



PROCESO DE TESTING FUNCIONAL DE SOFTWARE PARA LAS MIPYMES
DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA

LORENA CARDONA BENJUMEA

UNIVERSIDAD AUTONOMA DE MANIZALES
MAESTRÍA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE
FACULTAD DE INGENIERÍA
MANIZALES

2019

PROCESO DE TESTING FUNCIONAL DE SOFTWARE PARA LAS MIPYMES
DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA

Proyecto de grado para optar al título de
Magíster en Gestión y Desarrollo de Proyectos de Software

LORENA CARDONA BENJUMEA

Tutor

Dr. MAURICIO FERNANDO ALBA CASTRO

UNIVERSIDAD AUTONOMA DE MANIZALES
MAESTRÍA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE
FACULTAD DE INGENIERÍA
MANIZALES

2019

DEDICATORIA

A Dios, por darme la vida y la sabiduría para realizar este proyecto.

A la memoria de mi padre José Yeiro Cardona Osorio.

A mi madre María Emir Benjumea Loaiza y a mi hermana Gloria Orfadi Cardona

Benjumea, por su amor y apoyo incondicional.

Por último, pero no menos importante, a mis amigas: Juliana Giraldo Marín, Luz Marina Mejía Giraldo y Luisa Fernanda Corredor Arias, sin cuyo apoyo y motivación no habría logrado ésta meta.

AGRADECIMIENTOS

*Agradezco sinceramente a mi Director de Tesis,
Doctor Mauricio Fernando Alba Castro,
por su orientación y apoyo en el desarrollo
de esta investigación.*

*A las MiPymes desarrolladoras de software de la ciudad de Pereira
por permitirme conocer su proceso de pruebas.*

*A la empresa que colaboró
con la validación del proceso propuesto.*

*A mis compañeros y amigos, por apoyarme y contribuir
de una u otra forma a la consecución de esta meta.*

RESUMEN

El presente documento pretende dar cuenta de un trabajo de investigación que responde a la necesidad de indagar sobre los problemas de calidad del sector del software, específicamente a lo relacionado con la forma en que la industria del software de la ciudad de Pereira aplica el proceso de pruebas, con el fin de desarrollar una propuesta que contribuya al mejoramiento de la calidad del software producido. Para tal fin, el proyecto plantea la creación de un proceso de pruebas funcionales¹ que se puede aplicar de manera independiente al proceso de desarrollo del software elaborado por las MiPymes de la industria local.

La investigación plantea el desarrollo de las siguientes fases:

- Construcción del referente teórico.
- Caracterización de las prácticas de pruebas de software empleadas por las MiPymes desarrolladoras de software de la ciudad de Pereira.
- Diseño del proceso de pruebas funcionales de software para las MiPymes desarrolladoras de software de la ciudad de Pereira, utilizando el metamodelo SPEM.
- Validación del proceso planteado mediante dos pilotos de aplicación.
- Realización de mejoras al proceso propuesto.
-

Como resultado se obtuvo un proceso estructurado a través de una secuencia de actividades puntuales, que establecen la aplicación de técnicas soportadas en estudios previos, los atributos de calidad planteados por la norma ISO/IEC/25000 y los lineamientos definidos por el estándar ISO/IEC/IEEE 29119.

Finalmente, y como conclusión, la aplicación del proceso en dos proyectos elaborados por una empresa local que hizo parte de la validación, logró que el software producido

¹ Pruebas basadas en un análisis de la especificación de funcionalidad de un componente o sistema (ISTQB, 2017)

alcanzara mayores niveles de satisfacción por parte de los usuarios finales, situación que da cuenta de los buenos resultados obtenidos.

Palabras clave: Pruebas de software, calidad de software, mejora de procesos de software.

ABSTRACT

This document aims to present the research that response the need of investigate about the problems of the quality of the software sector specifically related to the way of the software industry apply the test process in Pereira City with the purpose of developing a proposal which contribute to the improvement of the quality of the software produced. For that objective the project set the creation of a functional process which can be applied independently of the process of development software elaborated by the SMEs (Small and Medium Enterprises) of local industry.

The research establishes the development of the following phases:

- Construction of the theoretical referent
- Characterization of the software testing practices used by the software developers of the city of Pereira.
- Design of the functional software testing process for the SMEs software developers of the city of Pereira, using the SPEM metamodel.
- Validation of the process proposed through two application pilots.
- Realization of improvements to the proposed process:

As a result it was obtained a structured process through a sequencer punctual activities which establish the application of supported techniques in previous studies, the attributes of quality set by the norm ISO/IEC/9126 and the guidelines defined by the standard ISO/IEC/IEEE 29119.

To conclude the application of the process in two projects elaborated by a local company which was part of the validation achieve that the produced software reached the highest levels of satisfaction by the final users, this situation demonstrates the excellent results obtained.

Keywords: Software testing, software quality, improvement of software process.

CONTENIDO

1. PRESENTACIÓN	18
2. INTRODUCCIÓN	19
3. ÁREA PROBLEMÁTICA	22
4. JUSTIFICACIÓN	33
5. REFERENTE TEÓRICO	35
5.1 ANTECEDENTES	35
5.1.1 Estrategia para desarrollar la perspectiva “Procesos internos en un laboratorio de pruebas de software”	35
5.1.2 Marco Metodológico para la Mejora de las Actividades de Verificación y Validación de Productos Software.	38
5.1.3 Proceso de testing funcional independiente.	41
5.1.4 Modelo formal de pruebas funcionales de software para alcanzar el nivel de madurez integrado 2.	47
5.1.5 Proceso de pruebas para pequeñas organizaciones desarrolladoras de software.	50
5.1.6 Tmap - Test management approach.	53
5.1.7 Resumen de antecedentes encontrados.	58
5.2 DEFINICIONES	61
5.2.1 Prueba.	61
5.2.2 Caso de prueba.	61
5.2.3 Error.	61
5.2.4 Defecto.	61
5.2.5 Fallo.	62
5.2.6 Validación.	62
5.2.7 Verificación.	62
5.2.8 Plan de pruebas.	62
5.2.9 Niveles de prueba.	63
5.2.10 Tipos de prueba.	64
5.2.11 Técnicas de prueba.	66
5.2.12 Pruebas durante el ciclo de vida del software.	68
5.3 ORGANIZACIONES RELACIONADAS CON LAS PRUEBAS DE SOFTWARE	70
5.3.1 ISTQB: International software testing qualification board.	70
5.3.2 ISO: International organization for standardization.	70

5.4	PRINCIPALES ESTÁNDARES, MODELOS Y METODOLOGÍAS DE PRUEBAS	71
5.4.1	TMM – Test maturity model.....	71
5.4.2	Test Maturity Model integration (TMMi).....	73
5.4.3	TPI – Test process improvement.	76
5.4.4	ISO/IEC/IEEE 29119 Estándar Internacional de Pruebas de Software.	79
5.5	SOFTWARE PROCESS ENGINEERING METAMODEL (SPEM).....	82
5.6	ESTADO ACTUAL DEL PROBLEMA.....	83
6.	OBJETIVOS	85
6.1	OBJETIVO GENERAL.....	85
6.2	OBJETIVOS ESPECÍFICOS	85
7.	ALCANCES Y LIMITACIONES	86
8.	METODOLOGÍA	87
8.1	ENFOQUE.....	87
8.2	TIPO DE ESTUDIO	90
8.3	TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE LA INFORMACIÓN	90
8.4	PROCEDIMIENTOS.....	91
8.5	ANÁLISIS DE RESULTADOS	92
9.	RESULTADOS	93
9.1	CARACTERIZACIÓN DE LAS PRÁCTICAS DE PRUEBAS DE SOFTWARE EN LAS MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA.....	93
9.1.1	Tabulación y gráficos de la información.....	93
9.1.2	Análisis de resultados de la encuesta aplicada a las empresas.....	114
9.2	FORMULACIÓN DEL PROCESO DE TESTING FUNCIONAL DE SOFTWARE “QUALITEST” PARA LAS MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA.....	119
9.2.1	Fases.....	119
9.2.2	Roles.	124
9.3	APLICACIÓN DEL PROCESO DE TESTING FUNCIONAL DE SOFTWARE PARA LAS MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA – “QUALITEST”	130
9.3.1	Evaluación inicial.....	130
9.3.2	Selección de proyectos para la prueba piloto.....	132
9.3.3	Medición inicial.	133

9.3.4	Aplicación del proceso en Proyecto 1.....	133
9.3.5	Aplicación del proceso en Proyecto 2.....	137
9.3.6	Análisis de los resultados obtenidos al aplicar el proceso.	141
9.3.7	Evaluación del proceso aplicado.....	144
10.	DISCUSIÓN DE RESULTADOS	146
11.	CONCLUSIONES	148
12.	RECOMENDACIONES	150
13.	RESULTADOS EN GENERACIÓN DE CONOCIMIENTO	151
14.	BIBLIOGRAFÍA	152
15.	ANEXOS	157

LISTA DE ILUSTRACIONES

Ilustración 1: Porcentaje de empresas que utilizaron computador, internet y página o sitio web sector comercio e industria manufacturera - Total Nacional 2015	22
Ilustración 2: Porcentaje de empresas que utilizaron aplicaciones o programas informáticos según forma de obtención Sector Industrial Manufacturero - Total nacional 2015	23
Ilustración 3: Ventas de la Industria TI como porcentaje del PIB en Colombia	23
Ilustración 4: Tamaño de las empresas por valor de ventas	24
Ilustración 5: Activos a nivel nacional	24
Ilustración 6: Participación de las empresas en distintos modelos de calidad – Total Nacional 566	25
Ilustración 7: Comparativo participación de las empresas colombianas en distintos modelos de calidad Año 2015 vs Año 2018.....	26
Ilustración 8: Exportaciones de software y servicios de TI - Dinámica trimestral (2013-2015)	29
Ilustración 9: Productos y/o Servicios TI Ofrecidos por Número de Empresas – Región ...	30
Ilustración 10: Líneas de Negocio Principales en la Región Eje Cafetero y Antioquia	30
Ilustración 11: Representación gráfica de las etapas de la estrategia para desarrollar los Procesos internos	35
Ilustración 12: Valores de cumplimiento de cada requisito	38
Ilustración 13: Visión General del Modelo de Procesos.	39
Ilustración 14: Etapas del Protest	42
Ilustración 15: Actividades involucradas en la Etapa del Estudio Preliminar y artefactos generados en cada una	43
Ilustración 16: Actividades involucradas en la Etapa de Planeación y artefactos generados en cada una	43
Ilustración 17: Sub-etapas en cada Ciclo de Prueba.....	44
Ilustración 18: Actividades involucradas en la Sub-Etapa Seguimiento del Ciclo y artefactos generados en cada una	44

Ilustración 19: Actividades involucradas en la Sub-Etapa de Configuración del Entorno y artefactos.....	45
Ilustración 20: Actividades involucradas en la Sub-Etapa Diseño de Pruebas y artefactos generados en cada una	45
Ilustración 21: Actividades involucradas en la Sub-Etapa Ejecución de las Pruebas y artefactos.....	45
Ilustración 22: Actividades involucradas en la Etapa de Evaluación y artefactos generados en cada una	46
Ilustración 23: Modelo formal de pruebas funcionales de software para alcanzar el nivel de madurez integrado 2	49
Ilustración 24: Business Driven Test Management.....	55
Ilustración 25: Tmap Test Process.....	56
Ilustración 26: Partición de Equivalencia.....	68
Ilustración 27: Modelo en V o Secuencial.....	69
Ilustración 28: Objetivos de madurez de TMM.....	72
Ilustración 29: Niveles de madurez y áreas de proceso de TMMi	74
Ilustración 30: Niveles de madurez y áreas de proceso de TMMi	76
Ilustración 31: Matriz de madurez de TPI.....	78
Ilustración 32: Facturación por línea de negocio – Regiones.....	83
Ilustración 33: Etapas del proceso de pruebas Qualitest	119
Ilustración 34: Fases del proceso en las que participa el Rol de Líder del Proceso de Pruebas y Productos de Trabajo generados.....	125
Ilustración 35: Fases del proceso en las que participa el Ingeniero de Pruebas y Productos de Trabajo generados.....	127
Ilustración 36: Fases del proceso en las que participa el Rol de Líder del Proceso de Desarrollo y Productos de Trabajo generados.....	128
Ilustración 37: Fases del proceso en las que participa el Rol de Ingeniero de Desarrollo y Productos de Trabajo generados.....	129

LISTA DE TABLAS

Tabla 1: Relación de empresas colombianas certificadas en distintos modelos de calidad .	25
Tabla 2: Promedio de Salarios de los Integrantes de un Área de Pruebas	27
Tabla 3: Salarios promedio existentes en una organización desarrolladora de software	27
Tabla 4: Resumen de antecedentes encontrados.....	58
Tabla 5: Elementos de la fase Análisis Inicial.....	120
Tabla 6: Elementos de la fase de Planeación.....	121
Tabla 7: Elementos de la fase de Diseño.....	122
Tabla 8: Elementos de la fase de Ejecución	123
Tabla 9: Elementos de la fase de Cierre	124
Tabla 10: Clasificación de proyectos según su duración.....	132
Tabla 11: Valores de medición inicial.....	133
Tabla 12: Lecciones aprendidas y buenas prácticas identificadas durante la aplicación del proceso Proyecto 1	136
Tabla 13: Lecciones aprendidas y buenas prácticas identificadas durante la aplicación del proceso Proyecto 2	140
Tabla 14: Indicadores obtenidos después de aplicar el proceso de pruebas propuesto	141
Tabla 15: Resultados y productos esperados	151

LISTA DE GRÁFICOS

Gráfico 1: Años de creación de la empresa	94
Gráfico 2: Número de empleados que tiene la empresa	94
Gráfico 3: Principal actividad de la empresa.....	95
Gráfico 4: Metodología, modelo o marco de trabajo de desarrollo de software utilizado por las empresas Fuente: Elaboración propia	95
Gráfico 5: Certificaciones de calidad	96
Gráfico 6: Certificaciones de calidad de las empresas encuestadas	96
Gráfico 7: Respuesta a la pregunta sobre administración de configuraciones de software..	97
Gráfico 8: Respuesta a la pregunta sobre administración de versiones de software	97
Gráfico 9: Técnicas que utilizan las empresas para llevar a cabo la especificación de software	98
Gráfico 10: Utilización de proceso o metodología para el desarrollo de las pruebas de software	98
Gráfico 11: Respuesta a la pregunta sobre tenencia de un área especializada de pruebas de software	99
Gráfico 12: Forma en que llevan a cabo las pruebas las empresas encuestadas	100
Gráfico 13: Número de personas que conforman el Área de Pruebas en las empresas encuestadas	100
Gráfico 14: Respuesta a la pregunta sobre contar con personal certificado en pruebas de software	101
Gráfico 15: Características y atributos de calidad que se tienen en cuenta al realizar pruebas.....	101
Gráfico 16: Tipos de pruebas que realizan las empresas encuestadas.....	102
Gráfico 17: Relación del número de empresas que cuentan con ambiente independiente de pruebas.....	102
Gráfico 18: Planificación de pruebas de software	103
Gráfico 19: Establecimiento de la duración del ciclo de pruebas en el Plan de Pruebas ...	103

Gráfico 20: Respuesta a pregunta sobre planificación de pruebas desde el inicio del proyecto	104
Gráfico 21: Respuesta a pregunta sobre establecimiento de atributos de calidad en las pruebas.....	104
Gráfico 22: Respuesta a pregunta sobre establecimiento de tipos de pruebas en el Plan de Pruebas	105
Gráfico 23: Empleo de Técnicas de Diseño de Casos de Prueba	105
Gráfico 24: Técnicas de diseño de pruebas utilizadas por las empresas encuestadas	106
Gráfico 25: Socialización de casos de prueba con el cliente.....	106
Gráfico 26 Respuesta sobre el registro de tiempo invertido y costos de las pruebas.....	107
Gráfico 27: Respuesta pregunta sobre presupuestación de la fase de pruebas en un proyecto	107
Gráfico 28: Respuesta sobre costo del proceso de pruebas para la organización.....	108
Gráfico 29: Porcentaje que representan las pruebas dentro de la duración de un proyecto de software	108
Gráfico 30: Respuesta a pregunta sobre registro de pruebas junto con el esfuerzo invertido.....	109
Gráfico 31: Respuesta a la pregunta sobre clasificación de defectos encontrados en la fase de pruebas.....	109
Gráfico 32: Clasificación de defectos encontrados en la fase de pruebas	110
Gráfico 33: Empleo de indicadores para evaluar la eficiencia del proceso de pruebas.....	110
Gráfico 34: Herramientas empleadas para diseñar y ejecutar las pruebas	111
Gráfico 35: Herramientas empleadas para gestionar los defectos encontrados en las pruebas.....	111
Gráfico 36: Consideración sobre mejoramiento del proceso de pruebas en las organizaciones	112
Gráfico 37: Consideración sobre tener talento humano especializado en pruebas.....	112
Gráfico 38: Consideración sobre contar con un área de pruebas en la organización	113
Gráfico 39: Consideración sobre contratar el servicio de pruebas de software.....	114
Gráfico 40: Bugs por categoría - Proyecto 1	135
Gráfico 41: Bugs por categoría - Proyecto 2	139

Gráfico 42: Esfuerzo invertido Proyecto 1	142
Gráfico 43: Cantidad de Bugs Reportados - Proyecto 1.....	142
Gráfico 44: Producto no conforme - Proyecto 1.....	142
Gráfico 45: Esfuerzo invertido Proyecto 2	143
Gráfico 46: Cantidad de Bugs Reportados - Proyecto 2.....	143
Gráfico 47: Producto no conforme - Proyecto 2.....	144

LISTA DE ANEXOS

ANEXO A: PRESUPUESTO.....	157
ANEXO B: ENCUESTA APLICADA A EMPRESAS DE LA CIUDAD DE PEREIRA	160
ANEXO C: PROCESO DE TESTING FUNCIONAL DE SOFTWARE PARA LAS MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA – “QUALITEST”	170
ANEXO D: FORMATO DE ENCUESTA APLICADO A LA EMPRESA EN LA QUE SE LLEVÓ A CABO LA APLICACIÓN DEL PROCESO.....	175

1. PRESENTACIÓN

La presente investigación denominada “*PROCESO DE TESTING FUNCIONAL DE SOFTWARE PARA LAS MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA*”, obedece al interés que se tiene en mejorar las buenas prácticas que la industria local del software lleva actualmente a cabo alrededor del proceso de pruebas. En ella se plantea un proceso estructurado a través de una secuencia de actividades puntuales, que establecen la aplicación de técnicas soportadas en estudios previos y en los lineamientos definidos por el estándar ISO/IEC/IEEE 29119.

El documento presenta inicialmente el área problema, la justificación y el referente teórico; en este último el lector podrá encontrar antecedentes, definiciones, organizaciones relacionadas con las pruebas de software, así como los principales estándares, modelos y metodologías de pruebas.

Seguidamente, se relacionan los objetivos, alcances y limitaciones de la investigación, al igual que la metodología utilizada, en la cual se hace énfasis en el enfoque, el tipo de estudio, las técnicas e instrumentos de recolección de información, así como los procedimientos llevados a cabo para el desarrollo del presente trabajo.

En la parte final del documento, se hace la presentación y discusión de los resultados obtenidos, posteriormente, se relacionan las conclusiones y recomendaciones derivadas de la investigación y, por último, las referencias bibliográficas y los anexos.

2. INTRODUCCIÓN

Con la gran expansión de las TIC ha aumentado considerablemente la producción de equipos que facilitan la sistematización de los procesos realizados por el hombre en diferentes campos y con ellos, también ha incrementado el desarrollo e implementación de soluciones a través del software, las cuales incluyen redes sociales, aplicaciones financieras, sistemas de alerta temprana y sistemas de información médica, por tan solo mencionar algunos. Es en este contexto donde la calidad se convierte en un aspecto clave, ya que las exigencias por parte de los clientes y usuarios son cada vez mayores, dado que a medida que avanza y crece la industria del software, se espera que las técnicas de prueba utilizadas contribuyan con la disminución del número de fallas encontradas durante la puesta en marcha de los sistemas creados.

En este contexto, se puede decir que, a lo largo de la historia, el mundo ha sido testigo del impacto que puede causar el mal funcionamiento del software, un gran ejemplo de ello lo ilustra (Tassey, 2002) en la investigación denominada *The Economic Impacts of Inadequate Infrastructure for Software Testing* en la cual se relaciona que en los últimos años la industria aeroespacial estadounidense ha perdido más de mil millones de dólares atribuidos a software problemático. También menciona que en varios proyectos las fallas suelen ser muy visibles y a menudo resultan en pérdida de reputación y pérdida de negocios futuros (pág. 35).

Por lo anterior y teniendo en cuenta el punto de vista tanto del cliente como de los usuarios finales, es posible mencionar que la calidad de un producto software es percibida en gran medida por la cantidad de fallas que se encuentran en su puesta en marcha y sobre todo, por la gravedad e impacto que éstas tienen para el usuario final, situación que obliga a las empresas desarrolladoras de software a implementar técnicas de prueba que contribuyan con el aseguramiento de la calidad de sus productos, más aun si se tiene presente que “el costo de encontrar y reparar defectos aumenta considerablemente a lo largo del ciclo de vida” (Graham, Van Veenendaal, Evans, & Black, 2009, pág. 8). Es en este punto donde cobra

importancia el concepto de Software Testing o Pruebas de Software, las cuales según (IEEE Computer Society, 2014) “consisten en la verificación dinámica de que un programa proporciona comportamientos esperados en un conjunto finito de casos de prueba adecuadamente seleccionado del dominio de ejecución generalmente infinito”.

Por otra parte, al analizar el panorama nacional con respecto al tema en mención, se puede decir que no es ajeno al escenario descrito, prueba de ello es que las empresas productoras de software se han preocupado por mejorar sus procesos de desarrollo, situación que se evidencia con el número de empresas certificadas en modelos de calidad como: CMMI, ISO/IEC, IT Mark, Moprosoft, MPS.Br, entre otras. De igual manera, es importante resaltar la evolución que ha tenido el sector, ya que, al analizar el comportamiento de las ventas de la industria con respecto al PIB, se evidencia un crecimiento positivo en los últimos años, pasando del 0,64% en el 2012 al 1,19 para el 2014 (Observatorio de Tecnologías de Información MinTIC, 2015, págs. 12-15). A la par del crecimiento del mercado, se incrementa también la necesidad al interior de la industria del software de implementar diferentes medidas que promuevan la producción de software de calidad y conduzcan a la obtención de ventajas competitivas, especialmente para las PYMES.

En cuanto al panorama de las empresas desarrolladoras de software locales se refiere, se encuentra que “la mayoría de ellas coinciden en que deben mejorar los procesos que garanticen la calidad del software” (Ramírez Aguirre & Ramírez Arias, 2010, pág. 76), además según avances de diferentes trabajos de investigación, se considera que “las MiPymes de la ciudad de Pereira requieren adoptar mejores prácticas en todas las fases del proceso de desarrollo, de tal manera que éstas puedan redundar en un producto software de más y mejor calidad” (Peláez & Toro Lazo, 2016, pág. 122).

Tomando como punto de referencia la situación anteriormente expuesta y con el ánimo de contribuir con el mejoramiento de la calidad del software producido por la industria local, se realizó el presente trabajo de investigación, mediante el cual se define un proceso de testing funcional de software para las MIPYMES desarrolladoras de software de la ciudad de Pereira, el cual se realizó siguiendo la estrategia de mejora continua PHVA (Planear-Hacer-

Verificar-Actuar)², la caracterización de las empresas locales, la aplicación de técnicas soportadas en estudios previos, los atributos de calidad planteados por la norma ISO/IEC/25000 y los lineamientos definidos por el estándar ISO/IEC/IEEE 29119.

² El nombre del Ciclo PDCA (o Ciclo PHVA) viene de las siglas Planificar, Hacer, Verificar y Actuar, en inglés “Plan, Do, Check, Act”. También es conocido como Ciclo de mejora continua o Círculo de Deming, por ser Edwards Deming su autor (PDCA Home, s.f.).

3. ÁREA PROBLEMÁTICA

Desde su aparición, las tecnologías de la información y la comunicación han influido notablemente en diversos aspectos de la sociedad. Según el DANE (2016):

Tanto para los países desarrollados como para los países en vías de desarrollo se ha evidenciado que las TIC son un factor clave para mejorar el desempeño productivo y, con esto, el crecimiento económico y desarrollo social al hacer parte de estrategias integrales de negocios (pág. 2).

Colombia no es un país ajeno a dicho comportamiento, ya que con el pasar de los años son cada vez más las organizaciones que incorporan a sus procesos productivos el uso de la tecnología. Prueba de ello es que, en 2015, un gran porcentaje de empresas del sector de comercio e industria manufacturera, utilizaron computador, internet y página o sitio web.

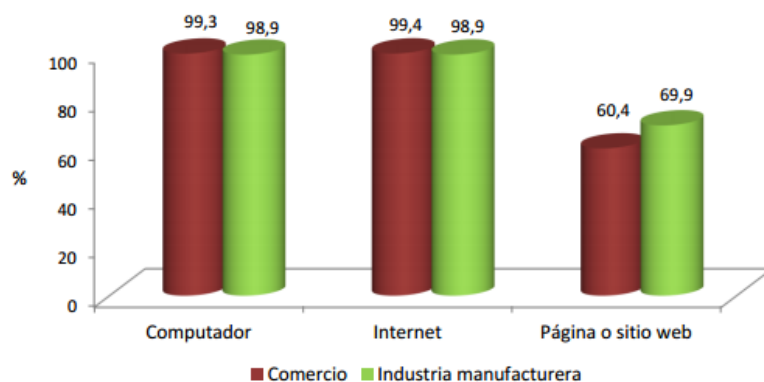


Ilustración 1: Porcentaje de empresas que utilizaron computador, internet y página o sitio web sector comercio e industria manufacturera - Total Nacional 2015

Fuente: (DANE, 2016, pág. 1)

De igual manera, según el Boletín Técnico Indicadores Básicos de TIC en Empresas 2015, publicado por el (DANE, 2016), en el cual se aplicó una encuesta que indagó por la forma de obtención de las aplicaciones o programas informáticos que utilizaron las empresas, “la compra para uso exclusivo tuvo el mayor porcentaje de respuesta con 84,8%; le siguieron el alquiler con pago periódico con 30,7% y la descarga o acceso gratuito con 20,5%” (pág. 6).

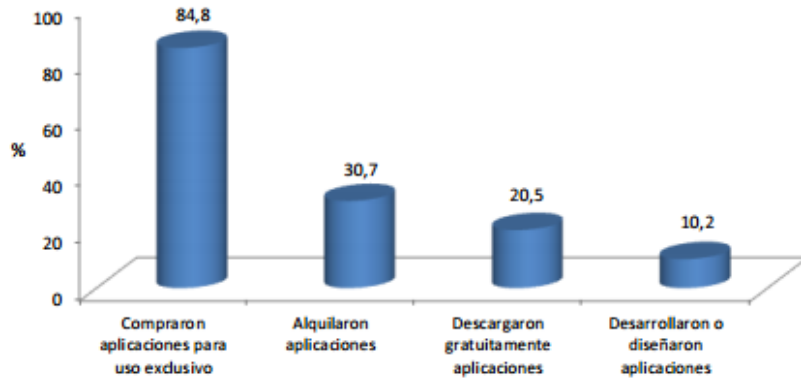


Ilustración 2: Porcentaje de empresas que utilizaron aplicaciones o programas informáticos según forma de obtención Sector Industrial Manufacturero - Total nacional 2015

Fuente: (DANE, 2016, pág. 7)

Al analizar las ventas de la industria de TI como porcentaje del PIB, se puede apreciar que dicho indicador ha tenido un crecimiento positivo en los últimos años, pasando del 0,64% en el 2012 al 1,19 para el 2.014 como se ilustra a continuación:

	2012	2013	2014
PIB a precios corrientes	\$664,240,000,000,000	\$710,257,000,000,000	\$756,152,000,000,000
Ventas de la Industria TI	\$4,252,000,000,000	\$5,945,000,000,000	\$8,980,163,013,000
Ventas de la Industria como porcentaje del PIB	0.64%	0.84%	1.19%

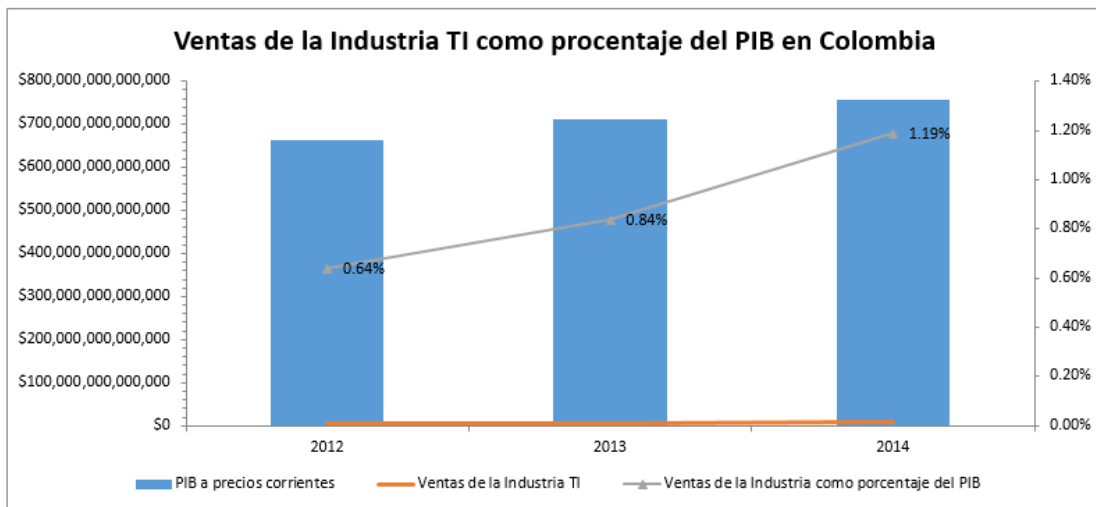


Ilustración 3: Ventas de la Industria TI como porcentaje del PIB en Colombia

Fuente: (Observatorio de la Industria TI, 2016)

Sin embargo, la industria aún se encuentra en proceso de crecimiento, ya que según (FEDESOF, 2015), “el 53% de las empresas del sector vende anualmente menos de 294 millones de pesos. Sólo el 4% de las empresas se encuentran en “grandes ligas” con ventas superiores a 17.000 millones de pesos anuales”.

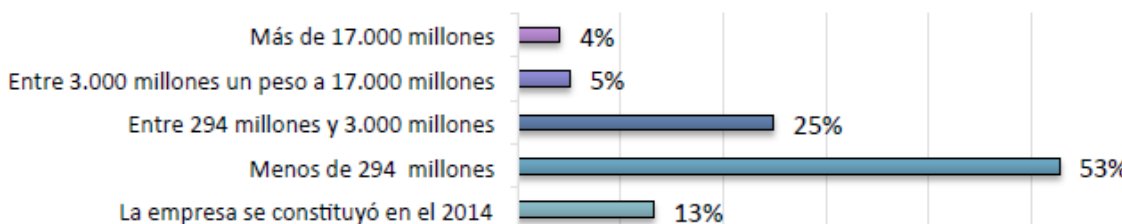


Ilustración 4: Tamaño de las empresas por valor de ventas

Fuente: (FEDESOF, 2015, pág. 12)

Otro factor que es importante considerar es el tamaño de las empresas del sector por rango de activos, ya que al realizar el análisis correspondiente, se encuentra que el 60% de las empresas a nivel nacional cuentan con menos de 294 millones de pesos en activos, el 20% entre 294 y 3.000 millones de pesos y el 9% tiene más de 3.000 millones de pesos.

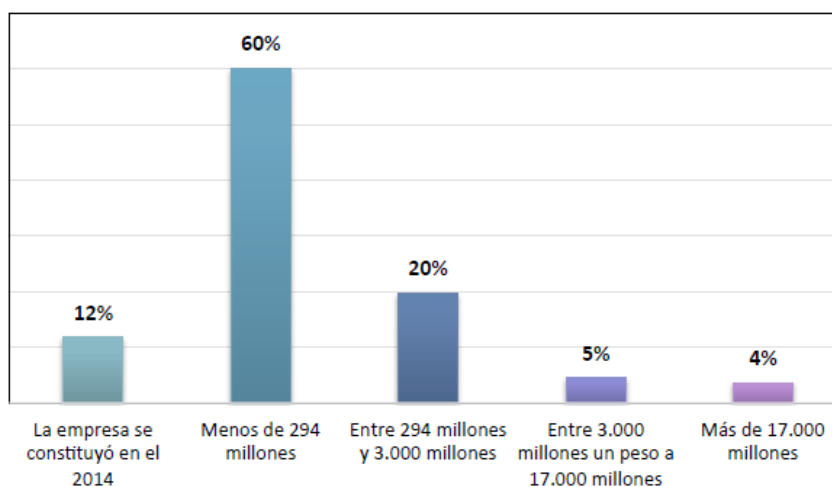


Ilustración 5: Activos a nivel nacional

Fuente: (FEDESOF, 2015, pág. 12)

En lo que respecta a las certificaciones de calidad, las cifras son más alentadoras, ya que tomando como base las 566 empresas que participaron del Informe de Caracterización

de Productos y Servicios del Sector de Tecnologías de Información en Colombia (FEDESOF, 2015), se destaca que todas afirman contar con al menos un modelo de calidad implementado, y algunas de ellas cuentan con más de un certificado. Dentro de las certificaciones con las que cuentan las empresas se encuentran: CMMI, ISO/IEC, IT Mark, Moprosoft, MPS.Br y una agrupación de otras certificaciones diferentes a las mencionadas.

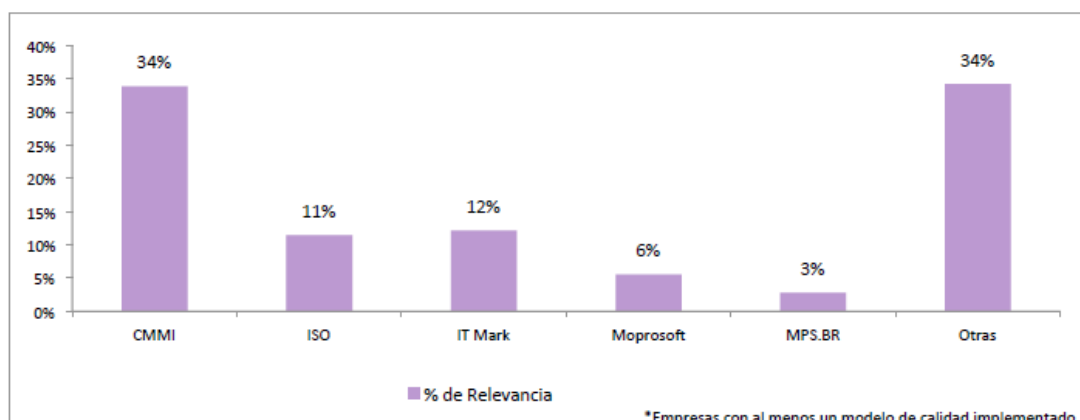


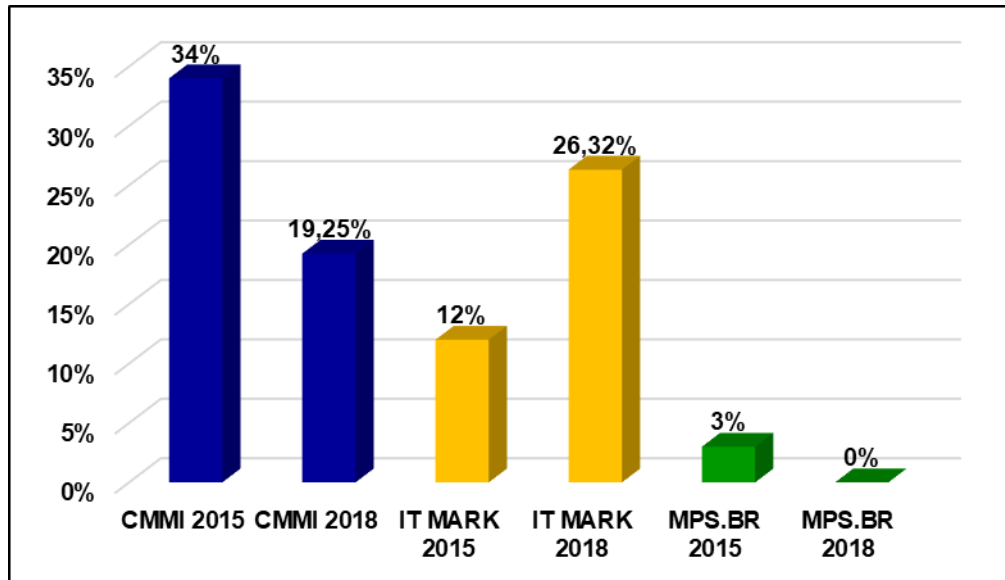
Ilustración 6: Participación de las empresas en distintos modelos de calidad – Total Nacional 566

Fuente: (Observatorio de Tecnologías de Información MinTIC, 2015, pág. 15)

Al consultar los reportes oficiales de entes certificadores de CMMI, IT Mark y MPS.BR, se encontró una reducción de empresas colombianas certificadas en CMMI y MPS.BR, mientras que aumentó el número de certificaciones de IT Mark, como se ilustra a continuación:

Tabla 1: Relación de empresas colombianas certificadas en distintos modelos de calidad

Estándar	Número de Empresas Colombianas Certificadas	Fuente
CMMI	109	(CMMI Institute, 2018)
IT Mark	149	(IT Mark, 2018)
MPS.BR	0	(Softex, 2018)



**Ilustración 7: Comparativo participación de las empresas colombianas en distintos modelos de calidad
Año 2015 vs Año 2018**

Fuente: Elaboración Propia

Por lo anterior y con el ánimo de continuar fortaleciendo la industria, se hace cada vez más importante producir software que funcione impecablemente, sea fácil de utilizar y, sobre todo, que cumpla con los estándares de calidad y satisfaga ampliamente las necesidades de quienes lo utilizan.

En este contexto, es donde se hace necesario implementar al interior de las empresas desarrolladoras de software técnicas de prueba que contribuyan con el aseguramiento de la calidad de sus productos. Sin embargo, el hecho de aplicar un proceso de testing al interior de una organización trae consigo costos que no todas las empresas están en capacidad de solventar, tal y como se ilustra a continuación en la proyección salarial de una organización cuya Área de Pruebas está compuesta por cuatro (4) integrantes, la cual fue realizada tomando como base el *Estudio de Salarios y Profesionales del Sector de Software y TI de Colombia*, llevado a cabo por el Observatorio de la Industria TI (2016):

Tabla 2: Promedio de Salarios de los Integrantes de un Área de Pruebas

Cargo	Salario Promedio año 2.016	Proyección Salario según el incremento del SMMLV para el año 2.017
Líder de Calidad Senior	\$2.838.117	\$ 3.036.785
Analista de Pruebas Senior	\$2.401.132	\$ 2.569.211
Analista de Pruebas Estándar	\$1.766.368	\$ 1.890.014
Analista de Pruebas Junior	\$1.268.388	\$ 1.357.175
Total Salarios	\$ 8.274.005	\$ 8.853.185

Si se continúan aplicando los salarios relacionados en el estudio antes mencionado para elaborar la proyección salarial de los demás cargos de la organización, se puede evidenciar que la carga salarial de un Área de Pruebas representa aproximadamente el 14% del total de la nómina mensual, mientras que la gestión del proyecto y la toma de requerimientos representan el 9,3% cada uno respectivamente.

Tabla 3: Salarios promedio existentes en una organización desarrolladora de software

Cargo	Salario Promedio año 2.016	Proyección Salario según el incremento del SMMLV para el año 2.017
Gerente de Operaciones	\$4.629.965	\$ 4.954.063
Gerente de Desarrollo	\$4.110.835	\$ 4.398.593
Arquitecto de Software Senior	\$4.636.914	\$ 4.961.498
Arquitecto de Software Estándar	\$3.797.398	\$ 4.047.315
Líder de Desarrollo Senior	\$3.782.537	\$ 2.871.889
Líder de Desarrollo Estándar	\$2.684.008	\$ 3.036.785
Líder de Calidad Senior	\$2.838.117	\$ 3.036.785
Ingeniero de Desarrollo Senior	\$3.467.650	\$ 3.710.386

Ingeniero de Desarrollo Estándar	\$2.411.787	\$ 2.580.612
Ingeniero de Desarrollo Junior	\$1.811.731	\$ 1.938.552
Administrador de la Configuración	\$3.775.094	\$ 4.039.351
Analista Funcional Senior	\$3.316.118	\$ 3.548.246
Analista Funcional Estándar	\$2.184.236	\$ 2.337.133
Administrador de Base de Datos	\$2.234.286	\$ 2.390.686
Director del Proyecto	\$3.748.557	\$ 4.010.956
Ingeniero de Soporte y Mantenimiento	\$2.231.336	\$ 2.387.530
Líder de Calidad Senior	\$2.838.117	\$ 3.036.785
Analista de Pruebas Senior	\$2.401.132	\$ 2.569.211
Analista de Pruebas Estándar	\$1.766.368	\$ 1.890.014
Analista de Pruebas Junior	\$1.268.388	\$ 1.357.175
Total	\$ 59.934.574	\$ 63.103.565

Es importante tener en cuenta que adicional a la carga salarial, el hecho de contar con un área dedicada a las pruebas de software al interior de una organización, trae consigo gastos prestacionales y de inversión, estos últimos relacionados con la infraestructura requerida para el montaje y mantenimiento de los ambientes de prueba.

Otro factor a resaltar, es el tamaño de las empresas existentes en Colombia de acuerdo a su rango de activos (Ver Ilustración 5), ya que según el *Estudio de Caracterización de Productos y Servicios del Sector de Tecnologías de la Información en Colombia*, realizado por el Ministerio de Tecnologías de la Información y las Comunicaciones de Colombia MinTic y la Federación Colombiana de la Industria de Software y TI en adelante FEDESOFIT (2015), “*el sector está conformado en su mayoría por Pymes*” (pág. 11), es decir por un grupo de empresas cuya planta de personal tiene entre once (11) y cincuenta (50) trabajadores, o con activos totales por valor entre quinientos uno (501) y menos de cinco mil (5.000) salarios mínimos mensuales legales vigentes (Senado de la República de Colombia, 2004).

Sumado a la problemática anterior, en el mismo estudio se realiza un análisis de las exportaciones de software y servicios de TI, en el cual se menciona que éstas presentan una tendencia decreciente entre los años 2.013 y 2.015, ya que pasaron de US \$ 292 millones en el 2.103 a US \$ 250 millones. Lo que “indica que la industria local de TI adolece de capacidades y herramientas suficientes para provechar las oportunidades de mercado a nivel externo” (MinTic - FEDESOFTEC, 2015).

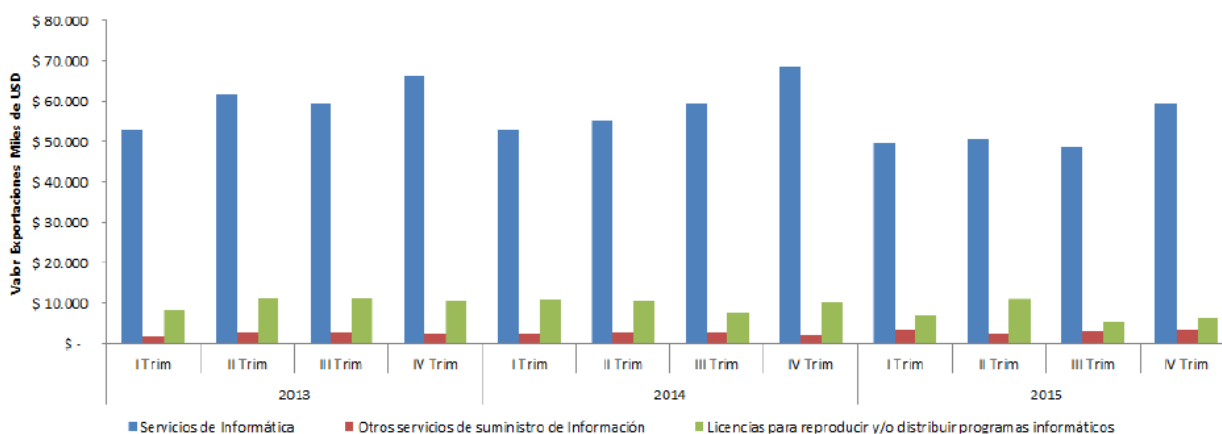


Ilustración 8: Exportaciones de software y servicios de TI - Dinámica trimestral (2013-2015)

Fuente: (Observatorio de Tecnologías de Información MinTIC, 2015, pág. 18)

Al plasmar la problemática anteriormente descrita en el plano regional, se puede decir que el Eje Cafetero no es ajeno a dicha situación, dado que a pesar de que en la región hacen presencia algunas multinacionales dedicadas a la producción de software, el sector está conformado en su gran mayoría por PYMES cuya línea de negocio es principalmente el Desarrollo de Software, tal y como se puede observar en los gráficos que se muestran a continuación:

SERVICIOS	REGIONES									Total General
	ANT	BOY	CAR	CUN	EJE C.	MET	PAC	SAN	Otras	
Manejo de centros de datos (data center)	129	1	47	538	28	2	53	37	16	851
Desarrollo / fábrica de software	119	4	35	464	52	4	54	25	15	772
Mesas de ayuda (Otras)	91	0	18	304	13	6	21	17	7	477
Testing de software	63	0	19	198	12	0	26	8	4	330
Infraestructura como servicio	43	1	18	186	14	0	23	9	6	300
Consultoría e implementación	18	0	2	106	3	1	9	2	2	143
Mantenimiento o soporte de aplicaciones	21	1	4	98	4	0	9	6	0	143
Software como servicio	15	0	5	80	7	0	6	3	0	116
Otro ¿Cuál?	10	0	2	85	3	1	8	4	2	115
Plataformas tecnológicas como servicio	13	0	4	57	3	1	10	2	0	90
Cloud computing	6	0	0	16	1	0	2	1	1	27
Gerencia	1	0	0	2	0	0	1	2	0	6
TOTAL GENERAL	529	7	154	2134	140	15	222	116	53	3370

Ilustración 9: Productos y/o Servicios TI Ofrecidos por Número de Empresas – Región

Fuente: (Observatorio de Tecnologías de Información MinTIC, 2015, pág. 14)

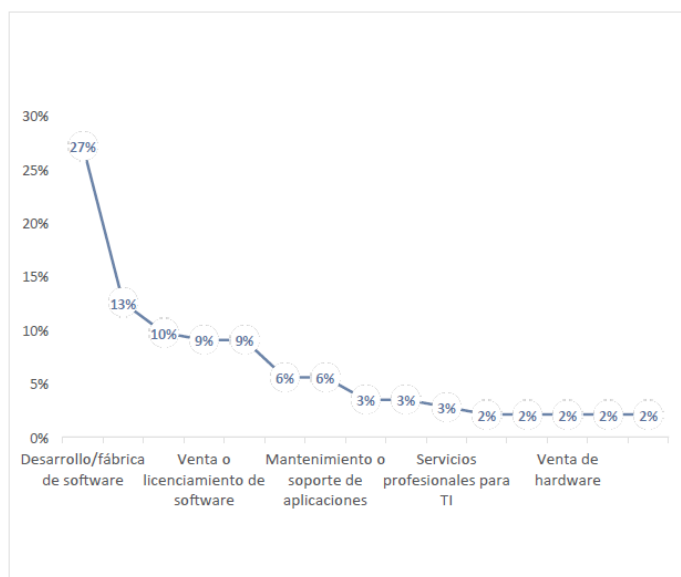


Ilustración 10: Líneas de Negocio Principales en la Región Eje Cafetero y Antioquia

Fuente: (Observatorio de Tecnologías de Información MinTIC, 2015, pág. 44)

De acuerdo con lo plasmado en las ilustraciones 9 y 10, de las 3.370 empresas que formaron parte del estudio a nivel nacional, un total de 772 ofertan principalmente servicios de Desarrollo/Fábrica de Software, cantidad que representa el 23% de la muestra, mientras

que 330 indicaron que ofertan principalmente el Testing de Software, lo cual corresponde al 9,79% del total de la muestra. En lo que respecta a la Zona del Eje Cafetero, se puede apreciar que la actividad principal es el Desarrollo/Fábrica de Software, mientras que el Testing ni siquiera figura dentro de los principales servicios ofertados.

Por otra parte, también es importante resaltar que la ciudad de Pereira no es ajena a dicha situación, dado que a pesar de que varias multinacionales dedicadas a la producción de software tienen su sede en este Municipio, se puede decir que el sector Software está conformado en su gran mayoría por PYMES, las cuales probablemente no cuentan con los recursos suficientes para conformar un área dedicada exclusivamente al testing del software producido. Según (Ramírez Aguirre & Ramírez Arias, 2010), la gran mayoría de estas empresas desarrollan software empaquetado y software a la medida y están orientadas a desarrollar software para el mercado nacional, sin embargo, varias de ellas contemplan dentro de su visión abarcar el mercado internacional y coinciden en que deben mejorar los procesos que garanticen la calidad del software.

Por lo anterior, se considera pertinente plantear un proceso de pruebas de software funcionales para ser ejecutado de manera independiente, es decir, que permita llevar a cabo el testing por parte de organizaciones diferentes a las que realizan el desarrollo del software, con el fin de contribuir en primera medida, con el mejoramiento de los procesos de pruebas aplicados a los productos elaborados por las empresas desarrolladoras de software de la ciudad de Pereira y en segunda medida, con la especialización del sector, la cual es muy necesaria para crear una real cadena de valor y con ello, constituir un referente para el sector a nivel nacional que posibilite el incremento de la calidad y la competitividad de los productos software producidos por la industria colombiana.

Para tal fin, se tendrá en cuenta el ciclo PHVA, también conocido como Ciclo de mejora continua o Círculo de Deming, por ser este último su autor. De acuerdo con (García P, Quispe A, & Ráez G, 2003):

A partir del año 1950, y en repetidas oportunidades durante las dos décadas siguientes, Deming empleó el Ciclo PHVA como introducción a todas y cada una de las

capacitaciones que brindó a la alta dirección de las empresas japonesas. De allí hasta la fecha, este ciclo (que fue desarrollado por Shewhart), ha recorrido el mundo como símbolo indiscutido de la Mejora Continua. Las Normas NTP-ISO 9000:2001 basan en el Ciclo PHVA su esquema de la Mejora Continua del Sistema de Gestión de la Calidad.

Es importante destacar que se selecciona el área de pruebas dentro de los elementos que conforman la calidad del software porque es una de las áreas más conocidas y aplicadas, razón por la cual es posible encontrar importantes referentes como modelos, estándares, certificaciones, etc, que se pueden tomar como base para plantear un proceso de pruebas funcionales que contribuya a mejorar en la región la problemática expuesta anteriormente. De igual manera, vale la pena resaltar que se plantea como solución un proceso de pruebas funcionales porque estas abarcan en gran medida los requerimientos y solicitudes realizadas por los usuarios, razón por la cual es más fácil determinar si la solución que se está construyendo cumple o no con los niveles de aceptación descritos por el cliente y con los criterios de calidad establecidos. A su vez, el hecho de que sea independiente hace que el proceso se ejecute de una manera más objetiva y lejos del control gerencial de la organización que realiza el desarrollo, adicional a que se fomenta la creación de empresas independientes cuya actividad principal sea la ejecución de las pruebas, situación que contribuirá con la especialización de profesionales en dicho tema al igual que con el mejoramiento de la calidad del software producido por las PYMES desarrolladoras de la ciudad de Pereira.

4. JUSTIFICACIÓN

La constante evolución tecnológica hace a la industria de TI cada vez más competitiva. Día a día son mayores los retos que las empresas pertenecientes a este sector deben afrontar. Diferentes modelos, estándares, metodologías, normas y guías que hacen parte de la ingeniería del software surgen a la par de los avances tecnológicos con el ánimo de garantizar la calidad y el óptimo funcionamiento de las soluciones propuestas.

Así pues, se hace evidente que el uso de dichas herramientas proporcionadas por la ingeniería del software, constituye un elemento fundamental para garantizar que tanto la construcción como el mantenimiento de los productos software se haga de manera eficiente, con el fin de avalar que estos cumplen en gran medida con los niveles de calidad establecidos. Por lo anterior, se considera de gran importancia que las diferentes organizaciones productoras de software en Colombia, conozcan y utilicen en sus procesos de desarrollo uno o varios de los modelos de calidad que actualmente proporciona la ingeniería del software a nivel mundial.

Es en este contexto, donde la aplicación de pruebas de software funcionales es de gran relevancia y más aún si estas se llevan a cabo de manera independiente dado que según el Informe de Caracterización del Sector Software y Tecnologías de la Información en Colombia, realizado por FEDESOFTE en el año 2015, el 60% de las empresas a nivel nacional cuentan con menos de 94 millones de pesos en activos, situación que evidencia que el sector está conformado en su gran mayoría por Pymes, a las cuales les resulta costoso contratar personal especializado en pruebas de software para sostener un área de únicamente dedicada al testing. Lo anterior, complementado con lo que dice (Pérez Lamancha, 2006), quien afirma que:

La falta de objetividad que ocurre cuando el equipo de desarrollo es el mismo que el que realiza las pruebas sumado a las presiones institucionales por salir al mercado hace que en muchos casos la calidad del software se conozca recién cuando se pone en producción (pág. 18).

Sin embargo, es importante resaltar que los desarrolladores pueden probar sus propios códigos.

Aunque normalmente esta responsabilidad se transfiere a un probador con vistas a contribuir a centralizar los esfuerzos y obtener mayores beneficios, tales como una visión independiente por parte de recursos formados y profesionales del ámbito de las pruebas. Un determinado nivel de independencia (evitando el sesgo del autor) a menudo hace del probador la figura más efectiva a la hora de detectar defectos y fallos (ISTQB, 2011, pág. 19).

También es importante resaltar la importancia de aplicar pruebas no funcionales, las cuales permiten evaluar los diferentes atributos de calidad del software. Sin embargo, vale la pena aclarar que dicho componente no forma parte de la presente investigación porque no hace parte del alcance establecido, no obstante, podría incluirse como parte de una investigación posterior con el fin de mejorar la propuesta inicial.

Por lo anterior, el fundamento de la presente investigación se centra en contribuir con el desarrollo del sector software a partir de dos aportes significativos: El primero, consiste en establecer un estado del arte en torno a las pruebas de software, abordando de ésta manera definiciones, elementos, estrategias y técnicas para la ejecución de pruebas de software, que sirvan como referente en la construcción de un proceso de pruebas que se pueda ejecutar de manera independiente, el cual constituirá el segundo aporte de éste trabajo investigativo, que tiene como finalidad principal, contribuir con el mejoramiento de los procesos de testing aplicados a los productos elaborados por las empresas desarrolladoras de software de la ciudad de Pereira y con ello, constituir un punto de partida para el sector a nivel nacional que posibilite la creación de más empresas especializadas en pruebas de software, al igual que el incremento de la calidad y la competitividad de los productos software producidos por la industria colombiana.

5. REFERENTE TEÓRICO

5.1 ANTECEDENTES

5.1.1 Estrategia para desarrollar la perspectiva “Procesos internos en un laboratorio de pruebas de software”.

Iniciativa liderada por la Universidad de las Ciencias Informáticas de la Habana Cuba, la cual presenta una estrategia para desarrollar la perspectiva *Procesos internos en un laboratorio de pruebas de software que brinda servicios outsourcing*, con el fin de facilitar su acreditación y certificación, así como un aumento en la eficiencia de las pruebas de software. Dicha iniciativa parte de la identificación de las normas ISO que tributan la acreditación y certificación de un laboratorio así como aquellas que son relativas a las pruebas de software, estas son: NC-ISO/IEC 9001: 2008, NC-ISO/IEC 17025:2006 y NC-ISO/IEC 9126-1:2005 (Capote García, Brito Riverol, Yzquierdo Herrera, & Febles Estrada, 2014).

5.1.1.1 Fases.

La estrategia plantea cuatro fases o etapas en las que se lleva a cabo su desarrollo, las cuales se relacionan en la siguiente gráfica mediante un diagrama de actividades:

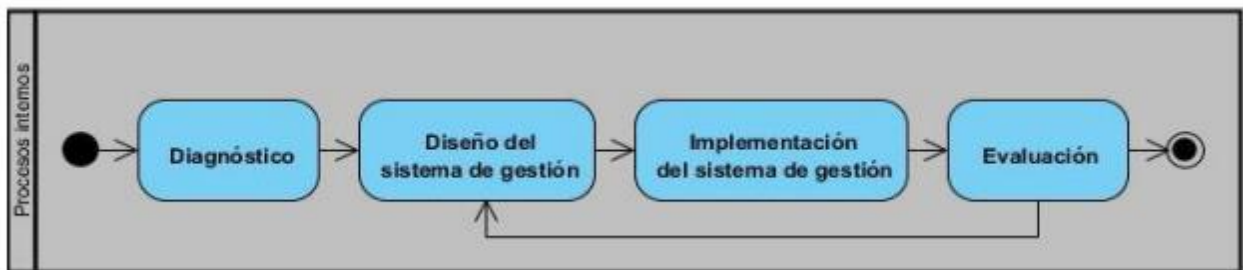


Ilustración 11: Representación gráfica de las etapas de la estrategia para desarrollar los Procesos internos

Fuente: (Capote García, Brito Riverol, Yzquierdo Herrera, & Febles Estrada, 2014, pág. 148)

Fase 1: Inicio-diagnóstico.

En esta fase se lleva a cabo la realización del diagnóstico de la organización. Dicha actividad se lleva a cabo mediante tres actividades: Diagnóstico, reunión con la dirección y constitución del equipo de trabajo (Capote García, Brito Riverol, Yzquierdo Herrera, & Febles Estrada, 2014, págs. 148-149).

Fase 2: Diseño del sistema de gestión.

Fase en la que se establecen cuatro actividades: Capacitar al personal, Diseñar el sistema de gestión, Identificar la documentación de gestión y técnica y Definir plantillas, documentos tipo (Capote García, Brito Riverol, Yzquierdo Herrera, & Febles Estrada, 2014, págs. 149-150).

Fase 3: Implementación del sistema de gestión.

En esta fase los procesos deben ser documentados con el fin de lograr la estandarización y el establecimiento de una cultura de organización. Las actividades que se desarrollan son: Elaborar la documentación, Elaborar el Manual de Calidad y Pilotar el sistema de gestión. (Capote García, Brito Riverol, Yzquierdo Herrera, & Febles Estrada, 2014, págs. 150-151)

Fase 4: Evaluación.

En esta fase se lleva a cabo un control exhaustivo y la detección de no conformidades, definidas como incumplimientos de requisitos. La evaluación se debe realizar mediante visitas de control, así como auditorías internas y externas, contempladas en un Plan de visitas y auditorías. Como evidencia del proceso, se elaboran informes con los hallazgos de la evaluación y los planes de acciones preventivas, correctivas y/o de mejora. Después de haber ejecutado las actividades antes mencionadas y cuando el resultado de las evaluaciones indiquen que la organización cumple con los requisitos de las normas, puede pasarse al paso formal de la acreditación y certificación (Capote García, Brito Riverol, Yzquierdo Herrera, & Febles Estrada, 2014, págs. 151-152).

5.1.1.2 Implementación de la estrategia desarrollada.

Todas las actividades propuestas en la estrategia desarrollada, fueron implementadas en el Laboratorio de Pruebas del Centro Nacional de Calidad de Software (CALISOFT) de Cuba, como parte de un modelo para aumentar la eficiencia de las pruebas de software (Capote García, Brito Riverol, Yzquierdo Herrera, & Febles Estrada, 2014, pág. 8).

5.1.1.3 Validación de la estrategia.

Para evidenciar la aplicación y validez de la estrategia propuesta, se aplicaron las siguientes técnicas y métodos de investigación:

a. Entrevista con especialistas y directivos de las áreas de informática y normalización

Una vez diseñada la estrategia, se realizó una entrevista a profundidad a especialistas y directivos de las áreas de informática y normalización, con conocimientos y experiencia en temas de acreditación de laboratorios, que pertenecen a tres entidades nacionales. Se realizó una explicación sobre la estrategia y fueron ejecutadas ocho preguntas, como guía para la entrevista (Capote García, Brito Riverol, Yzquierdo Herrera, & Febles Estrada, 2014, pág. 8).

b. Comparación de resultados obtenidos

“Se contrastaron los resultados obtenidos en el diagnóstico inicial y la última auditoría externa realizada, con relación a la implementación de los procesos” (Capote García, Brito Riverol, Yzquierdo Herrera, & Febles Estrada, 2014, pág. 8). Para tal fin se consideraron los valores de cumplimiento definidos en la siguiente imagen:

Valor	Estado del requisito
0%	A- No aplicado, ni documentado/No existe.
25%	B- Documentado, no aplicado.
50%	C- Aplicado, no documentado.
75%	D- Aplicado y documentado.
100%	E- Aplicado, documentado y controlado.
NA	No aplica.

Ilustración 12: Valores de cumplimiento de cada requisito

Fuente: (Capote García, Brito Riverol, Yzquierdo Herrera, & Febles Estrada, 2014, pág. 5)

a. Realización de un cuasi experimento con pre-prueba, post prueba y grupos intactos

Una vez aplicadas las técnicas anteriores, se realizó un cuasi experimento con pre-prueba, post-prueba y grupos intactos, para demostrar que hubo un aumento de la eficiencia de las pruebas de software, posterior a la aplicación de la propuesta (Capote García, Brito Riverol, Yzquierdo Herrera, & Febles Estrada, 2014, pág. 9).

5.1.2 Marco Metodológico para la Mejora de las Actividades de Verificación y Validación de Productos Software.

Tesis doctoral elaborada por (Sanz Esteban, 2012), la cual presenta una investigación que se centra en la definición de un marco metodológico para la mejora de las actividades relacionadas con el proceso de pruebas de software. Se destaca como componente principal el modelo de procesos para la realización de las actividades de verificación y validación de productos software, el cual puede ser implantado en las organizaciones de modo que, independientemente de la metodología de desarrollo y de gestión de proyectos aplicadas, se disponga de un conjunto de actividades o buenas prácticas que guíen a los integrantes del equipo de pruebas sobre las tareas que tienen que llevar a cabo a lo largo del ciclo de vida del producto.

Otro componente importante es el modelo de competencias que permite identificar los roles que mejor se ajustan a cada actividad, así como las habilidades y conocimientos requeridos

por los integrantes del equipo de pruebas, de forma que puedan llevar a cabo su trabajo de forma satisfactoria.

De igual manera, se destaca como objetivo principal del marco metodológico el proporcionar el conjunto de prácticas eficientes para desarrollar las tareas relacionadas con las pruebas de software y formar y gestionar equipos de pruebas eficientes, es decir que tengan conocimiento acerca del trabajo que han de desarrollar y dispongan de las competencias requeridas para ello (Sanz Esteban, 2012, pág. 69).

5.1.2.1 Módulos que componen el marco metodológico.

El modelo de procesos propuesto consta de 5 módulos independientes y complementarios, los cuales se ilustran a continuación:



Ilustración 13: Visión General del Modelo de Procesos.

Fuente: (Sanz Esteban, Marco Metodológico para la mejora de las actividades de Verificación y Validación de Productos Software (Tesis Doctoral), 2012, pág. 69)

Procesos de gestión del servicio.

“Agrupa el conjunto de procesos que definen las actividades requeridas para implantar el marco metodológico propuesto en una estructura del tipo Factoría de Pruebas Software” (Sanz Esteban, Marco Metodológico para la mejora de las actividades de Verificación y Validación de Productos Software (Tesis Doctoral), 2012, pág. 69)

Procesos de soporte.

“Agrupa el conjunto de procesos que definen las actividades requeridas para controlar y asegurar la correcta ejecución de los procesos de pruebas software así como permitir implantar acciones de mejora continua de dichos procesos” (Sanz Esteban, Marco Metodológico para la mejora de las actividades de Verificación y Validación de Productos Software (Tesis Doctoral), 2012, pág. 70).

Procesos de pruebas software.

“Agrupa el conjunto de procesos que definen las actividades requeridas para llevar a cabo la verificación y validación de los productos software, tanto intermedios como finales” (Sanz Esteban, Marco Metodológico para la mejora de las actividades de Verificación y Validación de Productos Software (Tesis Doctoral), 2012, pág. 70).

Infraestructura.

“Agrupa el conjunto de procesos necesarios para proporcionar y gestionar la infraestructura requerida para dar soporte tecnológico a los procesos anteriores” (Sanz Esteban, Marco Metodológico para la mejora de las actividades de Verificación y Validación de Productos Software (Tesis Doctoral), 2012, pág. 70).

Personas.

“Define el modelo de competencias necesario para identificar los roles que mejor se ajustan a cada actividad así como las habilidades y conocimientos requeridos por los integrantes del equipo de pruebas” (Sanz Esteban, Marco Metodológico para la mejora de las actividades de Verificación y Validación de Productos Software (Tesis Doctoral), 2012, pág. 70).

5.1.2.2 Validación.

La tesis doctoral utiliza el método empírico para validar la solución propuesta, proponiendo una teoría formal que es evaluada a través de estudios empíricos que se realizan mediante casos de estudio y experimentos. Para tal fin, la validación de la solución propuesta se descompone en dos fases:

a. Fase I: Validación del Modelo de Procesos para Verificación y Validación de productos software.

Esta fase se subdivide a su vez en dos sub-fases:

Sub-Fase I.I: Realización de los proyectos software sin el uso de los procesos definidos en el Modelo de Procesos y Sub-Fase I.II: Realización de los proyectos software con el uso de los procesos definidos en el Modelo de Procesos. La razón por la que existen éstas dos fases, básicamente se centra en poder validar cómo influye en la calidad del software el uso de las actividades relacionadas con la verificación y validación de productos software definidas en la tesis doctoral. Para tal fin, en cada fase se tomaron como base 12 proyectos de gestión con características y complejidad similares, siguiendo la metodología MÉTRICA v3 con una duración aproximada de seis meses y realizando el mismo conjunto de actividades de desarrollo del producto (Sanz Esteban, 2012, págs. 201 - 203).

a. Fase II: Validación del Modelo de Competencias.

El objetivo de la Fase II es comprobar que las competencias incluidas en el modelo así como los niveles requeridos para cada uno de los roles identificados son las requeridas para la realización de las actividades relativas a la verificación y validación de productos software. Dicha validación se lleva a cabo por medio de una encuesta que es aplicada a personas con experiencia en uno o varios de los roles incluidos en el modelo (Sanz Esteban, 2012, pág. 18).

5.1.3 Proceso de testing funcional independiente.

Tesis de Maestría en Informática elaborada en la ciudad de Montevideo Uruguay por (Pérez Lamancha, 2006), en la cual se lleva a cabo una investigación que plantea la definición del ProTest (proceso de pruebas funcionales de productos de software realizadas en el Laboratorio de Testing Funcional del CES³), el cual sirve de guía para realizar pruebas

³ El CES, Centro de Ensayos de Software, es una organización uruguaya especializada en proveer servicios de testing a empresas de tecnologías de la información y de otras industrias que la utilizan intensamente, permitiendo el

funcionales a partir de las especificaciones del producto. Entre sus principales características se encuentra que es independiente del ciclo de vida utilizado para desarrollar el producto, basado en el análisis de riesgo del producto para definir las funcionalidades a probar y la prioridad con que serán probadas, además organiza el proyecto de prueba en ciclos, donde cada ciclo se corresponde con una versión del producto a probar (Pérez Lamancha, 2006, pág. 59).

5.1.3.1 Etapas.

ProTest consta de cuatro etapas: Estudio Preliminar, Planificación, Ciclo de Prueba y Evaluación del proyecto, como se ilustra a continuación:



Ilustración 14: Etapas del ProTest

XCCFuente: (Pérez Lamancha, 2006, pág. 62)

a. **Etapa 1: Estudio preliminar.** De acuerdo con (Pérez Lamancha, 2006), en esta etapa:

Se estudian las principales funcionalidades del producto, con el objetivo de definir el alcance de las pruebas y un primer cronograma de los ciclos de prueba. A partir de estos datos se realiza la propuesta de servicio de prueba (la forma en que se realiza la cotización no forma parte de ProTest). Si el cliente aprueba la propuesta de servicio de prueba, se sigue con la etapa de Planificación, en caso contrario se pasa a la etapa de Evaluación, donde se analiza cuáles fueron los problemas en este proyecto y se archiva para su consulta en futuros proyectos (pág. 62).

incremento de su capacidad productiva mediante la mejora en la calidad, diversidad de plataformas e innovación de sus productos de software (CES, 2017).

Actividades	Artefactos generados
I1 - Definición del Servicio	RR - Resumen de Reunión PS - Propuesta de Servicio
I2 - Revisión preliminar de requerimientos	IP - Inventario de Prueba
I3 - Análisis preliminar de riesgo	IP - Inventario de Prueba
G1 - Estimación de las tareas	ET - Estimación de Tareas
G2 - Reportar el Esfuerzo	PE - Planilla de Esfuerzo

Ilustración 15: Actividades involucradas en la Etapa del Estudio Preliminar y artefactos generados en cada una

Fuente: (Pérez Lamanca, 2006, pág. 63)

b. Etapa 2: Planificación. El objetivo de esta etapa es planificar el proyecto de prueba, una vez que fue aprobado por el cliente. Se define los ciclos de prueba y las funcionalidades a probar en cada ciclo en función del análisis de riesgo del producto. Se genera el Plan de Pruebas que resume toda la información del proyecto de prueba y las decisiones tomadas durante la etapa de Planificación (Pérez Lamanca, 2006, pág. 62).

Actividades	Artefactos generados
P1 - Negociación con el Cliente	RR - Resumen de Reunión
P2 - Revisión de requerimientos	IP - Inventario de Pruebas
P3 - Análisis de riesgo	IP - Inventario de Pruebas
P4 - Exploración del Producto	IP - Inventario de Pruebas
P5 - Definición de los Ciclos de Prueba	ACP - Agenda de Ciclos de Prueba
P6 - Definición del Testware	TW - Testware
P7 - Planificación de las Pruebas	PP - Plan de Pruebas
P8 - Definición del Proceso de Incidentes	PI - Proceso de Incidentes
G1 - Estimación de las tareas	ET - Estimación de Tareas
G2 - Reportar el Esfuerzo	PE - Planilla de Esfuerzo

Ilustración 16: Actividades involucradas en la Etapa de Planeación y artefactos generados en cada una

Fuente: (Pérez Lamanca, 2006, pág. 64)

c. Etapa 3: Ciclo de prueba. El objetivo de esta etapa es generar y ejecutar las pruebas para una versión determinada del producto. El proyecto de prueba es guiado por los ciclos de prueba, cada ciclo de prueba está asociado a una versión del producto a probar. Esta etapa se divide a su vez en cuatro sub-etapas: Seguimiento del Ciclo, Configuración del Entorno, Diseño de las Pruebas y Ejecución de las Pruebas (Pérez Lamanca, 2006, págs. 62-63).

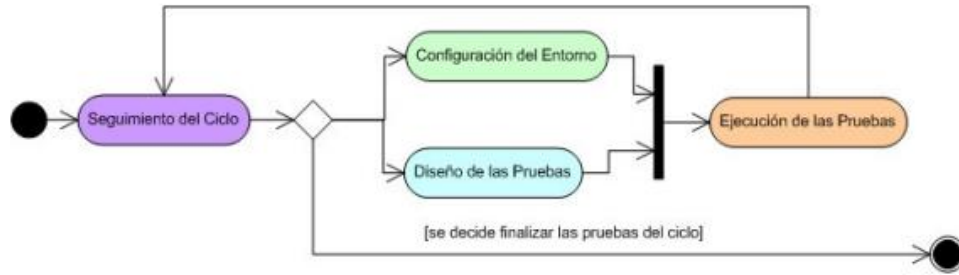


Ilustración 17: Sub-etapas en cada Ciclo de Prueba

Fuente: (Pérez Lamancha, 2006, pág. 65)

Seguimiento del ciclo. El objetivo de ésta sub-etapa es realizar el seguimiento y control del ciclo de prueba. La planificación realizada al principio del proyecto es revisada al comenzar cada ciclo de prueba, se planifican en detalle las tareas para el ciclo y se ajusta la planificación según las estimaciones y posteriores mediciones realizadas en los ciclos anteriores. Luego que se realizaron las otras tres subetapas del ciclo, se vuelve a esta sub-etapa, donde se evalúa el ciclo y se decide si se finaliza o si es necesario generar y ejecutar nuevas pruebas para el ciclo (Pérez Lamancha, 2006, pág. 65).

Actividades	Artefactos generados
S1 – Planificación del Ciclo	PPC – Plan de Pruebas del Ciclo
P3 – Análisis de Riesgo del Producto	IP – Inventario de Pruebas
S2 – Administración de la Configuración	RC – Reporte de Configuración
S3 – Seguimiento y Control del Ciclo	RA – Reporte de Avance del Ciclo
S4 – Evaluación del Ciclo de Prueba	RE – Reporte de Evaluación del Ciclo
G1 – Estimación de las tareas	ET – Estimación de Tareas
G2 – Reportar el Esfuerzo	PE – Planilla de Esfuerzo

Ilustración 18: Actividades involucradas en la Sub-Etapa Seguimiento del Ciclo y artefactos generados en cada una

Fuente: (Pérez Lamancha, 2006, pág. 66)

Configuración del entorno. “El objetivo de esta sub-etapa es configurar el ambiente de pruebas, separando los ambientes de desarrollo y prueba, instalar las herramientas necesarias y el software a probar en la versión correspondiente a cada ciclo de prueba” (Pérez Lamancha, 2006, pág. 66).

Actividades	Artefactos generados
C1 – Instalación de Herramientas	RO – Reporte de Obstáculos
C2 – Instalación y Configuración	RO – Reporte de Obstáculos

Ilustración 19: Actividades involucradas en la Sub-Etapa de Configuración del Entorno y artefactos generados en cada una

Fuente: (Pérez Lamancha, 2006, pág. 67)

Diseño de las pruebas. “Consiste en el diseño de los casos de prueba a partir de la especificación del producto. En caso de ser necesario, en esta sub-etapa se mejoran o generan las especificaciones junto con el Cliente” (Pérez Lamancha, 2006, pág. 66).

Actividades	Artefactos generados
P2 – Revisión de Requerimientos	IP – Inventario de Pruebas
D1 – Diseño de los Casos de Prueba	CP – Casos de Prueba MT – Matriz de Trazabilidad
D2 – Validación de los Casos de Prueba	RR – Resumen de Reunión
D3 – Asignación de los Casos de Prueba	PE – Plan de Ejecución

Ilustración 20: Actividades involucradas en la Sub-Etapa Diseño de Pruebas y artefactos generados en cada una

Fuente: (Pérez Lamancha, 2006, pág. 68)

Ejecución de las pruebas. “El objetivo de esta etapa es contrastar el comportamiento esperado del software con su comportamiento real, analizar las diferencias y reportar los resultados” (Pérez Lamancha, 2006, pág. 66).

Actividades	Artefactos generados
E1 – Pruebas de Humo	RP – Reporte de Ejecución de las Pruebas
E2 – Ejecución de las Pruebas	RP – Reporte de Ejecución de las Pruebas
E3 – Testing Exploratorio	RP – Reporte de Ejecución de las Pruebas
E4 – Reporte de Incidentes	RI – Reporte de Incidente
E5 – Validación de los Incidentes	RI – Reporte de Incidente
E6 – Verificación de las Correcciones	RP – Reporte de Ejecución de las Pruebas

Ilustración 21: Actividades involucradas en la Sub-Etapa Ejecución de las Pruebas y artefactos generados en cada una

Fuente: (Pérez Lamancha, 2006, pág. 69)

a. **Etapa 4: Evaluación.** “Esta etapa tiene como objetivo conocer el grado de satisfacción del cliente, realizar el informe final, evaluar el proceso de prueba para su mejora y almacenar los elementos del proyecto de prueba para su uso en proyectos posteriores” (Pérez Lamancha, 2006).

Actividades	Artefactos generados
V1- Evaluación de la Satisfacción del Cliente	ISC – Informe de Satisfacción del Cliente
V2 – Reporte Final del Proyecto	RF – Reporte Final del Proyecto de Prueba
V3 – Archivar el Testware	TW – Testware
V4 – Ajustes y Mejoras del Proceso de Prueba	APT – Ajustes al Proceso de Prueba
G1 – Estimación de las tareas	ET – Estimación de Tareas
G2 – Reportar el Esfuerzo	PE – Planilla de Esfuerzo

Ilustración 22: Actividades involucradas en la Etapa de Evaluación y artefactos generados en cada una

Fuente: (Pérez Lamancha, 2006, pág. 70)

De igual manera, es importante resaltar que el ProTest puede ser usado para realizar pruebas funcionales independientes de productos desde su comienzo, en sucesivos ciclos de desarrollo o en productos que se encuentra en mantenimiento. A su vez, define un conjunto de roles responsables de realizar las actividades del proceso y generar sus artefactos, estos son:

- Líder de Proyecto de Prueba
- Diseñador de Pruebas
- Tester
- Cliente
- Desarrollador o contraparte técnica.

El equipo de prueba está conformado por:

- El Líder de Prueba
- Los Diseñadores de Pruebas
- Los Testers

5.1.3.2 Metodología aplicada en la investigación. Tomando como base las cuatro etapas del ProTest, se puede intuir que la metodología aplicada en dicha investigación es la estrategia de mejora continua PHVA.

5.1.3.3 Técnicas de pruebas empleadas en el proceso. El proceso ejecuta pruebas funcionales con el fin de validar cuándo el comportamiento observado del software cumple o no con sus especificaciones. Para tal fin, se tienen en cuenta las especificaciones del producto al igual que se emplea una versión ejecutable del mismo. En caso de que no existan especificaciones o que las mismas estén incompletas, se trabaja junto al cliente y los desarrolladores para generarlas y mejorarlas (Pérez Lamancha, 2006, pág. 60).

5.1.3.4 Aplicación del Proceso. El proceso sólo se aplicó en un caso de estudio para el que se utilizó una aplicación de gestión que se encontraba en producción a la cual se le hacían mantenimientos y mejoras.

5.1.4 Modelo formal de pruebas funcionales de software para alcanzar el nivel de madurez integrado 2.

Artículo realizado por investigadores de la Universidad de las Fuerzas Armadas ESPE (Latacuna, Ecuador), en el cual se evidencian los resultados de una investigación enfocada en diseñar un modelo formal para desarrollar pruebas funcionales de software que permitan alcanzar el nivel de calidad 2 del Modelo de Madurez de Pruebas Integrado (TMMI).

El proceso se inició con un diagnóstico situacional, aplicando la norma ISO-9001-2000; luego, se evaluaron diversos modelos de prueba de calidad de software, como el ISO/IEC 9126, el TMM, el TMMI, el Proceso de Mejoramiento de Pruebas (TPI) y el Enfoque de Gestión de Pruebas (TMAP), realizando una comparación bajo algunos criterios como año de publicación, licenciamiento, niveles, factorías y riesgos. Con esta información se diseñó el modelo propuesto, que es independiente del proceso de desarrollo de software. Concretamente, se fundamentó en el ciclo de prueba, y se compone de cuatro fases: Especificación, Planificación, Ejecución y Evaluación, en el que se contrasta en forma real el comportamiento esperado del software. Como caso de estudio y validación se aplicó este modelo a una PYME; los resultados mostraron la eficiencia del modelo y revelaron que es preciso desarrollar una cultura de calidad organizacional en esta empresa (Escobar Sánchez & Fuertes Díaz, 2015, pág. 31).

5.1.4.1 Fases.

A continuación, se detallan las fases planteadas por el modelo para la ejecución de las pruebas de software:

a. Fase de Especificación

El objetivo es analizar las funcionalidades más importantes del software que se va a probar, con el fin de determinar el alcance de las pruebas y, así, establecer un cronograma de los ciclos correspondientes. Luego, si el cliente está de acuerdo, se procede con la planificación; caso contrario, se vuelve a realizar la especificación, donde se analizan los inconvenientes encontrados por el cliente, lo cual se documenta y guarda para tener una referencia para futuros proyectos (Escobar Sánchez & Fuertes Díaz, 2015, pág. 36).

b. Fase de Planificación

“El objetivo es planificar y diseñar las pruebas, ya aceptadas por el usuario. Se establecen los ciclos, las funcionalidades y el análisis de riesgo; en esta parte se genera un plan de pruebas” (Escobar Sánchez & Fuertes Díaz, 2015, pág. 36).

c. Fase de Ejecución

“El objetivo es realizar las pruebas de una versión del software, relacionándolas con el ciclo correspondiente” (Escobar Sánchez & Fuertes Díaz, 2015, pág. 36).

d. Fase de Evaluación y Resultados

El objetivo es determinar el informe final sobre las pruebas realizadas, la satisfacción del usuario con respecto al software probado y al proceso utilizado, a fin de realizar una mejora continua. Cabe destacar que esta información es almacenada, a manera de histórico, para pruebas futuras (Escobar Sánchez & Fuertes Díaz, 2015, pág. 36).

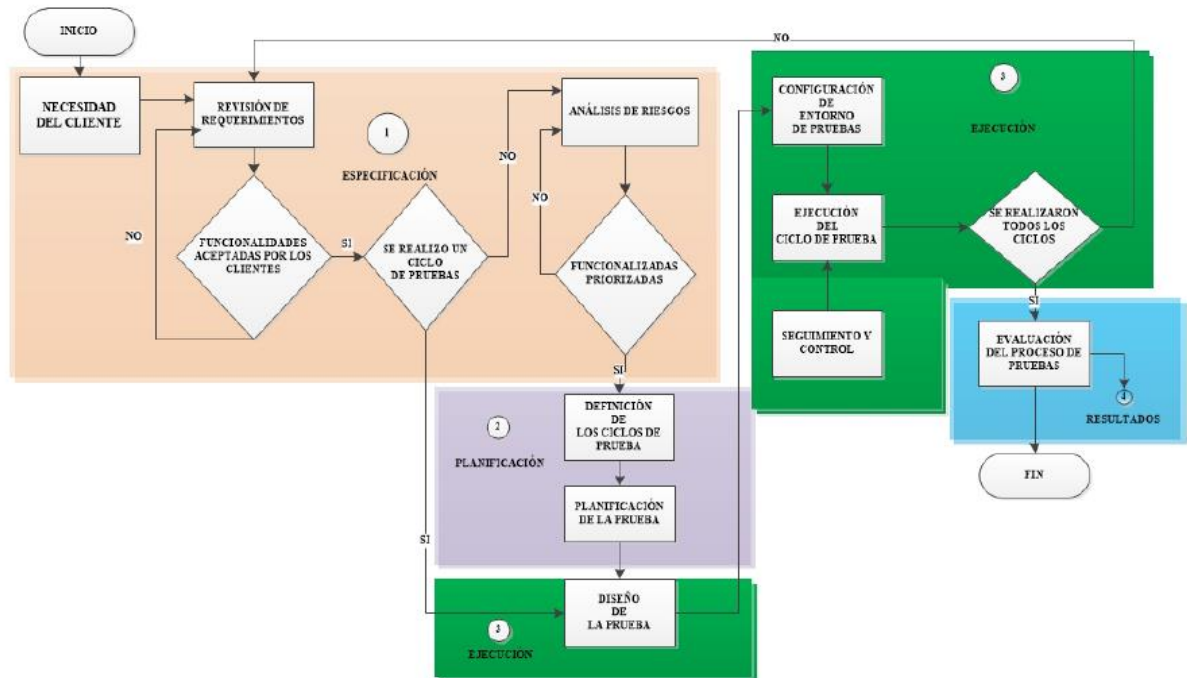


Ilustración 23: Modelo formal de pruebas funcionales de software para alcanzar el nivel de madurez integrado 2

Fuente: (Escobar Sánchez & Fuertes Díaz, 2015, pág. 37)

5.1.4.2 Aplicación del Modelo.

Tomando como base los resultados obtenidos en la Fase de Especificación, fue posible analizar las funcionalidades más importantes del software a probar, determinándose el alcance de las pruebas y un cronograma de los ciclos correspondientes. Una vez que el cliente estuvo de acuerdo, se procedió con la Planificación, donde se analizaron los inconvenientes encontrados por el cliente, al planificar y diseñar las pruebas; con este ajuste se procedió con la Ejecución y la evaluación de una versión del software; finalmente, se obtuvo el informe final sobre las pruebas realizadas. Las pruebas se desarrollaron en un sistema modular de administración educativa, y como resultado se evidenció la satisfacción del usuario con respecto al software probado y al proceso utilizado (Escobar Sánchez & Fuertes Díaz, 2015, pág. 37).

5.1.5 Proceso de pruebas para pequeñas organizaciones desarrolladoras de software.

Artículo realizado por investigadores de la Universidad del Cauca en Colombia, el cual propone un proceso liviano definido para guiar y apoyar la realización de las pruebas en pequeñas organizaciones desarrolladoras de software. Dicho proceso 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. De igual manera, es importante resaltar, que fue definido a partir del análisis de algunos procesos de pruebas existentes en modelos de referencia y ha sido aplicado exitosamente en una pequeña empresa de desarrollo de software (Rojas Montes, Pino Correa, & Martínez, 2015, pág. 57).

5.1.5.1 Fases.

El proceso propuesto consta de 6 fases: Inicio, Planeación, Diseño, Ejecución, Monitoreo y Control y Finalización; las cuales se describen a continuación:

a. 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 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 (Rojas Montes, Pino Correa, & Martínez, 2015, pág. 62).

b. 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 (Rojas Montes, Pino Correa, & Martínez, 2015, pág. 62).

c. Fase de Diseño

“Tiene como propósito construir los artefactos necesarios para ejecutar las pruebas en el proyecto (los artefactos de prueba hacen referencia a casos de prueba en lo relacionado a pruebas dinámicas y listas de chequeo en cuanto a pruebas estáticas)” (Rojas Montes, Pino Correa, & Martínez, 2015, pág. 62).

La Fase de Diseño se compone de las siguientes actividades:

FDA1. Análisis detallado de los elementos por probar: Esta actividad se lleva a cabo con el fin de obtener diseños más precisos que permitan encontrar un mayor número de errores.

FDA2. Seleccionar y validar las técnicas de pruebas: Se seleccionan y se validan las técnicas de pruebas más apropiadas para elaborar el conjunto de artefactos que evalúen los elementos descritos en la actividad anterior. Algunas de las técnicas consideradas son:

- Técnicas de pruebas funcionales.
- Partición equivalente.
- Análisis de valor límite.
- Tablas de decisión.
- Arreglos Ortogonales (Herramienta de ayuda allpairs).

FDA3. Definir los artefactos por diseñar y sus convenciones: Definir los artefactos a diseñar y sus convenciones, de tal manera que se conozca qué componente utilizar según sea el elemento por probar.

FDA4. 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 (Rojas Montes, Pino Correa, & Martínez, 2015, pág. 62).

d. 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 (Rojas Montes, Pino Correa, & Martínez, 2015, pág. 64).

e. 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 (Rojas Montes, Pino Correa, & Martínez, 2015, pág. 64).

f. 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” (Rojas Montes, Pino Correa, & Martínez, 2015, pág. 64).

5.1.5.2 Aplicación del proceso.

Se utilizó el método de casos de estudio para conducir la aplicación del proceso de pruebas en el contexto real de una empresa de software de la ciudad de Cali-Colombia, dedicada a brindar soluciones de software para “gobierno en línea”. El tipo de diseño de caso

de estudio empleado fue simple-holístico, ya que el proceso fue aplicado en la organización utilizando como unidad de análisis uno de sus proyectos de desarrollo relacionado con un portal para una Superintendencia en Colombia.

La ejecución del proceso de pruebas se llevó a cabo mediante tres ciclos. En el primer ciclo solo se ejecutaron 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 (Rojas Montes, Pino Correa, & Martínez, 2015, págs. 64-65).

5.1.6 Tmap - Test management approach.

TMap® (Test Management Approach) es la metodología de pruebas de SOGETI⁴, diseñada especialmente para abordar los principales problemas de calidad, tiempo y costes ocurridos durante todo el ciclo de vida de desarrollo de una solución tecnológica.

⁴ SOGETI es uno de los líderes en servicios informáticos y en ingeniería de proximidad, especializado en la gestión de aplicaciones e infraestructuras (*application and infrastructure management*) y en consultoría tecnológica (*high-tech engineering* – SOGETI HIGH TECH). SOGETI ofrece soluciones innovadoras sobre Testing, Business Intelligence Management, Movilidad, Cloud y Ciberseguridad, basándose en su metodología y su modelo mundial de prestaciones de servicios Rightshore® (Sogeti).

TMap® ofrece un enfoque global y coherente para proyectos y programas de **gestión de pruebas, ejecución y aseguramiento de la calidad**, disponible para organizaciones de todo tipo y tamaño. Es un método adaptable a todas las situaciones de testing y a la mayoría de entornos de desarrollo, como son: nuevos desarrollos, mantenimientos, desarrollos ágiles o paquetes de software customizados (SOGETI, 2017).

El enfoque estructurado de pruebas de TMap puede resumirse en cuatro componentes esenciales:

- Basado en un enfoque de gestión de pruebas conducidas por la empresa.
- Utiliza un proceso de pruebas estructurado.
- Utiliza una caja de herramientas completa.
- Es un método de prueba adaptativo.

5.1.6.1 Gestión de pruebas dirigida al negocio (Business Driven Test Management BDTM).

De acuerdo con (SOGETI, 2009), BDTM describe cómo organizar, administrar y ejecutar un proceso de prueba. BDTM traduce objetivos empresariales de una organización en los objetivos de la prueba, permitiendo a un cliente un control más efectivo sobre el proceso de prueba y por lo tanto los resultados de las pruebas. Basado en consideraciones racionales y económicas del negocio y en riesgos identificados, los componentes correctos son probados. A lo largo de todo, hay un fuerte enfoque en la comunicación clara y eficaz con el cliente.

BDTM tiene seis pasos, los cuales forman un bucle iterativo y se relacionan a continuación:

1. Formular objetivos de asignación y prueba.
2. Determinar la clase de riesgo para cada combinación de características y parte del objeto.
3. Determinar si una combinación de característica y parte del objeto debe ser probado minuciosamente o ligeramente.
4. Estimar y planificar la ejecución de las pruebas.
5. Asignar las técnicas de diseño de prueba más apropiadas.
6. Durante todo el proceso de ejecución de la prueba, proporcionar al cliente y a otras partes interesadas con un conocimiento adecuado de la prueba, procesos y objetos de prueba.

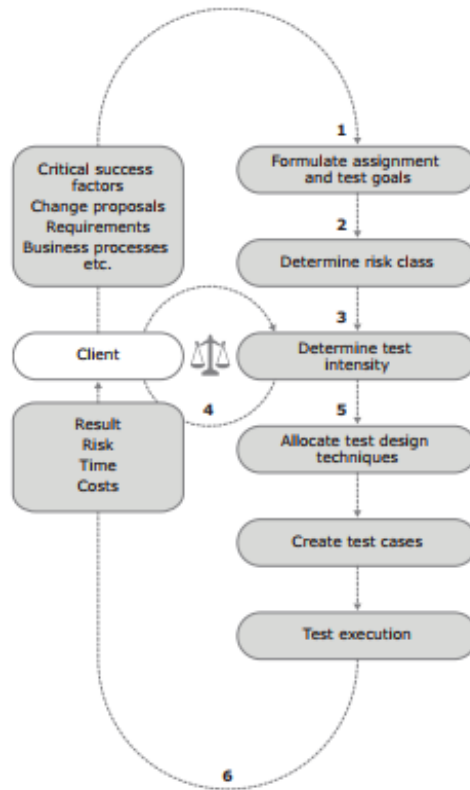


Ilustración 24: Business Driven Test Management

Fuente: (SOGETI, 2009, pág. 2)

5.1.6.2 *Proceso de pruebas estructurado.*

TMap proporciona una descripción completa de todos los aspectos del ciclo de vida del proceso de prueba, la cual comprende siete etapas desde la planificación hasta la finalización.

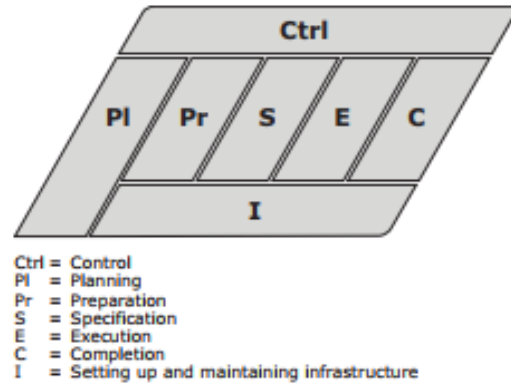


Ilustración 25: Tmap Test Process

Fuente: (SOGETI, 2009, pág. 3)

La fase de Planificación tiene un enfoque comprensivo y coherente para asegurar que el test asignado es ejecutado adecuadamente, como se detalla en el plan de prueba.

La fase de Preparación determina si las especificaciones de software escritas o no escritas son de suficiente calidad para lograr una Especificación y Ejecución exitosas de las pruebas.

La fase Control permite que las actividades del plan de prueba sean monitoreadas y ajustadas según sea necesario.

La asignación de prueba concluye en la fase de finalización, que ofrece la oportunidad de aprender lecciones de la experiencia adquirida en el proyecto. Además, las actividades son ejecutadas de tal manera que garantiza la reutilización de los productos de prueba (SOGETI, 2009, pág. 3).

5.1.6.3 Caja de herramientas completa.

TMap admite la ejecución correcta de un proceso de prueba estructurado con una caja de herramientas completa, que proporciona un extenso conjunto de listas de verificación y directrices para la configuración y ejecución de las actividades de prueba:

- Técnicas – Cómo probar
- Infraestructura - Dónde probar y con qué herramientas

- **Organización – Quién llevará a cabo las pruebas**

TMap también incluye más de 400 ejemplos, consejos y sugerencias para pruebas exitosas, basadas en la experiencia de Sogeti. Tiene en cuenta los nuevos desarrollos como las pruebas en organizaciones de prueba permanente (con o sin outsourcing), ambientes de prueba y herramientas de prueba (SOGETI, 2009, pág. 3).

5.1.6.4 Método de prueba adaptativo.

TMap puede aplicarse en todas las situaciones de prueba y en combinación con cualquier metodología de desarrollo. TMap está diseñado para abordar las diferencias en las aplicaciones y ofrece la flexibilidad para crear un enfoque de prueba a medida para satisfacer las necesidades particulares de una prueba específica. En síntesis, se puede decir lo siguiente acerca de la adaptabilidad de TMap:

- Responde a una multitud de cambios.
- Promueve el reuso de productos y procesos.
- Fomenta el aprendizaje de la experiencia del usuario y lo mantiene dentro de la organización.
- Permite 'probar antes de usar' (SOGETI, 2009, pág. 3).

5.1.7 Resumen de antecedentes encontrados.

Tabla 4: Resumen de antecedentes encontrados

AUTOR	TITULO	CLASIFICACIÓN/ ORGANIZACIÓN	FALENCIAS DE LA INVESTIGACIÓN	APORTES SIGNIFICATIVOS
(Capote García, Brito Riverol, Yzquierdo Herrera, & Febles Estrada, 2014)	Estrategia para desarrollar la perspectiva “Procesos internos en un laboratorio de pruebas de software”.	Artículo de Investigación. Universidad de las Ciencias Informáticas La Habana, Cuba	No se detalla la metodología empleada para llevar a cabo la investigación. No dan a conocer las técnicas de prueba empleadas.	<ul style="list-style-type: none"> ▪ División del proceso en fases, teniendo en cuenta la participación de la alta dirección y la formación requerida por el personal.
(Sanz Esteban, Marco Metodológico para la mejora de las actividades de Verificación y Validación de Productos Software (Tesis Doctoral), 2012)	Marco Metodológico para la Mejora de las Actividades de Verificación y Validación de Productos Software.	Tesis Doctoral. Universidad Carlos III Madrid, España	No se especifican las técnicas utilizadas para llevar a cabo la elaboración de los casos de prueba.	<ul style="list-style-type: none"> ▪ Implementación de un módulo enfocado a las personas, el cual define el modelo de competencias necesario para identificar los roles que mejor se ajustan a cada actividad, así como las habilidades y conocimientos requeridos por los integrantes del equipo de pruebas. ▪ Implementación de método empírico para validar la solución propuesta.

(Pérez Lamancha, 2006)	Proceso de Testing Funcional Independiente.	Tesis de Maestría. Instituto de Computación, Universidad de la República Montevideo, Uruguay.	El proceso genera demasiados artefactos como producto de las actividades ejecutadas, situación que lo puede hacer pesado y difícil de mantener.	<ul style="list-style-type: none"> ▪ Implementación de una etapa para llevar a cabo el ciclo de prueba
(Escobar Sánchez & Fuertes Díaz, 2015)	Modelo formal de pruebas funcionales de software para alcanzar el Nivel de Madurez Integrado 2.	Artículo de Investigación. Universidad de las Fuerzas Armadas ESPE Latacunga, Ecuador	<p>En el artículo no se detallan las actividades que se llevan a cabo en las fases que hacen parte del modelo propuesto.</p> <p>Es un modelo cuya aplicación va enfocada a alcanzar el nivel 2 de TMMI.</p>	<ul style="list-style-type: none"> ▪ Tipos de pruebas utilizadas en el modelo. ▪ Implementación de encuesta de satisfacción del cliente.
(Rojas Montes, Pino Correa, & Martínez, 2015)	Proceso de pruebas para pequeñas organizaciones desarrolladoras de software.	Artículo de Investigación. Universidad del Cauca Popayán, Colombia	En el artículo no se detalla la metodología empleada para llevar a cabo la investigación.	<ul style="list-style-type: none"> ▪ Técnicas de prueba consideradas en el proceso. ▪ Implementación de una fase para llevar a cabo el monitoreo y control de las pruebas. ▪ Criterios definidos para el análisis de las propuestas relacionadas con pruebas de software. ▪ Criterios a tener en cuenta para realizar el diseño de las pruebas.

<p>(SOGETI, 2017)</p>	<p>TMAP (Test Management Approach)</p>	<p>Metodología de Pruebas de SOGETI.</p>	<ul style="list-style-type: none"> ▪ Está diseñado para organizaciones grandes, y no es fácilmente adaptable a estructuras más pequeñas, como las PyMEs. ▪ Es menos difundido que otros modelos. ▪ No existe mucha documentación. ▪ No es un estándar internacional. (Sánchez Melchor, 2010, pág. 76) 	<ul style="list-style-type: none"> ▪ Se encuentra fundamentado en la Gestión del Proceso de Pruebas orientado hacia el Negocio de la Organización. ▪ Está basado en el análisis de riesgos, resultados, tiempo y coste para la Organización. ▪ Posee un conjunto de herramientas completo para el proceso de Pruebas: análisis de riesgos, estrategia de pruebas y priorización de éstas con una clara orientación al Negocio, técnicas de especificación de pruebas, etc. que facilitan que las pruebas sean reutilizables en el futuro. (Sánchez Melchor, 2010, pág. 76)
-----------------------	--	--	---	---

5.2 DEFINICIONES

A continuación, se relacionan algunos conceptos que son claves a la hora de enmarcar la investigación dentro del contexto de Pruebas de Software:

5.2.1 Prueba.

Según (Institute of Electrical and Electronics Engineers IEEE, 2002), “una prueba es una actividad en la que un sistema o un componente es ejecutado bajo condiciones especificadas, los resultados son observados o registrados, y una evaluación es realizada de un aspecto del sistema o componente” (IEEE Standard Glossary Of Software Engineering Terminology - Std 610.12-1990(R2002), pág. 78).

5.2.2 Caso de prueba.

De acuerdo con lo expuesto por (Institute of Electrical and Electronics Engineers IEEE, 2002), un caso de prueba se define como “un conjunto de entradas de prueba, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular, tales como el ejercicio de una determinada ruta del programa o para verificar el cumplimiento de un requisito específico” (Institute of Electrical and Electronics Engineers IEEE, 2002, pág. 78).

5.2.3 Error.

Según el (Institute of Electrical and Electronics Engineers IEEE, 2002), un error es “la diferencia entre un valor o condición calculado, observado o medido y el valor verdadero, especificado o teóricamente correcto” (IEEE Standard Glossary Of Software Engineering Terminology - Std 610.12-1990(R2002), 2002, pág. 29).

5.2.4 Defecto.

De acuerdo con (Institute of Electrical and Electronics Engineers IEEE, 2002), un defecto se define como una “imperfección en un componente o sistema que puede causar que el componente o sistema falle en desempeñar las funciones requeridas, por ejemplo una sentencia o una definición de datos incorrectas” (IEEE Standard Glossary Of Software Engineering Terminology - Std 610.12-1990(R2002), 2002, pág. 30).

5.2.5 Fallo.

Según el (Institute of Electrical and Electronics Engineers IEEE, 2002), un fallo se define como “la incapacidad de un sistema o componente para realizar sus funciones requeridas dentro de los requisitos de funcionamiento especificados” (IEEE Standard Glossary Of Software Engineering Terminology - Std 610.12-1990(R2002), 2002, pág. 30).

5.2.6 Validación.

El (Institute of Electrical and Electronics Engineers IEEE, 2002), define la validación como “el proceso de evaluación de un sistema o componente durante o al final del proceso de desarrollo para determinar si cumple los requisitos especificados” (IEEE Standard Glossary Of Software Engineering Terminology - Std 610.12-1990(R2002), 2002, pág. 85).

5.2.7 Verificación.

El (Institute of Electrical and Electronics Engineers IEEE, 2002), define la verificación como “el proceso de evaluación de un sistema o componente para determinar si los productos de una determinada fase de desarrollo satisfacen las condiciones impuestas en el inicio de esa fase” (IEEE Standard Glossary Of Software Engineering Terminology - Std 610.12-1990(R2002), 2002, pág. 85)

5.2.8 Plan de pruebas.

De acuerdo con (Black & Rueda Sandoval, 2011):

El plan de pruebas es un documento que describe el alcance, el método, los recursos y el cronograma de las actividades de pruebas previstas. Identifica entre otros ítems de pruebas, las características que tiene que ser probadas, las tareas de pruebas, quien hará cada tarea, el grado de independencia del probador, el entorno de pruebas, las técnicas de diseño de pruebas y los criterios de entrada y salida que serán utilizados, y las razones para su selección, y algunos riesgos que requieren planes de contingencia (Fundamentos de pruebas de software, 2011, pág. 39).

5.2.9 Niveles de prueba.

Según (IEEE Computer Society, 2014):

Las pruebas del software se realizan normalmente a diferentes niveles durante los procesos de desarrollo y mantenimiento. Lo cual significa que el objetivo de las pruebas puede cambiar: un módulo, un grupo de dichos módulos (relacionados por propósito, uso, comportamiento o estructura), o un sistema completo. Conceptualmente, se pueden distinguir tres grandes niveles de pruebas, llamadas de Unidad, de Integración y del Sistema. No hay un modelo de proceso implícito, ni se asume que alguno de estos tres niveles tiene mayor importancia que los otros dos (SWEBOK V3.0 Guide to the Software Engineering Body of Knowledge, 2014, págs. 4-5).

5.2.9.1 Pruebas de unidad.

Según lo expuesto en (SWEBOK V3.0 Guide to the Software Engineering Body of Knowledge, 2014):

Las pruebas de unidad verifican el funcionamiento aislado de partes del software que se pueden probar independientemente. Dependiendo del contexto, estas partes podrían ser subprogramas individuales o un componente más grande formado por unidades muy relacionadas. Normalmente las pruebas de unidad se realizan con acceso al código fuente y con el soporte de herramientas de depuración, pudiendo implicar a los programadores que escribieron el código (IEEE Computer Society, 2014, págs. 4-5).

5.2.9.2 Pruebas de integración.

En (SWEBOK V3.0 Guide to the Software Engineering Body of Knowledge, 2014), se definen las pruebas de integración como:

El proceso de verificar la interacción entre componentes de software. Estrategias clásicas de integración, como arriba-abajo o abajo-arriba, se usan, tradicionalmente, con software estructurado jerárquicamente. Las estrategias de integración sistemáticas son modernas normalmente arquitectura dirigida, que implica incrementalmente la integración de los componentes de software o subsistemas en base a las discusiones funcionales identificadas. Las pruebas de integración son una actividad continua en cada etapa del

desarrollo, que sucede en cada fase en que los ingenieros de software tienen que hacer abstracciones de las perspectivas de bajo nivel y concentrarse en las perspectivas del nivel que están integrando. (IEEE Computer Society, 2014, págs. 4-5).

5.2.9.3 Pruebas de sistema.

De acuerdo con (IEEE Computer Society, 2014):

Las pruebas del sistema se ocupan del comportamiento de un sistema completo. La mayoría de los fallos funcionales deberían haber sido identificados antes, durante las fases de pruebas de unidad y pruebas de integración. Las pruebas del sistema se consideran normalmente como las apropiadas para comparar el sistema con los requisitos no funcionales del sistema, como seguridad, velocidad, exactitud y confiabilidad. Las interconexiones externas con otras aplicaciones, utilidades, dispositivos hardware o con el sistema operativo, también se evalúan en este nivel (IEEE Computer Society, 2014, págs. 4-5).

5.2.10 Tipos de prueba.

En la actualidad, se puede encontrar una gran variedad de tipos de prueba, algunos de ellos se describen a continuación:

5.2.10.1 Pruebas de aceptación/calificación.

Las pruebas de aceptación comparan el comportamiento del sistema con los requisitos del cliente, sea cual sea la forma en que éstos se hayan expresado. El cliente realiza, o específica, tareas típicas para comprobar que se satisfacen sus requisitos o que la organización los ha identificado para el mercado al que se destina el software. Esta actividad de pruebas puede incluir o no a los desarrolladores del sistema (IEEE Computer Society, 2014, págs. 4-6).

5.2.10.2 Pruebas de instalación.

Normalmente, cuando las pruebas de aceptación han terminado, el software se puede comprobar una vez instalado en el entorno final. Las pruebas de instalación se pueden ver como pruebas del sistema realizadas en relación con los requisitos de la configuración de

hardware. Los procedimientos para la instalación también se podrían verificar (IEEE Computer Society, 2014, págs. 4-6).

5.2.10.3 Pruebas alfa y beta.

A veces, antes de poner el software en distribución, éste se proporciona a un grupo representativo de usuarios potenciales para que puedan usarlo en pruebas en las instalaciones del desarrollador (pruebas *alpha*) o externamente (pruebas *beta*). Dichos usuarios notifican problemas con el producto. Normalmente, el uso de versiones alfa y beta sucede en entornos no controlados y no siempre se le hace referencia en los planes de pruebas (IEEE Computer Society, 2014, págs. 4-6).

5.2.10.4 Pruebas de conformidad/Pruebas funcionales.

“Las pruebas de conformidad tienen el objetivo de verificar si el comportamiento del software corresponde con las especificaciones” (IEEE Computer Society, 2014, págs. 4-6).

5.2.10.5 Pruebas de regresión.

Según (Institute of Electrical and Electronics Engineers IEEE, 2002) las pruebas de regresión son “pruebas selectivas que se repiten en un componente para verificar que las modificaciones no han causado efectos no deseados y que el sistema o componente todavía cumple con los requisitos especificados (IEEE Standard Glossary Of Software Engineering Terminology - Std 610.12-1990(R2002), pág. 60).

5.2.10.6 Pruebas de rendimiento.

“Las pruebas de rendimiento verifican que el software cumple con los requisitos de comportamiento especificados y evalúa las características de rendimiento – Por ejemplo, la capacidad y el tiempo de respuesta” (IEEE Computer Society, 2014, págs. 4-6).

5.2.10.7 Pruebas de estrés.

“Las pruebas de estrés hacen funcionar el software a la máxima capacidad para la que fue diseñado, y por encima de ella, con el objetivo de determinar los límites de comportamiento

y probar mecanismos de defensa en sistemas críticos” (IEEE Computer Society, 2014, págs. 4-7).

5.2.10.8 Pruebas de recuperación.

El objetivo de las pruebas de recuperación es verificar la capacidad del software para reiniciarse después de un fallo del sistema o de otro "desastre" (IEEE Computer Society, 2014, págs. 4-7).

5.2.10.9 Pruebas de humo.

“Subconjunto de todos los casos de prueba definidos/planificados que cubren la funcionalidad principal de un componente o sistema, con el objeto de asegurar que las funciones cruciales de un programa funcionan. Pero sin preocuparse por los detalles finos” (Spanish Software Testing Qualifications Board (SSTQB), 2016, pág. 56).

5.2.11 Técnicas de prueba.

A continuación, se describen algunas de las Técnicas de Prueba existentes:

5.2.11.1 Pruebas de caja blanca.

La prueba de caja blanca, en ocasiones llamada *prueba de caja de vidrio*, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que: 1) garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez, 2) revisen todas las decisiones lógicas en sus lados verdadero y falso, 3) ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y 4) revisen estructuras de datos internas para garantizar su validez (Pressman, 2010, pág. 414).

Prueba de ruta básica. La prueba de la ruta o trayectoria básica es una técnica de prueba de caja blanca propuesta por primera vez por Tom McCabe. El método de ruta básica permite

al diseñador de casos de prueba derivar una medida de complejidad lógica de un diseño de procedimiento y usar esta medida como guía para definir un conjunto básico de rutas de ejecución. Los casos de prueba derivados para revisar el conjunto básico tienen garantía para ejecutar todo enunciado en el programa, al menos una vez durante la prueba (Pressman, 2010, págs. 414-415).

5.2.11.2 Pruebas de caja negra.

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software, es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entra que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello. Es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca.

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes: 1) Funciones incorrectas o faltantes, 2) errores de interfaz, 3) errores en las estructuras de datos o en el acceso a bases de datos externas, 4) errores de comportamiento o rendimiento y 5) errores de inicialización y terminación.

A diferencia de las pruebas de caja blanca, que se realizan tempranamente en el proceso de pruebas, la prueba de caja negra tiende a aplicarse durante las últimas etapas de la prueba. (Pressman, 2010, pág. 423).

Partición de Equivalencia. La partición de equivalencia es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. Un caso de prueba ideal descubre de primera mano una clase de errores (por ejemplo, procesamiento incorrecto de todos los datos carácter) que de otro modo podrían requerir la ejecución de muchos casos de prueba antes de observar el error general (Pressman, 2010, pág. 425).

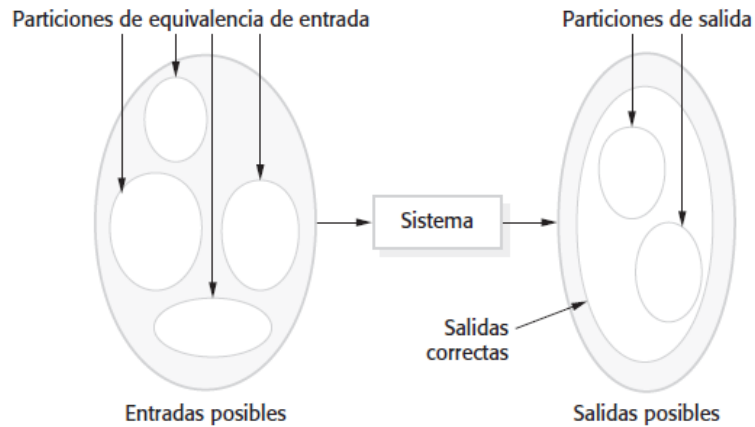


Ilustración 26: Partición de Equivalencia

Fuente: (Sommerville, 2005)

Análisis de Valores Límite. Casos de prueba se seleccionan en y cerca de los límites del dominio de las variables de la entrada de datos, basándose en la idea de que una gran parte de los errores se concentran cerca de los valores extremos de la entrada de datos. Una extensión de esta técnica son las *pruebas de robustez*, donde se seleccionan casos de prueba que se encuentran fuera del dominio de las variables de la entrada de datos, para comprobar la robustez del programa con entradas de datos erróneas e inesperadas. (IEEE Computer Society, 2014).

5.2.12 Pruebas durante el ciclo de vida del software.

De acuerdo con el (ISTQB International Software Testing Qualifications Board, 2011, pág. 23), “*el proceso de pruebas no es un proceso aislado; las actividades de pruebas están asociadas a actividades de desarrollo de software. Distintos modelos de ciclo de vida de desarrollo requieren distintos enfoques de pruebas*”. En este contexto, se puede decir que el modelo de ciclo de vida que se adopte para un proyecto, influye en gran medida en las pruebas que se llevan a cabo, ya que las actividades de pruebas están relacionadas con las actividades de desarrollo de software.

5.2.12.1 Modelo-V (*Modelo de desarrollo secuencial*)

Es un framework para describir las actividades del ciclo de vida del desarrollo de software desde la especificación de requisitos hasta el mantenimiento, dicho modelo ilustra la forma en que las actividades de pruebas pueden ser integradas en cada fase del ciclo de vida del desarrollo de software (Black & Rueda Sandoval, 2011, pág. 82).

En el modelo, las flechas horizontales de la V, denominadas “Desarrollar Pruebas”, indican la planificación, el análisis, el diseño y la implementación de las pruebas. Es así como la fase de especificación de requisitos es la base para desarrollar las pruebas de aceptación. A su vez, la especificación de requisitos y el diseño de alto nivel constituyen la base para desarrollar las pruebas de sistema. El diseño detallado y de alto nivel son la base para las pruebas de integración. De igual manera, el diseño detallado y el código en sí son la base para las pruebas unitarias.

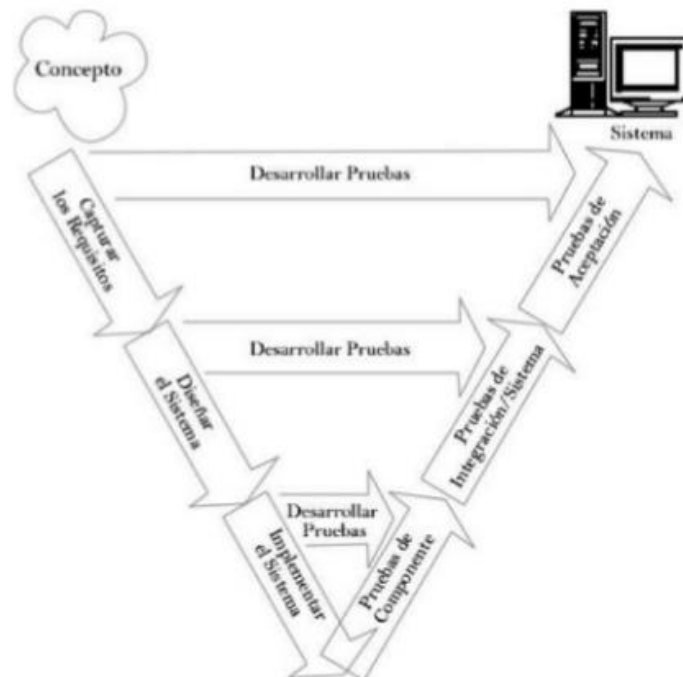


Ilustración 27: Modelo en V o Secuencial

Fuente: (Black & Rueda Sandoval, 2011, pág. 82)

5.3 ORGANIZACIONES RELACIONADAS CON LAS PRUEBAS DE SOFTWARE

Día a día surgen diferentes problemáticas a nivel internacional en torno al desarrollo de software y específicamente alrededor de la necesidad latente de garantizar su calidad. Es en este contexto donde muchas organizaciones se han unido a la búsqueda de nuevas alternativas que fomenten buenas prácticas de la ingeniería del software con el fin de garantizar la producción de soluciones que satisfagan los requerimientos de los clientes. Algunas de ellas son:

5.3.1 ISTQB: International software testing qualification board.

El Comité Internacional de Cualificación de Pruebas de Software (ISTQB) es una organización sin ánimo de lucro, creada en el año 2002 por empresas, instituciones, organizaciones y personas especializadas en el campo de las pruebas y la industria del software. El fin de la asociación es la profesionalización de las pruebas, mediante la definición de un esquema de certificación internacional de personas. Dicho comité suministra los planes de estudios y el glosario a partir de los cuales se definen los estándares internacionales por niveles y se establecen las guías para la acreditación de material y la evaluación de los profesionales de pruebas a cargo de los comités de cada país.

El esfuerzo actual de ISTQB está dirigido a homologar los términos y conceptos de pruebas, esto se realiza a través del trabajo de diversos comités (o “boards”) alrededor del mundo. Los “boards” de cada país o región se encargan de apoyar tanto el trabajo conceptual colaborando con el resto de los comités, como haciéndose cargo del proceso de certificación, además de realizar la traducción de los contenidos resultantes (SSTQB Spanish Software Testing Qualification Board, 2017).

5.3.2 ISO: International organization for standardization.

ISO es una organización internacional independiente, no gubernamental, con una membresía de 163 organismos nacionales de normalización. A través de sus miembros, reúne a expertos para compartir conocimientos y desarrollar estándares internacionales voluntarios, basados en el consenso y relevantes para el mercado, que apoyen la innovación y proporcionen soluciones a los retos globales (ISO, 2017).

5.4 PRINCIPALES ESTÁNDARES, MODELOS Y METODOLOGÍAS DE PRUEBAS

En la actualidad, existen diversos estándares, modelos y metodologías que definen buenas prácticas a implementar al interior de una organización con el fin de llevar a cabo un proceso de Pruebas de Software que contribuya con el desarrollo de productos de buena calidad que satisfagan las necesidades de las personas que los utilizan. En este contexto y con el fin de tener un marco de referencia para el proceso de pruebas que se desea definir con la presente investigación, se relacionan a continuación algunos de ellos con sus aspectos más representativos.

5.4.1 TMM – Test maturity model.

“El modelo TMM-Test Maturity Model (Burnstein 2003) ha sido desarrollado por el Instituto Tecnológico de Illinois como una guía para la mejora del proceso de pruebas” (Sanz Esteban, 2012, pág. 37).

Según el (Illinois Institute of Technology), TMM es un complemento al Modelo de Madurez de Capacidades CMM (Capability Maturity Model), basado en la evolución histórica del proceso de prueba y en las mejores prácticas de la industria. TMM tiene dos componentes principales:

1. Un conjunto de 5 niveles que definen la capacidad de prueba. Cada nivel se compone de:
 - Objetivos de madurez
 - Subobjetivos de madurez
 - Tareas y responsabilidades

2. Un modelo de evaluación. Esto consiste en:
 - Un cuestionario de madurez (tiene un componente basado en la web)
 - Un procedimiento de evaluación
 - Selección de Equipos y Guías de Entrenamiento

De acuerdo con Burnstein (2003), cada uno de los niveles tiene una serie de características. El nivel inicial se caracteriza porque el proceso de pruebas es un caos; no se encuentra definido y es considerado como una parte de la depuración. Las pruebas se desarrollan ad hoc después de haber terminado toda la codificación.

En el nivel 2, se separan las pruebas de la depuración, definiéndose una fase después de la codificación. El objetivo principal de este nivel es verificar que el software satisface los requisitos especificados. Sin embargo, surgirán problemas de calidad en este nivel, debido a que la planificación de las pruebas se realiza tarde. Además, los defectos se propagarán desde los requisitos y fases de diseño al código, ya que no existen programas de revisión.

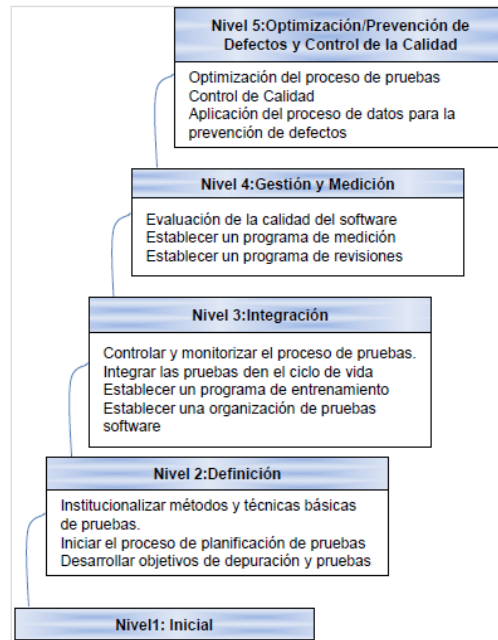


Ilustración 28: Objetivos de madurez de TMM

Fuente: (Sanz Esteban, 2012, pág. 38)

En el nivel 3, las pruebas se encuentran integradas en todo el ciclo de vida del software, no son una fase que se realiza después de la codificación. A diferencia del nivel 2, la planificación de las pruebas comenzará con la fase de requisitos y continuará durante todo el ciclo de vida. Existe una organización de pruebas, un programa de formación y se reconoce a las pruebas como una actividad profesional. Se realizarán revisiones aunque no exista un programa formal de revisión y éstas resulten inconsistentes.

El nivel 4 se caracteriza porque las pruebas son un proceso medido y cuantificado. Las revisiones de todas las fases del proceso de desarrollo son reconocidas como actividades de pruebas y control de calidad en este nivel. Se prueba el software atendiendo a los atributos de calidad, como fiabilidad, usabilidad y mantenimiento. Las deficiencias en este nivel se producen debido a la carencia en la prevención de defectos. Las actividades de prueba se expanden, incluyen revisiones, inspecciones y *walkthroughs* a través de las diferentes fases.

Finalmente, el nivel 5 se caracteriza porque el proceso de pruebas es repetible, definido, gestionado y medido, por lo que se pueden definir mecanismos que permitan una mejora continua en el proceso de pruebas. Se realiza prevención de defectos y control de calidad (Sanz Esteban, 2012, págs. 37-39).

5.4.2 Test Maturity Model integration (TMMi).

El TMMi es un modelo detallado para la mejora de los procesos de prueba (basado en algunos elementos de TMM) y se posiciona como un complemento a CMMI. Ha sido desarrollado por la Fundación TMMi como guía y marco de referencia para la mejora de procesos de prueba y se posiciona como un modelo complementario a la versión 1.2 CMMI al abordar esas cuestiones importantes para poner a prueba los gestores de pruebas, ingenieros de pruebas y los profesionales de la calidad del software (TMMi Foundation, 2015, pág. 6).

Al igual que la representación por etapas de CMMI, el TMMi también utiliza el concepto de niveles de madurez para la evaluación y mejora de procesos. Además se identifican las áreas de proceso, objetivos y prácticas. La aplicación de los criterios de madurez TMMi mejorará el proceso de prueba y tendrá un impacto positivo en la calidad del producto, la productividad de ingeniería de pruebas, y el esfuerzo en tiempo de los ciclos. El TMMi ha sido desarrollado para ayudar a las organizaciones a evaluar y mejorar su proceso de prueba. Con TMMi, las pruebas evolucionan de un proceso caótico, mal definido con la falta de recursos, herramientas y probadores bien formados a un proceso maduro y

controlado que tiene la prevención de defectos como su principal objetivo (TMMi Foundation, 2015, pág. 6).

“TMMi contiene una serie de etapas o niveles a través de los cuales una organización pasa a medida que su proceso de pruebas evoluciona desde uno ad hoc y no gestionado a uno que está gestionado, definido, medido y optimizado” (TMMi Foundation, 2015, pág. 9).

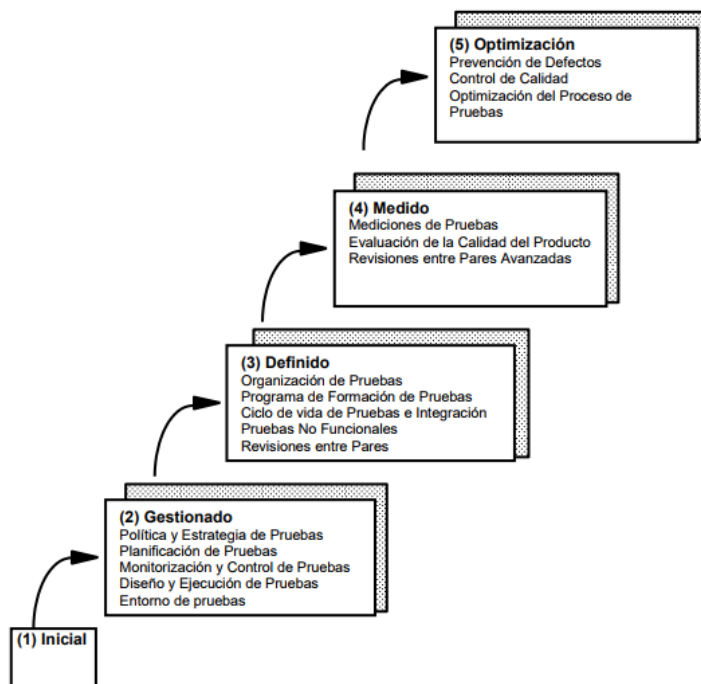


Figura 1: Niveles de madurez y áreas de proceso de TMMi

Ilustración 29: Niveles de madurez y áreas de proceso de TMMi

Fuente: (TMMi Foundation, 2015, pág. 9)

5.4.2.1 Nivel 1 Inicial.

Éste nivel se caracteriza porque las pruebas se realizan de forma caótica, con un proceso indefinido y frecuentemente se consideran parte de la depuración. Las pruebas se desarrollan ad hoc después de la codificación. Las pruebas y la depuración se intercalan para detectar errores en el sistema. El objetivo de las pruebas en este nivel es mostrar que el software funciona sin fallos mayores, sin embargo, los productos son entregados sin una adecuada visibilidad de su calidad y riesgos (TMMi Foundation, 2015, pág. 10).

5.4.2.2 Nivel 2 Gestionado.

En este nivel las pruebas se convierten en un proceso gestionado y está claramente separado de la depuración. El principal objetivo de las pruebas en una organización de nivel 2 es verificar que el producto satisface los requisitos definidos. Muchos de los problemas de calidad que pasan en este nivel de TMMi ocurren porque las pruebas se realizan muy tarde en el ciclo de vida de desarrollo, los defectos se propagan desde los requisitos y el diseño hasta el código. El nivel 2 de TMMi cuenta con las siguientes áreas de proceso: Política y Estrategia de Pruebas, Planificación de Pruebas, Monitorización y Control de Pruebas, Diseño y Ejecución de Pruebas y Entorno de Pruebas (TMMi Foundation, 2015, págs. 10-11).

5.4.2.3 Nivel 3 Definido.

En el nivel 3 de TMMI las pruebas están completamente integradas en el ciclo de vida de desarrollo y los hitos correspondientes. La planificación de pruebas se realiza en fases tempranas del proyecto por ejemplo, durante la fase de requisitos y se documenta en el plan de pruebas maestro. Las organizaciones en el nivel 3 comprenden la importancia de las revisiones en el control de la calidad.

Las áreas de proceso que hacen parte de este nivel son: Organización de Pruebas, Programa de Formación en Pruebas, Ciclo de vida de Pruebas e Integración, Pruebas No Funcionales y Revisiones entre Pares (TMMi Foundation, 2015, pág. 11).

5.4.2.4 Nivel 4 Medido.

En este nivel las organizaciones cuentan con un proceso de pruebas completamente definido, fundamentado y medible. Las pruebas se perciben como una evaluación, razón por la cual las organizaciones definen un programa de métricas de pruebas para evaluar la calidad del proceso de pruebas, medir la productividad y monitorizar las mejoras. Las áreas de proceso que hacen parte de este nivel son: Mediciones de Pruebas, Evaluación de la Calidad de Producto y Revisiones entre Pares Avanzadas (TMMi Foundation, 2015, págs. 11-12).

5.4.2.5 Nivel 5 Optimización.

En el nivel de madurez 5 de TMMi, una organización es capaz de mejorar de forma continua sus procesos basándose en un conocimiento cuantitativo de procesos estadísticamente controlados. Los métodos y técnicas de pruebas se optimizan y hay un foco continuo en afinar y mejorar el proceso. Para alcanzar el objetivo de este nivel, se dispone de tres áreas de proceso: Prevención de Defectos, Control de Calidad y Optimización del proceso de pruebas (TMMi Foundation, 2015, pág. 13).

5.4.3 TPI – Test process improvement.

El modelo Test Process Improvement (TPI) se desarrolló basándose en el conocimiento y la experiencia de SOGETI. Este modelo asiste en la mejora de los procesos de pruebas dentro de la organización. Basándose en esto, el modelo permite definir pasos de mejora controlables y graduales.

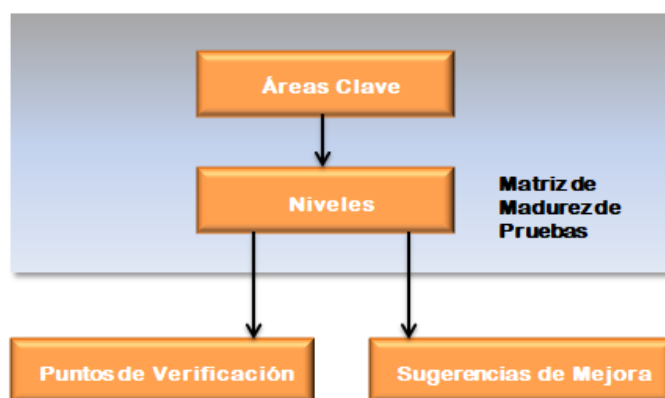


Ilustración 30: Niveles de madurez y áreas de proceso de TMMi

Fuente: (Sánchez Melchor, 2010, pág. 35)

TPI tiene en cuenta los diversos aspectos del proceso de pruebas, tales como el uso de herramientas de test, las técnicas de diseño de pruebas o la realización de informes. En su análisis, se hacen visibles los puntos fuertes y débiles del proceso de pruebas. Estos aspectos se denominan áreas clave (Sánchez Melchor, 2010, pág. 34).

El modelo TPI establece un total de 20 áreas clave, cada una de las cuales puede tener un determinado nivel de capacidad (ascendiendo desde la A hasta la D). El ascenso de nivel implica una mejora en tiempo, dinero y/o calidad. Los requisitos que permiten pasar de un

nivel a otro son definidos en forma de preguntas que deben ser respondidas positivamente para alcanzar cierto nivel, siendo necesario haber logrado el nivel anterior para pasar al siguiente; a dicho conjunto de preguntas se le denomina puntos de validación, ya que permiten determinar el nivel en el que se encuentra el proceso para cada área clave (Sanz Esteban, 2012, pág. 42).

Todos los niveles y áreas clave están interrelacionados en la Matriz de Madurez del proceso de pruebas. Esta proporciona una manera adecuada para definir las prioridades internas, teniendo en cuenta las dependencias entre las áreas clave y los niveles correspondientes. En la matriz, cada nivel está relacionado con una determinada escala de madurez del proceso de pruebas. En total, hay 13 niveles de madurez para el proceso de pruebas total. Las áreas clave, los niveles y la matriz se utilizan para identificar los puntos fuertes y débiles del proceso de pruebas actual, para definir el nivel de madurez que queremos alcanzar, y para establecer las acciones para llevar a cabo la mejora. El uso de la matriz permite seleccionar y evaluar propuestas de mejora con mayor facilidad (Sánchez Melchor, 2010, págs. 35-36).

Área Clave	Escala													
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Estrategia de pruebas		A					B				C		D	
Modelo del ciclo de vida		A			B									
Momento de implicación			A				B				C		D	
Estimación y planificación				A							B			
Técnicas de especificación de pruebas		A		B										
Técnicas de pruebas estáticas				A			B							
Métricas						A			B			C		D
Automatización de pruebas				A				B			C			
Entorno de pruebas				A				B						C
Entorno de trabajo				A										
Compromiso y motivación		A				B						C		
Funciones de prueba y entrenamiento				A			B			C				
Alcance de la metodología					A						B			C
Comunicación			A		B							C		
Informes		A			B		C						D	
Gestión de defectos		A				B		C						
Gestión de elementos de prueba			A			B				C				D
Gestión del proceso de pruebas		A		B								C		
Evaluación							A			B				
Pruebas de bajo nivel					A		B		C					

Ilustración 31: Matriz de madurez de TPI

Fuente: (Sanz Esteban, 2012, pág. 43)

Otro aspecto que es importante resaltar, es que TPI se basa en la metodología Tmap, situación que genera una limitante, ya que para su uso es necesario que la organización

implemente la metodología mencionada como mecanismo para la gestión del proceso de pruebas (Sanz Esteban, 2012, pág. 44).

5.4.4 ISO/IEC/IEEE 29119 Estándar Internacional de Pruebas de Software.

El propósito de la serie ISO/IEC/ IEEE 29119 de estándares de pruebas de software es definir un conjunto de estándares acordados internacionalmente para pruebas de software que puedan ser utilizados por cualquier organización al realizar cualquier tipo de pruebas de software (ISO, 2017).

La estructura de este estándar está conformada por cinco partes, las cuales se relacionan a continuación:

- ISO/IEC/IEEE 29119-1:2013: Conceptos y Definiciones.
- ISO/IEC/IEEE 29119-2:2013 - Procesos de prueba.
- ISO/IEC/IEEE 29119-3:2013 - Documentación de Pruebas.
- ISO/IEC/IEEE 29119-4:2015 – Técnicas de prueba.
- ISO/IEC/IEEE 29119-5:2016 – Pruebas dirigidas por palabras clave.

5.4.4.1 ISO/IEC/IEEE 29119-1:2013: Conceptos y Definiciones

ISO/IEC/IEEE 29119-1:2013 facilita el uso de las otras normas ISO/IEC/ IEEE 29119 mediante la introducción de los conceptos y el vocabulario sobre los cuales se construyen estos estándares, y proporciona ejemplos de su aplicación en la práctica. ISO / IEC / IEEE 29119-1: 2013 es informativo, proporcionando un punto de partida, contexto y guía para las otras partes (ISO, 2013) .

5.4.4.2 ISO/IEC/IEEE 29119-2:2013 - Procesos de prueba.

Éste estándar comprende descripciones de procesos de prueba que definen los procesos de prueba de software a nivel organizativo, nivel de gestión de prueba y niveles de prueba dinámicos. Es compatible con pruebas dinámicas, pruebas funcionales y no funcionales, pruebas manuales y automáticas, y pruebas con scripts y sin scripts. Los procesos definidos

en ISO/ IEC/IEEE 29119-2: 2013 se pueden usar en conjunción con cualquier modelo de ciclo de vida de desarrollo de software.

Dado que las pruebas son un enfoque clave para mitigar los riesgos en el desarrollo de software, ISO/IEC/IEEE 29119-2: 2013 sigue un enfoque basado en el riesgo para las pruebas. La evaluación basada en el riesgo es un enfoque común de la industria para elaborar estrategias y administrar pruebas. Las pruebas basadas en el riesgo permiten que las pruebas se prioricen y se centren en las características y funciones más importantes (ISO, 2013).

5.4.4.3 ISO/IEC/IEEE 29119-3:2013 - Documentación de Pruebas.

ISO/IEC/IEEE 29119-3: 2013 incluye plantillas y ejemplos de documentación de prueba. Las plantillas se organizan dentro de las cláusulas que reflejan la estructura general de descripción del proceso de prueba en ISO / IEC / IEEE 29119-2, es decir, mediante el proceso de prueba en el que se están produciendo. IEEE 29119-3: 2013 admite pruebas dinámicas, pruebas funcionales y no funcionales, pruebas manuales y automatizadas, y pruebas con scripts y sin scripts. Las plantillas de documentación definidas en ISO/IEC/IEEE 29119-3: 2013 se pueden usar junto con cualquier modelo de ciclo de vida de desarrollo de software (ISO, 2013).

El objetivo de ISO/IEC/IEEE 29119-3 es definir plantillas para la documentación de prueba que cubra todo el ciclo de vida de las pruebas de software. Cada plantilla puede adaptarse para satisfacer las necesidades únicas de cada organización que implementa el estándar, para respaldar la implementación del estándar dentro de cualquier modelo de ciclo de vida de desarrollo de software. Todas las plantillas se alinean con el proceso de prueba definido en ISO/IEC/IEEE 29119-2 y pueden producirse aplicando los procesos definidos en ese estándar.

El estándar de documentación de prueba IEEE 829, conocido y ampliamente utilizado, se utilizó como base para esta norma y, como tal, ISO/IEC/IEEE 29119-3 reemplaza al IEEE 829. Los documentos que se definen en ISO/IEC/IEEE 29119-3 son los siguientes:

Documentación del proceso de prueba de la organización:

- Política de prueba
- Estrategia de prueba organizacional

Documentación del proceso de gestión de prueba:

- Plan de prueba (incluida una estrategia de prueba)
- Informe de estado de prueba
- Informe de finalización de prueba

Documentación dinámica del proceso de prueba:

- Especificación del diseño de prueba
- Especificación del caso de prueba
- Especificación del procedimiento de prueba
- Requisitos de datos de prueba
- Probar el informe de preparación de datos
- Requisitos de entorno de prueba
- Probar el informe de preparación del entorno
- Resultados actuales
- Resultado de la prueba
- Registro de ejecución de prueba
- Informe de incidentes de prueba (Software Testing Standard, 2013).

5.4.4.4 ISO/IEC/IEEE 29119-4:2015 – Técnicas de prueba.

ISO/IEC/IEEE 29119-4: 2015 define las técnicas de diseño de prueba que pueden usarse durante el diseño de prueba y el proceso de implementación que se define en ISO/IEC/IEEE 29119-2. ISO/IEC/IEEE 29119-4: 2015 está destinado, entre otros, a probadores, administradores de pruebas y desarrolladores, en particular a los responsables de administrar e implementar pruebas de software (ISO, 2015).

5.4.4.5 ISO/IEC/IEEE 29119-5:2016 – Pruebas dirigidas por palabras clave.

El objetivo de ISO/IEC/IEEE 29119-5 es definir un estándar internacional para soportar pruebas conducidas por palabras clave. Las pruebas controladas por palabras clave son una forma de describir los casos de prueba mediante el uso de un conjunto predefinido de palabras clave. Estas palabras clave son nombres que están asociados con un conjunto de acciones que se requieren para realizar un paso específico en un caso de prueba. Al usar palabras clave para describir pasos de prueba en lugar de lenguaje natural, los casos de prueba pueden ser más fáciles de comprender, mantener y automatizar (Software Testing Standard, 2014).

5.5 SOFTWARE PROCESS ENGINEERING METAMODEL (SPEM)

SPEM fue creado por OMG (Object Management Group) como un estándar de alto nivel, que está basado en MOF (MetaObject Facility) y es un metamodelo UML (Uniform Model Language). Constituye un tipo de ontología de procesos de desarrollo de software (Wautelet, Kolp, & Achbany, 2006) . “Para ello provee un conjunto de elementos de modelado de procesos para describir cualquier proceso de desarrollo de software sin agregar modelos o restricciones de alguna área o disciplina específica, tal como gestión de proyectos o análisis” (Menéndez Domínguez & Castellanos Bolaños, 2015, pág. 92).

SPEM proporciona una sintaxis y estructura para cada aspecto de los procesos desarrollo, incluyendo:

- Roles.
- Tareas.
- Artefactos.
- Lista de verificación.
- Productos de trabajo.
- Técnicas y herramientas.
- Estructuras de trabajo.
- Capacidad de rastreo y refinamiento.
- Ayuda sensible al contexto, guía y lineamientos.
- Descripción textual de elementos (Menéndez Domínguez & Castellanos Bolaños, 2015).

Es importante resaltar que este estándar fue utilizado para modelar el proceso de testing funcional de software para las MIPYMES desarrolladoras de software de la ciudad de Pereira, como se verá en la sección de Resultados.

5.6 ESTADO ACTUAL DEL PROBLEMA

Según el (Observatorio de Tecnologías de Información MinTIC, 2015), la fabricación de software constituye la línea de negocio con mayor facturación para las empresas de Antioquia y el Eje Cafetero, prueba de ello es que el 36.2% de la facturación total de éstas regiones corresponde a dicho concepto, situación que sugiere un alto grado de especialización en la fabricación de software en dichas zonas del país.

LÍNEAS DE NEGOCIO	REGIONES			
	CENTRO ORIENTE	EJE CAFETERO ANTIOQUIA	PACÍFICO	CARIBE
Desarrollo/fábrica de software	21.8%	36.2%	16.6%	39.9%
Software como servicio (SaaS)	13.0%	12.7%	9.9%	1.9%
Consultoría e implementación	10.1%	5.3%	9.5%	3.0%
Venta o licenciamiento de software	7.8%	6.7%	16.0%	6.1%
Mantenimiento o soporte de aplicaciones	5.0%	2.8%	5.5%	6.1%
Plataformas tecnológicas como servicio (PaaS)	4.9%	1.5%	0.2%	5.0%
Servicios profesionales para TI	4.0%	5.9%	11.2%	1.5%
Venta de hardware	3.0%	1.2%	0.7%	0.9%
Integración de soluciones	2.8%	4.4%	2.5%	0.6%
Mesa de ayuda/ Soporte infraestructura	2.4%	2.0%	0.1%	0.7%
Seguridad informática	1.9%	3.2%	0.7%	0.1%
Cloud computing	1.9%	0.5%	4.0%	0.0%
Infraestructura como servicio (IaaS)	1.1%	0.3%	0.0%	0.7%
Servicios de conectividad	0.8%	1.1%	0.0%	0.0%
Servicios de cableado	0.7%	0.1%	0.5%	0.0%
Testing de software	0.5%	3.1%	1.0%	0.0%
Data Center	0.1%	0.1%	1.0%	0.0%

Ilustración 32: Facturación por línea de negocio – Regiones

Fuente: (Observatorio de Tecnologías de Información MinTIC, 2015, pág. 48)

A la hora de evaluar el contexto local y según el “*Estudio de las prácticas de calidad del software implementadas en las MIPYMES desarrolladoras de software de Pereira*”, elaborado por (Ramírez Aguirre & Ramírez Arias, 2010), en el cual evidencia que dentro de los tipos de producto o servicio producidos mayormente por las empresas de la ciudad de Pereira, se destacan el software empaquetado y el software a la medida. De igual manera se encuentra que el 37% sus ventas anuales oscilan entre los 50 y 100 millones de

pesos, mientras que tan solo el 11% vende más de 500 millones de pesos, abarcando en mayor medida la demanda del mercado local y regional. En lo que respecta a la forma en que se lleva a cabo la ejecución de las pruebas de software, se resalta que en mayor medida éstas son realizadas por el desarrollador, aunque algunas empresas cuentan con un Departamento de Pruebas. También es importante mencionar que, dentro de las áreas de mejora consideradas por las empresas para aplicar a un certificado de calidad, sobresale el área de Aseguramiento de la Calidad.

En este contexto y teniendo en cuenta que actualmente la región del eje cafetero y especialmente la ciudad de Pereira se está convirtiendo en el epicentro de diversas empresas de Desarrollo de Software reconocidas a nivel nacional e internacional, se hace necesario establecer procesos que contribuyan con la producción de software de calidad, que funcione impecablemente y que satisfaga las necesidades de las personas que lo utilizan.

Para tal fin y teniendo en cuenta que actualmente no se encuentra mucha información de estudios o investigaciones realizadas en la ciudad en torno a la forma en que se lleva a cabo el proceso de pruebas de software al interior de las organizaciones, se plantea como uno de los objetivos de la presente investigación, la caracterización del proceso utilizado para la ejecución de pruebas de software en las PYMES Desarrolladoras de Software de la Ciudad de Pereira, con el fin de que sirva como referente para proponer un proceso de pruebas funcionales de software cuya ejecución sea independiente del proceso de desarrollo, de tal manera que contribuya con el mejoramiento de los procesos de calidad del software producido e incentive el crecimiento de la línea de negocio correspondiente al Testing de Software en la región, la cual actualmente representa un 3.1% de la facturación total por línea de negocio.

6. OBJETIVOS

6.1 OBJETIVO GENERAL

Diseñar un proceso de pruebas funcionales de software para las PYMES de la ciudad de Pereira, independiente del proceso de desarrollo.

6.2 OBJETIVOS ESPECÍFICOS

- Caracterizar el uso de procesos para la ejecución de pruebas de software en las PYMES Desarrolladoras de Software de la Ciudad de Pereira.
- Diseñar un proceso mediante el cual se pueda llevar a cabo la ejecución de pruebas funcionales de software de manera independiente al proceso de desarrollo, teniendo en cuenta los resultados obtenidos en el diagnóstico realizado previamente.
- Realizar la aplicación del proceso diseñado mediante la ejecución de una prueba piloto, tomando como base dos proyectos de software elaborados por una o por dos empresas desarrolladoras de la ciudad de Pereira, con el fin de validar su efectividad.

7. ALCANCES Y LIMITACIONES

- Elaboración del referente teórico en torno al tema objeto de la presente investigación, el cual estará conformado por: Antecedentes, definiciones, organizaciones relacionadas con las pruebas de software y los principales estándares, modelos y metodologías de pruebas.
- Para llevar a cabo la construcción del informe de caracterización de las prácticas de pruebas de software en las MiPymes desarrolladoras de software de la ciudad de Pereira, se contactará la mayor cantidad de empresas posibles, sin embargo, el estudio se hará según la muestra por conveniencia de las empresas que manifiesten interés por participar.
- El proceso de pruebas planteado, estará basado en: i) la caracterización de las prácticas de pruebas empleadas por las MiPymes Desarrolladoras de Software de la ciudad de Pereira, ii) conceptos y buenas prácticas que hacen parte tanto del referente teórico como de los antecedentes de la investigación, iii) los atributos de calidad planteados por la norma ISO/IEC 9126 y iv) y los lineamientos definidos por el estándar ISO/IEC/IEEE 29119.
- Se implementará el proceso de pruebas propuesto en dos proyectos elaborados por PYMES desarrolladoras de software de la ciudad de Pereira.

8. METODOLOGÍA

8.1 ENFOQUE

Se realizará una investigación aplicada cuyo enfoque metodológico estará orientado hacia el desarrollo tecnológico de un proceso de software, el cual será validado mediante su aplicación en un caso de estudio, tomando como base la estrategia de mejora continua PHVA, la cual se puede enmarcar en la presente investigación de la siguiente manera:

Fase 1: Planificar

- **Plan de recolección y análisis de información**

Actividades:

- ✓ Elaboración de encuesta
- ✓ Envío de la encuesta a las PYMES desarrolladoras de software de la ciudad de Pereira que deseen contribuir con la investigación.
- ✓ Recolección y tabulación de la información obtenida.
- ✓ Elaboración de diagnóstico situacional de las PYMES desarrolladoras de software de la ciudad de Pereira a partir del análisis de la información recolectada en la encuesta.

- **Plan de elaboración del proceso de pruebas de software**

Actividades:

- ✓ Revisar las conclusiones obtenidas en la etapa de recolección y análisis de la información.
- ✓ Elaborar el proceso de pruebas de software teniendo en cuenta los antecedentes encontrados en el desarrollo de la investigación. Es importante tener en cuenta que el diseño del proceso se modelará bajo el estándar SPEM.

- **Plan de aplicación del proceso de pruebas**

Actividades:

- ✓ Selección de empresas en las cuales se llevará a cabo la aplicación del caso de estudio teniendo en cuenta los siguientes criterios: Empresas que se encuentren ubicadas en la ciudad de Pereira, prestas a suministrar la información requerida relacionada con el proyecto y cuya actividad económica esté catalogada en la División **62 DESARROLLO DE SISTEMAS INFORMÁTICOS (PLANIFICACIÓN, ANÁLISIS, DISEÑO, PROGRAMACIÓN, PRUEBAS), CONSULTORÍA INFORMÁTICA Y ACTIVIDADES RELACIONADAS** y específicamente *en el desarrollo de sistemas informáticos (planificación, análisis, diseño, programación, pruebas)* con código **CIIU 620 6201** de acuerdo con la clasificación Industrial Internacional Uniforme impartida por el Departamento Administrativo Nacional de Estadística (DANE).
 - ✓ Definir las variables que se van a medir antes y después de la aplicación del proceso de pruebas para establecer la línea base correspondiente. Para tal fin se tendrán en cuenta las siguientes categorías: Esfuerzo invertido, cantidad de defectos reportados, cantidad de productos no conformes.
 - ✓ Elaborar encuesta para conocer el estado actual y posterior de las empresas en las cuales se llevará a cabo la aplicación del caso de estudio.
 - ✓ Definir el cronograma y los tiempos para la aplicación del proceso.
 - ✓ Concertar las fechas de entrega por parte de la persona encargada de la aplicación del proceso y de los responsables del proyecto.
- **Plan de verificación de resultados al aplicar el proceso de pruebas propuesto**
 Actividades:
 - ✓ Revisar la documentación entregada con el fin de establecer que está diligenciada de acuerdo a lo establecido en el proceso de pruebas propuesto.
 - ✓ Analizar la información obtenida en las encuestas realizadas antes y después de la aplicación del proceso de pruebas propuesto.
 - ✓ Evaluar las mejoras que se podrían realizar al proceso de pruebas propuesto.

Fase 2: Hacer

- **Plan de capacitación para llevar a cabo la aplicación del proceso propuesto**
 Actividades:

- ✓ Capacitar al personal responsable de llevar a cabo la aplicación del proceso propuesto.
- ✓ Aplicación de la encuesta antes y después de llevar a cabo la aplicación del proceso de pruebas propuesto en las empresas que harán parte del caso de estudio.
- ✓ Socializar el cronograma de entregas a los responsables del proyecto.

- **Implementación del proceso de pruebas propuesto**

Actividades:

- ✓ Implementar el proceso de pruebas propuesto en dos proyectos elaborados por PYMES desarrolladoras de software de la ciudad de Pereira.

Fase 3: Verificar

- **Implementación del proceso de pruebas propuesto**

Actividades:

- ✓ Evaluación de los resultados obtenidos contra la línea base establecida al inicio de la aplicación del caso de estudio, la cual tiene en cuenta las siguientes categorías: Esfuerzo invertido, cantidad de defectos reportados y cantidad de productos no conformes.
- ✓ Realizar la evaluación del proceso aplicado con el fin de conocer las observaciones, ventajas y desventajas de su ejecución.

Fase 4: Actuar

- **Mejoramiento del proceso de pruebas propuesto**

Actividades:

- ✓ Establecer las oportunidades de mejora que se pueden tener en cuenta para ser desarrolladas en futuras investigaciones, con base en los resultados obtenidos en la fase de verificación.

8.2 TIPO DE ESTUDIO

La presente investigación pretende llevar a cabo un proyecto de mejora de procesos, cuyo tipo de estudio será exploratorio en primera instancia, ya que se pretende indagar sobre el uso de metodologías y técnicas de pruebas propuestas dentro de la ingeniería del software a nivel nacional e internacional, contemplando dentro de éste las organizaciones, los estándares, las técnicas y demás prácticas utilizadas en la producción de software de calidad. Lo anterior, teniendo en cuenta que éste es un tema sobre el cual hay diferentes posturas y por ende, la adopción de un procedimiento determinado o una metodología específica depende del contexto en el que se encuentre la organización. Por lo anterior, en la primera parte del estudio se pretende obtener una visión general acerca del concepto a través de un recorrido teórico y el estado de las prácticas mediante su contextualización en el ámbito nacional e internacional con el fin de servir como referente para la elaboración del proceso que se pretende plantear.

En segunda instancia, se llevará a cabo un tipo de estudio empírico, ya que será necesario utilizar un instrumento como la encuesta para recolectar la información requerida para realizar el diagnóstico situacional de las PYMES desarrolladoras de software de la ciudad de Pereira, con base en el cual se establecerá una línea base y se diseñará un proceso de pruebas, el cual será aplicado y validado mediante casos de estudio.

8.3 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE LA INFORMACIÓN

Las técnicas e instrumentos que se utilizarán para llevar a cabo la recolección de la información son los siguientes:

Encuesta: Se construirá una encuesta que será utilizada para llevar a cabo la primera fase de diagnóstico en las PYMES desarrolladoras de software de la ciudad de Pereira. Dicho documento será dividido en categorías, esto con el fin de que cada individuo encuestado se haga una idea de los aspectos que se van a preguntar sobre el proceso de pruebas de software que se lleva a cabo al interior de cada organización. Las categorías planteadas son las siguientes:

- Generalidades de la Organización.
- Generalidades del Proceso de Pruebas.
- Planeación de las Pruebas.
- Diseño de las Pruebas.
- Estimación de Costos.
- Métricas del Proceso de Pruebas.
- Herramientas utilizadas en el proceso de pruebas.
- Oportunidades de mejora del proceso de pruebas.

8.4 PROCEDIMIENTOS

Con el fin de cumplir con los objetivos propuestos en la presente investigación, se llevarán a cabo los siguientes procedimientos:

- a. Revisión bibliográfica.
- b. Recolección y análisis de la información: Se enviará a cada una de las PYMES desarrolladoras de software de la ciudad de Pereira, un correo en el cual se hará una breve exposición del objetivo, alcances del proyecto y se solicitará el diligenciamiento de la encuesta, cuyo contenido está dividido en 10 categorías y un máximo de 40 preguntas, las cuales se formularon teniendo en cuenta conceptos, procesos y tareas que enmarcan las pruebas de software mediante la revisión y el análisis bibliográfico. Posteriormente, se recolectará la información para llevar a cabo el análisis de los resultados obtenidos.
- c. Caracterización de las prácticas relacionadas con las pruebas de software en las PYMES Desarrolladoras de Software de la ciudad de Pereira, mediante los resultados obtenidos en la fase anterior. De igual manera, se identificarán las empresas que servirán de apoyo para la aplicación del proceso de pruebas propuesto y se elaborará el cronograma correspondiente.

- d. Formulación del proceso de pruebas de software, teniendo en cuenta los resultados obtenidos en el análisis de la encuesta aplicada a las PYMES desarrolladoras de software de la ciudad de Pereira.
- e. Aplicación del proceso propuesto. Se aplicará el proceso a dos proyectos elaborados por una o dos PYMES Desarrolladoras de Software de la Ciudad de Pereira.
- f. Validación del proceso propuesto. Se llevará a cabo un análisis comparativo de los resultados obtenidos en las PYMES Desarrolladoras de Software escogidas para la aplicación del proceso de pruebas planteado, esto con el fin analizar los resultados obtenidos, identificar las ventajas y desventajas del proceso propuesto y definir acciones de mejora que se puedan ejecutar en un futuro.

8.5 ANÁLISIS DE RESULTADOS

El análisis se realizará en dos fases. La primera se llevará a cabo una vez se recolecten los resultados correspondientes a la aplicación de la encuesta en las PYMES desarrolladoras de software de la ciudad de Pereira, esto con el fin de caracterizar las prácticas de pruebas utilizadas al interior de dichas organizaciones.

La segunda fase se realizará una vez se tengan los resultados obtenidos en las PYMES Desarrolladoras de Software escogidas para la aplicación del proceso de pruebas planteado. Con base en dicha información, se realizará un análisis comparativo con el fin de identificar las ventajas y desventajas del proceso planteado y definir acciones de mejora que se puedan ejecutar en un futuro.

9. RESULTADOS

9.1 CARACTERIZACIÓN DE LAS PRÁCTICAS DE PRUEBAS DE SOFTWARE EN LAS MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA

Con el objetivo de recolectar y analizar la información requerida para caracterizar la manera como las MiPymes desarrolladoras de software de la ciudad de Pereira están llevando a cabo el proceso de pruebas, se utilizó el método de la encuesta (**Ver Anexo B – Encuesta aplicada a Empresas de la ciudad de Pereira**). Dicho instrumento fue aplicado en 10 empresas y se planteó bajo los siguientes parámetros:

- Encabezado con el objetivo de la encuesta.
- Treinta y nueve (39) preguntas claras y concisas, distribuidas en las 8 categorías que se relacionan a continuación:
 - ✓ Generalidades de la Organización.
 - ✓ Generalidades del Proceso de Pruebas.
 - ✓ Planeación de las Pruebas.
 - ✓ Diseño de las Pruebas.
 - ✓ Estimación de Costos.
 - ✓ Métricas del Proceso de Pruebas.
 - ✓ Herramientas utilizadas en el proceso de pruebas.
 - ✓ Oportunidades de mejora del proceso de pruebas.

9.1.1 Tabulación y gráficos de la información.

La tabulación y los gráficos presentados a continuación están constituidos por las respuestas aportadas por las empresas encuestadas:

9.1.1.1 Respuestas a preguntas de la sección 1: Generalidades de la organización

Pregunta 1: ¿Cuántos años de creación tiene la empresa?

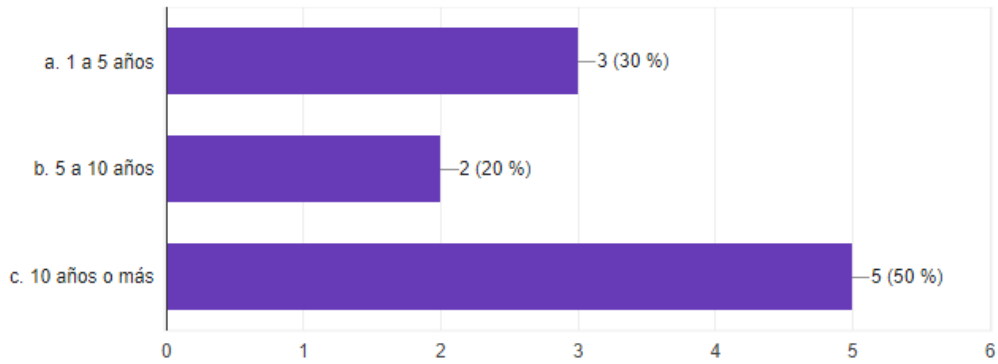


Gráfico 1: Años de creación de la empresa

Fuente: Elaboración propia

Pregunta 2: ¿Cuántos empleados tiene la empresa?

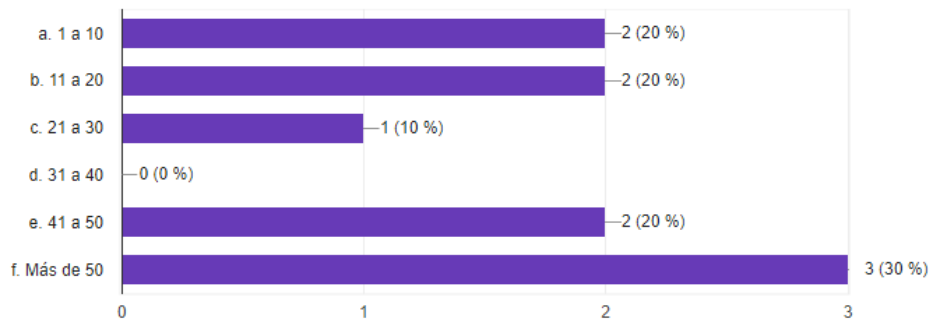


Gráfico 2: Número de empleados que tiene la empresa

Fuente: Elaboración propia

Pregunta 3: ¿Cuál es la principal actividad de la empresa?

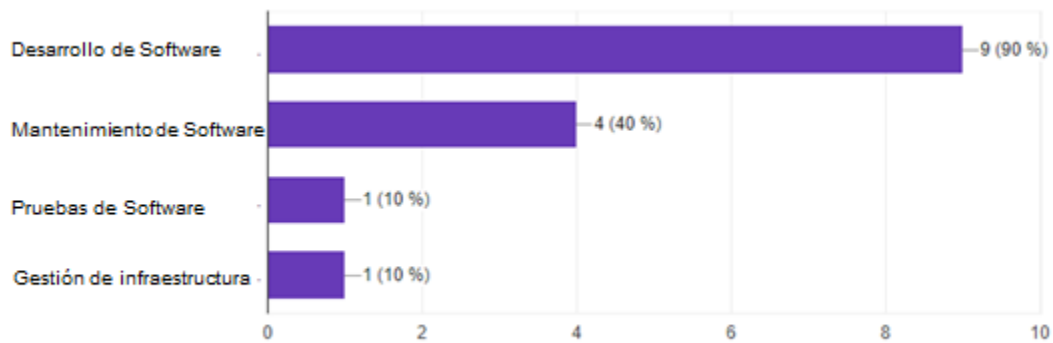


Gráfico 3: Principal actividad de la empresa

Fuente: Elaboración propia

Pregunta 4: ¿Qué Metodología, Modelo o Marco de Trabajo de Desarrollo de Software utiliza la Empresa?

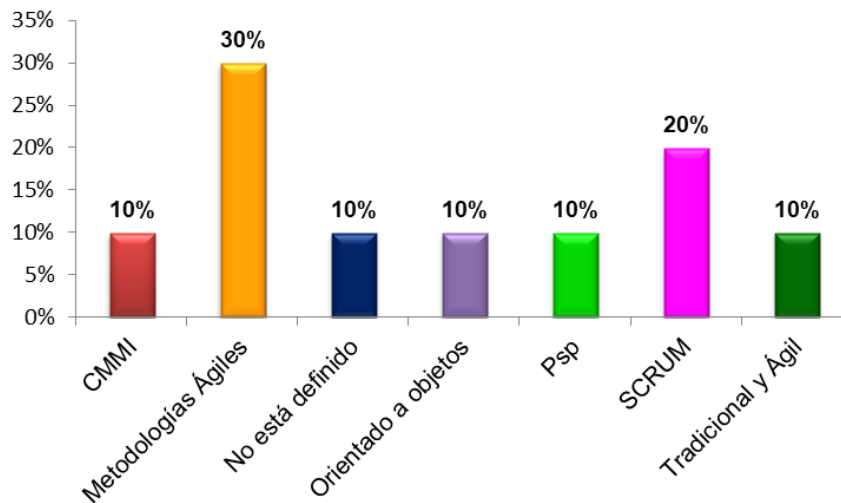


Gráfico 4: Metodología, modelo o marco de trabajo de desarrollo de software utilizado por las empresas

Fuente: Elaboración propia

Pregunta 5: ¿La Empresa tiene certificaciones de calidad?

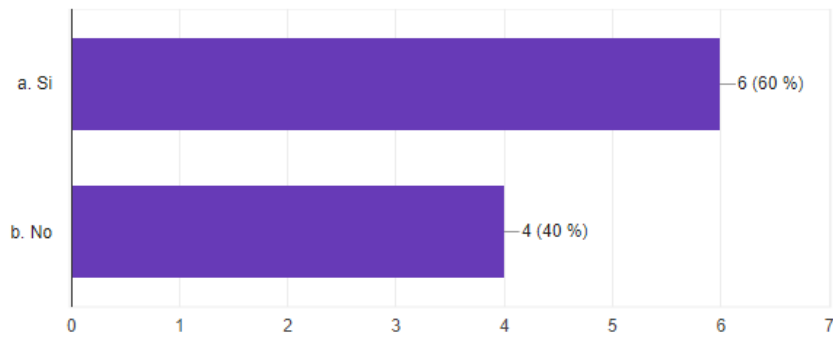


Gráfico 5: Certificaciones de calidad

Fuente: Elaboración propia

Pregunta 5.1: ¿Cuáles?

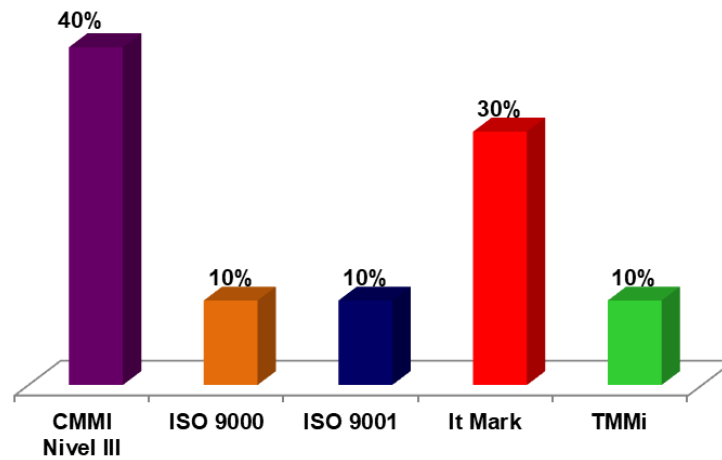


Gráfico 6: Certificaciones de calidad de las empresas encuestadas

Fuente: Elaboración propia

Pregunta 6: ¿La Empresa administra las configuraciones de software?

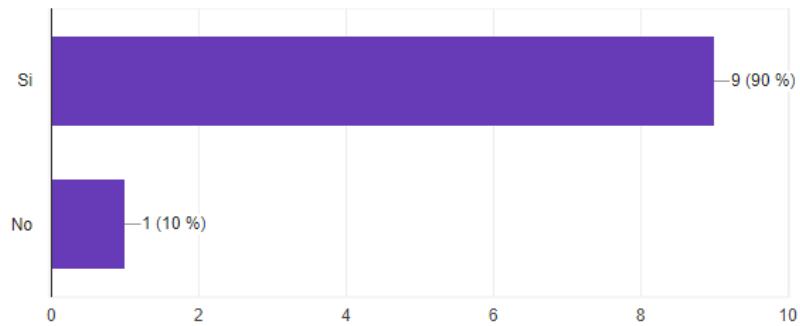


Gráfico 7: Respuesta a la pregunta sobre administración de configuraciones de software

Fuente: Elaboración propia

Pregunta 7: ¿La Empresa administra las versiones de software?

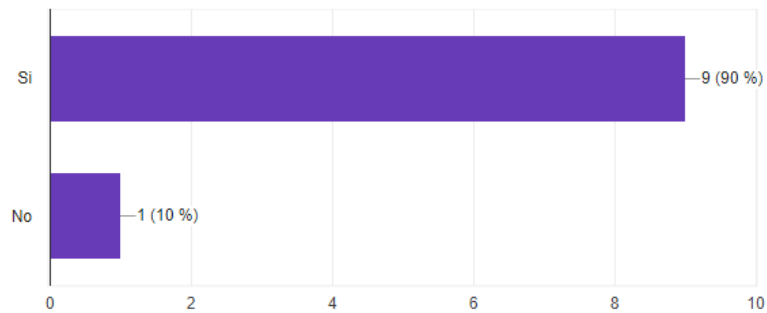


Gráfico 8: Respuesta a la pregunta sobre administración de versiones de software

Fuente: Elaboración propia

Pregunta 8: ¿Qué técnica utilizan para llevar a cabo la especificación del software?

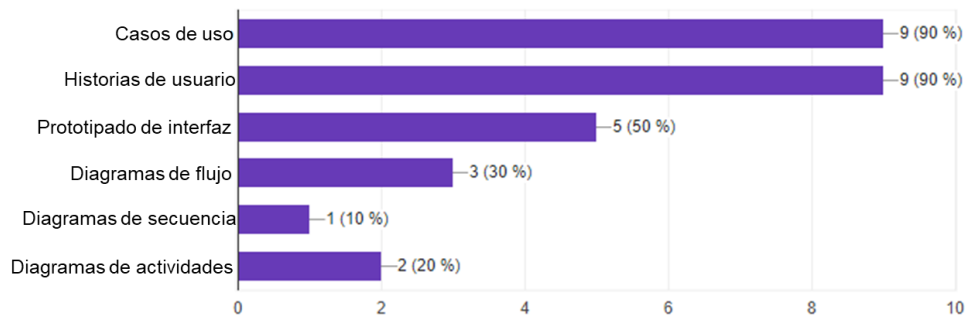


Gráfico 9: Técnicas que utilizan las empresas para llevar a cabo la especificación de software

Fuente: Elaboración propia

9.1.1.2 Respuestas a las preguntas de la sección 2: Generalidades del proceso de pruebas

Pregunta 9: ¿La empresa utiliza un proceso o metodología para el desarrollo de las pruebas de software?

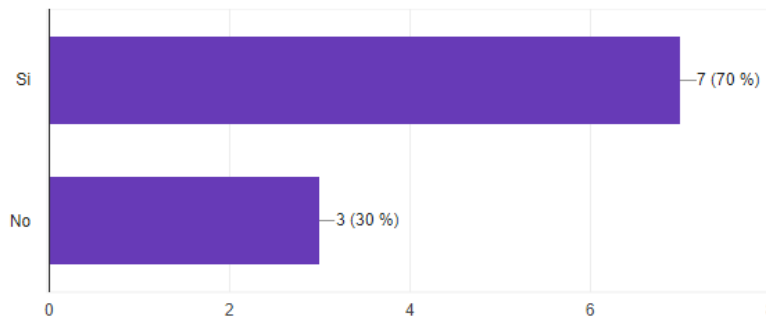


Gráfico 10: Utilización de proceso o metodología para el desarrollo de las pruebas de software

Fuente: Elaboración propia

Pregunta 9.1: ¿Cuáles?

- Basadas en TMMi
- Pruebas unitarias, funcionales, integrales y de estrés
- El proceso esta guiado por ISTQB y Agile Testing
- Automatización

Pregunta 10: ¿La empresa cuenta con un área especializada en la realización de pruebas de software?

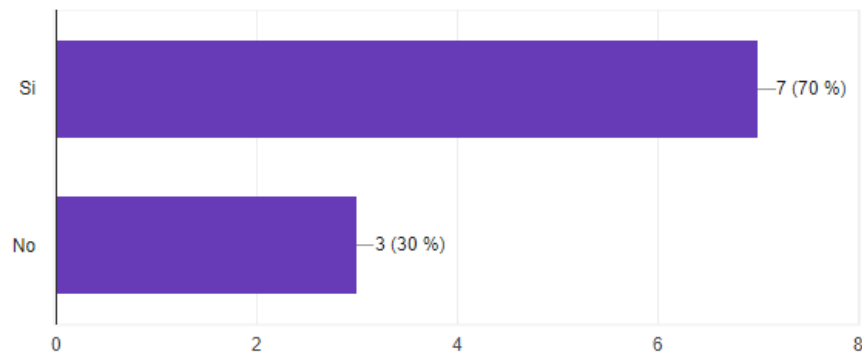


Gráfico 11: Respuesta a la pregunta sobre tenencia de un área especializada de pruebas de software

Fuente: Elaboración propia

Pregunta 11: ¿Cómo se llevan a cabo las pruebas de software en la organización?

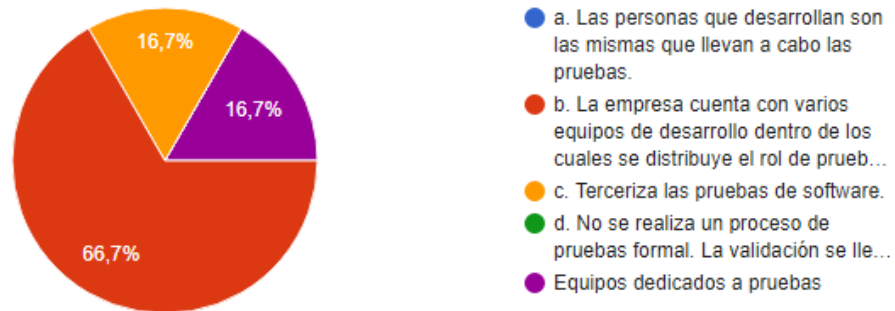


Gráfico 12: Forma en que se llevan a cabo las pruebas de software en las empresas encuestadas

Fuente: Elaboración propia

Pregunta 12: ¿Cuántas personas conforman el Área de Pruebas?

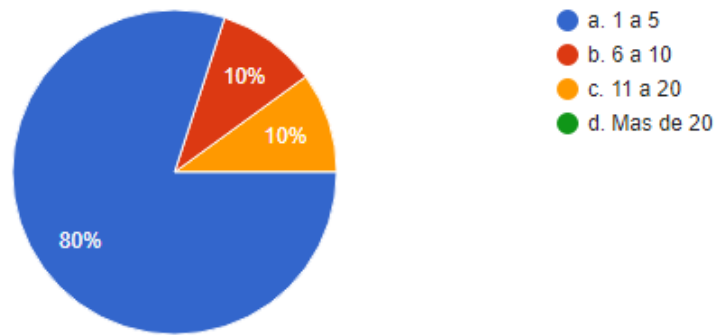


Gráfico 13: Número de personas que conforman el Área de Pruebas en las empresas encuestadas

Fuente: Elaboración propia

Pregunta 13: ¿La empresa cuenta con personal certificado en pruebas de software?

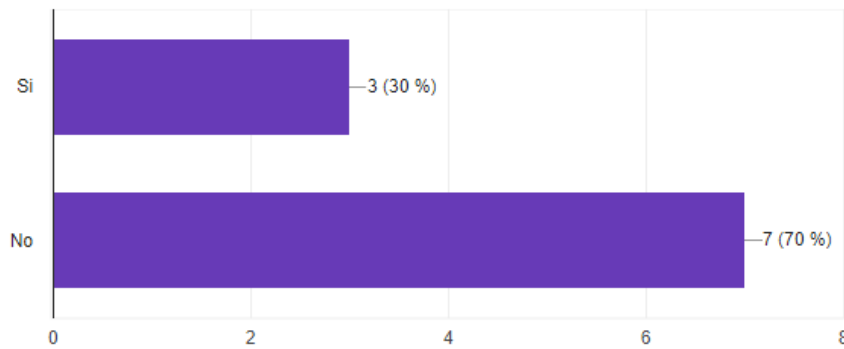


Gráfico 14: Respuesta a la pregunta sobre contar con personal certificado en pruebas de software

Fuente: Elaboración propia

Pregunta 13.1: ¿Cuáles?

- ISTQB – Agile Testing
- ISTQB – Foundation Level

Pregunta 14: ¿Qué características o atributos de calidad tienen en cuenta a la hora de realizar las pruebas de un artefacto?

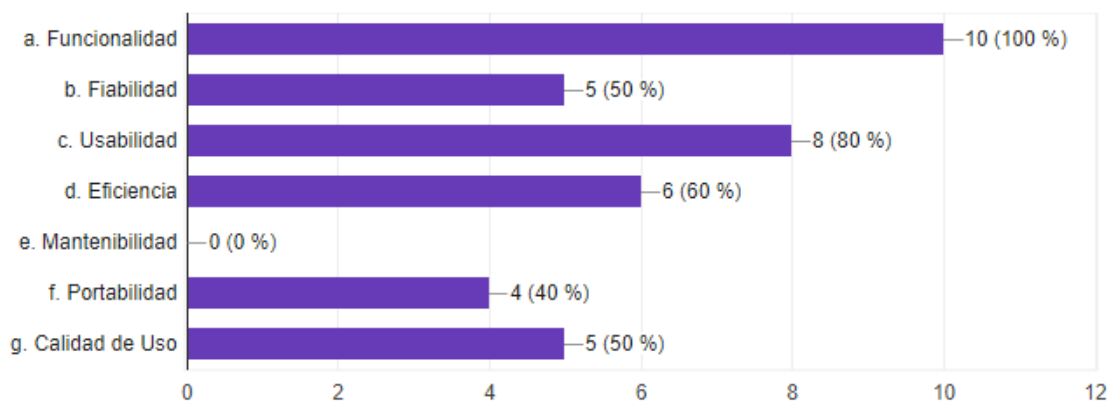


Gráfico 15: Características y atributos de calidad que se tienen en cuenta al realizar pruebas

Fuente: Elaboración propia

Pregunta 15: ¿Qué tipo de pruebas realiza la empresa?

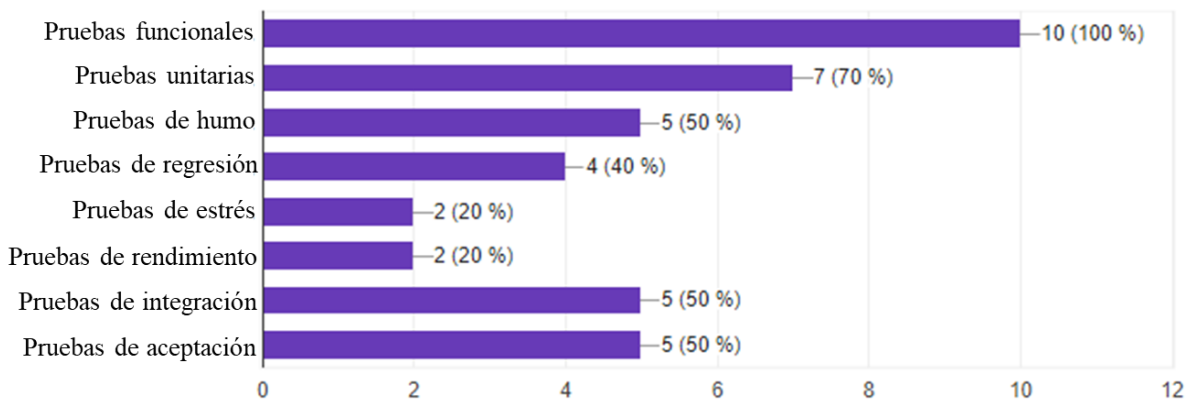


Gráfico 16: Tipos de pruebas que realizan las empresas encuestadas

Fuente: Elaboración propia

Pregunta 16: ¿La empresa cuenta con un ambiente de pruebas independiente del entorno de desarrollo?

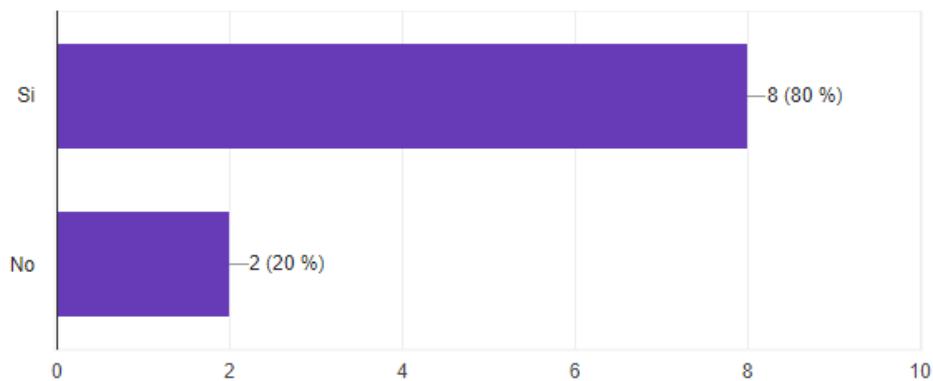


Gráfico 17: Relación del número de empresas que cuentan con ambiente independiente de pruebas

Fuente: Elaboración propia

9.1.1.3 Respuestas a las preguntas de la sección 3: Generalidades del proceso de pruebas

Pregunta 17: ¿La Empresa planifica las pruebas de software y asigna los recursos necesarios para su ejecución?

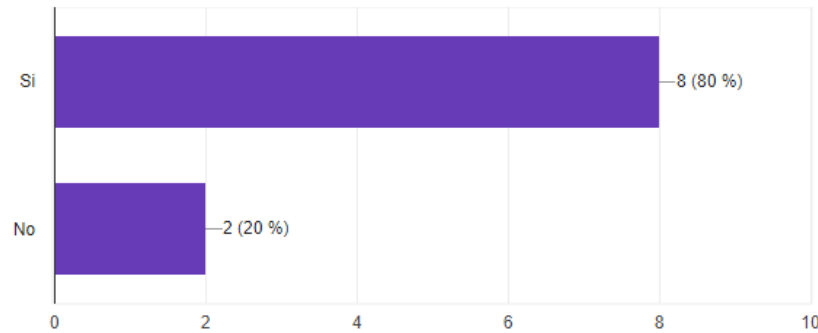


Gráfico 18: Planificación de pruebas de software

Fuente: Elaboración propia

Pregunta 18: ¿En el Plan de Pruebas se establece la duración del ciclo de pruebas?

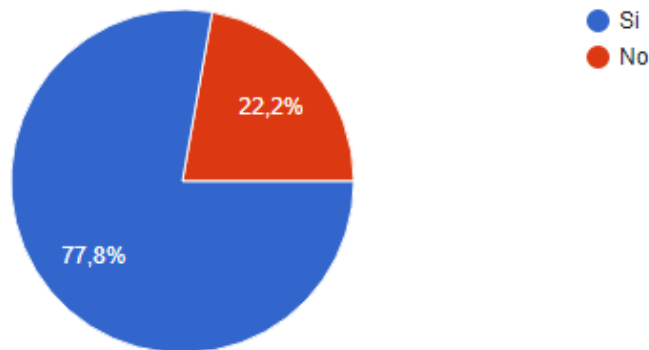


Gráfico 19: Establecimiento de la duración del ciclo de pruebas en el Plan de Pruebas

Fuente: Elaboración propia

Pregunta 19: ¿Las pruebas se planifican desde el inicio del proyecto?

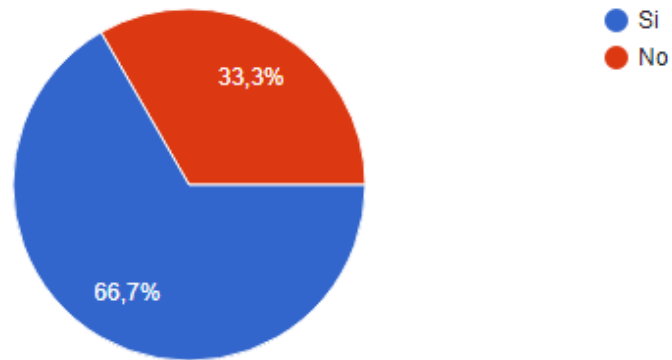


Gráfico 20: Respuesta a pregunta sobre planificación de pruebas desde el inicio del proyecto

Fuente: Elaboración propia

Pregunta 20: ¿En el Plan se establece qué atributos de calidad se considerarán en las pruebas?

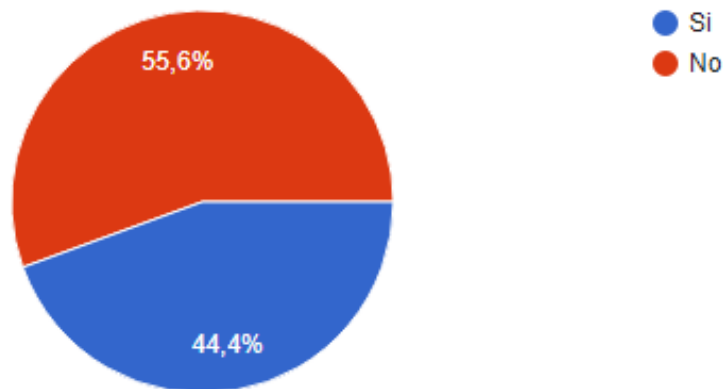


Gráfico 21: Respuesta a pregunta sobre establecimiento de atributos de calidad en las pruebas

Fuente: Elaboración propia

Pregunta 21: ¿En el Plan se establece qué tipos de pruebas se realizarán?

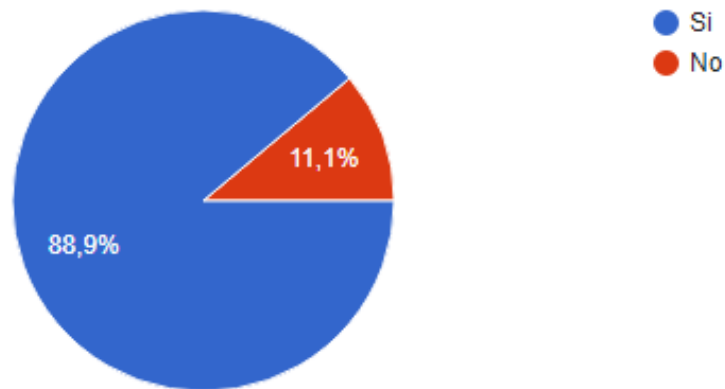


Gráfico 22: Respuesta a pregunta sobre establecimiento de tipos de pruebas en el Plan de Pruebas

Fuente: Elaboración propia

9.1.1.4 Respuestas a las preguntas de la sección 4: Diseño de las pruebas.

Pregunta 22: ¿Se emplean Técnicas de Diseño de Casos de Prueba?

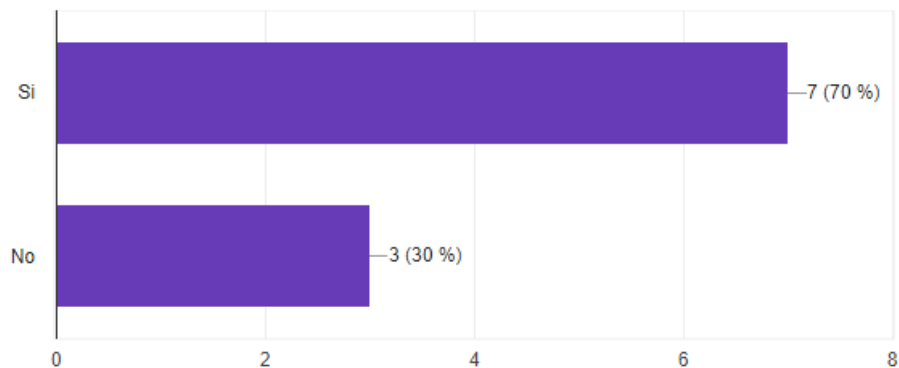


Gráfico 23: Empleo de Técnicas de Diseño de Casos de Prueba

Fuente: Elaboración propia

Pregunta 23: ¿Qué técnicas utiliza?

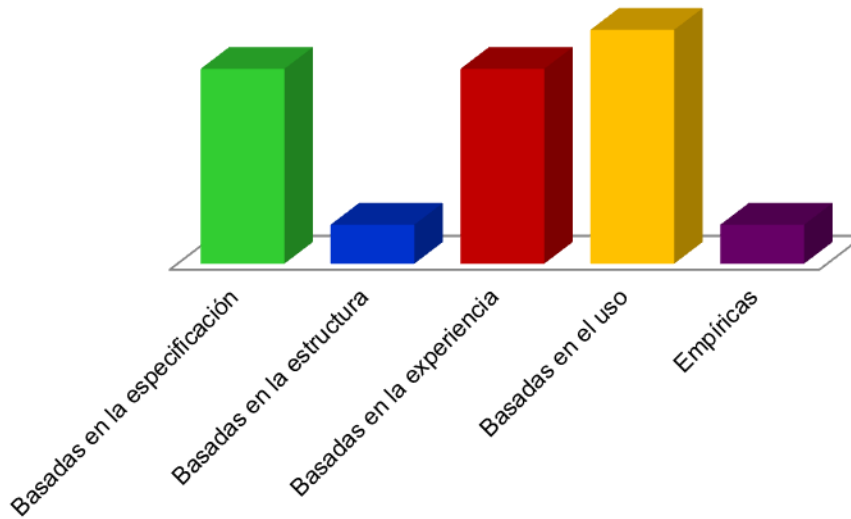


Gráfico 24: Técnicas de diseño de pruebas utilizadas por las empresas encuestadas

Fuente: Elaboración propia

Pregunta 24: ¿Socializa los casos de prueba con el cliente para que éste lleve a cabo su aprobación y/o haga observaciones al respecto?

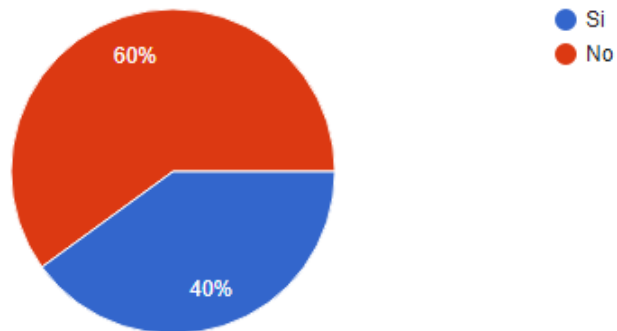


Gráfico 25: Socialización de casos de prueba con el cliente

Fuente: Elaboración propia

9.1.1.5 Respuestas a las preguntas de la sección 5: Estimación de costos.

Pregunta 25: ¿La empresa registra el esfuerzo invertido y los costos de las pruebas?

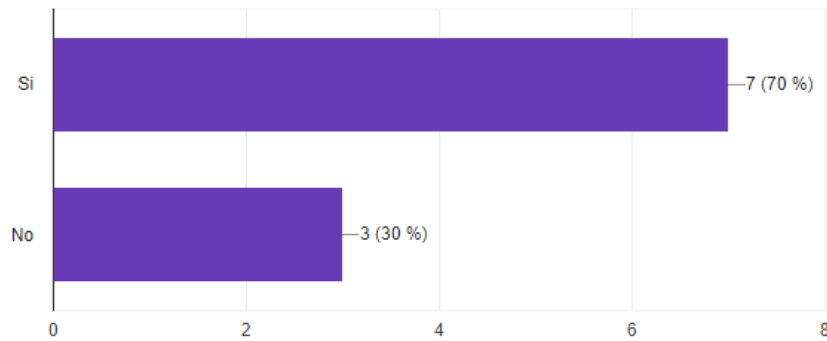


Gráfico 26 Respuesta sobre el registro de tiempo invertido y costos de las pruebas

Fuente: Elaboración propia

Pregunta 26: ¿La empresa presupuesta la fase de pruebas del software en un proyecto? Es decir, ¿incluye en el presupuesto un rubro para llevar a cabo el proceso de pruebas?

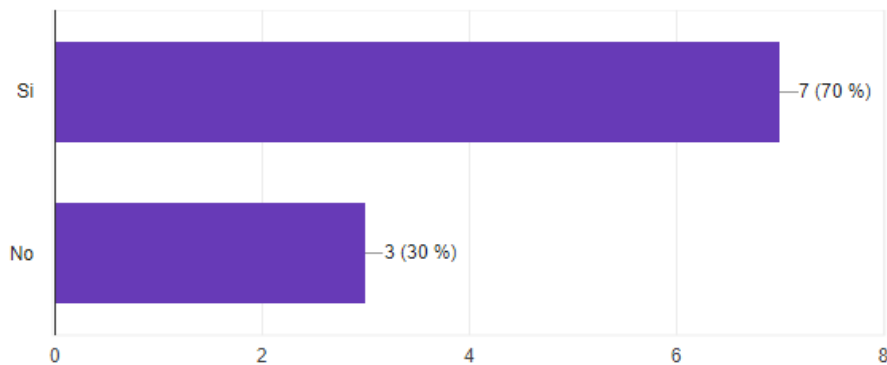


Gráfico 27: Respuesta pregunta sobre presupuestación de la fase de pruebas en un proyecto

Fuente: Elaboración propia

Pregunta 27: ¿Considera que el proceso de pruebas es demasiado costoso para la organización?

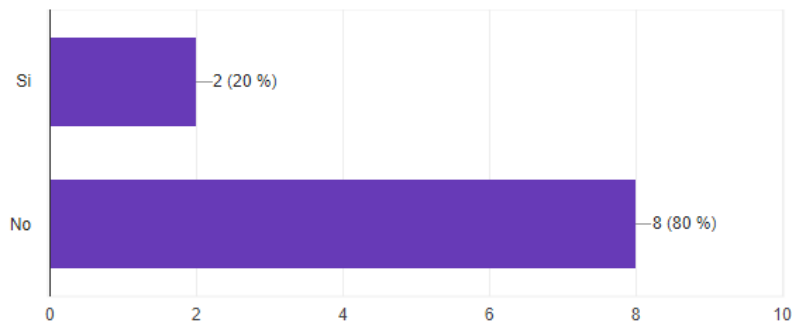


Gráfico 28: Respuesta sobre costo del proceso de pruebas para la organización

Fuente: Elaboración propia

Pregunta 28: En promedio, ¿Qué porcentaje representan las pruebas dentro de la duración total de un proyecto de software en su organización?

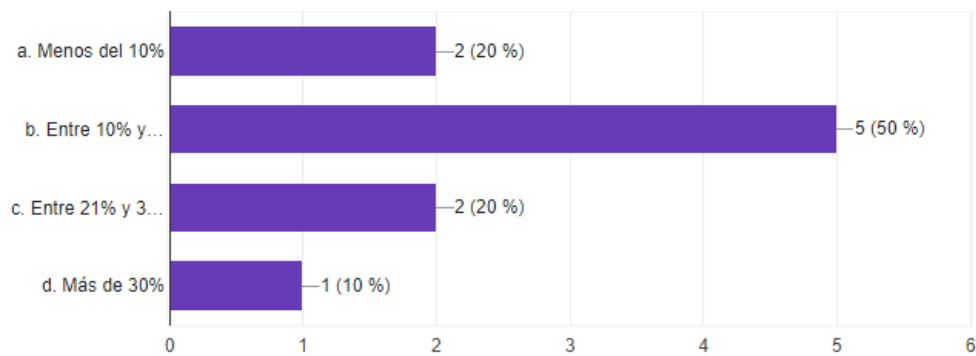


Gráfico 29: Porcentaje que representan las pruebas dentro de la duración de un proyecto de software

Fuente: Elaboración propia

9.1.1.6 Respuestas a las preguntas de la sección 6: Métricas del proceso de pruebas.

Pregunta 29: ¿Las pruebas que se realizan son registradas junto con el esfuerzo invertido en hacerlas?

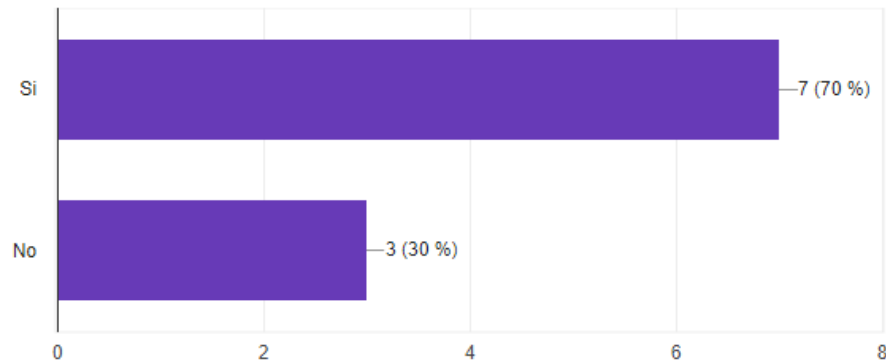


Gráfico 30: Respuesta a pregunta sobre registro de pruebas junto con el esfuerzo invertido

Fuente: Elaboración propia

Pregunta 30: ¿La empresa clasifica los defectos encontrados en la fase de pruebas?

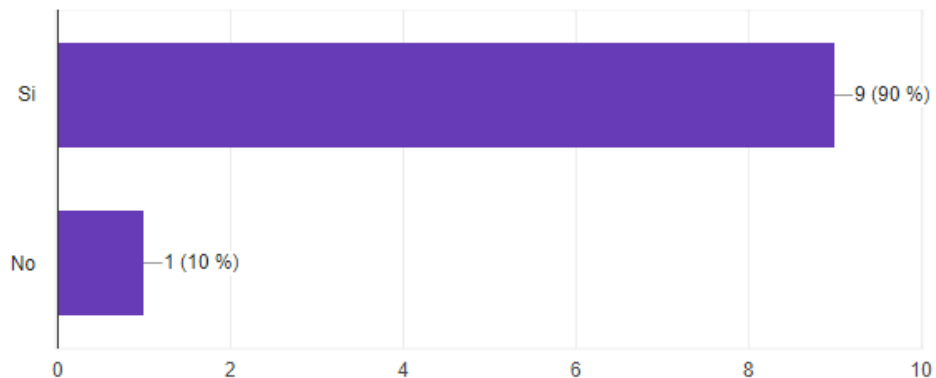


Gráfico 31: Respuesta a la pregunta sobre clasificación de defectos encontrados en la fase de pruebas

Pregunta 31: ¿Considera importante realizar la medición de la efectividad de las pruebas?

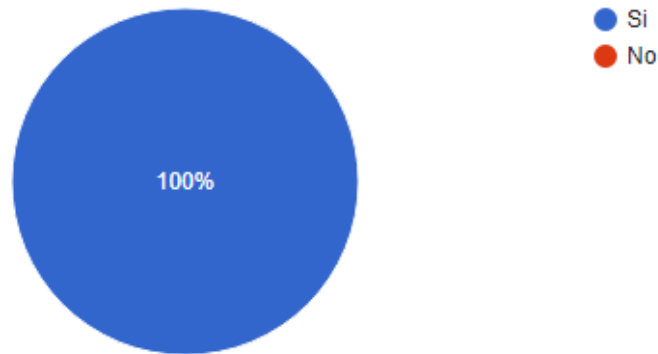


Gráfico 32: Clasificación de defectos encontrados en la fase de pruebas

Fuente: Elaboración propia

Pregunta 32: ¿La Empresa emplea indicadores para evaluar la eficiencia del proceso de pruebas?

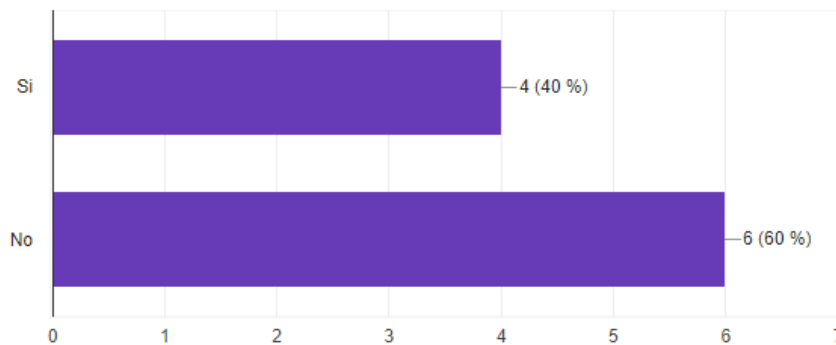


Gráfico 33: Empleo de indicadores para evaluar la eficiencia del proceso de pruebas

Fuente: Elaboración propia

9.1.1.7 Respuestas a las preguntas de la sección 7: Herramientas utilizadas en el proceso de pruebas.

Pregunta 33: ¿Qué herramientas emplea la organización para diseñar y ejecutar las pruebas?

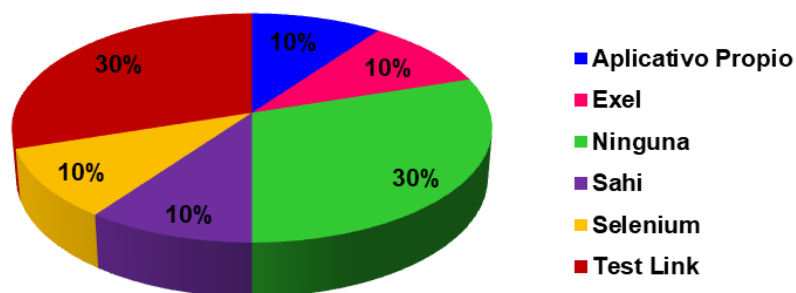


Gráfico 34: Herramientas empleadas para diseñar y ejecutar las pruebas

Fuente: Elaboración propia

Pregunta 34: ¿Qué herramientas emplea la organización para gestionar los defectos encontrados en las pruebas?

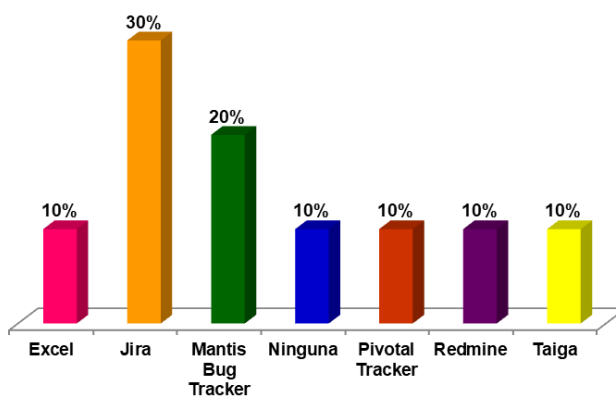


Gráfico 35: Herramientas empleadas para gestionar los defectos encontrados en las pruebas

Fuente: Elaboración propia

9.1.1.8 Respuestas a las preguntas de la sección 8: Oportunidades de mejora del proceso de pruebas.

Pregunta 35: ¿Considera usted que el proceso de pruebas de software debe ser mejorado en su empresa?

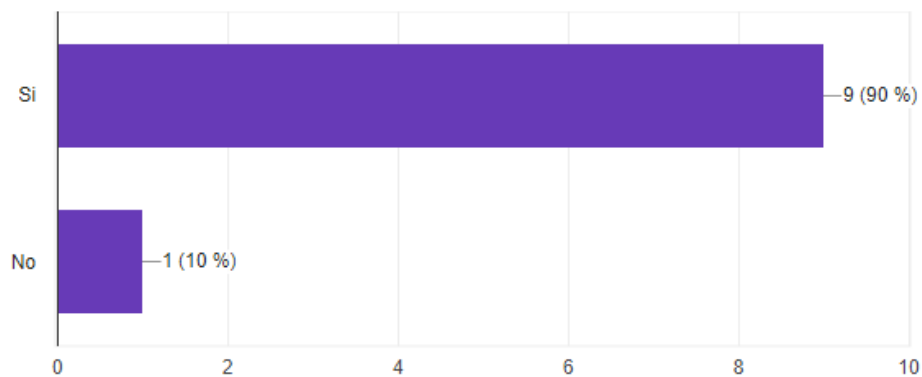


Gráfico 36: Consideración sobre mejoramiento del proceso de pruebas en las organizaciones

Fuente: Elaboración propia

Pregunta 36: ¿Considera usted que el hecho de contar con talento humano especializado en pruebas mejoraría la calidad del software desarrollado por la empresa?

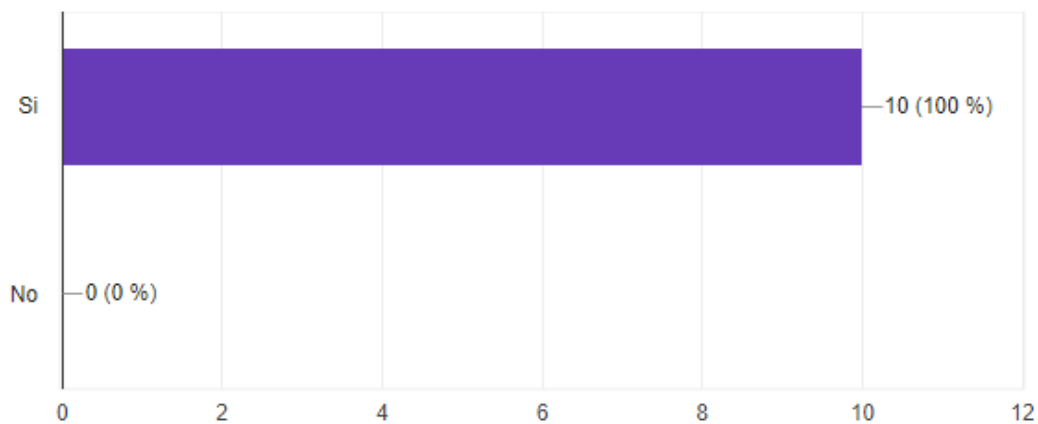


Gráfico 37: Consideración sobre tener talento humano especializado en pruebas

Fuente: Elaboración propia.

Pregunta 37: ¿Considera usted que el hecho de contar con un área de pruebas en su empresa representa o representaría un gran gasto para su organización?

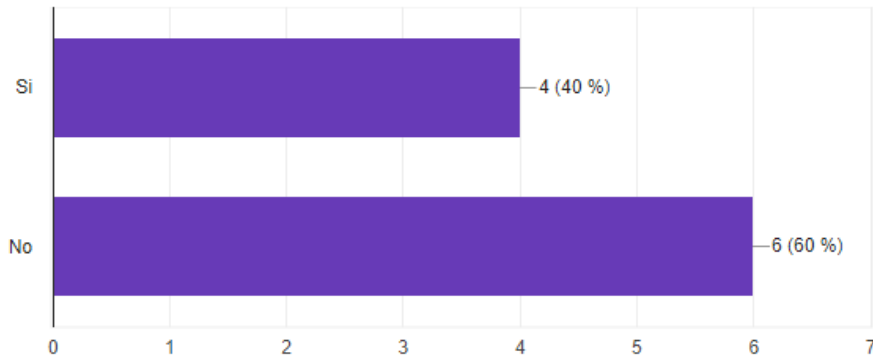


Gráfico 38: Consideración sobre contar con un área de pruebas en la organización

Fuente: Elaboración propia.

Pregunta 38: ¿Cuáles considera usted que son los elementos que se deben incluir en un proceso de pruebas cuya ejecución sea independiente del proceso de desarrollo?

- Acceso a especificación de requerimientos
- Ejecución en ambientes similares a los productivos
- Pruebas funcionales
- Automatización
- Pruebas de usabilidad
- Pruebas con el usuario

Pregunta 39: ¿Si existiera en la ciudad una empresa dedicada al Testing de Software, consideraría contratar dicho servicio?

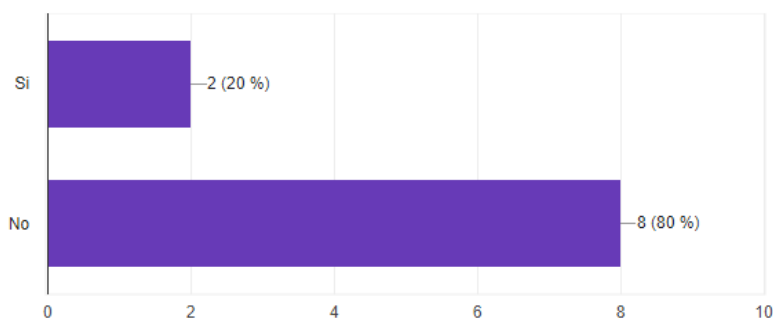


Gráfico 39: Consideración sobre contratar el servicio de pruebas de software

Fuente: Elaboración propia.

Pregunta 39.1: ¿Por qué no?

- El proceso se tiene internamente.
- La empresa ha realizado un esfuerzo significativo en el tema, contratarlo externamente sería retroceder en el avance logrado.
- Hay que invertir en la formación del personal dentro de la empresa.
- Experiencia en el sector donde se desarrolla.
- Costos y aplicación del producto.

9.1.2 Análisis de resultados de la encuesta aplicada a las empresas.

A continuación se detalla el análisis de los resultados obtenidos por cada una de las categorías que hacen parte de la encuesta:

a. Generalidades de la Organización.

- La mayoría de las empresas encuestadas lleva más de 10 años de permanencia en el mercado.

- La principal actividad económica de las empresas consultadas es el desarrollo de software con un 90%, seguido del mantenimiento de software con un 40%, la gestión de infraestructura con 1% y las Pruebas de Software también con 1%.
- En lo que respecta al uso de una Metodología, Modelo o Marco de Trabajo de Desarrollo de Software, se destaca que la mayoría de las organizaciones mencionaron la utilización de metodologías ágiles de forma parcial o total, haciendo énfasis en SCRUM particularmente.
- El 60% de las empresas encuestadas tienen certificaciones de calidad, predominando CMMI Nivel III e IT Mark.
- El 90% de las empresas, administra las configuraciones y las versiones del software.
- En lo que respecta a las técnicas utilizadas para llevar a cabo la especificación del software, se destaca el predominio de los Casos de Uso y las Historias de Usuario, seguidos del Prototipado de Interfaz y los Diagramas de Flujo.

b. Generalidades del Proceso de Pruebas.

- En lo que respecta al Proceso de Pruebas, el 70% de las empresas mencionó que utiliza un proceso o metodología para el desarrollo de las pruebas de software.
- Dentro de las metodologías más utilizadas, se destaca la aplicación de pruebas basadas en TMMi, el proceso guiado por el ISTQB y el Agil Testing. También se resalta la aplicación de pruebas unitarias, funcionales, integrales y de estrés.
- De igual manera, llama la atención que el 70% de las empresas manifestó contar con un área de pruebas de software. Sin embargo, al consultar sobre la forma como se llevan a cabo las pruebas de software, se destaca que el 66,7% de las empresas mencionó que cuenta con varios equipos de desarrollo dentro de los cuales se distribuye el rol de pruebas, teniendo como principio que quien prueba no sea la misma persona que desarrolla. En este último aspecto, también es importante destacar que el 16,7% de las organizaciones consultadas, indicó que tiene equipos dedicados a las pruebas de software y en igual medida (16,7%) que terceriza las pruebas.
- En lo que respecta a la cantidad de personas que hacen parte del área de pruebas, se destaca que el 80% de las empresas tiene de 1 a 5 personas.
- Al consultar sobre el personal certificado en pruebas de software al interior de cada organización, se encuentra que tan solo el 30% de las organizaciones consultadas, afirmó

contar con profesionales certificados en algunos de los niveles establecidos por el ISTQB (International Software Testing Qualifications Board).

- En cuanto a las características o atributos de calidad evaluados en las pruebas de software, se destaca la Funcionalidad con un 100%, seguido de usabilidad y eficiencia con 80% y 60% respectivamente.
- De igual manera se destaca que el 100% de las empresas manifestó realizar pruebas funcionales, el 70% pruebas unitarias y el 50% pruebas de humo, pruebas de integración y pruebas de aceptación.
- En lo que se refiere a los ambientes de pruebas, es importante resaltar que el 80% de las organizaciones consultadas, manifestó contar con ambientes de pruebas independientes del entorno de desarrollo.

c. Planeación de las Pruebas.

- Al consultar sobre la manera como se realiza la planeación de las pruebas, se destaca que el 80% de las empresas afirmó que planifica las pruebas de software al igual que los recursos requeridos para su ejecución.
- El 66,7% de las organizaciones consultadas, manifestó que planifica las pruebas desde el inicio del proyecto. Mientras que un 77,8% indicó que en el Plan de Pruebas establece la duración del ciclo de pruebas.
- En lo que se refiere a los atributos de calidad que se consideran en las pruebas, se encontró que el 55,6% no los define en el plan de pruebas correspondiente.
- En cuanto al tipo de pruebas a ejecutar, se destaca que el 88,9% de las empresas establece dicho criterio en el documento del plan de pruebas.

d. Diseño de las Pruebas.

- Al indagar sobre aspectos relacionados con el diseño de las pruebas de software, se destaca que el 70% de las empresas encuestadas, afirmó emplear técnicas para el diseño de casos de prueba, destacándose en primer lugar las técnicas basadas en el uso, seguidas de las basadas en la especificación y en la experiencia respectivamente.
- En cuanto a la socialización de los casos de prueba con el cliente, se encontró que el 60% de las organizaciones partícipes del estudio, no lleva a cabo dicha labor, argumentando la

poca disponibilidad de los clientes como el principal motivo para no realizar la validación correspondiente.

e. Estimación de Costos.

- En lo que a la Estimación de Costos se refiere, se destaca que el 70% de las empresas registra el esfuerzo invertido y los costos de las pruebas. En igual proporción (70%), las organizaciones consultadas afirmaron incluir en el presupuesto de cada proyecto un rubro para llevar a cabo el proceso de pruebas.
- También es importante resaltar que el 80% de las empresas que hacen parte de la muestra, indicó que no considera que el proceso de pruebas es demasiado costoso para la organización.
- Al indagar sobre el porcentaje que representan las pruebas dentro de la distribución total de un proyecto de software, el 50% de las empresas coincidieron en afirmar que éstas representan entre 10% y 20%.

f. Métricas del Proceso de Pruebas.

- En lo que a la utilización de métricas en el proceso de pruebas, se destaca que el 70% de las organizaciones afirmó que las pruebas que se realizan se registran junto con el esfuerzo invertido en hacerlas.
- De igual manera, se encontró que el 90% de las empresas, clasifica los defectos encontrados en la fase de pruebas.
- También es importante resaltar que el 100% de las empresas, considera importante realizar la medición de la efectividad de las pruebas. Sin embargo, tan solo el 60% de ellas, afirmó emplear métricas para tal fin. Algunas de ellas son: Productos no conformes, afectaciones en los pasos a producción, incidencias internas y externas, esfuerzo invertido en pruebas, defectos por puntos de función, calidad en el proceso de pruebas y eficacia en el reporte de bugs.

De igual manera, al indagar sobre el motivo de no emplear métricas para determinar la efectividad de las pruebas, las empresas afirmaron no contar con ninguna definida.

g. Herramientas utilizadas en el proceso de pruebas.

- En cuanto a la utilización de herramientas para gestionar las pruebas, se encontró que el 30% de las empresas afirmó emplear TestLink, mientras que un 20% no tiene ninguna. También se mencionan con un 10% Sahi, Excel y el uso de un aplicativo propio.
- También se consultó sobre el uso de una herramienta para la gestión de los defectos encontrados, destacándose Jira con un 30%, seguida de Mantis Bug Tracker.

h. Oportunidades de mejora del proceso de pruebas.

- Al consultar sobre aspectos relacionados con las oportunidades de mejora del proceso de pruebas se encontró que el 90% de las empresas encuestadas afirmó considera que el dicho proceso debe ser mejorado al interior de la organización. De igual manera, se destaca que el 100% de las empresas considera que el hecho de contar con talento humano especializado en pruebas de software mejoraría la calidad del software producido.
- También es importante resaltar que el 60% de las empresas considera que el hecho de contar con un área de pruebas no representaría un gran gasto para la organización, mientras que un 40% si lo afirma.
- Algunos de los elementos que las empresas encuestadas consideran que deben ser tenidos en cuenta en un proceso de pruebas cuya ejecución sea independiente son:
 - Acceso a especificación de requerimientos
 - Ejecución en ambientes similares a los productivos
 - Pruebas funcionales
 - Automatización
 - Pruebas de usabilidad
 - Pruebas con el usuario

9.2 FORMULACIÓN DEL PROCESO DE TESTING FUNCIONAL DE SOFTWARE “QUALITEST” PARA LAS MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA

El diseño del proceso de testing funcional de software para MiPymes desarrolladoras de la ciudad de Pereira, hace parte de la Fase de Planeación de la metodología. Se modeló bajo SPEM para una mejor presentación y entendimiento. A dicho proceso se le dio el nombre de “QUALITEST”.

QUALITEST es un proceso para la ejecución de pruebas funcionales de productos software cuya aplicación se lleva a cabo de manera independiente del proceso empleado para desarrollar el producto. La guía propuesta es el resultado de: i) La caracterización de las prácticas de pruebas empleadas por las MiPymes Desarrolladoras de Software de la ciudad de Pereira, ii) Conceptos y buenas prácticas que hacen parte tanto del referente teórico como de los antecedentes de la investigación, iii) Los atributos de calidad relacionados con la funcionalidad planteados por la norma ISO/IEC 9126 y iv) Lineamientos definidos por el estándar ISO/IEC/IEEE 29119.

El objetivo principal del proceso definido es proporcionar un conjunto de buenas prácticas que permitan llevar a cabo las tareas de pruebas de software de manera independiente del proceso de desarrollo, garantizando la eficiencia y eficacia de las mismas.

9.2.1 Fases.

El proceso está compuesto por cinco (5) fases: Análisis Inicial, Planeación, Diseño, Ejecución y Cierre:



Ilustración 33: Etapas del proceso de pruebas Qualitest

Fuente: Elaboración propia

Es importante resaltar que las fases de QUALITEST, se plantearon tomando como base algunas características definidas en el ProTest (proceso de pruebas funcionales de productos de software realizadas en el Laboratorio de Testing Funcional del CES) el cual se detalla en la sección de antecedentes. Sin embargo, ambos procesos difieren tanto en la cantidad de fases planteadas como en las actividades que hacen parte de cada una de ellas.

A continuación, se detallan las fases que hacen parte del proceso propuesto:

Análisis Inicial: En esta etapa se lleva a cabo una reunión inicial para realizar una revisión previa de los requerimientos que constituyen las funcionalidades que se van a probar para determinar el alcance de la prueba e identificar prioridades. Posteriormente, se realiza la estimación de tiempos de prueba, los cuales son tenidos en cuenta para elaborar la propuesta económica para contratar el servicio de prueba (La manera como se elabora dicha propuesta no hace parte de Qualitest), si ésta no se aprueba se analizan los inconvenientes presentados y se documentan las lecciones aprendidas para que sean tenidas en cuenta en futuros proyectos.

Tabla 5: Elementos de la fase Análisis Inicial

Fuente: Elaboración propia

ACTIVIDAD	ARTEFACTO GENERADO
Reunión inicial	AI1 – Acta de Reunión
Revisión de Requerimientos	AI2 - Relación de artefactos a probar
Aclaración de inquietudes sobre Documento de Especificación de Requerimientos	AI1 – Acta de Reunión
Identificación de herramientas requeridas	AI3 – Lista de herramientas requeridas
Estimación de tiempos de prueba	AI4 – Formato de Estimación de Tiempos

Planificación: Después de que se haya aprobado la propuesta económica elaborada en la fase de Análisis Inicial, se procede a elaborar la planeación de las pruebas, definiendo funcionalidades a probar, priorización de ejecución de casos de prueba (con base en los riesgos identificados, probabilidad de fallo de las funcionalidades que hacen parte de la prueba y tipo de prueba que se va a ejecutar), cronograma y herramientas requeridas para la ejecución del ciclo de prueba.

Tabla 6: Elementos de la fase de Planeación

Fuente: Elaboración propia

ACTIVIDAD	ARTEFACTO GENERADO
Identificación de riesgos	AI2 - Relación de artefactos a probar con registro de riesgos identificados.
Elaboración del plan de pruebas	P1 – Plan de pruebas
Aprobar plan de pruebas	P1 – Plan de pruebas (Con registro de aprobación)

Diseño: En esta etapa se lleva a cabo la construcción de los casos de prueba, los cuales deben ser aprobados por el equipo de desarrollo. Para tal fin, se recomienda la utilización de técnicas basadas en la especificación o de caja negra.

Tabla 7: Elementos de la fase de Diseño

Fuente: Elaboración propia

ACTIVIDAD	ARTEFACTO GENERADO
Elaboración de casos de prueba	D1 – Formato de caso de prueba
Revisión y aprobación de casos de prueba	D2 – Check list de revisión de caso de prueba
Ajuste de casos de prueba	D1 – Formato de caso de prueba (Actualizado)
Aprobación de casos de prueba	D1 – Formato de caso de prueba (Con registro de aprobación)

Ejecución: En esta etapa se lleva a cabo el desarrollo de las pruebas, iniciando por la configuración de los entornos de prueba y de las herramientas requeridas para la ejecución de las mismas. Posteriormente, se efectúa una prueba de humo de las funcionalidades que hacen parte del alcance de la prueba para verificar la estabilidad de la versión. Una vez realizada la prueba de humo, se procede con la ejecución de los casos de prueba y con el reporte y verificación de las evidencias reportadas. En esta fase también se realiza semanalmente un informe de monitoreo con el fin de tener informado a los interesados relevantes del estado actual de las pruebas.

De igual manera, se llevan a cabo las pruebas de regresión, las cuales consisten en revisar nuevamente todos los bugs generados para garantizar que están corregidos satisfactoriamente. Por último, como actividad opcional, se plantea la realización de pruebas de aceptación con el usuario final, éstas se llevarán a cabo siempre y cuando se consideren pertinentes por parte del equipo de desarrollo y se cuente con disponibilidad de tiempo del usuario.

Tabla 8: Elementos de la fase de Ejecución

Fuente: Elaboración propia

ACTIVIDAD	ARTEFACTO GENERADO
Configuración del entorno de prueba	Entorno de prueba configurado
Ejecución de prueba de humo	E1 – Check list de Prueba de Humo
Ejecución de caso de prueba	D1 – Formato de caso de prueba (Con registro de las corridas por ciclo de prueba)
Reporte de Bugs	E2 - Reporte de incidencia en la herramienta definida
Recibir solución de Bugs	Entorno de prueba configurado
Revisión de Bugs corregidos	E2 – Reporte de incidencia (Modificación del estado de la incidencia en la herramienta definida).
Monitoreo del Proyecto	E3 – Informe de estado de las pruebas
Pruebas de regresión	E2 – Reporte de incidencia (Registro de verificación de cada uno de los bugs en la herramienta definida).
Pruebas de aceptación con el usuario final (Actividad Opcional)	E4 – Carta de aceptación E5 - Encuesta de satisfacción

Cierre: Una vez se han finalizado correctamente los ciclos de prueba, se procede a realizar la reunión de cierre del proyecto, en la cual se lleva a cabo una retroalimentación para dar a conocer los resultados del proceso e identificar las lecciones aprendidas y las buenas prácticas.

Tabla 9: Elementos de la fase de Cierre

Fuente: Elaboración propia

ACTIVIDAD	ARTEFACTO GENERADO
Elaborar informe de finalización de las pruebas	C1 – Informe de finalización de prueba
Identificar lecciones aprendidas y mejores prácticas	C2 – Registro de lecciones aprendidas y mejores prácticas.

9.2.2 Roles.

El proceso de testing funcional de software para las MIPYMES desarrolladoras de software de la ciudad de Pereira QUALITEST, tiene en cuenta los siguientes roles:

- Líder del proceso de pruebas
- Ingeniero de Pruebas
- Líder del proceso de Desarrollo
- Ingeniero de Desarrollo

9.2.2.1 Líder del proceso de pruebas

La persona que desempeñe este rol, es la encargada de planificar, dirigir, monitorear y controlar las actividades y tareas de prueba. Dentro del proceso QUALITEST, realiza concretamente las siguientes actividades:

- Convoca la reunión inicial para realizar la revisión previa de los requerimientos que constituyen las funcionalidades que se van a probar.

- Revisa el documento de Especificación de Requerimiento para entender la lógica de cada una de las funcionalidades que harán parte de la prueba.
- En caso de que se presenten dudas relacionadas con la Especificación de Requerimientos, solicita reunión con el equipo de desarrollo para hacer las aclaraciones pertinentes.
- En compañía de los Ingenieros de Pruebas, identifica las herramientas requeridas para llevar a cabo la ejecución de las pruebas.
- En compañía de los Ingenieros de Pruebas, realiza la estimación de los tiempos requeridos para la ejecución de las fases que hacen parte del proceso de pruebas.
- Tomando como base la Especificación de Requerimientos, analiza los riesgos de cada funcionalidad con el fin de establecer prioridades y definir la estrategia de prueba.
- En conjunto con los Ingenieros de Pruebas, elabora el Plan de Pruebas correspondiente determinando objetivo, alcance y estrategia de pruebas requerida.
- Contribuye en la elaboración de casos de prueba para cada una de las funcionalidades que hacen parte del alcance.
- Elabora el Reporte de Monitoreo del Proyecto.
- Elabora el Informe de Cierre del Proyecto.
- Participa en la identificación de lecciones aprendidas y mejores prácticas que podrán ser aplicadas en proyectos futuros.



Ilustración 34: Fases del proceso en las que participa el Rol de Líder del Proceso de Pruebas y Productos de Trabajo generados.

Fuente: Elaboración propia

9.2.2.2 Ingeniero de Pruebas

Rol encargado de planificar, diseñar y ejecutar los casos de prueba de acuerdo al proceso establecido. Dentro del proceso QUALITEST, realiza concretamente las siguientes actividades:

- Participa en la revisión del documento de Especificación de Requerimiento para entender la lógica de cada una de las funcionalidades que harán parte de la prueba.
- Contribuye con la identificación de las herramientas requeridas para llevar a cabo la ejecución de las pruebas.
- Participa con el Líder de Pruebas en la realización de la estimación de los tiempos requeridos para la ejecución de las fases que hacen parte del proceso de pruebas.
- Revisa la Especificación de Requerimientos y analiza los riesgos de cada funcionalidad con el fin de establecer prioridades y definir la estrategia de prueba para determinar la lista completa de los artefactos a probar.
- En conjunto con el Líder de Pruebas, elabora el Plan de Pruebas correspondiente determinando objetivo, alcance y estrategia de pruebas requerida.
- Elabora los casos de prueba para cada una de las funcionalidades que hacen parte del alcance.
- Revisa los casos de prueba elaborados y en caso de que tenga observaciones al respecto las consigna en el check list de revisión.
- Ajusta los casos de prueba tomando como base las observaciones y/o sugerencias realizadas por el Equipo de Desarrollo.
- Realiza la configuración requerida en el ambiente de pruebas asignado.
- Ejecuta la Prueba de Humo con el fin de garantizar la estabilidad del ambiente de pruebas.
- Ejecuta los Diseños de Casos de Prueba asignados.
- Reporta los bugs encontrados en la herramienta definida para tal fin.
- Lleva a cabo la revisión de los bugs corregidos y actualiza el estado de los mismos en la herramienta definida para el reporte de bugs.

- Una vez se encuentran ejecutados exitosamente todos los casos de prueba, procede a hacer la revisión de cada uno de los bugs generados en la versión con el fin de garantizar que todos se encuentran corregidos.
- Presenta al usuario final las funcionalidades probadas para que éste verifique su funcionamiento y lleve a cabo la aceptación del desarrollo realizado.
- Participa en la identificación de lecciones aprendidas y mejores prácticas que podrán ser aplicadas en proyectos futuros.

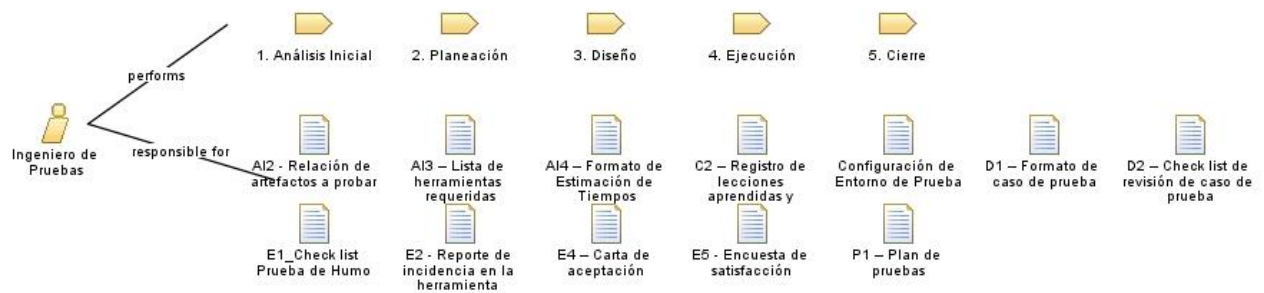


Ilustración 35: Fases del proceso en las que participa el Ingeniero de Pruebas y Productos de Trabajo generados.

Fuente: Elaboración propia

9.2.2.3 Líder del proceso de desarrollo

Rol encargado de planificar, dirigir, monitorear y controlar las actividades y tareas de Desarrollo. Dentro del proceso QUALITEST, realiza concretamente las siguientes actividades:

- Participa en la reunión inicial para realizar la revisión previa de los requerimientos que constituyen las funcionalidades que se van a probar.
- Brinda apoyo al Equipo de Pruebas para solucionar inquietudes sobre las funcionalidades a probar.
- Revisa y aprueba el Plan de Pruebas.
- Revisa el informe de estado de las pruebas.
- Revisa la Carta de Aceptación y la Encuesta de Satisfacción diligenciada por el usuario final.
- Revisa el informe de finalización de las pruebas.

- Participa en la identificación de lecciones aprendidas y mejores prácticas que podrán ser aplicadas en proyectos futuros.



Ilustración 36: Fases del proceso en las que participa el Rol de Líder del Proceso de Desarrollo y Productos de Trabajo generados.

Fuente: Elaboración propia

9.2.2.4 Ingeniero de Desarrollo

Rol encargado de desarrollar software de alta calidad siguiendo un proceso definido. Dentro del proceso QUALITEST, realiza concretamente las siguientes actividades:

- Participa en la reunión inicial para realizar la revisión previa de los requerimientos que constituyen las funcionalidades que se van a probar.
- Brinda apoyo al Equipo de Pruebas para solucionar inquietudes sobre las funcionalidades a probar.
- Soluciona los bugs reportados por el Ingeniero de Pruebas.
- Participa en la identificación de lecciones aprendidas y mejores prácticas que podrán ser aplicadas en proyectos futuros.



Ilustración 37: Fases del proceso en las que participa el Rol de Ingeniero de Desarrollo y Productos de Trabajo generados.

Fuente: Elaboración propia

Para tener mayor claridad con respecto a la secuencia de ejecución de las actividades que hacen parte del proceso propuesto, ver **Anexo C: *Proceso de testing funcional de software para las MiPymes desarrolladoras de software de la ciudad de Pereira.***

9.3 APLICACIÓN DEL PROCESO DE TESTING FUNCIONAL DE SOFTWARE PARA LAS MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA – “QUALITEST”

Con el fin de poder llevar a cabo la aplicación del proceso propuesto, se seleccionaron dos proyectos elaborados por una empresa que para efectos de la investigación será denominada como Empresa 1, quien fue la organización que aceptó ser parte de este estudio. Dicha organización fue constituida legalmente en el año 2.002 en la ciudad de Pereira y desde entonces, su principal actividad económica ha estado enfocada en el desarrollo, la implantación y la integración de soluciones de TI.

9.3.1 Evaluación inicial.

En esta etapa se llevó a cabo la aplicación de una encuesta (Ver Anexo D), con el fin de conocer las características tanto de la organización como del proceso de pruebas que ha venido aplicando. Los resultados arrojados por dicho instrumento, se presentan a continuación:

▪ Generalidades de la Organización

La metodología de desarrollo de software está basada en algunas prácticas de SCRUM.

La empresa cuenta actualmente con dos certificaciones de calidad: CMMI Nivel III e It Mark. De igual manera se destaca que la Organización administra las configuraciones y las versiones del software.

En lo que respecta a las técnicas utilizadas para especificar el software, se destacan las Historias de Usuario y el Prototipado de Interfaz.

▪ Generalidades del proceso de pruebas

Actualmente la empresa utiliza un proceso para el desarrollo de las pruebas, el cual se basa en una metodología propia. Sin embargo, no cuenta con un área especializada en la

realización de pruebas de software. Para llevar a cabo dicho proceso, cuenta con varios equipos de desarrollo dentro de los cuales se distribuye el rol de pruebas, teniendo como principio que quien prueba no sea la misma persona que desarrolla.

En lo que respecta a las características o atributos de calidad que se tienen en cuenta a la hora de realizar las pruebas de software, se destacan las siguientes: Funcionalidad, fiabilidad, usabilidad y calidad de uso.

Los tipos de prueba realizados son: Pruebas funcionales, pruebas unitarias, pruebas de humo, pruebas de regresión, pruebas de integración y pruebas de aceptación.

También se destaca el hecho de contar con un ambiente de pruebas independiente del entorno de desarrollo.

▪ **Planeación de las pruebas**

En lo que a la planeación de las pruebas se refiere, la empresa manifiesta que actualmente planifica las pruebas desde el inicio del proyecto, estableciendo en el plan de pruebas la duración del ciclo de pruebas y los tipos de pruebas que se aplicarán.

▪ **Diseño de las pruebas**

Al indagar sobre la utilización de técnicas de diseño, la empresa afirma utilizar las siguientes: Basadas en la especificación y basadas en el uso.

▪ **Estimación de costos**

En cuanto a la estimación de costos, se encuentra que la Organización registra el esfuerzo invertido en las pruebas. De igual manera se resalta que las pruebas representan entre el 21% y el 30% de la duración total de un proyecto.

▪ **Métricas**

En lo que a las métricas respecta, la empresa afirma tener las siguientes: Producto no conforme y afectaciones en pasos a producción.

▪ **Herramientas utilizadas en el proceso de pruebas**

La empresa utiliza la herramienta Taiga para gestionar los defectos de las pruebas, mientras que los casos de prueba los elabora utilizando Excel.

▪ **Oportunidades de mejora del proceso de pruebas**

La empresa considera que es necesario mejorar el proceso de software que tiene actualmente.

9.3.2 Selección de proyectos para la prueba piloto.

Para llevar a cabo la selección de los proyectos para la prueba piloto, se programó una reunión con el Coordinador de Desarrollo, en la cual se realizó la presentación del proceso planteado en la investigación, posteriormente se recibió la retroalimentación correspondiente a las mejoras que se podían implementar al proceso propuesto. Una vez concluido este punto, se acordó realizar la aplicación del proceso en dos proyectos, uno de tamaño pequeño (Denominado Proyecto 1) y otro de tamaño mediano (Denominado Proyecto 2).

Para tener mayor claridad sobre las dimensiones de los proyectos, se relaciona a continuación la clasificación que con la que cuenta actualmente la compañía de acuerdo a su duración:

Tabla 10: Clasificación de proyectos según su duración

Fuente: Empresa 1

TAMAÑO DE LOS PROYECTOS SEGÚN SU DURACIÓN		
Pequeño	Mediano	Grande
De 2 a 4 Semanas	De 4 a 8 Semanas	Superior a 8 Semanas

Posteriormente, se definieron las variables a medir antes y después de la aplicación del proceso de pruebas para establecer la línea base correspondiente, las cuales se relacionan a continuación:

Esfuerzo invertido durante todo el proceso de pruebas

Cantidad de bugs reportados

Cantidad de productos no conformes.

Una vez definidas las variables, se concretaron las fechas de aplicación y se programó la reunión inicial con los equipos de desarrollo correspondientes.

9.3.3 Medición inicial.

Con el fin de tener una línea base para comparar los resultados obtenidos en la aplicación del proceso de pruebas, se presentan a continuación las métricas correspondientes a los proyectos de tamaño pequeño y mediano:

Tabla 11: Valores de medición inicial

Fuente: Empresa 1

Tamaño del proyecto	Esfuerzo invertido en pruebas	Producto no conforme	Cantidad de Bugs Reportados
Pequeño	35 Horas	2	6
Mediano	84 Horas	5	16

9.3.4 Aplicación del proceso en Proyecto 1.

▪ Fase de análisis inicial

En esta fase se llevó a cabo una reunión con el equipo de desarrollo para presentar el proyecto de investigación y capacitar a los integrantes del proyecto en cada una de las fases que hacen parte del proceso. La capacitación fue dictada a un total de cuatro personas, tuvo una duración de 4 horas y los temas tratados fueron los siguientes:

- Contextualización del proceso propuesto
- Fases del proceso (Explicación de cada una de las fases y de las actividades que hacen parte de ellas)

- Productos derivados de la ejecución del proceso (Explicación de cada uno de los formatos en los cuales se lleva a cabo el registro de las actividades realizadas en cada una de las fases).

De igual manera se llevó a cabo la revisión del Documento de Especificación de Requerimientos, el cual no estaba claramente definido, situación que fue solventada con la presentación detallada de las funcionalidades implementadas por parte del equipo de desarrollo. Posteriormente, se definieron las herramientas requeridas para las pruebas y se identificaron las prioridades de las mismas. Por último, se llevó a cabo la estimación correspondiente. Los artefactos resultantes de esta fase fueron los siguientes:

- ✓ AI1 – Acta de Reunión
- ✓ AI2 - Relación de artefactos a probar
- ✓ AI3 – Lista de herramientas requeridas

▪ **Fase de Planificación**

Una vez realizada la reunión inicial y tomando como insumo la Especificación de Requerimientos y los artefactos generados en la etapa anterior, se procedió a identificar los riesgos asociados a las pruebas y se elaboró el Plan de pruebas con su respectiva aprobación. Los artefactos resultantes de esta fase fueron:

- ✓ AI2 - Relación de artefactos a probar (Actualizada)
- ✓ P1 – Plan de pruebas con registro de aprobación

▪ **Fase de Diseño**

En esta fase se llevó a cabo el Diseño de Casos de Prueba correspondientes a 2 Historias de Usuario que hacían parte del proyecto. Una vez elaborados, se procedió con el envío de

los mismos al equipo de desarrollo para que éste efectuara la revisión correspondiente e hiciera las observaciones que considerara pertinentes. Posterior a esto, el equipo de pruebas realizó los ajustes en los casos de pruebas teniendo en cuenta las sugerencias del equipo de desarrollo y efectuó el envío correspondiente al equipo de desarrollo quien después de revisarlos nuevamente dio su aprobación. Los artefactos resultantes de esta fase fueron:

- ✓ D1 – Formatos de caso de prueba aprobados
- ✓ D2 – Check list de revisión de caso de prueba

▪ Fase de Ejecución

En esta fase inicialmente se hizo la configuración de los entornos de prueba, posteriormente se efectuó una prueba de humo para garantizar la estabilidad de las funcionalidades a probar. Se procedió con la ejecución de los casos de prueba y el reporte de bugs, los cuales fueron un total de diez (10), discriminados en las categorías que se relacionan a continuación:

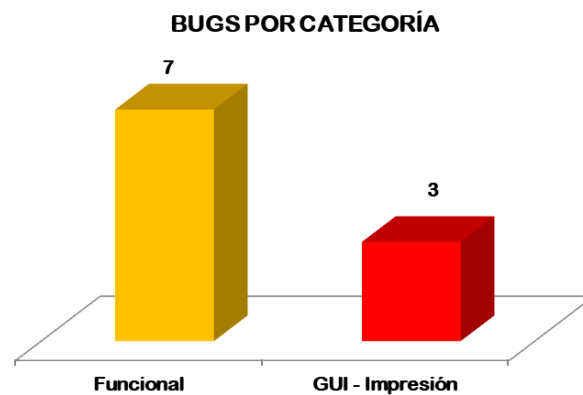


Gráfico 40: Bugs por categoría - Proyecto 1

Fuente: Elaboración propia

Posteriormente, se recibió la solución de los bugs y se procedió con la validación correspondiente, razón por la cual fueron necesarios dos ciclos de pruebas. Después, se elaboró el informe de avance correspondiente y se llevaron a cabo las pruebas de regresión.

No se ejecutaron pruebas de aceptación debido a que el usuario final no tenía disponibilidad para llevar a cabo dicha tarea.

▪ **Fase de cierre**

Una vez verificados todos los bugs y realizada la regresión correspondiente, se procedió a realizar el informe de finalización de las pruebas y se efectuó una reunión con el equipo de desarrollo para socializar los resultados de las mismas e identificar las lecciones aprendidas y las buenas prácticas utilizadas. Los artefactos resultantes de esta fase fueron:

- ✓ C1 – Informe de finalización de pruebas
- ✓ C3 – Registro de lecciones aprendidas y mejores prácticas.

A continuación, se relacionan algunas de las lecciones aprendidas y buenas prácticas identificadas durante la aplicación del proceso:

Tabla 12: Lecciones aprendidas y buenas prácticas identificadas durante la aplicación del proceso Proyecto 1

Tipo	Descripción	Fase del Proceso
Lección aprendida	Para una óptima aplicación del proceso propuesto, es de vital importancia que el documento de Especificación de Requerimientos se encuentre claramente definido.	Análisis Inicial
Buena práctica	Se destaca como buena práctica la realización de una reunión inicial, ya que facilita la comunicación efectiva entre los miembros del equipo de desarrollo y las personas encargadas de realizar las tareas de testing.	Análisis Inicial

Buena práctica	Se considera una buena práctica la aprobación de los casos de prueba por parte del equipo de desarrollo, ya que sus miembros pueden aportar sus conocimientos sobre las funcionalidades desarrolladas para proponer nuevos casos de prueba.	Diseño
Lección aprendida	Una buena redacción de los bugs encontrados es de vital importancia para dar solución a los mismos.	Ejecución
Buena práctica	Se destaca como buena práctica la realización de pruebas de aceptación con el usuario final.	Cierre

9.3.5 Aplicación del proceso en Proyecto 2.

▪ Fase de análisis inicial

Al igual que en el Proyecto 1, en esta fase se llevó a cabo una reunión con el equipo de desarrollo para presentar el proyecto de investigación y capacitar a los integrantes del proyecto en cada una de las fases que hacen parte del proceso. La capacitación fue dictada a un total de tres personas, tuvo una duración de 4 horas y los temas tratados fueron los siguientes:

- Contextualización del proceso propuesto
- Fases del proceso (Explicación de cada una de las fases y de las actividades que hacen parte de ellas)
- Productos derivados de la ejecución del proceso (Explicación de cada uno de los formatos en los cuales se lleva a cabo el registro de las actividades realizadas en cada una de las fases).

De igual manera, se llevó a cabo la revisión del Documento de Especificación de Requerimientos, el cual no estaba claramente definido, situación que fue solventada con la presentación detallada de las funcionalidades implementadas por parte del equipo de desarrollo. Posteriormente, se definieron las herramientas requeridas para las pruebas y se identificaron las prioridades de las mismas. Por último, se llevó a cabo la estimación correspondiente. Los artefactos resultantes de esta fase fueron los siguientes:

- ✓ AI1 – Acta de Reunión
- ✓ AI2 - Relación de artefactos a probar
- ✓ AI3 – Lista de herramientas requeridas

▪ **Fase de Planificación**

Una vez realizada la reunión inicial y tomando como insumo la Especificación de Requerimientos y los artefactos generados en la etapa anterior, se procedió a identificar los riesgos asociados a las pruebas y se elaboró el Plan de pruebas con su respectiva aprobación. Los artefactos resultantes de esta fase fueron:

- ✓ AI2 - Relación de artefactos a probar (Actualizada)
- ✓ P1 – Plan de pruebas con registro de aprobación

▪ **Fase de Diseño**

En esta fase se llevó a cabo el Diseño de Casos de Prueba correspondientes a 4 Historias de Usuario que hacían parte del proyecto. Una vez elaborados, se procedió con el envío de los mismos al equipo de desarrollo para que éste efectuara la revisión correspondiente e hiciera las observaciones que considerara pertinentes. Posterior a esto, el equipo de pruebas realizó los ajustes en los casos de pruebas teniendo en cuenta las sugerencias del equipo de

desarrollo y efectuó el envío correspondiente al equipo de desarrollo quien después de revisarlos nuevamente dio su aprobación. Los artefactos resultantes de esta fase fueron:

- ✓ D1 – Formatos de caso de prueba aprobados
- ✓ D2 – Check list de revisión de caso de prueba

▪ Fase de Ejecución

En esta fase inicialmente se hizo la configuración de los entornos de prueba, posteriormente se efectuó una prueba de humo para garantizar la estabilidad de las funcionalidades a probar. Se procedió con la ejecución de los casos de prueba y el reporte de bugs, los cuales fueron un total de veintiuno (21), discriminados en las categorías que se relacionan a continuación:

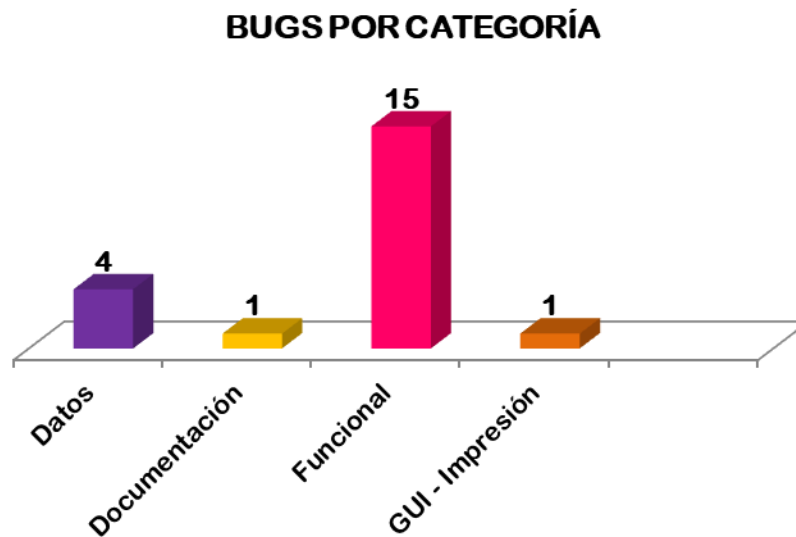


Gráfico 41: Bugs por categoría - Proyecto 2

Fuente: Elaboración propia

Posteriormente, se recibió la solución de los bugs y se procedió con la validación correspondiente, razón por la cual fueron necesarios 3 ciclos de pruebas. Después, se elaboró el informe de avance correspondiente y se llevaron a cabo las pruebas de regresión. No se

ejecutaron pruebas de aceptación debido a que el usuario final no tenía disponibilidad para llevar a cabo dicha tarea.

▪ **Fase de Cierre**

Una vez verificados todos los bugs y realizada la regresión correspondiente, se procedió a realizar el informe de finalización de las pruebas y se efectuó una reunión con el equipo de desarrollo para socializar los resultados de las mismas e identificar las lecciones aprendidas y las buenas prácticas utilizadas. Los artefactos resultantes de esta fase fueron:

- ✓ C1 – Informe de finalización de pruebas
- ✓ C3 – Registro de lecciones aprendidas y mejores prácticas.

A continuación, se relacionan algunas de las lecciones aprendidas y buenas prácticas identificadas durante la aplicación del proceso:

*Tabla 13: Lecciones aprendidas y buenas prácticas identificadas durante la aplicación del proceso
Proyecto 2*

Tipo	Descripción	Fase del Proceso
Lección aprendida	Para una óptima aplicación del proceso propuesto, es de vital importancia que el documento de Especificación de Requerimientos se encuentre claramente definido.	Análisis Inicial
Buena práctica	Se destaca como buena práctica la realización de una reunión inicial con el equipo de desarrollo.	Análisis Inicial
Lección aprendida	Es importante plantear en la fase de ejecución una reunión con el equipo de desarrollo en la	Ejecución

	cual sea posible aclarar dudas relacionadas con los bugs reportados.	
Lección aprendida	Enviar releases con correcciones de bugs por funcionalidad con el fin de agilizar la ejecución de los ciclos de prueba.	Ejecución

9.3.6 Análisis de los resultados obtenidos al aplicar el proceso.

A continuación, se presenta el resumen de los indicadores obtenidos en los dos proyectos después de la aplicación del proceso propuesto:

Tabla 14: Indicadores obtenidos después de aplicar el proceso de pruebas propuesto

Fuente: Elaboración propia

Tamaño del proyecto	Esfuerzo invertido en pruebas	Producto no conforme	Cantidad de Bugs Reportados
Pequeño (Proyecto 1)	26 Horas	0	10
Mediano (Proyecto 2)	78 Horas	0	21

Al realizar el análisis correspondiente se encuentra que:

- Para el proyecto 1, el esfuerzo invertido se redujo en un 25,7%, la tasa de bugs reportados aumentó en un 40%, mientras que el producto no conforme presentó una reducción del 100%.

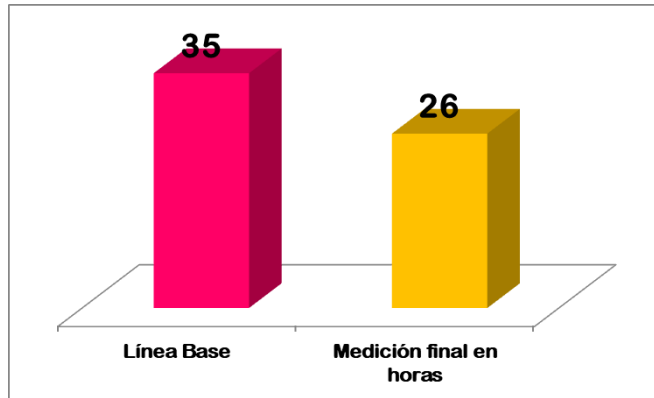


Gráfico 42: Esfuerzo invertido Proyecto 1

Fuente: Elaboración propia

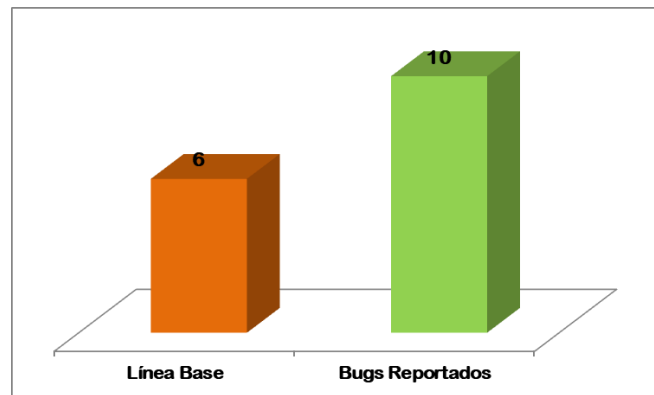


Gráfico 43: Cantidad de Bugs Reportados - Proyecto 1

Fuente: Elaboración propia

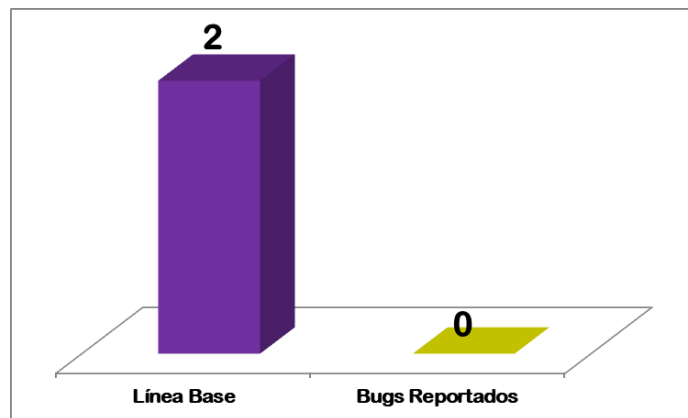


Gráfico 44: Producto no conforme - Proyecto 1

Fuente: Elaboración propia

- Para el proyecto 2, el esfuerzo invertido se redujo en un 20%, la tasa de bugs reportados aumentó en un 31,25%, mientras que el producto no conforme presentó una reducción del 100%.

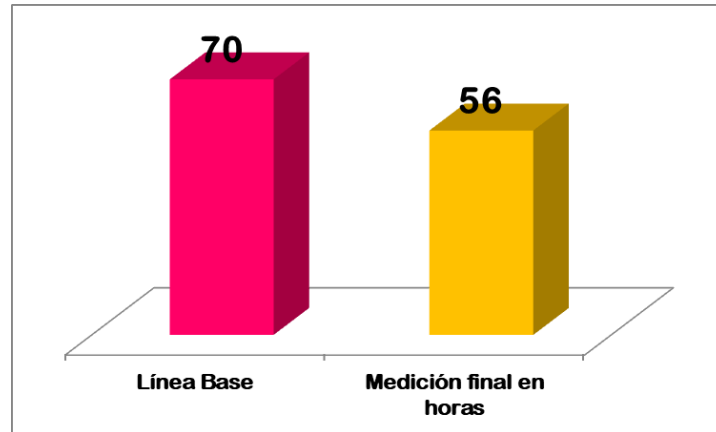


Gráfico 45: Esfuerzo invertido Proyecto 2

Fuente: Elaboración propia

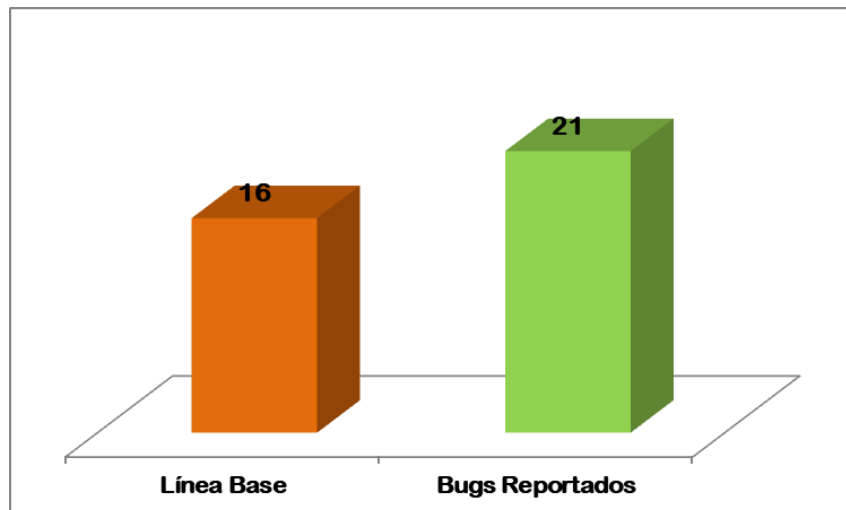


Gráfico 46: Cantidad de Bugs Reportados - Proyecto 2

Fuente: Elaboración propia

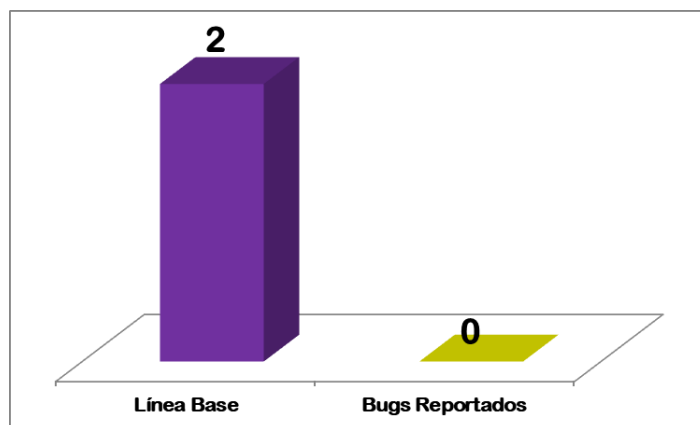


Gráfico 47: Producto no conforme - Proyecto 2

Fuente: Elaboración propia

9.3.7 Evaluación del proceso aplicado.

Después de haber llevado a cabo la aplicación del proceso, se realizó una Encuesta de Satisfacción, cuyo resultado permite destacar lo siguiente:

- Los documentos entregados cumplieron satisfactoriamente con las expectativas de la empresa que hizo parte de la validación del proceso, en cuanto a especificaciones y criterios de calidad.
- En lo que respecta a los canales de comunicación, se resalta que la información entregada acerca del estado del proyecto fue clara, transparente, frecuente y entendible.
- El proceso propuesto se considera entendible y fácil de aplicar, siempre y cuando se tengan especificados correctamente los requerimientos del sistema.
- Se considera que la aplicación del proceso fue altamente satisfactoria para la organización, ya que permitió identificar las fortalezas y oportunidades de mejora en los procesos de la compañía.

De igual manera, al evaluar el proceso tomando como punto de partida las métricas obtenidas, es posible destacar los siguientes aspectos:

▪ **Aumento en la efectividad con respecto a la cantidad de bugs reportados.** Se considera como un aporte valioso el hecho de que se encuentren más errores en la fase de pruebas, ya que disminuye la probabilidad de que el cliente reporte productos no conformes, y, por ende, reduce los costos de la compañía con respecto a la solución de las no conformidades detectadas en el ambiente productivo.

▪ **Reducción del esfuerzo invertido en la fase de pruebas y de los productos no conformes reportados por parte del cliente.** Se considera que el proceso es eficiente y eficaz, ya que su aplicación no solo permitió aumentar la tasa de bugs reportados, también redujo el tiempo empleado en la ejecución de las pruebas.

También es importante destacar como parte de la evaluación del proceso propuesto y tomando como base los resultados obtenidos, al igual que las lecciones aprendidas y las mejores prácticas identificadas durante su aplicación, se consideró importante adicionar una actividad en la fase de Ejecución, la cual consiste en realizar una reunión entre el equipo de desarrollo y el equipo encargado de la ejecución de las pruebas, con el fin de aclarar inquietudes con respecto a los bugs reportados.

10. DISCUSIÓN DE RESULTADOS

Los resultados obtenidos en este proyecto permiten evidenciar los logros alcanzados en cada una de las etapas de la metodología propuesta. Inicialmente, se planteó la elaboración de un plan de recolección y análisis de la información con el fin de caracterizar las prácticas relacionadas con las pruebas de software en las PYMES Desarrolladoras de Software de la ciudad de Pereira. Posteriormente, se formuló un proceso de pruebas de software, teniendo en cuenta los resultados obtenidos en el análisis de la encuesta aplicada a las PYMES desarrolladoras de software de la ciudad de Pereira. Después, se llevó a cabo la validación correspondiente mediante su aplicación en dos proyectos de software. Por último, se realizó un análisis comparativo de los resultados obtenidos en los proyectos seleccionados para la aplicación del proceso planteado.

Al analizar los resultados de la caracterización realizada en torno a las prácticas relacionadas con pruebas de software en las PYMES de la ciudad de Pereira, se destaca que el 70% de ellas utiliza un proceso o metodología para el desarrollo de las pruebas, mientras que un 60% posee una certificación de calidad.

En lo que respecta a la formación del personal que lleva a cabo las pruebas, se pudo constatar que tan solo el 30% de las empresas que hicieron parte del estudio, cuentan con personal certificado en la ejecución de pruebas de software, situación que pone en evidencia la necesidad de formación que tiene la región en este tema.

A la hora de evaluar la efectividad de las pruebas, el 100% de las empresas encuestadas manifestó que es importante medir la efectividad de las pruebas, sin embargo, es preocupante que tan solo el 30% de ellas emplea métricas para tal fin.

También llama la atención el hecho de evidenciar la utilización de marcos de trabajo ágiles como SCRUM, razón por la cual el 66,7% de las empresas cuenta con varios equipos de desarrollo dentro de los cuales distribuye el rol de pruebas.

De igual manera, después de ejecutar el proceso propuesto, se identificaron falencias en la forma en que se lleva a cabo la especificación de requerimientos, situación que evidencia la necesidad de aplicar al interior de las organizaciones un procedimiento para especificar y validar los requisitos del software.

Otro aspecto a destacar después de haber realizado la ejecución del proceso, es el hecho de que los formatos propuestos como evidencia resultante de la ejecución de cada una de las fases, fueron diligenciados en su totalidad en cada uno de los pilotos realizados y sirvieron de referente a la hora de evaluar el estado de ejecución de las pruebas.

En lo que a propuestas de mejora se refiere, se resalta que fue necesario adicionar una actividad en la fase de ejecución del proceso propuesto, con el fin de mejorar la comunicación entre el equipo de desarrollo y el equipo encargado de la ejecución de las pruebas. Sin embargo, se considera que el proceso es entendible y fácil de aplicar, siempre y cuando se tengan especificados correctamente los requerimientos del sistema.

11. CONCLUSIONES

El trabajo realizado con el fin de construir un proceso de testing funcional para las MIPYMES desarrolladoras de software de la ciudad de Pereira, permitió establecer las siguientes conclusiones:

- La encuesta realizada a la muestra por conveniencia permitió conocer aspectos relevantes sobre el proceso de pruebas que actualmente aplican las empresas desarrolladoras de software de la ciudad de Pereira, lo cual sirvió de base para llevar a cabo el planteamiento del proceso objeto de esta investigación. Sin embargo, se considera que dicha información puede servir como referente para futuros estudios.
- Se llevó a cabo el proceso de caracterización de las prácticas relacionadas con las pruebas de software en las MIPYMES Desarrolladoras de Software de la ciudad de Pereira, mediante la aplicación de una encuesta, con la cual se pudo evidenciar que en el ámbito local existen varias empresas aunando esfuerzos para mejorar sus procesos internos de pruebas de software, con el fin de aumentar la calidad del software producido.
- Se logró validar el proceso propuesto mediante la aplicación de dos pilotos en dos proyectos elaborados por una MIPYME desarrolladora de software de la ciudad de Pereira, obteniendo como resultado indicadores favorables y representativos a la hora de medir la calidad del software producido.
- La aplicación del procedimiento en una de las empresas desarrolladoras de software de la ciudad de Pereira, contribuyó en el fortalecimiento del sector productivo y la academia. De igual manera, sirvió en gran medida para que los equipos de desarrollo que fueron partícipes de la validación, conocieran nuevas técnicas y/o alternativas para llevar a cabo el proceso de pruebas de software.

- Con la aplicación del procedimiento, también se evidenció que la especificación de requerimientos juega un papel fundamental a la hora de planear y ejecutar las pruebas de software, ya que de su claridad y fácil entendimiento, depende que el equipo de pruebas escoja las técnicas adecuadas de validación y diseñe los escenarios de prueba correctos.

12. RECOMENDACIONES

- Elaborar especificaciones de requerimientos que sean claras, concisas y que correspondan en gran medida a las necesidades manifestadas por el usuario final de la solución implementada.
- Se considera pertinente que las empresas desarrolladoras de software tengan ambientes de prueba independientes del entorno de desarrollo, que simulen en gran medida al ambiente productivo.
- Fortalecer los procesos de formación en torno a pruebas de software del personal encargado de ejecutar dicha labor al interior de las organizaciones.
- Antes de liberar el software en el ambiente productivo, se recomienda realizar pruebas de aceptación con los usuarios finales del mismo.
- Se considera importante mejorar el proceso propuesto para tenga en cuenta la evaluación de aspectos no funcionales del software.
- Como trabajo futuro, se recomienda la elaboración de procedimientos que planteen la automatización de casos de prueba y la implementación de pruebas ágiles al interior de las empresas desarrolladoras de software de la ciudad de Pereira.

13. RESULTADOS EN GENERACIÓN DE CONOCIMIENTO

A continuación, se relacionan los resultados y los productos obtenidos con la realización del presente proyecto de investigación:

Tabla 15: Resultados y productos esperados

ASPECTO	RESULTADO	INDICADOR	BENEFICIARIO
GENERACIÓN DE CONOCIMIENTO	Caracterización de las prácticas de pruebas de software empleadas por las PYMES desarrolladoras de software de la ciudad de Pereira	Informe de caracterización elaborado.	<ul style="list-style-type: none"> ▪ Empresas de Desarrollo de Software. ▪ Grupos de Investigación. ▪ Comunidad Académica.
	Artículo de Investigación	Artículo de investigación elaborado.	<ul style="list-style-type: none"> ▪ Grupos de Investigación. ▪ Comunidad Académica. ▪ Gremios del Sector del Software.
	Proceso de pruebas de software para ser ejecutado de manera independiente.	Proceso elaborado.	<ul style="list-style-type: none"> ▪ Grupos de Investigación. ▪ Comunidad Académica. ▪ Gremios del Sector del Software.
APLICACIÓN DE CONOCIMIENTO	Aplicación de un nuevo proceso de pruebas de software.	Aplicación del proceso en dos proyectos elaborados por una empresa desarrolladora de software de la ciudad de Pereira.	<ul style="list-style-type: none"> ▪ Empresa de Desarrollo de Software. ▪ Grupos de Investigación.
APORTE SOCIAL	Proceso base para Constituir una Empresa dedicada a la ejecución de pruebas en la Ciudad de Pereira.	Proceso de pruebas aplicado y validado.	<ul style="list-style-type: none"> ▪ Empresas de Desarrollo de Software. ▪ Gremios del sector de Software.

14. BIBLIOGRAFÍA

- Black, R., & Rueda Sandoval, G. (2011). *Fundamentos de pruebas de software*. Santa Cruz, Bolivia: RBCS, Inc.
- Capote García, T., Brito Riverol, Y., Yzquierdo Herrera, R., & Febles Estrada, A. (Octubre - Diciembre de 2014). *Scielo Cuba*. Recuperado el 01 de Marzo de 2017, de <http://scielo.sld.cu/pdf/rcci/v8n4/rcci09414.pdf>
- CMMI Institute*. (2018). Recuperado el 22 de Noviembre de 2018, de <https://cmmiinstitute.com/>
- DANE. (29 de Diciembre de 2016). *Departamento Administrativo Nacional de Estadística*. Recuperado el 20 de Marzo de 2017, de http://www.dane.gov.co/files/investigaciones/boletines/tic/bol_empresas_2015.pdf
- Escobar Sánchez, M. E., & Fuertes Díaz, W. M. (Mayo - Agosto de 2015). Modelo formal de pruebas funcionales de software para alcanzar el Nivel de Madurez Integrado 2. *Revista Facultad de Ingeniería*, 24(39), 31-42.
- FEDESOF. (2015). *Informe de Caracterización del Sector Software y Tecnologías de la Información en Colombia*.
- García P, M., Quispe A, C., & Ráez G, L. (1 de Agosto de 2003). Mejora continua de la calidad en los procesos. *Industrial Data*, 6(1), 89-94. Recuperado el 22 de Noviembre de 2018, de <http://www.redalyc.org/pdf/816/81606112.pdf>
- Graham, D., Van Veenendaal, E., Evans, I., & Black, R. (2009). *Universitatea Tehnica din Cluj-Napoca*. Obtenido de https://www.utcluj.ro/media/page_document/78/Foundations%20of%20software%20testing%20-%20ISTQB%20Certification.pdf
- IEEE Computer Society. (2006). *IEEE Standard Dictionary of Measures of the Software Aspects of Dependability*. New York, NY: the Institute of Electrical and Electronics Engineers.
- IEEE Computer Society. (2014). *SWEBOK V3.0 Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society.
- IEEE Computer Society. (2014). *SWEBOK V3.0 Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society.

IEEE Computer Society. (2014). *SWEBOK V3.0 Guide to the Software Engineering Body of Knowledge*. Recuperado el 18 de Enero de 2018

Illinois Institute of Technology. (s.f.). *Illinois Institute of Technology*. Recuperado el 14 de Septiembre de 2017, de <https://science.iit.edu/computer-science/research/testing-maturity-model-tmm>

Institute of Electrical and Electronics Engineers IEEE. (2002). *IEEE Standard Glossary Of Software Engineering Terminology - Std 610.12-1990(R2002)* (1990 ed.). New York.

ISO. (Septiembre de 2013). *International Organization for Standardization*. Recuperado el 18 de Septiembre de 2017, de <https://www.iso.org/standard/45142.html>

ISO. (Septiembre de 2013). *International Organization for Standardization*. Recuperado el 18 de Septiembre de 2017, de <https://www.iso.org/standard/56736.html>

ISO. (Septiembre de 2013). *International Organization for Standardization*. Recuperado el 18 de Septiembre de 2017, de <https://www.iso.org/standard/56737.html>

ISO. (Diciembre de 2015). *International Organization for Standardization*. Obtenido de <https://www.iso.org/standard/60245.html>

ISO. (2017). *International Organization for Standardization*. Recuperado el 25 de Junio de 2017, de <https://www.iso.org/about-us.html>

ISO. (2017). *International Organization for Standardization*. Recuperado el 29 de Septiembre de 2018, de <https://www.iso.org/search/x/query/29119>

ISTQB. (2011). *ISQTB International Software Testing Qualifications Board*. Recuperado el 22 de Noviembre de 2018, de http://www.sstqb.es/ficheros/sstqb_file95-637831.pdf

ISTQB. (16 de Agosto de 2017). *Glosario estándar de términos utilizados en pruebas de software*. Recuperado el 22 de Noviembre de 2018, de http://www.sstqb.es/ficheros/sstqb_file94-445368.pdf

ISTQB International Software Testing Qualifications Board. (31 de Marzo de 2011). *SSTQB Spanish Software Testing Qualification Board*. Obtenido de http://www.sstqb.es/ficheros/sstqb_file95-637831.pdf

IT Mark. (2018). Recuperado el 22 de Noviembre de 2018, de http://it-mark.eu/wordpress/?page_id=64&lang=es

- Martínez Marín, S. J., Arango Aramburo, S., & Robledo Velásquez, J. (2015). el crecimiento de la industria del software en Colombia: un análisis sistémico. *Revista EIA, ISSN 1794-1237 / Año XII / Volumen 12 / Edición N.23*, 95-116.
- Menéndez Domínguez, V. H., & Castellanos Bolaños, M. E. (2015). SPEM: Software Process Engineering Metamodel. *Revista Latinoamericana de Ingeniería del Software*, 3(2), 92-100.
- Mera Paz, J. (Octubre de 2016). Análisis del proceso de pruebas de calidad de software. *Ingeniería Solidaria*, 12(20), 163-176. Recuperado el 21 de Noviembre de 2018, de <https://revistas.ucc.edu.co/index.php/in/article/view/1482>
- MinTic - FEDESOFTE. (2015). *Observatorio TI*. Recuperado el 02 de Marzo de 2017, de <http://observatorioti.co/wp-content/uploads/2016/02/EstudiodeCaracterizaci%C3%B3n2015.pdf>
- Observatorio de la Industria TI. (2016). *Observatorio de la Industria TI*. Recuperado el 20 de Marzo de 2017, de http://observatorioti.co/k_course/representatividad-del-cluster-en-la-region/
- Observatorio de la Industria TI. (Diciembre de 2016). *Observatorio TI*. Recuperado el 18 de Marzo de 2017, de http://observatorioti.co/k_course/estudio-de-salarios-2016/
- Observatorio de Tecnologías de Información MinTIC. (2015). *CARACTERIZACIÓN DE PRODUCTOS Y SERVICIOS DEL SECTOR DE TECNOLOGIAS DE INFORMACIÓN EN COLOMBIA 2015*. Obtenido de https://www.observatorioti.gov.co/uploads/files/71_e9c52991-061b-4f30-b01c-78f9c828a83b.pdf
- PDCA Home. (s.f.). Recuperado el 22 de Noviembre de 2018, de <https://www.pdcahome.com/5202/CICLO-PDCA/>
- Peláez, L. E., & Toro Lazo, A. (2016). Ingeniería de Requisitos: de la especificación de requisitos de software al aseguramiento de la calidad. Cómo lo hacen las Mipymes desarrolladoras de software de la ciudad de Pereira. *Entre Ciencia e Ingeniería*(20), 117-123. Recuperado el 24 de Noviembre de 2018, de <http://www.scielo.org.co/pdf/ecei/v10n20/v10n20a16.pdf>

- Pérez Lamancha, B. (2006). *Proceso de Testing Funcional Independiente - (Tesis de Maestría)*. Centro de Ensayos de Software (CES). Recuperado el 30 de Mayo de 2016, de http://www.ces.com.uy/documentos/imasd/Tesis-Beatriz_Perez_2006.pdf
- Pressman, R. (2010). *Ingeniería del Software un Enfoque Práctico Séptima Edición*. México: McGrawHill.
- Ramírez Aguirre, P., & Ramírez Arias, C. (2010). *Repositorio Universidad Tecnológica de Pereira*. Recuperado el 29 de Enero de 2018, de repositorio.utp.edu.co/dspace/bitstream/11059/1977/1/0053R173e.pdf
- Rojas Montes, M. L., Pino Correa, F. J., & Martínez, J. M. (2015). Proceso de pruebas para pequeñas organizaciones de software. *Revista Facultad de Ingeniería*, 24(39), 55-70.
- Sánchez Melchor, S. (2010). *Universidad Rey Juan Carlos*. Recuperado el 18 de Septiembre de 2017, de <https://eciencia.urjc.es/bitstream/handle/10115/4115/Una%20revisi%C3%B3n%20y%20comparativa...%20Sandra%20S%C3%A1nchez%20Melchor.pdf?sequence=1&isAllowed=y>
- Sanz Esteban, A. (2012). *Marco Metodológico para la mejora de las actividades de Verificación y Validación de Productos Software (Tesis Doctoral)*. Universidad Carlos III de Madrid. Recuperado el 01 de Marzo de 2017, de http://e-archivo.uc3m.es/bitstream/handle/10016/15464/tesis_ana_sanz_esteban_2012.pdf;jsessionid=02072170894DD7EEC47807548B676217?sequence=1
- Sanz Esteban, A. (2012). *Universidad Carlos III de Madrid*. Recuperado el 01 de Marzo de 2017, de http://e-archivo.uc3m.es/bitstream/handle/10016/15464/tesis_ana_sanz_esteban_2012.pdf;jsessionid=02072170894DD7EEC47807548B676217?sequence=1
- Senado de la República de Colombia. (02 de Agosto de 2004). *Alcandía de Bogotá*. Recuperado el 25 de Marzo de 2017, de <http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=14501>
- Softex. (2018). Recuperado el 22 de Noviembre de 2018, de <https://www.softex.br/mpsbr/avaliacoes/>
- Software Testing Standard. (2013). *Software Testing Standard*. Recuperado el 18 de Septiembre de 2017, de <http://www.softwaretestingstandard.org/part3.php>

- Software Testing Standard. (Diciembre de 2014). *softwaretestingstandard.org*. Obtenido de <http://www.softwaretestingstandard.org/part5.php>
- SOGETI. (Junio de 2009). *SOGETI*. Obtenido de https://www.sogeti.es/globalassets/spain/soluciones/testing/tmap_brochure-.pdf
- SOGETI. (2017). *SOGETI*. Recuperado el 18 de Septiembre de 2017, de <https://www.sogeti.es/soluciones/calidad-de-software/metodologias-de-pruebas/tmap/>
- Sogeti. (s.f.). *Sogeti Hightech*. Recuperado el 27 de Septiembre de 2018, de <https://www.sogeti-hightech.es/sobre-sogeti-hightech/Grupo-sogeti/>
- Sommerville, I. (2005). *Ingeniería del Software* (Séptima ed.). Madrid: Pearson Education S.A.
- Spanish Software Testing Qualifications Board (SSTQB). (12 de Septiembre de 2016). *Spanish Software Testing Qualifications Board (SSTQB)*. Recuperado el 17 de Julio de 2016, de http://www.sstqb.es/ficheros/sstqb_file94-53f0b3.pdf
- SSTQB Spanish Software Testing Qualification Board. (13 de Marzo de 2017). *SSTQB Spanish Software Testing Qualification Board*. Obtenido de <http://www.sstqb.es/acerca-de-sstqb/que-es-sstqb.html>
- Tassey, G. (2002). *The Economic Impacts of Inadequate Infrastructure for Software Testing*. National Institute of Standards and Technology (NIST) , Gaithersburg. Recuperado el 21 de Noviembre de 2018, de <https://www.nist.gov/sites/default/files/documents/director/planning/report02-3.pdf>
- TMMi Foundation. (2015). *TMMi*. Recuperado el 25 de Junio de 2017, de <https://www.tmmi.org/wp-content/uploads/2016/09/TMMi-Framework-R1.0-Spanish.pdf>
- Wautelet, Y., Kolp, M., & Achbany, Y. (2006). *UCL Université Catholique de Louvain*. Recuperado el 29 de Diciembre de 2017, de http://www.isys.ucl.ac.be/staff/youssef/Articles/WP_SPEM.pdf

15. ANEXOS

ANEXO A: PRESUPUESTO

A continuación, se relacionan todos los ítems que hacen parte del presupuesto requerido para llevar a cabo la realización del proyecto.

1) Presupuesto Global

RUBRO	VALOR
Personal	53.748.600
Materiales	2.183.000
Viajes	220.000
Salidas de Campo	450.400
TOTAL	\$ 56.602.000

Descripción de Gastos de Personal

PERSONAL	FORMACIÓN ACADÉMICA	DEDICACIÓN EN MESES	VALOR	CANTIDAD	VALOR TOTAL
Investigador principal	Ingeniero de Sistemas y Telecomunicaciones	15	1.000.000	1	15.000.000
Asesor del proyecto	Ingeniero de Sistemas	15	2.500.000	1	37.500.000
Pares evaluadores		2	624.300	1	1.248.600
TOTAL					\$ 53.748.600

Descripción de materiales, suministros y útiles de oficina

MATERIAL	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Argollados	4	6.000	24.000
Computadores	1	1.500.000	1.500.000
Empastado	1	12.000	12.000
Fotocopias	100	50	5.000
Hojas Digitadas	180	150	27.000
Papelería y útiles de oficina			15.000
Uso de internet (Horas)	500	1.200	600.000
TOTAL			\$ 2.183.000

Descripción y justificación de los viajes

LUGAR	JUSTIFICACIÓN	CANTIDAD	VALOR DEL DESPLAZAMIENTO	VALOR TOTAL
UAM Manizales	Presentación del proyecto para retroalimentación en Talleres de Línea.	4	20.000	80.000
UAM Manizales	Asistencia a asesorías	6	20.000	120.000
UAM Manizales	Sustentación pública del proyecto.	1	20.000	20.000
TOTAL				\$ 220.000

Relación de salidas de Campo

CONCEPTO	CANTIDAD	VALOR	VALOR TOTAL
Aplicación de encuestas para el análisis inicial.	6	3.600	21.600
Aplicación del proceso en dos empresas.	8	3.600	28.800
Número de horas invertidas en las salidas de campo	40	10.000	400.000
TOTAL			\$ 450.400

ANEXO B: ENCUESTA APLICADA A EMPRESAS DE LA CIUDAD DE PEREIRA



INSTRUMENTO PARA CARACTERIZAR LA MANERA COMO LAS MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA ESTÁN LLEVANDO A CABO EL PROCESO DE PRUEBAS

Objetivo: Caracterizar las prácticas del proceso de Pruebas de Software en las Empresas Desarrolladoras de Software de la ciudad de Pereira.

Señor empresario/productor de software, el presente instrumento de levantamiento de información obedece al interés que se tiene en mejorar las buenas prácticas que la industria local del software lleva actualmente a cabo alrededor del proceso de pruebas. Por lo anterior, acudimos a su voluntad, para que entre todos encontremos la forma de diagnosticar, proponer y describir mejoras en pro de tener un proceso de pruebas de software más eficiente y eficaz en las Empresas Desarrolladoras de Software de la Ciudad de Pereira.

Marque por favor con una X la respuesta que considere correcta para su empresa:

I. GENERALIDADES DE LA ORGANIZACIÓN

1. ¿Cuántos años de creación tiene la empresa?

a. 1 a 5 años

b. 5 a 10 años

c. 10 años o más

2. ¿Cuántos empleados tiene la empresa?

- a. 1 a 10 b. 11 a 20 c. 21 a 30 d. 31 a 40
- e. 41 a 50 f. 51 a 60 g. 61 o más

3. ¿Cuál es la principal actividad de la empresa?

- a. Desarrollo b. Mantenimiento c. Pruebas de d. Otro
- de Software de Software Software

¿Cuál? _____

4. ¿Qué Metodología, Modelo o Marco de Trabajo de Desarrollo de Software utiliza la Empresa?

5. ¿La Empresa tiene certificaciones de calidad?

- a. Si b. No

¿Cuáles?

6. ¿La Empresa administra las configuraciones de software?

- a. Si b. No

7. ¿La Empresa administra las versiones de software?

- a. Si b. No

8. ¿Qué técnica utilizan para llevar a cabo la especificación del software?

(Pregunta con múltiple respuesta).

- a. Casos de uso
 - b. Historias de usuario
 - c. Prototipado de interfaz
 - d. Diagramas de flujo
 - e. Diagramas de secuencia
 - f. Diagramas de actividades
 - g. Otro. ¿Cuál?
-
-

II. GENERALIDADES DEL PROCESO DE PRUEBAS

9. ¿La empresa utiliza un proceso o metodología para el desarrollo de las pruebas de software?

a. Si
¿Cuál(es)?

b. No
¿Por qué?

10. ¿La empresa cuenta con un área especializada en la realización de pruebas de software?

a. Si

b. No

- c. Usabilidad
 - d. Eficiencia
 - e. Mantenibilidad
 - f. Portabilidad
 - g. Calidad de Uso
 - h. Otro. ¿Cuál?
-
-

15. ¿Qué tipo de pruebas realiza la empresa?

(Pregunta con múltiple respuesta).

- a. Pruebas funcionales
- b. Pruebas unitarias
- c. Pruebas de humo
- d. Pruebas de regresión
- e. Pruebas de estrés
- f. Pruebas de rendimiento
- g. Pruebas de integración
- h. Pruebas de aceptación
- i. Otra. ¿Cuál? _____

16. ¿La empresa cuenta con un ambiente de pruebas independiente del entorno de desarrollo?

a. Si

b. No

¿Por qué?

III. PLANEACIÓN DE LAS PRUEBAS

17. ¿La Empresa planifica las pruebas de software y asigna los recursos necesarios para su ejecución?

a. Si

b. No

Si la respuesta es “Si”, responda por favor las siguientes preguntas:

18. ¿En el Plan de Pruebas se establece la duración del ciclo de pruebas?

a. Si

b. No

19. ¿Las pruebas se planifican desde el inicio del proyecto?

a. Si

b. No

20. ¿En el Plan se establece qué atributos de calidad se considerarán en las pruebas?

a. Si

b. No

21. ¿En el Plan se establece qué tipos de pruebas se realizarán?

a. Si

b. No

IV. DISEÑO DE LAS PRUEBAS

22. ¿Se emplean Técnicas de Diseño de Casos de Prueba?

a. Si

b. No

Si la respuesta es “Si”, responda por favor la siguiente pregunta:

23. ¿Qué técnicas utiliza?

- a. Basadas en la especificación (Partición de equivalencias, análisis de valores límite, tablas de decisión, diagramas de transición de estados).
- b. Basadas en la estructura (Pruebas de caja blanca)
- c. Basadas en la experiencia
- d. Basadas en el uso

24. ¿Socializa los casos de prueba con el cliente para que éste lleve a cabo su aprobación y/o haga observaciones al respecto?

a. Si

b. No

¿Por qué?

V. ESTIMACIÓN DE COSTOS

25. ¿La empresa registra el esfuerzo invertido y los costos de las pruebas?

a. Si

b. No

26. ¿La empresa presupuesta la fase de pruebas del software en un proyecto? Es decir, ¿incluye en el presupuesto un rubro para llevar a cabo el proceso de pruebas?

a. Si

b. No

27. ¿Considera que el proceso de pruebas es demasiado costoso para la organización?

a. Si

b. No

- b. Selenium
- c. QAManager
- d. Otra. ¿Cuál? _____

34. ¿Qué herramientas emplea la organización para gestionar los defectos encontrados en las pruebas?

- a. Mantis Bug Tracker
- b. Bugzila
- c. Taiga
- d. Otra. ¿Cuál? _____

VIII. OPORTUNIDADES DE MEJORA DEL PROCESO DE PRUEBAS

35. ¿Considera usted que el proceso de pruebas de software debe ser mejorado en su empresa?

- a. Si
- b. No

36. ¿Considera usted que el hecho de contar con talento humano especializado en pruebas mejoraría la calidad del software desarrollado por la empresa?

- a. Si
- b. No

37. ¿Considera usted que el hecho de contar con un área de pruebas en su empresa representa o representaría un gran gasto para su organización?

- c. Si
- d. No

38. ¿Cuáles considera usted que son los elementos que se deben incluir en un proceso de pruebas cuya ejecución sea independiente del proceso de desarrollo?

39. ¿Si existiera en la ciudad una empresa dedicada al Testing de Software, consideraría contratar dicho servicio?

a. Si


b. No

¿Por qué?

ANEXO C: PROCESO DE TESTING FUNCIONAL DE SOFTWARE PARA LAS MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA – “QUALITEST”.



**PROCESO DE TESTING FUNCIONAL
DE SOFTWARE PARA LAS MIPYMES
DESARROLLADORAS DE SOFTWARE DE
LA CIUDAD DE PEREIRA**

Título:	PROCESO DE TESTING FUNCIONAL DE SOFTWARE PARA LAS PYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA	
Descripción:	<i>Documento que describe el proceso mediante el cual se puede llevar a cabo la ejecución de pruebas funcionales de software de manera independiente al proceso de desarrollo de una Organización</i>	
Responsables:	<i>Líder del proceso de pruebas Ingenieros de Pruebas</i>	
Precondiciones:	<i>Haber realizado la especificación de requerimientos del software que se desea probar.</i>	
Entradas:	<i>Documento de especificación de requerimientos</i>	

ITEM	ACTIVIDAD	ENTRADAS	DESCRIPCIÓN	RESPONSABLE	DOCUMENTO O REGISTRO
1	ANÁLISIS INICIAL				
1.1	Reunión inicial	Documento de Especificación de Requerimientos	Reunión inicial para realizar la revisión previa de los requerimientos que constituyen las funcionalidades que se van a probar.	Líder del proceso de pruebas	AI1 – Acta de Reunión
1.2	Revisión de Requerimientos		Revisión del documento de Especificación de Requerimiento para entender la lógica de cada una de las funcionalidades que harán parte de la prueba.	Líder del proceso de pruebas Ingeniero de Pruebas	AI2 - Relación de artefactos a probar
1.3	Aclaración de inquietudes sobre la Especificación de Requerimientos.		En caso de que se presenten dudas relacionadas con la Especificación de Requerimientos, el Líder del Pruebas solicita reunión con el equipo de desarrollo para hacer las aclaraciones pertinentes.	Líder del proceso de pruebas	AI1 - Acta de Reunión
1.4	Identificación de herramientas requeridas		Se identifican las herramientas requeridas para llevar a cabo la ejecución de las pruebas.	Líder del proceso de pruebas Ingeniero de Pruebas	AI3 – Lista de herramientas requeridas
1.5	Estimación de tiempos de prueba		Se realiza la estimación de los tiempos requeridos para la ejecución de las fases que hacen parte del proceso de pruebas.	Líder del proceso de pruebas Ingeniero de Pruebas	AI4 – Formato de Estimación de Tiempos

ITEM	ACTIVIDAD	ENTRADAS	DESCRIPCIÓN	RESPONSABLE	DOCUMENTO O REGISTRO
2 PLANEACIÓN					
2.1	Identificación de riesgos	Documento de Especificación de Requerimientos AI2 - Relación de Artefactos a probar	Tomando como base la Especificación de Requerimientos, se analizan los riesgos de cada funcionalidad con el fin de establecer prioridades y definir la estrategia de prueba.	Líder del proceso de pruebas Ingenieros de Pruebas	AI2 - Relación de artefactos a probar con registro de riesgos identificados.
2.2	Elaboración del Plan de Pruebas	AI4 - Formato de Estimación de Tiempos	Elaboración del Plan de Pruebas correspondiente determinando objetivo, alcance y estrategia de pruebas requerida.	Líder del proceso de pruebas Ingenieros de Pruebas	P1 – Plan de pruebas
2.3	Aprobación del Plan de Pruebas		El líder del proceso de desarrollo lleva a cabo la revisión y aprobación del Plan de Pruebas.	Líder del Proceso de Desarrollo.	P1 – Plan de pruebas (Con registro de aprobación)
3 DISEÑO					
3.1	Elaboración de casos de prueba	Documento de Especificación de Requerimientos P1 – Plan de pruebas	Elaboración de casos de prueba para cada una de las funcionalidades que hacen parte del alcance.	Líder del proceso de pruebas Ingeniero de Pruebas	D1 – Formato de caso de prueba
3.2	Revisión de los casos de prueba	D1 – Formato de caso de prueba	El equipo de desarrollo revisa los casos de prueba elaborados y en caso de que tenga observaciones al respecto las consigna en el check list de revisión.	Ingeniero de Desarrollo Líder del Proceso de Desarrollo	D2 – Check list de revisión de caso de prueba
3.3	Ajuste de casos de prueba	D2 – Check list de revisión de caso de prueba	El equipo de pruebas lleva a cabo el ajuste de los casos de prueba tomando como base las observaciones y/o sugerencias realizadas por el Equipo de Desarrollo.	Líder del proceso de pruebas Ingeniero de Pruebas	D1 – Formato de caso de prueba (Actualizado)
3.4	Aprobación de casos de prueba	D1 – Formato de caso de prueba	El equipo de desarrollo lleva a cabo la aprobación de los casos de prueba.	Ingeniero de Desarrollo Líder del Proceso de Desarrollo	D1 – Formato de caso de prueba (Con registro de aprobación)

ITEM	ACTIVIDAD	ENTRADAS	DESCRIPCIÓN	RESPONSABLE	DOCUMENTO O REGISTRO
4	EJECUCIÓN				
4.1	Configuración del entorno de prueba	Documento de configuración del entorno de pruebas entregado por el Equipo de Desarrollo.	El equipo de pruebas lleva a cabo la configuración requerida en el ambiente de pruebas asignado.	Equipo de Pruebas	Entorno de prueba configurado
4.2	Ejecución de prueba de humo		Ejecución de Prueba de Humo con el fin de garantizar la estabilidad del ambiente de pruebas.	Ingeniero de Pruebas	E1 – Check list de Prueba de Humo
4.3	Ejecución de caso de prueba	D1 – Formato de caso de prueba	El ingeniero de pruebas asignado lleva a cabo la ejecución del Diseño de Casos de Prueba.	Ingeniero de Pruebas	D1 – Formato de caso de prueba (Con registro de las corrida por ciclo de prueba)
4.4	Reporte de Bugs		El ingeniero de pruebas lleva a cabo el reporte de bugs en la herramienta definida para tal fin.	Ingeniero de Pruebas	E2 - Reporte de incidencia en la herramienta definida
4.5	Recibir solución de Bugs	Notificación de nueva versión generada	El equipo de pruebas lleva a cabo la configuración del ambiente de pruebas en caso de ser requerida.	Ingeniero de Pruebas	Entorno de prueba configurado
4.6	Revisión de Bugs corregidos	Notificación de bug corregido en la herramienta definida	El ingeniero de pruebas lleva a cabo la revisión de los bugs corregidos y actualiza el estado de los mismos en la herramienta definida para el reporte de bugs.	Ingeniero de Pruebas	E2 – Reporte de incidencia (Modificación del estado de la incidencia en la herramienta definida).
4.7	Monitoreo del Proyecto		El Líder del Proyecto de Pruebas elabora semanalmente el Reporte de Monitoreo del Proyecto	Líder del Proceso de Pruebas	E3 – Informe de estado de las pruebas
4.8	Pruebas de regresión	D1 – Formato de caso de prueba	Una vez se encuentren ejecutados exitosamente todos los casos de prueba, el equipo de pruebas procede a hacer la revisión de cada uno de los bugs generados en la versión con el fin de garantizar que todos se encuentran corregidos.	Equipo de Pruebas	E2 – Reporte de incidencia (Registro de verificación de cada uno de los bugs en la herramienta definida).
4.9	Pruebas de aceptación con el usuario final (Actividad opcional)		En caso de que se cuente con la disponibilidad de tiempo del usuario final del sistema, el equipo de pruebas llevará a cabo la presentación de las funcionalidades probadas para que éste verifique su funcionamiento y lleve a cabo la aceptación del desarrollo realizado.	Equipo de Pruebas	E4 – Carta de aceptación E5 - Encuesta de satisfacción

ITEM	ACTIVIDAD	ENTRADAS	DESCRIPCIÓN	RESPONSABLE	DOCUMENTO O REGISTRO
5	CIERRE				
5.1	Elaborar informe de finalización de las pruebas	E2 – Reporte de incidencia E3 – Carta de aceptación E4 - Encuesta de satisfacción	El líder del proyecto de pruebas lleva a cabo la elaboración del Informe de Finalización de pruebas	Líder del Proceso de Pruebas	C1 – Informe de finalización de pruebas
5.2	Identificar lecciones aprendidas y mejores prácticas	C1 – Informe de finalización de pruebas	Se lleva a cabo una reunión entre los equipos de Desarrollo y Pruebas con el fin de identificar lecciones aprendidas y mejores prácticas que podrán ser aplicadas en proyectos futuros.	Equipo de Desarrollo Equipo de Pruebas	C2 – Registro de lecciones aprendidas y mejores prácticas.

ANEXO D: FORMATO DE ENCUESTA APLICADO A LA EMPRESA EN LA QUE SE LLEVÓ A CABO LA APLICACIÓN DEL PROCESO.



INSTRUMENTO PARA CARACTERIZAR LA MANERA COMO LA EMPRESA ESTÁ LLEVANDO A CABO EL PROCESO DE PRUEBAS

Objetivo: Caracterizar las prácticas del proceso de Pruebas de Software en la Empresa.

Marque por favor con una X la respuesta que considere correcta para su empresa:

I. GENERALIDADES DE LA ORGANIZACIÓN

1. ¿Qué Metodología, Modelo o Marco de Trabajo de Desarrollo de Software utiliza la Empresa?

2. ¿La Empresa tiene certificaciones de calidad?

a. Si

b. No

¿Cuáles?

3. ¿La Empresa administra las configuraciones de software?

a. Si

b. No

4. ¿La Empresa administra las versiones de software?

a. Si

b. No

5. ¿Qué técnica utilizan para llevar a cabo la especificación del software?

(Pregunta con múltiple respuesta).

- a. Casos de uso
- b. Historias de usuario
- c. Prototipado de interfaz
- d. Diagramas de flujo
- e. Diagramas de secuencia
- f. Diagramas de actividades
- g. Otro. ¿Cuál?

II. GENERALIDADES DEL PROCESO DE PRUEBAS

6. ¿La empresa utiliza un proceso o metodología para el desarrollo de las pruebas de software?

a. Si
¿Cuál(es)?

b. No
¿Por qué?

7. ¿La empresa cuenta con un área especializada en la realización de pruebas de software?

a. Si

b. No

Si la respuesta es “No”, responda por favor la siguiente pregunta:

8. ¿Cómo se llevan a cabo las pruebas de software en la organización?

- a. Las personas que desarrollan son las mismas que llevan a cabo las pruebas.
- b. La empresa cuenta con varios equipos de desarrollo dentro de los cuales se distribuye el rol de pruebas, teniendo como principio que quien prueba no sea la misma persona que desarrolla.
- c. Terceriza las pruebas de software.
- d. No se realiza un proceso de pruebas formal. La validación se lleva a cabo en el ambiente productivo.
- e. Otro. ¿Cuál?

9. ¿Cuántas personas conforman el Área de Pruebas?

a. 1 a 5

b. 6 a 10

c. 11 a 20

d. Más de 20

10. ¿Qué características o atributos de calidad tienen en cuenta a la hora de realizar las pruebas de un artefacto?

(Pregunta con múltiple respuesta).

a. Funcionalidad

b. Fiabilidad

- c. Usabilidad
- d. Eficiencia
- e. Mantenibilidad
- f. Portabilidad
- g. Calidad de Uso
- h. Otro. ¿Cuál?

11. ¿Qué tipo de pruebas realiza la empresa?

(Pregunta con múltiple respuesta).

- a. Pruebas funcionales
- b. Pruebas unitarias
- c. Pruebas de humo
- d. Pruebas de regresión
- e. Pruebas de estrés
- f. Pruebas de rendimiento
- g. Pruebas de integración
- h. Pruebas de aceptación
- i. Otra. ¿Cuál? _____

12. ¿La empresa cuenta con un ambiente de pruebas independiente del entorno de desarrollo?

a. Si

b. No

¿Por qué?

III. PLANEACIÓN DE LAS PRUEBAS

13. ¿La Empresa planifica las pruebas de software y asigna los recursos necesarios para su ejecución?

a. Si

b. No

Si la respuesta es “Si”, responda por favor las siguientes preguntas:

14. ¿En el Plan de Pruebas se establece la duración del ciclo de pruebas?

a. Si

b. No

15. ¿Las pruebas se planifican desde el inicio del proyecto?

a. Si

b. No

16. ¿En el Plan se establece qué atributos de calidad se considerarán en las pruebas?

a. Si

b. No

17. ¿En el Plan se establece qué tipos de pruebas se realizarán?

a. Si

b. No

IV. DISEÑO DE LAS PRUEBAS

18. ¿Se emplean Técnicas de Diseño de Casos de Prueba?

a. Si

b. No

Si la respuesta es “Si”, responda por favor la siguiente pregunta:

19. ¿Qué técnicas utiliza?

- a. Basadas en la especificación (Partición de equivalencias, análisis de valores límite, tablas de decisión, diagramas de transición de estados).
- b. Basadas en la estructura (Pruebas de caja blanca)
- c. Basadas en la experiencia
- d. Basadas en el uso

20. ¿Socializa los casos de prueba con el cliente para que éste lleve a cabo su aprobación y/o haga observaciones al respecto?

a. Si

b. No

¿Por qué?

V. ESTIMACIÓN DE COSTOS

21. ¿La empresa registra el esfuerzo invertido y los costos de las pruebas?

a. Si

b. No

22. En promedio, ¿Qué porcentaje representan las pruebas dentro de la duración total de un proyecto de software en su organización?

a. Menos del
10%

b. Entre 10%
y 20%

c. Entre 21% y
30%

d. Más de
30%

VI. MÉTRICAS DEL PROCESO DE PRUEBAS

23. ¿Las pruebas que se realizan son registradas junto con el esfuerzo invertido en hacerlas?

a. Si

b. No

24. ¿La empresa clasifica los defectos encontrados en la fase de pruebas?

a. Si

b. No

25. ¿La Empresa emplea indicadores para evaluar la eficiencia del proceso de pruebas?

a. Si
¿Cuál(es)?

b. No
¿Por qué?

VII. HERRAMIENTAS UTILIZADAS EN EL PROCESO DE PRUEBAS

26. ¿Qué herramientas emplea la organización para diseñar y ejecutar las pruebas?

a. TestLink

b. Selenium

c. QAManager

d. Otra. ¿Cuál? _____

27. ¿Qué herramientas emplea la organización para gestionar los defectos encontrados en las pruebas?

e. Mantis Bug Tracker

f. Bugzila

g. Taiga

h. Otra. ¿Cuál? _____

VIII. OPORTUNIDADES DE MEJORA DEL PROCESO DE PRUEBAS

28. ¿Considera usted que el proceso de pruebas de software debe ser mejorado en su empresa?

c. Si

d. No