacionados con Pruebas de Software y VSEs*: ISO/IEC 29119 Pruebas de Software [16] e ISO/IEC 29110 Perfiles de ciclo de vida para pequeñas organizaciones (VSEs) [4].
- *Procesos de Mejora de Pruebas*: TMMI[17], TPI[18], Propuesta para el testeo de PyMEs [19], y Minimal Test Practice Framework[20].
- *Procesos de pruebas*: ISO/IEC 29119 Pruebas de Software. Parte 2 – Proceso de Pruebas [21], Software Qualification Testing Process [13], Proceso de pruebas de software para el modelo de referencia Competisoft [22], TestPAI [23] y Proceso de pruebas para pequeñas empresas: en un escenario Brasileño [24].

Para realizar el análisis de las propuestas citadas se han definido los ocho criterios (con su correspondiente definición) presentados en la Tabla 1. Luego, en la Tabla 2 se presenta la relación de las propuestas revisadas y analizadas, tratando los aspectos más

relevantes de cada una de ellas, de tal manera que sea posible hacer una comparación y evidenciar de forma clara la diferencia con el proceso de pruebas propuesto en este artículo.

**TABLA 1**  
CRITERIOS DEFINIDOS PARA EL ANÁLISIS DE LAS PROPUESTAS RELACIONADAS

Criterio	Definición
Tipo	Alcance: <ol style="list-style-type: none"> <li>1. Procesos de todo el ciclo de vida del <i>software</i></li> <li>2. Proceso exclusivamente de pruebas</li> <li>3. Mejora de los procesos de todo el ciclo de vida del <i>software</i></li> <li>4. Mejora del proceso de pruebas</li> <li>5. Pruebas de Software</li> </ol>
Licencia	Propietario (P): Una propuesta es de tipo propietario si el usuario tiene limitaciones para usarla, modificarla o redistribuirla y requiere permiso de una organización privada. Abierto (A): Una propuesta es de tipo abierto si el usuario NO tiene limitaciones para usarla o modificarla y se puede redistribuir libremente.
Exclusivo de pruebas	Una propuesta es exclusiva de pruebas si se enfoca hacia todas las prácticas relativas a pruebas.
Completamente Definida	Una propuesta está completamente definida si todas las prácticas descritas en la misma están completamente especificadas.
Incorpora actividades	Una propuesta que define un proceso que contiene un conjunto de tareas definidas, que se llevan a cabo para cumplir los objetivos fijados.
Incorpora tareas	Son el nivel de detalle con que se desglosan las actividades.
Incorpora técnicas de pruebas	La propuesta incluye procedimientos técnicos y de gestión que ayudan a desarrollar las actividades planteadas.
Adecuada para VSEs	La propuesta es adecuada a VSEs si su implantación en este tipo de organizaciones es relativamente sencilla.

**TABLA 2**  
**EVALUACIÓN DE CRITERIOS**

Criterios	Tipo	Licencia	Especifica de pruebas	Completamente Definida	Incorpora Actividades	Incorpora Tareas	Incorpora Técnicas de pruebas	Adecuada para VSE'S
<i>ISO/IEC 12207</i> [13]	Procesos de todo el ciclo de vida del <i>software</i>	A	NO	SÍ	SÍ	SÍ	NO	NO
<i>CMMI</i> [14]	Mejora de los procesos de todo el ciclo de vida del <i>software</i>	P	NO	SÍ	SÍ	SÍ	NO	NO
<i>Competisofit</i> [15]	Mejora de los procesos de todo el ciclo de vida del <i>software</i>	P	NO	SÍ	SÍ	SÍ	NO	SÍ
<i>ISO/IEC 29119</i> [16]	Pruebas de Software	A	SÍ	NO	SÍ	SÍ	SÍ	NO
<i>ISO/IEC 29110</i> [4]	Procesos de todo el ciclo de vida del <i>software</i> de las VSEs	A	NO	SÍ	SÍ	SÍ	NO	SÍ
<i>TMMI</i> [17]	Mejora del proceso de pruebas	P	SÍ	SÍ	SÍ	NO	NO	NO
<i>TPI</i> [18]	Mejora del proceso de pruebas	P	SÍ	SÍ	SÍ	SÍ	NO	NO
<i>Propuesta para el testeo de PYMES</i> [19]	Mejora del proceso de pruebas	A	SÍ	NO	SÍ	NO	NO	SÍ
<i>Minimal Test Practice Framework</i> [20]	Mejora del proceso de pruebas	P	SÍ	NO	SÍ	SÍ	NO	SÍ
<i>ISO/IEC 29119 Parte 2</i> [21]	Proceso exclusivamente de Pruebas	A	SÍ	NO	SÍ	SÍ	NO	NO
<i>Software Qualification Testing Process</i> [13]	Proceso exclusivamente de pruebas	A	SÍ	SÍ	SÍ	SÍ	NO	NO
<i>Proceso de pruebas de software para el modelo de referencia de Competisofit</i> [22]	Proceso exclusivamente de pruebas	P	SÍ	SÍ	SÍ	SÍ	NO	SÍ
<i>TestPAI</i> [23]	Mejora del proceso de pruebas	P	SÍ	SÍ	SÍ	SÍ	NO	SÍ
<i>Proceso de pruebas para pequeñas empresas: En un escenario brasileño</i> [24]	Proceso exclusivamente de pruebas	P	SÍ	SÍ	SÍ	SÍ	NO	SÍ
<i>Nuestra propuesta</i>	Proceso exclusivamente de pruebas	A	SÍ	SÍ	SÍ	SÍ	SÍ	SÍ

Se puede observar en la tabla anterior que la literatura hace explícitos diferentes propuestas que tienen que ver con procesos que cubren todo el ciclo de vida de un producto *software*, como las presentadas en [4, 13 - 15], que cubren todo el ciclo de vida, pero describen de manera muy general las actividades por realizar durante las pruebas de *software*, y brindan pocos elementos sobre ellas, por lo que pasan casi inadvertidas; aun cuando son conscientes de su importancia, no ofrecen en detalle elementos que determinen cómo realizar las pruebas de software para asegurar la calidad de los productos desarrollados.

También de la Tabla 2 se evidencian propuestas relacionadas con mejora de procesos [17 - 20], que establecen cómo mejorar un proceso de pruebas; sin embargo, no muestran explícitamente el proceso de pruebas por mejorar ni el proceso de pruebas obtenido a partir de la mejora realizada.

Aunque en la literatura analizada hay estudios como [13, 21 - 24], que describen procesos de pruebas, entre estos no se ha identificado uno enfocado al contexto de las pequeñas organizaciones, que defina explícitamente las actividades y técnicas de pruebas funcionales que deben utilizarse oportuna y eficazmente en el proceso de desarrollo de un producto *software*. En este sentido, a partir de la comparación presentada en la Tabla 2 se extrae que la mayor contribución de este trabajo es la definición de un proceso de pruebas adecuado a las pequeñas organizaciones, el cual incorpora técnicas de pruebas funcionales al proceso planteado. En este trabajo se han identificado, analizado y relacionado un conjunto de técnicas de pruebas funcionales que apoyan la ejecución del proceso de pruebas propuesto; estas técnicas de pruebas ofrecen los lineamientos que pretenden asegurar la funcionalidad especificada, que es a menudo el criterio más importante para la aceptación del producto software.

Con este panorama en mente, en este artículo se propone un proceso de pruebas enfocado a la pequeña empresa desarrolladora de software, el cual integra, de forma sencilla, técnicas de pruebas funcionales, con el fin de ofrecer una guía a este tipo de organizaciones para llevar a cabo pruebas funcionales de software a lo largo del ciclo de vida de desarrollo del software. El proceso es adecuado para pequeñas organizaciones ya que es: i) liviano –incorpora pocas actividades, roles y productos de trabajo– y ii) descriptivo –involucra el qué y el cómo hacer–. No se trata de que estas organizaciones tengan que invertir más tiempo o contratar nuevo personal, lo que se propone es un proceso que define ¿qué hacer? y ¿cómo hacerlo?, mediante actividades, tareas, técnicas, responsabilidades y herramientas bien definidas y explícitas; además, las técnicas de pruebas funcionales propuestas no son complejas para este tipo de empresas.

### III. PROCESO DE PRUEBAS

El proceso de pruebas propuesto es el resultado de: i) la clasificación y análisis de los procesos de pruebas existentes, ii) la identificación, análisis e integración de técnicas de pruebas funcionales y iii) la identificación, observación y análisis de las pruebas de acuerdo con los procesos y formas de trabajo para el desarrollo de software en la pequeña empresa. Además, el proceso de pruebas definido adecúa sus prácticas pensando en la pequeña empresa, usando como referencia los estudios clasificados como *Procesos de pruebas* mostrados en la sección anterior y la experiencia adquirida tras la aplicación del caso de estudio en una pequeña empresa (ver Fig. 1). El referente fundamental es la parte 2 de la norma ISO/IEC 29119 [21], que define un proceso de pruebas genérico que busca ser utilizado dentro de cualquier desarrollo de *software* o ciclo de vida.

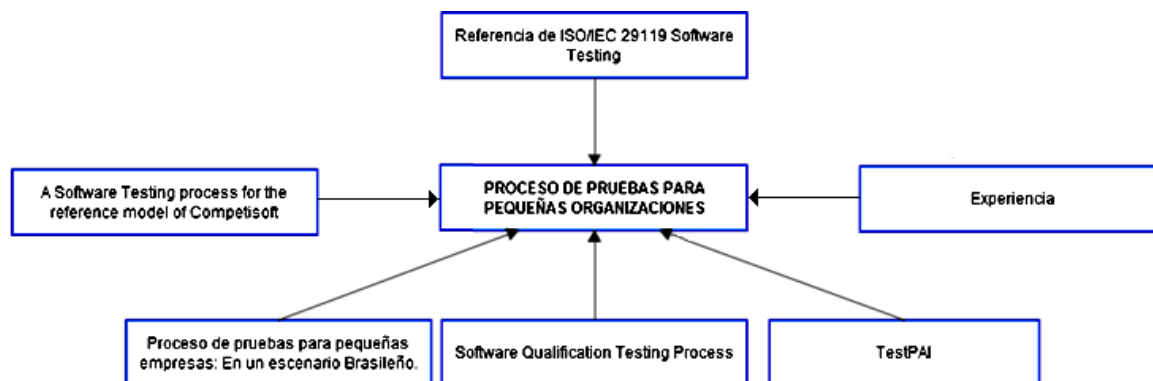


FIG. 1. Fuentes para la definición del proceso de pruebas[12, 20-24]

El proceso de pruebas propuesto describe 6 fases: i) Inicio, ii) Planeación, iii) Diseño, iv) Ejecución, v)

Monitoreo y control, y vi) Finalización; en cada una de las cuales se especifican sus actividades, tareas,

productos de trabajo y roles. Además, este proceso se describe siguiendo el patrón de procesos sugerido en el proyecto COMPETISOFT [13], el cual está enfocado para pequeñas organizaciones.

### Definición general del proceso de pruebas

Se presenta el proceso de pruebas propuesto, en términos de: propósito, objetivos, diagrama de flujo, descripción general, productos de trabajo, roles, técnicas de pruebas y herramientas.

**Propósito:** El propósito del proceso es incorporar actividades y técnicas de pruebas que sean fácilmente aplicables a pequeñas organizaciones desarrolladoras

de *software* y que conduzcan a la obtención de productos de alta calidad.

### Objetivos:

- Asegurar la calidad de los productos desarrollados.
- Lograr casos de pruebas exitosos tras la aplicación de las técnicas de pruebas funcionales que dan soporte a algunas de las actividades descritas por el proceso de pruebas.

**Diagrama de flujo de trabajo:** Se presenta en la Fig. 2. Las actividades del proceso se muestran como secuenciales, pero en la práctica algunas deben hacerse de manera iterativa.

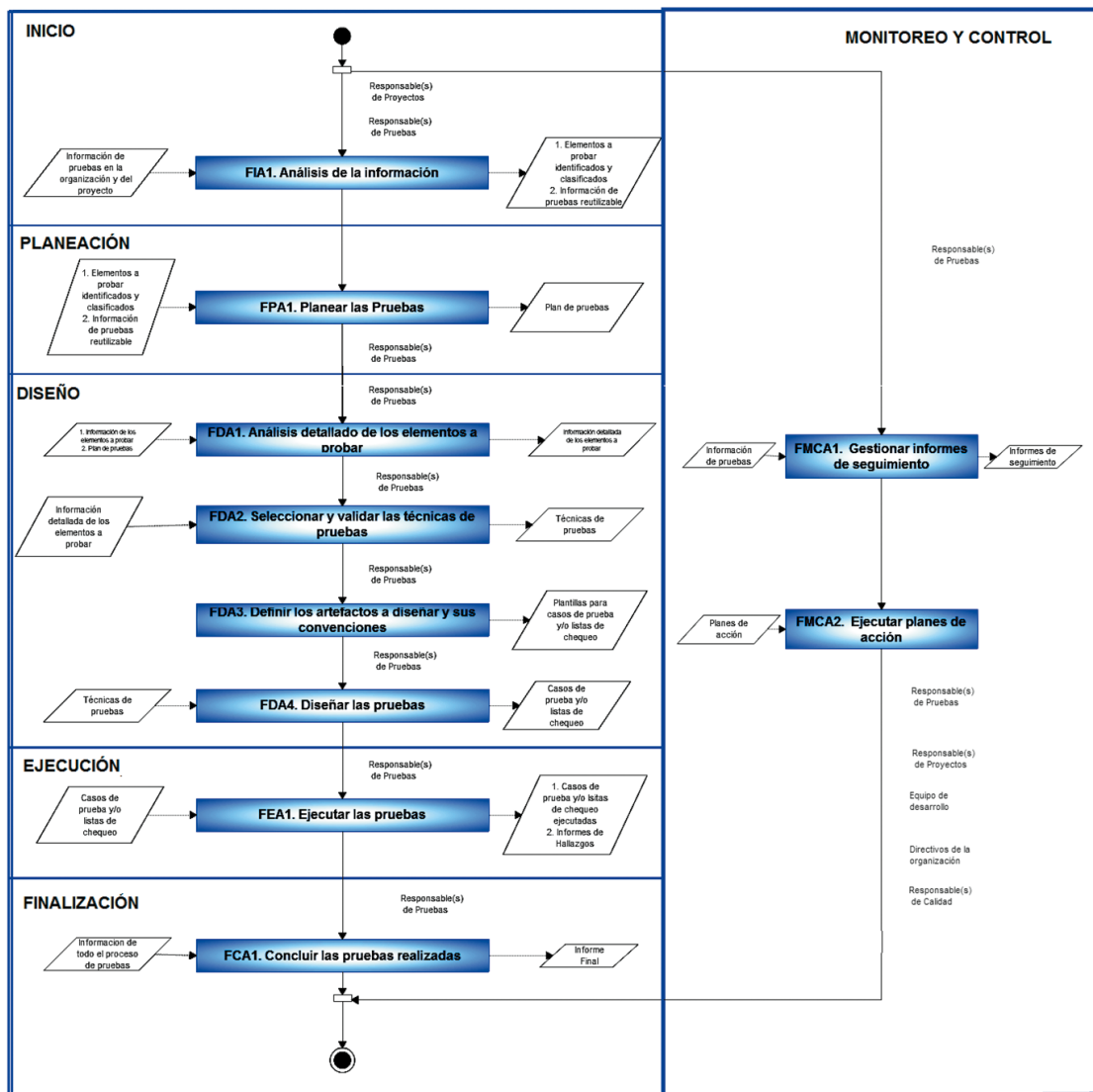


FIG. 2. Diagrama de flujo de trabajo



**Descripción del proceso:** Por restricciones de espacio, a continuación se presenta la descripción de las fases de manera general; sin embargo, la fase de diseño se presenta de manera detallada, bajo el patrón de procesos de Competisoft. El proceso de pruebas completamente definido, el detalle de las técnicas y el modelado se describen pormenorizadamente en [25].

*Fase de Inicio:*

- *FIA1. Análisis de la información:* Se recoge la información del proyecto al que se aplicarán las pruebas, con el objetivo de identificar los elementos que se van a probar en el software en reposo (hace referencia a requerimientos, modelos y demás artefactos que pueden ser evaluados sin que para ello se haya construido el producto software) y en el software ejecutable; esto permite establecer una visión global de las pruebas en el sistema y, en

ese sentido, prepararse para establecer un plan de pruebas que sirva como soporte para el proceso de pruebas que se llevará a cabo. A partir de esto, es posible encontrar información de pruebas en la organización que aporte contenido reutilizable.

*Fase de Planeación:*

- *FPA1. Planear las pruebas:* Establece las decisiones sobre qué es lo más importante para lograr el éxito de las pruebas, definiendo un plan de pruebas con los siguientes elementos: Elementos priorizados, Alcance, Riesgos, Tipos y técnicas de pruebas, Roles, Indicadores de pruebas, Estimaciones de tiempo y esfuerzo, Herramientas para pruebas, Atributos de calidad a probar y Cronograma.

*Fase de Diseño:* Esta fase se describe de manera detallada en la Tabla 3

**TABLA 3**  
FASE DE DISEÑO DEL PROCESO PROPUESTO

Fase	FASE DE DISEÑO	Abreviatura	FD
Propósito	La fase de diseño tiene como propósito construir los artefactos necesarios para ejecutar las pruebas en el proyecto (los artefactos de pruebas hacen referencia a casos de prueba en lo relacionado a pruebas dinámicas y listas de chequeo en cuanto a pruebas estáticas).		
Descripción de las actividades	La fase de diseño se compone de las siguientes actividades: <ul style="list-style-type: none"><li>Análisis detallado de los elementos por probar, para obtener diseños más precisos que permitan encontrar un mayor número de errores.</li><li>Seleccionar y validar las técnicas de pruebas más apropiadas para elaborar el conjunto de artefactos que evalúen los elementos descritos en la actividad anterior.</li><li>Definir los artefactos a diseñar y sus convenciones, de tal manera que se conozca qué componente utilizar según sea el elemento por probar.</li><li>Diseñar las pruebas: el diseño de las pruebas se basa en la creación de casos de prueba y/o listas de chequeo cuya ejecución permita observar posibles síntomas de defectos. Los casos de prueba se diseñan de acuerdo a las técnicas de pruebas funcionales seleccionadas.</li></ul>		
Roles involucrados	Abreviatura	Nombre	
	RPruebas	Responsable(s) de pruebas	
Entradas	Información de los elementos a probar; Plan de pruebas		
Salidas	Casos de prueba diseñados; Listas de chequeo		
ACTIVIDADES Y TAREAS DE LA FASE			
Rol		Descripción	
FDA1. Análisis detallado de los elementos por probar			
Entradas		Información de los elementos por probar; Plan de pruebas	
RPruebas	FDA1.1 Recoger información detallada <ul style="list-style-type: none"><li>Ubicar los elementos a prueba</li><li>Buscar información detallada del elemento</li><li>Identificar las variables y los valores del elemento a prueba (para el caso de pruebas funcionales)</li><li>Identificar los atributos de calidad a revisar (para revisiones)</li></ul>		

Fase	FASE DE DISEÑO		Abreviatura	FD
Salidas		Información detallada de los elementos por probar		
FDA2. Seleccionar y validar las técnicas de pruebas				
Entradas		Información detallada de los elementos por probar		
RPruebas	FDA2.1 Identificar las técnicas apropiadas			
	<ul style="list-style-type: none"><li>Analizar las variables y valores (para el caso de pruebas funcionales)</li><li>Analizar los atributos de calidad por revisar (para las listas de chequeo)</li><li>De acuerdo al análisis hecho identificar la técnica más apropiada.</li></ul>			
	Técnicas de pruebas funcionales	Partición equivalente		
	Análisis de valor límite			
	Tablas de decisión			
	Arreglos ortogonales (Herramienta de ayuda allpairs)			
Salidas		Técnicas seleccionadas		
FDA3. Definir los artefactos por diseñar y sus convenciones				
RPruebas	FDA3.1 Definir los artefactos que soportarán la ejecución de las pruebas			
	<ul style="list-style-type: none"><li>Identificar el artefacto por utilizar para las pruebas. En caso de revisiones formales se sugiere el uso de listas de chequeo y para pruebas dinámicas el uso de casos de prueba.</li></ul> <p><b>Nota:</b> Para esto se proporcionan plantillas como anexos al documento.</p>			
RPruebas	FDA3.2 Definir la estructura del artefacto			
	<ul style="list-style-type: none"><li>Identificar los campos que tendrá el artefacto (Identificación, descripción, prerequisites, actores, responsables, resultado esperado, resultado obtenido)</li><li>Identificar estados (Ej. En construcción, Exitoso, Fallido, Frenado, Pendiente de ejecución)</li></ul>			
Salidas		Plantillas para casos de prueba o listas de chequeo		
FDA4. Diseñar las pruebas				
Entradas		Técnicas		
RPruebas	FDA4.1 Construir los artefactos			
	<ul style="list-style-type: none"><li>Registrar información básica solicitada en la plantilla definida</li><li>Diseñar Listas de chequeo: (i) Enumerar los ítems de chequeo, (ii) Descripción del ítem, (iii) Descripción que permita evaluar el atributo de calidad para cada ítem (puede ser una pregunta)</li><li>Diseñar Casos de prueba: (i) Describir el proceso o escenario que debe llevarse a cabo para ejecutar la prueba (uso de la técnica), (ii) Indicar las precondiciones para ejecutar la prueba (si las hay), (iii) Indicar los pasos necesarios para ejecutar la prueba, (iv) Indicar el resultado esperado, (v) Indicar el estado del caso de prueba</li></ul>			
RPruebas	FDA4.2 Listar los datos necesarios para las pruebas			
Salidas		Casos de prueba o Listas de chequeo		

### *Fase de Ejecución*

- **FEA1. Ejecutar las pruebas:** Se prepara y organiza el entorno específico para las pruebas estáticas, y otro para las pruebas dinámicas, ya que cada tipo de prueba requerirá un entorno con unas características específicas para que su ejecución sea lo más efectiva posible. Se establecen las condiciones bajo las cuales un caso de prueba o lista de chequeo pasa o falla, y finalmente se ejecutan las pruebas estáticas o dinámicas, y se reportan los hallazgos encontrados.

### *Fase de Monitoreo y Control*

- **FMCA1. Gestionar informes de seguimiento:** Se analizan los indicadores de producto y proceso para emitir juicios acerca del progreso de las pruebas y la calidad del producto.
- **FMCA2. Ejecutar planes de acción:** Se ejecutan los planes de acción producto del análisis de los informes de seguimiento, y se verifica que estas acciones se hayan ejecutado de manera exitosa.

### *Fase de Finalización*

- **FFA1. Concluir las pruebas:** Se analizan los indicadores de producto y proceso de todos los informes generados para finalmente emitir un juicio acerca de la calidad del producto total.

**Productos de trabajo:** Se definen a partir de entradas y salidas, en algunas tareas se definen entradas particulares, y en otras con las entradas de cada actividad es suficiente para llevar a cabo las tareas; no necesariamente una salida de cada actividad es un producto de trabajo del proceso. Los productos de trabajo del proceso definido son: Plan de pruebas, Casos de pruebas o listas de chequeo, Informes de hallazgos e Informes de seguimiento.

**Responsabilidad y autoridad:** El responsable de llevar a cabo el proceso de pruebas es el rol: Responsable(s) de pruebas, y la autoridad a la cual se le debe reportar el trabajo realizado durante las pruebas (si la hay) es el rol: Líder de pruebas, quien en cualquier momento puede asumir el rol de Responsable de pruebas.

**Técnicas de pruebas:** Las técnicas de pruebas funcionales: Partición equivalente, Análisis de valor límite, Tablas de decisión y Arreglos Ortogonales, son las que se incorporan directamente con la Fase de Diseño de pruebas previamente descrita; estas

técnicas fueron seleccionadas e integradas al proceso, puesto que tienen un bajo nivel de complejidad, no son complicadas, no son difíciles de entender y pueden ser aplicadas por las pequeñas organizaciones. Además, desde la experiencia práctica se observó que la utilización de estas técnicas trae beneficios a estas organizaciones en corto tiempo e invirtiendo pocos recursos. La incorporación de este tipo de técnicas al proceso pretende incrementar la calidad del producto software, haciendo de las pruebas un proceso continuo y paralelo al proceso de construcción del software. Para cada una de estas técnicas se ha creado una guía de utilización, que describe aspectos teóricos de la técnica, consideraciones para aplicarla y un apartado sobre cuándo utilizarla. El objetivo de estas guías es apoyar la ejecución de las pruebas funcionales mediante la estructuración, diseño y aplicación de un conjunto de casos de prueba para verificar si el comportamiento del software cumple o no con lo esperado de acuerdo con su especificación.

**Herramientas:** A continuación se presentan algunas herramientas de código abierto que en la experiencia práctica han sido utilizadas con éxito como apoyo a la implantación del proceso de pruebas propuesto: (i) Las fases de Planeación y Monitoreo y control pueden ser apoyadas por herramientas que brinden soporte en la gestión de las actividades de pruebas –un ejemplo es Libre Plan (<http://www.libreplan.com>) –; (ii) La Fase de Ejecución puede ser apoyada por herramientas que ayuden a sistematizar el seguimiento de hallazgos –un ejemplo es Mantis (<http://www.mantisbt.org/>)–; (iii) La Fase de Diseño puede ser apoyada por herramientas que soporten el diseño de casos de prueba y que brinden ayuda en la optimización de ellos –un ejemplo es TestLink (<http://www.testlink.org/>)–. Además, la herramienta ALLPAIRS –Test Case Generation Tool (<http://allpairstesting.com/>)– es útil para aplicar la técnica de Arreglos ortogonales. Estas herramientas son un apoyo importante al proceso, pues proporcionan información útil a las personas que intervienen en él, permitiendo tomar decisiones y ejercer acciones sobre los proyectos de desarrollo.

## **IV. APLICACIÓN DEL PROCESO DE PRUEBAS**

El método de casos de estudio [26] se ha utilizado para conducir la aplicación del proceso de pruebas

en el contexto real de una pequeña empresa caleña de desarrollo *software* dedicada a brindar soluciones para ‘gobierno en línea’. El tipo de *diseño* del caso de estudio es simple-holístico, ya que el proceso de pruebas ha sido aplicado en la pequeña empresa utilizando como *unidad de análisis* uno de sus proyectos de desarrollo relacionado con un portal para una Superintendencia de Colombia. El *objeto de estudio* es el proceso de pruebas propuesto, y la pregunta de investigación principal es: ¿El proceso de pruebas propuesto es adecuado para llevar a cabo las pruebas en una pequeña empresa? Las *medidas* usadas para indagar sobre la pregunta de investigación son: (i) el esfuerzo de las actividades realizadas para llevar a cabo el proceso de pruebas y (ii) el número de errores encontrados antes y después de la ejecución del proceso de pruebas. Además, también se consideraron las apreciaciones de los miembros de la empresa relacionados con la aplicación del proceso de pruebas. También se definió el correspondiente *procedimiento de campo* para: (i) guiar la ejecución del proceso de pruebas en el contexto de la organización y (ii) recolectar la información generada de la aplicación del proceso. Todo el detalle del caso de estudio se encuentra en [25].

#### **A. Ejecución del proceso**

La ejecución del proceso de pruebas se llevó a cabo mediante tres ciclos. En el primer ciclo solo se ejecutan pruebas dinámicas, utilizando las fases de diseño y ejecución del proceso, es decir, se diseñaron y ejecutaron pruebas sobre los módulos ya desarrollados del proyecto piloto. Lo anterior se hizo porque se necesitaba hacer una entrega rápida, lo cual llevó finalmente a demostrar con resultados reales la necesidad de un proceso de pruebas que acompañara

el ciclo de vida de la construcción de los productos en la organización. A partir del análisis de los resultados de este primer ciclo, se refinó tanto el proceso de pruebas definido inicialmente como otros procesos de la organización; por tanto, la ejecución del proceso de pruebas definitivo se lleva a cabo durante el segundo y tercer ciclo de pruebas. Es importante destacar que se sometió cada ciclo de prueba a revisión, con el objetivo de concluir sobre su efectividad. Tras cada revisión se generaron informes que después de ser analizados permitieron establecer hallazgos sobre oportunidades de mejora latentes en el desarrollo del producto del proyecto piloto, a partir de los cuales se levantaban acciones correctivas o preventivas, según el caso.

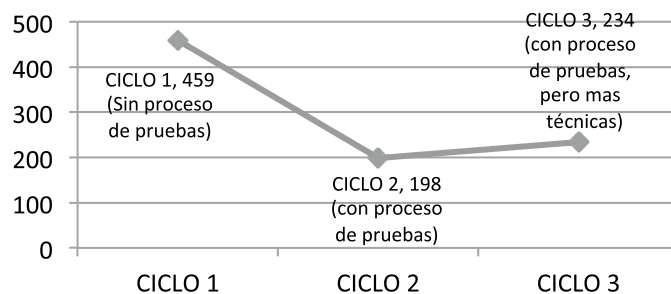
#### **B. Recolección de datos**

La recolección de datos se hizo mediante los productos de trabajo generados por las actividades del procedimiento de campo del caso de estudio y el proceso de pruebas. A partir de los datos recolectados se consolidó la información sobre el esfuerzo involucrado en ejecutar las fases/actividades del proceso de pruebas seleccionadas por la empresa (ver Tabla 4 y Fig. 3) y el número de errores del desarrollo encontrados antes y después de la ejecución del proceso de pruebas (ver Tabla 5 y Fig. 4). La información relacionada con el esfuerzo involucrado en llevar a cabo los ciclos de pruebas se obtuvo después de sintetizar los datos que se tenían individualmente sobre cada una de las actividades del procedimiento de campo y las actividades del proceso de pruebas. Además, la información relacionada con el porcentaje de errores antes y después de la ejecución del proceso de pruebas se obtuvo al sintetizar los datos que se tenían individualmente sobre los casos de prueba ejecutados y el número de errores de cada módulo del proyecto.

**TABLA 4**  
**ESFUERZO INVOLUCRADO EN CADA CICLO DE PRUEBAS EJECUTADO EN LA EMPRESA**

Esfuerzo (horas x 1 persona)														
Actividades del procedimiento de campo (Diagnóstico, Preparación, Formulación del proceso, Asignación del proyecto y Contextualización)												D	H	T
												38	9	343
CICLO 1	Actividades	D	H	T	CICLO 2	Actividades	D	H	T	CICLO 3	Actividades	D	H	T
	Diseño	17	9	153		Inicio	2	9	18		Inicio	3	9	27
	Ejecución	14	9	126		Planificación	3	9	27		Planificación	2	9	18
	Informe	1	9	9		Diseño	5	9	45		Diseño	10	9	90
	Plan Acción	4	9	36		Ejecución	10	9	90		Ejecución	9	9	81
	P. Regresión	15	9	135		Monitoreo y control	2	9	18		Monitoreo y control	2	9	18
	TOTAL					459	TOTAL				198	TOTAL		
Duración del ciclo (semanas)				9	Duración del ciclo (semanas)				5	Duración del ciclo (semanas)				5

D: Días H: Horas T: Total Horas



**FIG. 3.** Gráfica de esfuerzo por ciclo de pruebas ejecutado en la empresa

**TABLA 5**  
**PORCENTAJE DE DEFECTOS ENCONTRADOS TRAS LA EJECUCIÓN DE CADA CICLO DE PRUEBA**

Ciclos	CP	DEFECTOS	PORCENTAJE
<i>Antes de la ejecución del proceso de pruebas</i>			
CICLO 1	293	156	53%
<i>Después de la ejecución del proceso de pruebas</i>			
CICLO 2	101	26	26%
CICLO 3	195	43	22%
<i>Incluyendo técnicas de pruebas diferentes</i>			
CICLO 3 Plus (con más técnicas)	270	100	37%

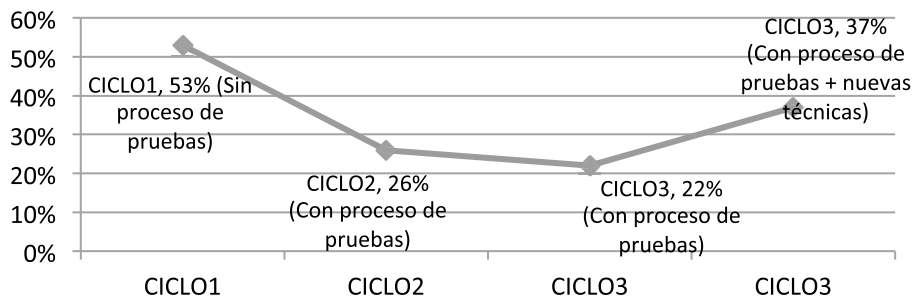


FIG. 4. Gráfico de porcentaje de errores por ciclo de prueba

### C. Análisis

De la Tabla 4 se puede extraer el esfuerzo invertido por semana en horas/persona para llevar a cabo cada ciclo de pruebas en la empresa. Como se puede observar, en el primer ciclo se ejecutaron 5 actividades (algunas de ellas no se ven como actividades propias del proceso de pruebas): Diseño, Ejecución, Informe, Plan de acción y Pruebas de regresión. Gran parte del esfuerzo invertido en este ciclo se enfatizó en las pruebas de regresión, comprobando que los cambios sobre los módulos no introducían defectos en otros módulos. Estas pruebas se realizaban cada vez que se hacía un cambio en el sistema, tanto para corregir un defecto como para realizar una mejora en el desarrollo de una funcionalidad específica; para su realización se incluía la repetición de los casos de pruebas realizados con anterioridad y que estaban directamente relacionados con la parte del sistema modificada.

La organización venía haciendo entrega de prototipos operativos de cada módulo, con el objetivo de que el cliente opinara acerca del producto que iba a usar, y de esta manera se refinaban los requerimientos. Cuando se asignó este proyecto al responsable del proceso de pruebas (primer autor del artículo), quedaba muy poco tiempo para ejecutar dicho proceso desde la fase de inicio y con todas las actividades (se estaba a unas pocas semanas de entregar un módulo), razón por la cual, y como se muestra en la Tabla 4, se empezó a ejecutar el primer ciclo de pruebas desde la fase de diseño (sin hacer un análisis previo de los elementos por probar, sin planificar y sin establecer unas políticas de pruebas). El diseño y la ejecución de los casos de prueba se hicieron sobre la marcha de la entrega de los módulos, generando así no conformidades a

partir de los defectos encontrados y alertando a los implicados responsables de tomar acciones correctivas al respecto. Inicialmente se realizó una reunión en la cual algunos desarrolladores señalaron las pruebas realizadas como culpables de ver defectos donde realmente no los había, razón por la cual se decidió realizar una clasificación de tales defectos. Al realizar esta actividad, se citó a otra reunión y se demostró que aunque era necesaria la clasificación, este no era el problema de fondo, pues seguían siendo muchos los hallazgos. Se decidió entonces hacer un ranking de defectos y buscar las causas, y posterior a esto las soluciones. Argumentados en estos hallazgos, se vio la necesidad de establecer pruebas de *software* desde los requerimientos, de tal manera que se eliminaran ambigüedades y se optimizaran los futuros desarrollos de la organización. Es a partir de aquí cuando empieza la ejecución completa del proceso de pruebas.

En los ciclos 2 y 3 de pruebas se ejecutaron 5 de las 6 fases del proceso: Inicio, Planificación, Diseño, Ejecución y Monitoreo y Control. La fase de Finalización no se ha llevado a cabo, porque el proyecto aún no ha finalizado. Al finalizar el ciclo 2 es cuando la dirección de la organización (tercer autor del artículo) entendió la necesidad latente de apropiar e implantar el proceso de pruebas a todos sus proyectos, lo anterior gracias a los resultados, es decir, al número de defectos encontrados antes y después de la ejecución del proceso (como se muestra en la Tabla 5). A partir de allí se ve el proceso de pruebas como una herramienta fuerte para la toma de decisiones y acciones frente a posibles desviaciones desde el proceso de Gestión de Proyectos y Construcción del Producto, pues el acompañamiento mutuo de pruebas

y construcción permitió mejorar los tiempos y las entregas de los módulos.

Basados en la premisa de que “*el objetivo de las pruebas es la detección de defectos en el software y no demostrar que el software no tiene defectos*”, para el ciclo 3 de pruebas se acogieron más técnicas de pruebas de las especificadas en el proceso general (hasta el momento solo se habían aplicado dos de ellas: partición equivalente y análisis de valor límite). El ciclo 3 hace referencia a la ejecución de los casos de prueba que ya habían sido definidos previamente, y tras su ejecución se obtuvieron resultados muy similares a los del ciclo 2. De ahí surgió la iniciativa de mejorar los diseños de las pruebas con más técnicas de pruebas (tablas de decisión y arreglos ortogonales) y, de esta manera, descubrir que las técnicas de pruebas sugeridas en el proceso de pruebas propuesto eran apropiadas para una pequeña organización. De este trabajo se evidenció que las técnicas de prueba sugeridas por el proceso propuesto son adecuadas para este tipo de organizaciones, ya que su diseño no implicó más de ocho días para cuatro módulos, lo que demuestra que son sencillas y de fácil uso. La incorporación de más técnicas al proceso de pruebas le aportó más calidad al producto; cuando los errores tendieron a disminuir lo que sucedía en realidad era que no se habían encontrado más errores bajo las técnicas de prueba utilizadas hasta ese momento.

Basados en las experiencias recopiladas del caso de estudio, en el número de errores encontrados antes y después de la ejecución del proceso de pruebas, en el esfuerzo involucrado para llevar a cabo este proceso y en los beneficios obtenidos por la empresa a partir de la utilización del proceso, consideramos que el proceso propuesto es adecuado para llevar a cabo actividades de pruebas de software en pequeñas empresas (respondiendo así a la pregunta de investigación principal).

## V. DISCUSIÓN

Como resultado de la utilización del proceso de pruebas, se llegó a consolidar una forma de trabajo de la cual la organización sacó provecho y la extendió a otros procesos. Cambios en el proceso de levantamiento de requerimientos surgieron tras la intervención que se hizo en el proyecto piloto en el que se aplicó el proceso de pruebas. Además de contribuir en la calidad de los

procesos de la organización y en la implementación del producto, el proceso de pruebas sirvió para identificar la necesidad de mejorar la especificación de requisitos para obtener una buena calidad en el *software*. De ahí que los beneficios de las pruebas no solo se vieron en el área de desarrollo de proyectos, sino que también se dejó notar en el resto de las áreas de negocio y especialmente en la satisfacción del cliente final.

Además, se han identificado diferentes factores que influenciaron la ejecución de las actividades de pruebas para dar pie a la institucionalización de un proceso de pruebas organizado y sistemático:

- Es necesario diseñar planes de pruebas y crear conciencia entre los implicados en el proceso de pruebas de las diferencias que existen entre planificar y diseñar pruebas. La planificación es el medio que permite mejorar la toma de decisiones para lograr el fin buscado tras el diseño y ejecución de las pruebas.
- Es necesario establecer desde el proceso de pruebas una priorización para la solución de las incidencias encontradas, de tal manera que el desarrollo de ellas no afecte los tiempos de entrega, por lo que es fundamental utilizar un solo mecanismo de comunicación entre los procesos de pruebas y construcción para evitar así reprocesos y demoras en la solución de incidentes urgentes.
- Los encargados de realizar las pruebas de *software* no deberían ser los mismos desarrolladores, ya que estos, en la mayoría de los casos, no cuentan con suficientes fundamentos teóricos y tienen un total desconocimiento de la existencia de procesos de pruebas y técnicas de pruebas formales. Es importante, por tanto, fortalecer el equipo de pruebas con personal calificado para ejercer dicha labor.
- Para el monitoreo y control del hallazgo de incidentes es fundamental hacerles un seguimiento hasta su solución, no se debe dar pie a incidentes olvidados o, en el peor de los casos, eliminados sin brindarles solución. Es importante contar con repositorios de información referente a incidentes y sus soluciones.

Los aspectos descritos anteriormente deben tenerse en cuenta a la hora de institucionalizar un proceso de pruebas con el fin de: i) concientizar en la importancia que tiene el proceso de pruebas sobre la calidad del

producto y ii) superar la etapa de construcción de software de forma artesanal, empírica y caótica. No abordar estos aspectos origina problemas en la calidad de sus productos, tiempos de desarrollo inadecuados (re-trabajo por falta de calidad), costos de producción mal estimados (y por tanto pérdidas económicas), actividades de operación y mantenimiento del software difíciles, y, desde luego, la insatisfacción de los clientes y usuarios finales.

## VI. CONCLUSIONES Y TRABAJO FUTURO

El aporte del proceso de pruebas descrito en este artículo es la adecuación y adaptación de las técnicas de pruebas funcionales en la Fase de diseño y el ajuste de las actividades del proceso para realizar revisiones a diferentes artefactos software (también conocidas como pruebas estáticas). Además, las tareas descritas en las actividades del proceso propuesto permiten clarificar no solo el quehacer en un proyecto que requiere revisiones y pruebas funcionales de software, sino también visualizar el cómo hacerlo (especialmente en el diseño de las pruebas), diferencia importante entre los procesos de pruebas encontrados en la literatura para el contexto de las pequeñas organizaciones.

Por otra parte, una de las mayores barreras a las que se enfrenta la implantación de un proceso en la pequeña empresa es la actitud de los directivos, que, por lo general, son reacios a invertir recursos en actividades que no muestren una ganancia económica notable a corto plazo. En este sentido, es importante tener en cuenta que la pequeña empresa necesita procesos con la mínima documentación posible y que deriven productos de trabajo que les generen un valor agregado; en lo posible, estos productos deben ser reutilizables, para agilizar el despliegue del proceso en los diferentes proyectos de la organización. A partir del trabajo realizado se ha observado que las técnicas de pruebas funcionales son fácilmente reutilizables y adaptables a los diferentes niveles de pruebas unitarias, de integración y de sistema.

Además, la comunicación en un proceso de pruebas es fundamental, sobre todo cuando se quiere implantar en una organización que no ha incorporado este tipo de prácticas. Generar un buen ambiente en el equipo de trabajo es fundamental para el éxito de la apropiación de un proceso “nuevo” en una organización. Dicha

comunicación involucra detalles tan pequeños como la descripción de un hallazgo o su clasificación, por tanto, se recomienda que para este tipo de procesos se establezcan formas de lenguaje comunes que ayuden al éxito del proyecto y a la calidad del producto desarrollado.

Finalmente, es importante destacar que las organizaciones pequeñas reciben con gran aprecio la incorporación de herramientas libres en sus procesos, de ahí que es importante trabajar en investigar y adaptar las herramientas existentes a las formas de trabajo de una organización. Como trabajo futuro se propone la adaptación e integración de algunas de estas herramientas, de tal forma que en su conjunto faciliten al responsable de pruebas obtener indicadores a partir de los cuales se puedan tomar decisiones sobre el proceso y sobre la calidad del producto; esto implica investigación sobre las herramientas de prueba y desarrollo. Además, el área de técnicas de pruebas es muy amplia, por lo que se sugiere a futuros investigadores abordar estas técnicas desde el punto de vista de los procesos, ya que las pequeñas organizaciones los acogen muy bien.

## AGRADECIMIENTOS

Los autores agradecen el apoyo brindado por la empresa Nexura International S.A.S. para la realización de este trabajo. Además, Francisco J. Pino agradece el apoyo del Grupo IDIS y de la Universidad del Cauca, donde él trabaja como profesor titular.

## REFERENCIAS

- [1] C. García, A. Dávila, and M. Pessoa, “Test process models: Systematic literature review”, *Communications in Computer and Information Science*, vol. 477, ed, 2014, pp. 84-93.
- [2] J. Kasurinen, “Elaborating Software Test Processes and Strategies”, in *Third International Conference on Software Testing, Verification and Validation (ICST)*, París, France, 2010, pp. 355-358; doi: 10.1109/ICST.2010.25.
- [3] K. Heshusius Rodríguez, “Colombia: desafíos de una industria en formación”, in *Desafíos y oportunidades de la industria del software en América Latina*, CEPAL, Ed., ed Colombia: Mayol Ediciones, 2009, pp. 139-170.



- [4] ISO, "ISO/IEC 29110. Software engineering-Lifecycle profiles for Very Small Entities (VSEs)", International Organization for Standardization, Genova, 2011.
- [5] C. Valencia Pinzón, "Las TIC, pieza clave para resolver problemas de competitividad nacional", Universidad Militar Nueva Granada (<http://hdl.handle.net/10654/7594>), Colombia2012.
- [6] P. Bastos Tigre y F. Silveira Marques, "América Latina en la industria global de software y servicios: una visión de conjunto", in *Desafíos y oportunidades de la industria del software en América Latina*, CEPAL, Ed., ed Colombia: Mayol Ediciones, 2009, pp. 249-292.
- [7] P. Clarke and R. V. O'Connor, "An empirical examination of the extent of software process improvement in software SMEs", *Journal of Software: Evolution and Process*, vol. 25, pp. 981-998, 2013.
- [8] J. Kasurinen, O. Taipale, and K. Smolander, "How Test Organizations Adopt New Testing Practices and Methods?", in *Software Testing, Verification and Validation Workshops (ICSTW), 2011 IEEE Fourth International Conference on*, 2011, pp. 553-558.
- [9] M. Piattini, F. García, I. García Rodríguez y F. Pino, *Calidad de Sistemas de Información*. Segunda edición, Madrid: Rama, 2014.
- [10] A. Fuggetta, "Software process: a roadmap", in *International Conference on Software Engineering (ICSE)*, Limerick. Ireland, 2000, pp. 25-34.
- [11] W. Humphrey, "Acquiring Quality Software", *CrossTalk The Journal of Defense Software Engineering*, vol. 18, pp. 19-23, 2005.
- [12] C. Fernández y M. Piattini, *Modelo para el gobierno de las TIC basado en las normas ISO*. Madrid, España: AENOR ediciones, 2012.
- [13] ISO, "ISO/IEC 12207:2008 Systems and software engineering - Software life cycle processes", International Organization for Standardization, Geneva, 2008.
- [14] SEI, "CMMI for Development, Version 1.3. Technical Report CMU/SEI-2010-TR-033 ESC-TR-2010-033", Software Engineering Institute (SEI), Pittsburgh2010.
- [15] H. Oktaba, M. Piattini, F. Pino, M. J. Orozco, and C. Alquicira, *COMPETISOFT: Mejora de Procesos Software para Pequeñas y Medianas Empresas y Proyectos*. Madrid: Ra-Ma, 2008.
- [16] ISO, "ISO/IEC/IEEE 29119: Software and systems engineering -- Software testing (Parts 1, 2, 3, 4 and 5)", International Organization for Standardization, Geneva, 2013.
- [17] E. v. Veenendaal, R. Hendriks, J. v. d. Laar, and B. Bouwers, "Test Process Improvement using TMMi", *The Magazine for Professional Testers*, vol. September, pp. 21-25, 2008.
- [18] J. Andersin, "TPI - a model for Test Process Improvement", Department of Computer Science, University of Helsinki (<http://www.cs.helsinki.fi/u/paakki/Andersin.pdf>), Helsinki, 2004.
- [19] E. J. V. Tanja, J. Sánchez Sánchez y M. Mannise, "Mejorando el testeo en las Pyme ¿Cómo empezar?", in *V Jornadas sobre Testeo de Software*, Valencia, España, 2008, pp. 71-80.
- [20] D. Karlström, P. Runeson, and S. Norden, "A minimal test practice framework for emerging software organizations", *Software Testing, Verification and Reliability*, vol. 15, pp. 145-166, 2005.
- [21] ISO, "ISO/IEC/IEEE 29119: Software and systems engineering -- Software testing -- Part 2: Test processes", International Organization for Standardization, Geneva, 2013.
- [22] P. Cruz, R. Villarroel, F. Mancilla, and M. Visconti, "A Software Testing process for the reference model of Competisoft", in *Chilean Computer Science Society (SCCC), 2010 XXIX International Conference of the*, 2010, pp. 51-59.
- [23] A. Sanz, J. Saldaña, J. García, and D. Gaitero, "TestPAI: A proposal for a testing process area integrated with CMMI", *European Systems and Software Process Improvement and Innovation (EuroSPI 2008)*, Dublin, Ireland, 2008.
- [24] A. Rodrigues, P. R. Pinheiro, and A. Albuquerque, "The definiton of a testing process to small-sized companies: the Brazilian scenario", in *Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the*, 2010, pp. 298-303.
- [25] M. Rojas, "Proceso de pruebas pequeñas organizaciones desarrolladoras de software "Ingeniería de Sistemas Tesis de pregrado, Departamento de Sistemas, Universidad del Cauca, Popayán, 2013.
- [26] R. K. Yin, *Case Study Research: Design and Methods*. Newbury Park: Sage Publications, 2003.