



ESCUELA UNIVERSITARIA DE POSGRADO

**“MODELO DE CALIDAD PARA LA GESTIÓN DE PROYECTOS
DE SOFTWARE EN MICROEMPRESAS”**

LÍNEAS DE INVESTIGACIÓN:

INGENIERÍA DE SOFTWARE, SIMULACIÓN Y DESARROLLO DE TICS

TESIS PARA OPTAR EL GRADO ACADÉMICO DE:

MAESTRO EN INGENIERÍA DE SISTEMAS

AUTOR:

UBALDE ENRIQUEZ, REMBRANDT

ASESOR:

DR. BELTRAN CASTAÑON, CESAR ARMANDO

JURADOS:

DR. MAYHUASCA GUERRA, JORGE VICTOR

DR. RODRIGUEZ RODRIGUEZ, CIRO

DR. CONTRERAS ARANDA, SANTIAGO ESTEBAN

LIMA – PERÚ

2020

ÍNDICE

	Pág.
DEDICATORIA.....	i
AGRADECIMIENTOS.....	ii
RESUMEN.....	iii
ABSTRACT.....	iv
I. INTRODUCCIÓN.....	15
1.1 PLANTEAMIENTO DEL PROBLEMA.....	17
1.2 DESCRIPCIÓN DEL PROBLEMA.....	20
1.3 FORMULACIÓN DEL PROBLEMA.....	22
- PROBLEMA GENERAL.....	22
- PROBLEMAS ESPECÍFICOS.....	22
1.4 ANTECEDENTES.....	24
1.5 JUSTIFICACIÓN DE LA INVESTIGACIÓN.....	28
1.6 LIMITACIONES DE LA INVESTIGACIÓN.....	30
1.7 OBJETIVOS.....	31
- OBJETIVO GENERAL.....	31
- OBJETIVOS ESPECÍFICOS.....	31
1.8 HIPÓTESIS.....	31
II. MARCO TEÓRICO.....	32
2.1 MARCO CONCEPTUAL.....	32
2.2 TEORÍAS GENERALES RELACIONADOS CON EL TEMA.....	40
2.3 BASES ESPECIALIZADAS TEORICAMENTE SOBRE EL TEMA.....	55
	75

III. MÉTODO.....	
3.1 TIPO DE INVESTIGACIÓN.....	75
3.2 POBLACIÓN Y MUESTRA.....	75
3.3 OPERACIONALIZACIÓN DE VARIABLES.....	83
3.4 INSTRUMENTOS.....	85
3.5 PROCEDIMIENTOS.....	89
3.6 ANÁLISIS DE DATOS.....	90
IV. RESULTADOS.....	101
V. DISCUSIÓN DE RESULTADOS.....	146
VI. CONCLUSIONES.....	150
VII. RECOMENDACIONES.....	151
VIII. REFERENCIAS	153
IX. ANEXOS.....	155

ÍNDICE DE TABLAS

	Pág
Tabla 1: Modelo de Calidad.....	37
Tabla 2: La Ley N° 30056 - Julio del 2013.....	38
Tabla 3: Modelos de Calidad más usados por Pymes.....	48
Tabla 4: Procesos relacionados con los modelos de calidad.....	54
Tabla 5: Número de preguntas del cuestionario dirigido a Jefes y Profesionales.....	80
Tabla 6: Tamaño de la muestra para poblaciones finitas con intervalo de confianza del 95,5%	81
Tabla 7: Tamaño de Muestra.....	82
Tabla 8: Número de preguntas del cuestionario dirigido a Jefes y Profesionales	86
Tabla 9: Número de preguntas del cuestionario dirigido a Clientes PMBOK.....	87
Tabla 10: Resultados de la Validación de Contenido.....	88
Tabla 11: Resultados de Opinión del Modelo por Expertos.....	89
Tabla 12: Apreciación de los Jefes y Profesionales sobre el área: Conocimiento de las metodologías de PMBOK, PRINCE2,ISTQB Test Manager y SCRUM.....	92
Tabla 13: Apreciación de los Usuarios sobre el área: Infraestructura Tecnológica en Calidad de Software.....	93
Tabla 14: Apreciación de los Jefes y Profesionales sobre el área: Aplicación de los modelamientos de Gestión de Proyectos.....	93
Tabla 15: Apreciación de los Jefes y Profesionales sobre el área: Gestión de Proyectos	93
Tabla 16: Apreciación de los Clientes sobre el área: Gestión de Proyectos....	94
Tabla 17: Apreciación de los Usuarios sobre el área: Calidad de Software.....	95
Tabla 18: Apreciación de los Usuarios sobre el área: Infraestructura Tecnológica en Calidad de Software	112

Tabla 19: Respuesta de los Usuarios sobre el área: Expectativas.....	97
Tabla 20: Nivel de Éxito del uso de metodologías de Gestión de Proyectos de Calidad de software en las empresas clientes	99
Tabla 21: Beneficios obtenidos en las Empresas Clientes	100
Tabla 22: Áreas de Conocimiento de la Investigación.....	110
Tabla 23: Resultado aplicando Chi cuadrado a la Investigación.....	112
Tabla 24: Resultado aplicando Chi cuadrado a las Hipótesis.....	115
Tabla 25: Resultado cuestionario sección "Política Organizacional".....	117
Tabla 26: Resultado cuestionario sección "Testing de Bajo Nivel".....	118
Tabla 27: Resultado cuestionario sección "Diseño de Escenarios".....	118
Tabla 28: Resultado cuestionario sección "Integración al Proyecto".....	119
Tabla 29: Resultado cuestionario sección "Prevención".....	119
Tabla 30: Resultado cuestionario sección "Testing No Funcional".....	120
Tabla 31: Resultado cuestionario sección "Mejora Continua".....	121
Tabla 32: Antes de la aplicación del Sub Modelo Genérico para Mejora de Software como Producto.....	122
Tabla 33: Después de la aplicación del Sub Modelo Genérico para Mejora de Software como Producto	122
Tabla 34: Comentarios de respuestas.....	122
Tabla 35: Antes de la aplicación del Sub Modelo Genérico para Gestión de Proyectos de Software Microempresarial.....	124
Tabla 36: Después de la aplicación del Sub Modelo Genérico para Gestión de Proyectos de Software Microempresarial.....	124
Tabla 37: Mejoras propuestas y aceptadas	125
Tabla 38: Formulario de inspección.	129
Tabla 39: Formulario de descripción/resultados de la prueba.....	130
Tabla 40: Esquema para diseño de pruebas	131
Tabla 41: Métricas para aprobación de requerimientos	132
Tabla 42: Antes de la aplicación del Sub Modelo Genérico para Gestión de Proyectos de Calidad de Software Complejo.....	133

ÍNDICE DE FIGURAS

	Pág.
Figura 1: Proyección del Crecimiento de las Pruebas de Software	22
Figura 2: Porcentajes de Onshore y Offshore para las Pruebas de Software en el mundo	23
Figura 3: Tendencia creciente de las ganancias por outsourcing para las Pruebas de Software en el mundo	23
Figura 4: Cantidad de Microempresas Peruanas Productoras de Software.....	24
Figura 5: Porcentaje de proyectos de software fracasados según categoría de empresa	24
Figura 6: Problema principal y sus causas.....	28
Figura 7: El Modelo de Gestión	37
Figura 8: Mapa General de las bases teóricas especializadas sobre el tema...	44
Figura 9: Mapa Detallado de las bases teóricas especializadas sobre el tema..	45
Figura 10: Procesos del PMBoK 5ta. Edición	47
Figura 11: El Modelo de Procesos PRINCE2.....	48
Figura 12: Los Procesos PRINCE 2.....	49
Figura 13: El Modelo de Gestión de Proyectos SCRUM	50
Figura 14: Procesos SCRUM	51
Figura 15: Estructura de los niveles y áreas de TMMi	53
Figura 16: Estructura de TPI Next	54
Figura 17: Partes de la ISO/IEC/IEEE 29119.....	57
Figura 18: Estructura del ISTQB Nivel Avanzado	59
Figura 19: Fases de Evaluación de STEP	60
Figura 20: Procesos de Pruebas Críticos (CTP).	61
Figura 21: Procesos de Pruebas Críticos 2 (CTP).....	63
Figura 22: Complejidad de las Pruebas en Dispositivos Móviles por su Fragmentación de Sistema Operativo Android	64
Figura 23: Complejidad de las Pruebas al Software para Dispositivos Móviles	64
Figura 24: Complejidad de las Pruebas al Software en Cloud Computing.....	66
Figura 25: La Ley N° 30056 - Julio del 2013.....	75

DEDICATORIA

A los padres que trabajan – [Mi familia, Alcira y yo]

A las hijas e hijos que continúan el trabajo - [Fiorella]

Al Dios que les da la vida para trabajar – [...].

AGRADECIMIENTOS

Esta tesis no hubiera sido posible, sin el apoyo del Dr. Cesar Armando Beltrán Castañón, por su disposición, por la admiración que le tengo, como ejemplo profesional a seguir.

Mis agradecimientos al Dr. Víctor Mayhuasca , al Dr. Ciro Rodríguez y al Dr. Esteban Contreras , distinguidos profesionales , dignos y honorables, en todo momento me sentí honrado desde que los conocí.

Agradezco el apoyo de mi ilustre amigo Dr. Iván Petrlik, quien, con su constancia y perseverancia al éxito, fueron para mi ejemplo en este camino.

RESUMEN

Se tomó como génesis del modelamiento de gestión en proyectos propuesto por el PMI (Project Management Institute) en su guía de PMBOK (*Project Management Body of Knowledge*) 5ta. Edición, de manera adicional el PRINCE2 y SCRUM.

La determinación de los procesos, los procedimientos, las técnicas y las herramientas se basaron en estándares de clase mundial de las pruebas de software como: ISTQB (*International Software Testing Qualifications Board*), *Advanced Level for the Test Manager*, *ISO/IEC/IEEE 29119 Software Testing Standard*, *Business Driven Test Process Improvement* (TPI-Next) y *Test Maturity Model Integration* (TMMi).

Se propuso una arquitectura en pruebas de software basada en artefactos de desarrollo, de gestión y componentes técnicos de pruebas a software en arquitecturas de críticas como SOA, Mobile, Cloud Computing o la integración de estas.

Para respaldar la efectividad del modelamiento, se han incluido las opiniones expertas de profesionales en el ámbito de las pruebas y calidad de software a nivel nacional e internacional.

Finalmente se efectuará una evaluación del impacto en la mejora de los procesos y procedimientos en gestión de proyectos de calidad de software, producto de la utilización de un modelamiento propuesto por parte de una microempresa peruana productora de software: ARWEBSYSTEMS SAC. La evaluación se llevó a cabo gracias a la aplicación de dos modelos internacionales de evaluación del mejoramiento de gestión en pruebas de software: el Systematic Test and Evaluation Process (STEP) y el Critical Testing Process (CTP).

Palabras claves: *proyectos de calidad, modelo, calidad del software, calidad en microempresas.*

ABSTRACT

The project management model proposed by the PMI (Project Management Institute) was taken as a starting point in its PMBOK (Project Management Body of Knowledge) 5th guide. Edition, in addition the PRINCE2 and SCRUM.

The definition of the processes, procedures, techniques and tools will be based on the world-class standards of software testing such as: International Software Testing Qualifications Board (ISTQB) Advanced Level for the Test Manager, ISO / IEC / IEEE 29119 Software Testing Standard, Business Driven Test Process Improvement (TPI-Next) and Test Maturity Model Integration (TMMi).

Was proposed in critical architectures such as SOA, Mobile, Cloud Computing or the integration of these.

To support the effectiveness of the modeling, the expert opinions of professionals in the field of testing and quality of software at national and international level have been included.

Finally, an evaluation of the impact on the improvement of software quality project management processes will be done, in a result to use the modeling proposed by a Peruvian micro-company that produces software: ARWEBSYSTEMS SAC. The evaluation was carried out through the application of two international models for evaluation of software test management processes: The Systematic Test and Evaluation Process (STEP) and the Critical Testing Process (CTP).

Keywords: *quality projects, model, software quality, quality in microenterprises.*

I. INTRODUCCIÓN

La industria de la producción del software en el Perú según Apesoft y Prompex Perú (2015) está mayoritariamente integrado por microempresas. Para que estas microempresas de software puedan sobrevivir en un mercado altamente competitivo, deben producir software de alta calidad que garantice un modelo de negocio sostenible.

El modelo de calidad de software propuesto en esta tesis proporciona las ventajas competitivas a un nivel microempresarial citadas por Arciniega Alemán, Flores Rosales, & García Paucar (2014): Incrementa la calidad del software producido, facilita la relación comunicativa y comprensiva entre las personas en el equipo del proyecto en software, apoya el mejoramiento en secuencia de los procesos y procedimientos de testing de software, dando soporte la ejecución automática de ciertas tareas de pruebas de software.

El modelamiento aborda las limitaciones que enfrentan las microempresas productoras de software en el Perú según , las cuales son: bajo presupuesto , falta de recursos, corto tiempo para los entregables, estructuras organizativas informales, desordenas y hasta caóticas; falta de habilidades y experiencia de sus colaboradores para probar software crítico y complejo, así como también que las metodologías internacionales de pruebas de software no están enfocados sobre las necesidades y los problemas de las microempresas peruanas productoras de software.

El modelamiento tiene dentro si mismo mecanismos de testing de software con la meta objetiva de hacer posibles tareas necesarias para realizar una evaluación de las funcionalidades del software como producto con una connotación organizada, así como también y sistematizada, con un énfasis en las de diseño y ejecución de testing. Este modelamiento propuesto se estableció en base al análisis de las metodologías internacionales de testing en pocos modelamientos como referencia y en distintas investigaciones.

En esta tesis se estudia críticamente los procesos de pruebas de software de una microempresa productora de software en Lima. La microempresa ARWEBSYSTEMS SAC está plagada de software lleno de defectos, a menudo causa que sus clientes abandonen el

software producido después de ser comprado y cortar sus lazos comerciales con ARWEBSYSTEMS SAC. En esta tesis se investigó el estado del testing de software en general y se proporcionó a la microempresa ARWEBSYSTEMS SAC un modelamiento con procesos de prueba prácticos y simples que permiten la creación de software de mayor calidad.

Los problemas concretos con para los procesos de testing de software en una microempresa como ARWEBSYSTEMS SAC que se determinan con el uso de cuestionarios y también son confirmados por expertos en calidad de software en el Perú y el mundo y la literatura de ingeniería de software con énfasis en pruebas de software.

1.1. PLANTEAMIENTO DEL PROBLEMA

Actualmente, la calidad de software es el factor diferenciador más importante que mide la competitividad entre las microempresas productoras de software a nivel mundial.

Quintana(2017) menciona que en los años 2000 el grado de complejidad del desarrollo software aumentó debido a la aparición de nuevas arquitecturas de software como SOA (Arquitectura Orientada a Servicios), la necesidad de desarrollo de software para nuevos dispositivos como los smartphones y la necesidad del uso de nuevas plataformas para el funcionamiento del software desarrollado como es la tecnología cloud computing. La complejidad del desarrollo de software convierte complejo también a los procesos de calidad de software. La complejidad del software ha determinado en la actualidad la aparición del outsourcing y offshoring a través de proveedores altamente especializados en desarrollo software y pruebas de software a nivel mundial.

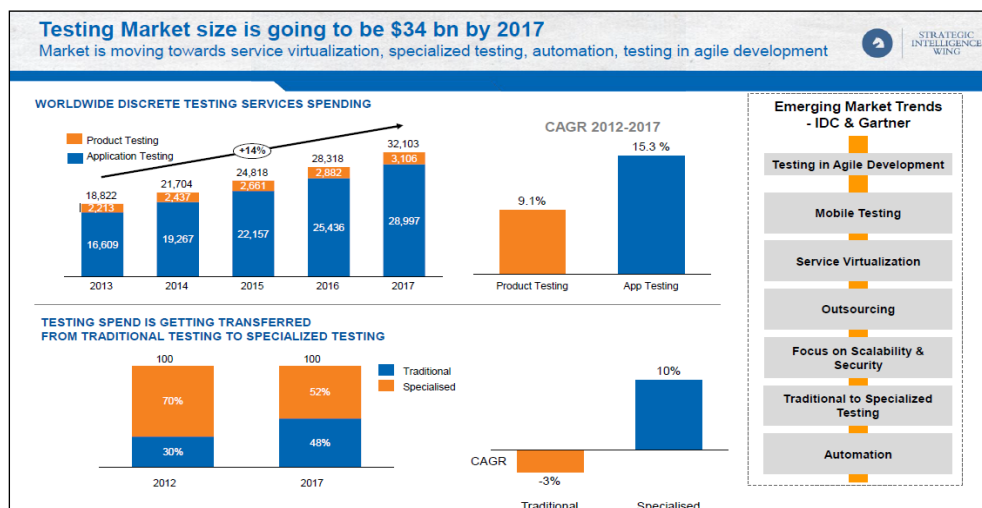


Figura 1. Proyección del Crecimiento de las Pruebas de Software
Fuente: Gartner Inc., IDC , NelsonHall y HCL Technologies.

Actualmente, el desarrollo y las pruebas de software están dominadas por los modelos de outsourcing: offshoring, onshoring o nearshoring impulsadas por el objetivo de restar gastos, las dificultades en la administración, y reducir las consecuencias producidas por la depreciación tecnológica. Debido a esto se han producido modificaciones en modelos y las fases de desarrollo y el testing de los sistemas que usan las microempresas productoras de

software. Ello demuestra un ámbito en constante crecimiento, lugar donde las microempresas creadoras de software tratan de adaptarse de mejor forma a las circunstancias del medio, adoptando el patrón idóneo con la finalidad de crear productos de software eficientes y de calidad.

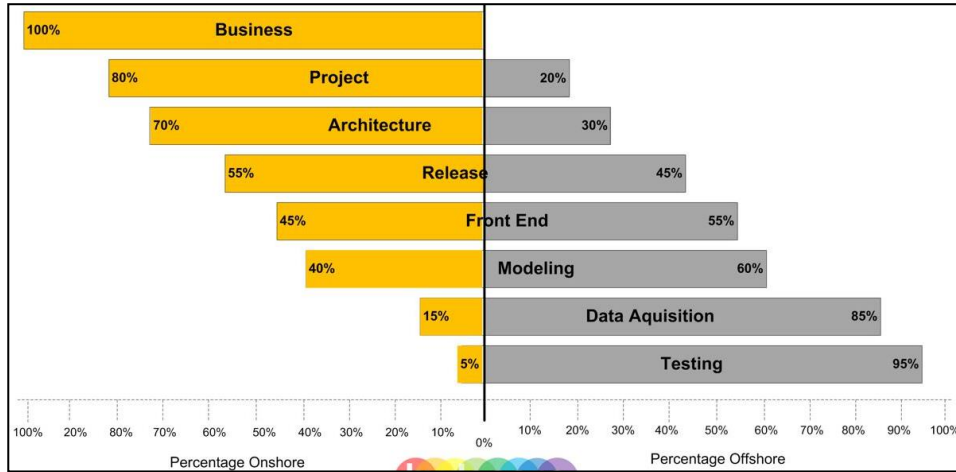


Figura 2. Porcentajes de Onshore y Offshore en Pruebas de Software Mundial.
Fuente: Hydecare consulting.

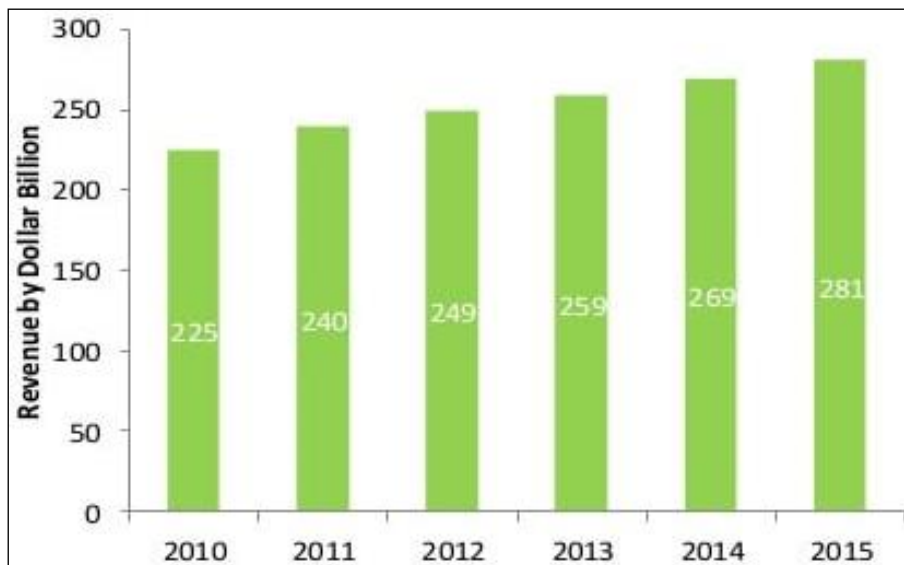


Figura 3. Tendencia creciente de las ganancias por outsourcing para las Pruebas de Software en el mundo.

Según Larrucea, O'Connor, Colomo-Palacios, & Laporte (2016) las microempresas productoras de software, son en mayor porcentaje en todo el mundo, como muestra en Canadá aproximadamente el 85% de empresas productoras de software están conformadas por menos de 20 trabajadores, en el continente Europeo, el 90% de las organizaciones del rubro de tecnologías de información están conformadas por menos de 15 trabajadores y en Latinoamérica el 80% de las organizaciones productoras de software tienen menos de 40

empleados, de estos porcentajes las microempresas especializadas en servicios de calidad de software representan solo un 30%. En el Perú existen son muy pocas las empresas dedicadas exclusivamente a servicios de calidad de software la mayoría son empresas productoras de software.

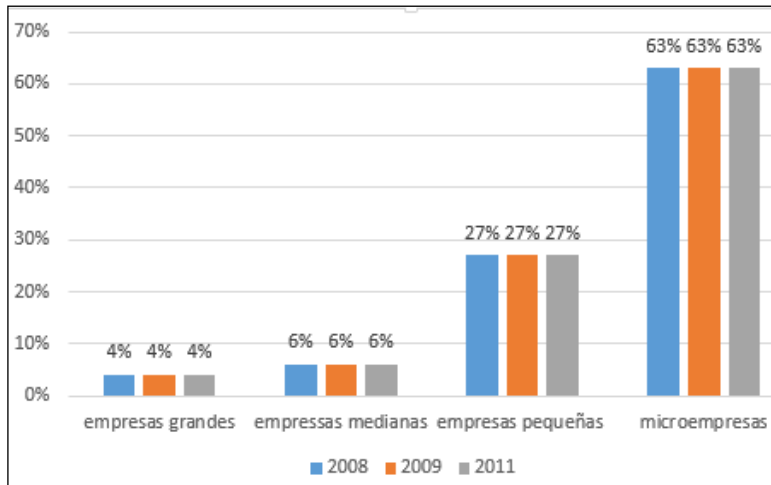


Figura 4. Cantidad de Microempresas Peruanas Productoras de Software
Fuente: Promperú - Perú Service Summit 2011

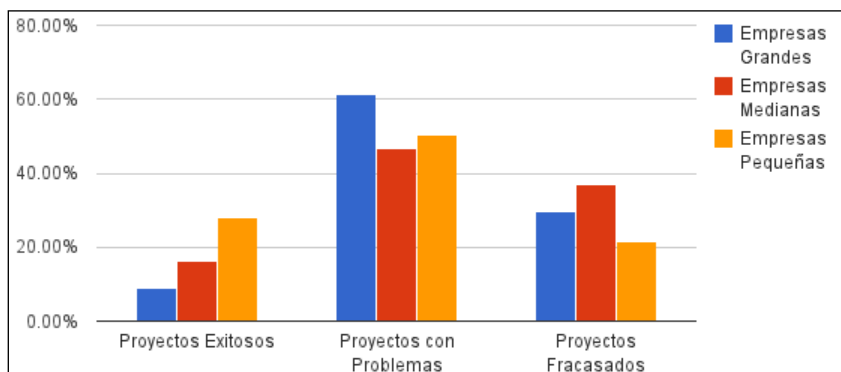


Figura 5. Porcentaje de proyectos de software fracasados según categoría de empresa.

Fuente: Standish Group – Chaos Report 2015 – En base a 50,000 empresas alrededor del mundo.

1.2. DESCRIPCIÓN DEL PROBLEMA

De lo expresado en la sección anterior es una necesidad la administración eficiente de los procedimientos de las pruebas por parte de microempresas productoras de sistemas informáticos. En entornos similares, las pruebas de software se maximizan, generando mayores dificultades en el aseguramiento y el control de los proyectos de producción de software. Lo anterior expuesto genera en las microempresas desafíos a nivel global y local como:

1.2.1. A NIVEL GLOBAL

Restricciones de tiempo y costo en los proyectos de calidad de software

Aplicar estándares complejos de calidad de sistemas es problemático. Esto es a razón de que su utilización implica una alta inversión; y sus características están más inclinadas para las grandes empresas.

Calá(2018) menciona en su documento de trabajo que las microempresas, al comenzar, afrontan el requerimiento obligatorio de entregables en tiempos muy cortos a sus clientes, entregables de pruebas de software bajo procesos innovadores y creativos. Para alcanzar este objetivo, utilizan metodologías modernas como las ágiles, como la manera más rápida de alcanzar objetivos, puesto que las metodologías ágiles son “ligeras” y a veces se ajustan mejor a los requerimientos y a la de la producción industrial de software, para tener resultados veloces y acordes a las constantes modificaciones en los requerimientos.

El seguir estándares es un mecanismo diferenciador y estratégico para las microempresas del rubro de producción de software, más aun tratándose de microempresas que se encuentran en la pugna por evolucionar. Asimismo, la utilización eficiente de prácticas buenas impide retrabajo y por ende reduce los costos.

Dificultad en la administración de calidad de software complejo

Para poder proveer servicios especializados de calidad de software complejo como software basado en arquitectura SOA, software para dispositivos móviles o software en cloud computing, según Quintana (2017) las microempresas se obligan a administrar de forma eficiente los procedimientos de pruebas de sistemas informáticos y su uso dentro de un ciclo de vida de proyectos de software. Dicho de otro modo, se requiere precisar distintas fases de administración y dar cumplimiento a los estándares internacionales CMMI, ISO y

otros; esto se incrementa en dificultad por el manifiesto ágil de los procesos que se utilizan y la razón de no aceptar el nivel de documentos y tecnología necesaria en los niveles dos y tres, para el modelo en CMMI.

1.2.2. A NIVEL LOCAL

Deficiente gestión del equipo humano en los proyectos

Según Coque Villegas, Jurado Vite, Avendaño Sudario, & Pizarro (2016) la distribución física y geográfica de las etapas del ciclo de vida de la construcción de un software, ofrecen muchas variaciones respecto a la realización conjunta o separada por parte de un outsourcing y un offshore; por tanto, obligan una alta dedicación para administrar los proyectos de construcción de software. Debido a esto la etapa de pruebas en software debe gestionarse como un proyecto aparte.

La implementación de proyectos de sistemas en simultaneo, con distintos ciclos de vida y frecuentemente con casos de alteraciones en la utilización de las metodologías, plantea varias situaciones como la coordinación en todos los niveles, inspecciones y labores no planificadas, ausencia de entendimiento en procesos, desconcierto debido a la terminología utilizada, determinación errada de roles, ausencia de coordinación entre los equipos y las actividades, y una pésima disposición de tareas.

1.3. FORMULACIÓN DEL PROBLEMA

1.3.1. PROBLEMA GENERAL

¿Qué efecto tendrá la implementación de un modelo para mejorar la eficiencia en la gestión de los proyectos de calidad de software en las microempresas?

1.3.2. PROBLEMAS ESPECIFICOS

- a. ¿Por qué los proyectos de calidad de software gestionados por las microempresas peruanas son económica, operativa y técnicamente inviables?
- b. ¿Por qué los proyectos microempresariales de calidad de software no se gestionan de manera flexible, adaptable y escalable a las diferentes metodologías de desarrollo de software moderno?
- c. ¿Por qué las microempresas peruanas no pueden gestionar proyectos de calidad de software complejo de los tipos SOA, Móviles o Cloud Computing o la integración de todos estos?

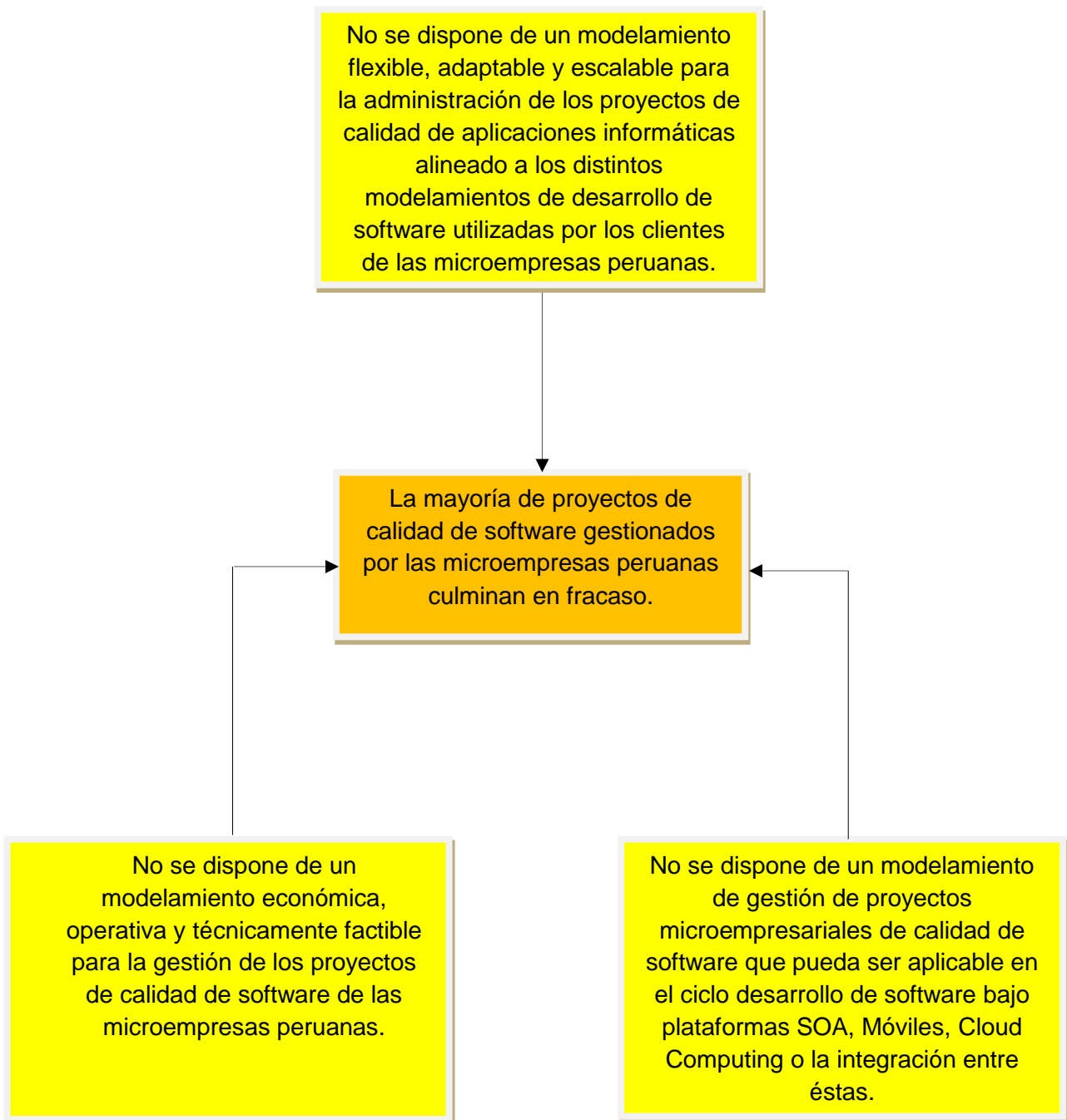


Figura 6: Problema principal y sus causas
Fuente: Esta investigación

1.4. ANTECEDENTES

1.4.1. ANTECEDENTES NACIONALES

En primer lugar, se tiene que, en marzo de 2017; fue publicado por Abraham Dávila, Cecilia García y Sandra Córdor (2017) en la Revista Ibérica de Sistemas e Tecnologías de Información, y luego registrada en la biblioteca virtual SCIELO Portugal, el artículo: “Análisis exploratorio en adopción de prácticas de testing de software de la ISO/IEC 29119-2 en organizaciones de Lima, Perú. “

El artículo aborda a la ISO/IEC 29119 en el mundo del testing de software. Revela que, en microempresas productoras de software limeñas, el testing de es caótico respecto al orden, no logrando adecuada cobertura y presentando errores en el software cuando entra a su fase operativa. Esta fue una investigación a nivel exploratorio en microempresas que testean software, para identificar aspectos externos que recaen en un conjunto de buenas prácticas en testing de software. La investigación se efectuó utilizando investigación de casos considerando una microempresa tipo.

El artículo concluye para las microempresas que se desarrollaron como fábricas de producción de software en una perspectiva exigente mostraron un alto grado de inclinación hacia prácticas estandarizadas frente a las que participaron como una parte importante en el equipo de construcción de software. Asimismo, las variables externas son influenciadores positivos para el primer grupo y negativos para el segundo grupo.

Este artículo se relaciona con la presente investigación, puesto que se realizan un estudio en el ámbito local relacionado a las pruebas de software en pequeñas empresas, la cual sirve como parte del fundamento de la realidad local para la presente investigación.

En segundo lugar, se tiene que, en noviembre del 2015 fue aceptado en el prestigioso Sistema de Información Científica Redalyc; de la Universidad Autónoma del Estado de México; el artículo: “Desarrollo de un Marco Metodológico del proceso de Verificación y Validación de Software para pequeñas y medianas empresas”, elaborado por Aguirre (2015).

El trabajo de investigación realiza primordialmente el objetivo de construir un Marco Modelo de los Procesos de Verificación & Validación en la realidad socio económica de las Microempresas, la tesis alcanza un camino metodológico, funciones y menesteres en general de los equipos de trabajo, así como también plantillas, checklists y indicadores de medición

de pruebas. El modelo fue aplicado a un software de Control de Inventarios de una microempresa peruana llamada QBOSS SAC, obteniéndose como resultado grandes mejorías en calidad en software.

El trabajo de investigación se enfoca en la aplicación de Modelos de pruebas y Calidad, como los modelos CMMI, CMM, MOPROSOFT, COMPETISOFT, TMMi, TMap para una Pyme orientada al desarrollo de Software en Lima – Perú.

Este artículo se relaciona con la investigación, debido al análisis de los procedimientos V&V (Verificación y Validación) como los procedimientos claves en las pruebas de software, desde 3 perspectivas: El grado de complejidad de utilización de modelos por parte de las Pymes, las condiciones cortas de presupuesto y la falta de conocimiento que se tiene sobre las utilidades que se obtendrían de utilizar estos modelos por parte de las microempresas.

En tercer lugar, se menciona que, en septiembre del 2014 fue publicado en la Escuela de Postgrado de la UPC (Universidad Peruana de Ciencias Aplicadas) de Lima - Perú, el trabajo de postgrado: “Integrando la oferta de las microempresas peruanas con la demanda de soluciones de software” Arciniega , Flores y García (2014); para dar cumplimiento a los requisitos para conseguir el grado de MBA.

La tesis está orientada hacia un objetivo en conjunto, en puntos generales, está implementando por las microempresas que brindan servicios relacionados con la implementación de diversas soluciones de software de sus empresas clientes. La investigación propone un plan para el negocio como una solución global para estas microempresas y ayude a su crecimiento. La propuesta plantea la implementación de la Norma ISO/IEC 29110 la cual tiene, como meta principal, mejorar los procesos de Gestión de Proyectos e Implementación de Software en microempresas.

La tesis describe a la microempresa ATIX IT Research & Innovation S.A.C., como la microempresa responsable en la propuesta. La meta es colocarse como la principal microempresa que da un servicio global a las microempresas productoras de software del Perú, garantizando la calidad y eficiencia para la mejora de sus procedimientos y procesos, elevando su competitividad vía la implementación de una Norma ISO / IEC 29110.

La tesis concluye con resultados viables; debido a la tendencia de crecimiento del sector y del consumo en el Perú; la originalidad y exclusividad del producto, que es la ventaja competitiva para la microempresa; el conocimiento que se tiene del sector y por las altas barreras de entrada se mitigará la aparición de posibles competidores.

Esta investigación se vincula con la tesis, ya que se aprovecha el conocimiento que se tiene del sector de desarrollo de software por parte de microempresas en el Perú.

1.4.2. ANTECEDENTES INTERNACIONALES

Inicialmente se tiene que Rojas, Pino y Martínez (2015) mencionan en su publicación: “Proceso de pruebas para pequeñas organizaciones desarrolladoras de software”, un planteamiento de procesos para conducir la realización de las pruebas en las microempresas productoras de software. Se mencionan técnicas de testing funcional con la premisa de plantear las tareas necesarias con el objetivo de plantear una evaluación funcional del software como producto de una forma organizadamente sistémica, con un énfasis en las que son de diseño y ejecución del testing.

Concluye con ejecuciones exitosas de procedimientos y procesos de testing en microempresas productoras de software, en la que se observó que estos procesos pueden ser prácticos y fácilmente utilizables para hacer pruebas al software como producto por microempresas dedicadas al mencionado rubro.

Este artículo se encuentra en la línea de la investigación de la presente tesis puesto que propone un proceso de pruebas de software base, a medida y ejecutado exitosamente en una pequeña organización, lo cual es objetivo similar respecto al ámbito y alcance de la presente investigación.

En segundo lugar, se tiene que, en mayo de 2014 fue presentado en la Unidad de Gestión de Postgrados en la Universidad de Fuerzas Armadas de Latacunga - Ecuador, el trabajo especial de postgrado de Escobar(2014): “Modelo de pruebas de software para el segundo nivel de TMMI en la organización Sirecom, Quito Ecuador” , como requisito para obtener el nivel académico de Maestro con una mención especial de Ingeniería de Software.

La Tesis trata de la utilización de listas de chequeo, basadas en la ISO-9001-2000 que permitieron establecer evaluaciones de calidad a la empresa Siredcom, alcanzando índices

pronunciados de inconformidad. Asimismo, en las evaluaciones hechas referentes al nivel 2 de TMMI, los procesos involucrados, determinaron mayor inconformidad que conformidad.

La Tesis concluye que el TMMI, en la microempresa SIREDCOM está en un nivel inicial ya que no se cuenta con un proceso definido que permita diferenciarlo claramente de un debugging formal y no se determinan las especificaciones del software bajo prueba. El personal de la microempresa no posee la capacidad idónea. Lo anteriormente mencionado refleja la carencia del diseño de un modelamiento que permita ejecutar testing funcional de software, a fin de alcanzar el nivel 2 de Test Maturity Integration como una forma alternativa más adecuada a la situación problemática de la microempresa SIREDCOM.

La Tesis se encamina con el trabajo de investigación presente, puesto que se basa como una propuesta en el ciclo de prueba y componiéndose de cuatro partes: Especificación, Planificación, Ciclo y Resultado, las mismas que sirven de base como experiencia exitosa con excelentes resultados.

En tercer lugar, se tiene que, en marzo de 2010 fue elaborado en el Departamento de Ciencias de la Computación de la Universidad de Pretoria – Sudáfrica, el Reporte Técnico: “Software testing en una pequeña compañía: un caso de estudio” por Van Zyl (2010).

El reporte técnico trata acerca de las microempresas de software en todo el mundo. Menciona que las microempresas de software para sobrevivir en un mercado altamente competitivo, deben producir software de alta calidad que garantice un modelo de negocio sostenible. En el reporte se muestra críticamente el procedimiento de testing en software de una microempresa de software en Sudáfrica. Esta empresa está plagada por software que está lleno de defectos, a menudo causando a los clientes a abandonar el software

completamente después de ser comprado y cortar sus lazos comerciales con esta empresa. Los problemas concretos con el proceso de prueba en la microempresa tratada se determinan con el uso de un cuestionario y también son confirmados por la literatura de ingeniería de software.

Este reporte técnico se relaciona con la presente tesis, ya que sirve de base a nivel microempresarial, para las mejoras de los procesos en pruebas para software de una

microempresa dedicada a este rubro, con grandes deficiencias al inicio, pero que luego de implantarse los nuevos procesos, está microempresa mejora notablemente.

1.5. JUSTIFICACIÓN DE LA INVESTIGACIÓN

La calidad de un software está basada en el proceso de las pruebas de software, la cual es una especialidad de la ingeniería de software amplia y profunda que ha madurado en la actualidad, que ha avanzado proporcionalmente a los métodos y las técnicas de producción industrial de software. En concordancia a esto se encuentra en la investigación realizada por Bertolino (2007): “Software Testing Research: Achievements, Challenges, Dreams, Future of Software Engineering”; donde se descubre la presencia de distintos procedimientos que pretenden dar resolución a situaciones como el seguimiento de los requisitos con los testcases, la automatización del testing de software y la administración total del proceso de las pruebas. Ahí también se demuestra el análisis asociado a los beneficios contados a la fecha, producto del crecimiento y las investigaciones; Asimismo muestra un mapa que refleja los desafíos que afrontan actualmente la industria y las instituciones educativas con la finalidad de contar con una teoría de testing estándar, el testing sustentado en prototipos, con realización automatizada, y un proceso de testing con la mayor eficacia.

Con la finalidad de recalcar el valor de la necesidad de procesos perfeccionados en la administración de las pruebas, se indica adicionalmente una investigación realizada por el Instituto Nacional de Estándares y de Tecnologías de los Estados Unidos de América, en el cual se indica que el valor sobre la economía nacional de Estados Unidos, de no contar con un apropiado servicio de calidad de software es aproximado a los 22 millones de dólares. Es claramente evidenciado que esto es un ámbito crítico, en el cual se pueden plantear medidas de solución para el desarrollo y creación de un modelamiento de pruebas de software.

El modelamiento de administración de proyectos de calidad de software que se plantea, busca dar una solución centrada en concordancia con las otras etapas de los ciclos de vida y los modelos aplicados por cualquier microempresa productora de sistemas-software. Enseguida se señalan los puntos del modelamiento de proyectos de calidad de software:

El proceso de testing de software ha implementado el total del ciclo de vida de construcción; en este sentido inclusive su administración. El modelo de administración del ciclo de vida de software aplicado, aqueja los procesos de testing de software y por ende en

su calidad, por tanto, el modelamiento propuesto se inicia desde modelos heterogéneos de desarrollo de software.

El testing de software, son una disciplina vital en la implementación de cualquier proyecto de producción de software. Estas establecen el nivel de consecución del software solicitado. Asimismo, establecen la culminación de las etapas o el nivel de evolución en la producción de un software. El modelo propuesto considerará el grado de cumplimiento de requisitos.

El uso del testing de software, no se restringe a una sola etapa que se inicia únicamente después de la programación del software. En la actualidad las modernas metodologías de testing de software son posibles para hacer testing de software antes del desarrollo y del mantenimiento al software. El modelamiento asegurará que se realicen las pruebas al software desde la fase inicial del proyecto de software.

Para la ejecución de un testing es importante la planificación, el diseño, la codificación, la ejecución y evaluación de resultados finales. Se conforman en un ciclo de pruebas con fases establecidas que hacen sus metas. Según Salazar Montes, Cárdenas Gaviria & Franco Bedoya (2018) , el modelamiento contemplará en su estructura procesos de pruebas bien definidos basados en estándares de clase mundial para dicho fin.

La gestión de la calidad de software requiere una amplia dedicación para hacer su aplicación y la trazabilidad de la superación de los errores detectados. El modelamiento contemplará una sólida fase para la estimación de tiempos para todos los procedimientos y procesos de testing en software.

Las pruebas de modelos y estigmas de producción de software, exige una transformación de los procesos, técnicas y herramientas. Este escenario es obvio en software bajo arquitectura SOA, software para dispositivos móviles y software en cloud computing. El modelamiento tendrá procesos exclusivos para pruebas de software bajo arquitectura SOA, software para dispositivos móviles y software en Cloud Computing.

Los mecanismos que se utilizan en el software común como a los componentes SOA, software para dispositivos móviles y software en cloud computing, se utilizan aisladamente. No existe un modelamiento integrado que simplifica la administración de los procesos de pruebas y que sea accesible para las microempresas productoras de software. El modelamiento planteado es aplicado en un proyecto de calidad de software de una

microempresa a manera de caso de estudio. El caso de estudio incluye un software bajo arquitectura SOA, integrado con un software para dispositivos móviles y algunos procesos de software Cloud Computing, el mencionado proyecto serán aplicados en la empresa ARWEBSYSTEMS SAC.

1.6. LIMITACIONES DE LA INVESTIGACIÓN

ALCANCE

El alcance de esta tesis es desarrollar un modelamiento para gestionar proyectos microempresariales de calidad de software en microempresas que se orientan a la construcción de software, este modelamiento propuesto está fundamentada en modelos y estándares de calidad de clase mundial. En este sentido, se añadirá una guía detallada, funciones, competencias del equipo de trabajo, las principales métricas, plantillas y listas de cotejo base que considerarían en la implementación de un Software con Calidad.

Se eligió un proyecto de desarrollo de software de la microempresa ARQEBSYSTEMS, para aplicar el modelamiento se evaluaron los resultados logrados. Para la presente tesis se seleccionó el protocolo de interoperabilidad que desarrolló la empresa ARWEBSYSTEMS para su cliente, un banco de prestigio local, al cual por confidencialidad no se nombrará en esta tesis.

LIMITACIONES

Temporal: La investigación se realizará durante el periodo comprendido entre el mes de enero hasta diciembre de 2018.

Espacial: La tesis de investigación se llevará a cabo en la microempresa productora de software ARWEBSYSTEMS SAC, para un proyecto desarrollado para uno de sus clientes, un prestigioso banco local.

Conceptual: La investigación tiene como delimitaciones conceptuales las metodologías PMBOK, PRINCE2, ISTQB, TMMI, SCRUM, TPI e ISO / IEEE 29119.

1.7. OBJETIVOS

1.7.1. OBJETIVO GENERAL

Implementar un modelamiento para mejorar la eficiencia de la gestión de proyectos de calidad de software en las microempresas.

1.7.2. OBJETIVOS ESPECÍFICOS

- a. Diseñar procesos flexibles, adaptables y escalables con sus respectivas actividades y herramientas de soporte a un nivel de gestión y técnico para la gestión eficiente de los proyectos microempresariales de calidad de software, alineadas a las metodologías de desarrollo de software más modernas.
- b. Diseñar procesos, actividades y herramientas que permitan la factibilidad económica, operativa y técnica para la gestión eficiente de los proyectos de calidad de software de las microempresas productoras de software.
- c. Diseñar una arquitectura tecnológicamente moderna que haga posible el gestionar proyectos de la calidad de software complejo bajo plataformas SOA, Móviles, Cloud Computing o la integración entre éstas.

1.8. HIPÓTESIS

1.8.1. HIPÓTESIS GENERAL

La implementación del modelamiento propuesto permitirá mejorar la eficiencia de la gestión de los proyectos de calidad de software de las microempresas.

1.8.2. HIPÓTESIS ESPECÍFICOS

- a. El diseño de procesos flexibles, adaptables y escalables con sus respectivas actividades y herramientas de soporte a nivel gestionable y técnicos permitirá que proyectos de calidad de los sistemas estén alineados a las metodologías más modernas de desarrollo de software.
- b. El diseño de procesos, actividades y herramientas factibles de gestión permitirá a las microempresas productoras de software gestionar proyectos de calidad de software con factibilidad económica, operativa y técnica.
- c. El diseño de una arquitectura tecnológicamente moderna permitirá a las microempresas productoras de software gestionar proyectos de calidad de software complejo bajo plataformas SOA, Móviles, Cloud Computing o la integración entre éstas.

II. MARCO TEÓRICO

2.1 MARCO CONCEPTUAL

En esta sección se describen algunos conceptos relacionadas con el presente trabajo de investigación y que se utilizaron a lo largo del desarrollo de este, tales como:

EFICIENCIA:

Es el análisis del volumen de los recursos consumidos para alcanzar los objetivos. Las actividades eficientes hacen uso optimizado de los recursos y tienen un costo menor.

En administración de proyectos, la eficiencia es el enlace entre medios utilizados en proyectos específicos junto con resultados obtenidos. Eficiencia se muestra cuando los pocos recursos se utilizan para lograr un fin en común; así como también, cuando los objetivos son alcanzados con el consumo de menos recursos.

Comienza con un estudio de opciones y alternativas para dar ejecución dentro de un proyecto. Se mide la capacidad de lograr objetivos y metas estimadas con pocos recursos con los que se cuenta, llegando de esta manera a su mejora. Es necesario explicar cómo la eficiencia puede influir en lo atractivo de un proyecto ya que de usarla existe gran oportunidad de invertir arriesgando más.

El análisis eficiente se hace uso de indicadores costo-eficiencia y costo-beneficio utilizando también medir diferentes alternativas de acciones pudiéndose ejecutar antes o después. En definitiva, se deben evaluar alternativas de formas como iniciar un proyecto, la búsqueda de un presupuesto eficiente es necesario.

La eficiencia evalúa la existencia de 2 tipos de análisis: el costo-efectividad y el costo-beneficio.

Análisis costo-beneficio: es un vínculo de los beneficios logrados y los costos hechos utilizando como unidad de medida al dinero. En definitiva, el análisis busca un vínculo beneficio-costo más grande que uno cuando los beneficios son más grandes que los costos.

$$\text{Relación beneficio-costo} = \frac{\text{Valor presente de los beneficios}}{\text{Valor presente de los costos}} \times 100$$

En el análisis costo-beneficio se hace uso del indicador Retorno de la Inversión (ROI) para evaluar el rendimiento de las inversiones que se hacen en un proyecto.

También la Tasa Interna de Retorno (TIR) define un valor de por encima del cual se considera atractivo económicamente un proyecto. Esta tasa hace que el valor presente de los beneficios y presupuesto es cero. La TIR se compara con tasas de interés de otros

proyectos alternativos; normalmente se usa para decidir a favor de aquel que ofrezca una TIR superior.

MODELAMIENTO:

Se determina la diferencia entre método y técnica, debido a que son muchas las veces que se tiene en cuenta sinónimos, pero una técnica se considera como la parte interactiva de un modelamiento. Considerando para el modelamiento los mejores resultados cuando los grupos tengan una conducción sólida. Se menciona que el centro de un modelamiento se centra en transformar la pretensión en intento y el intento en un hecho concreto y real.

Asimismo, se puede tomar la definición de modelamiento como el enfoque de un problema global organizada, metódica y bajo de cumplimiento de reglas de comportamiento. Con este enunciado se demuestra la diferencia de "modelamiento" y "técnica". La técnica se define como un elemento del modelamiento, un medio o forma que se utiliza para efectuar el modelamiento mismo. En conclusión, definimos a la técnica como uno de los elementos que interactúa en un modelamiento.

El modelamiento es un segmento de la lógica, estudia los métodos. Se subdivide en 2 partes: la sistémica, que fija los estándares conceptuales, la clasificación y las pruebas, y la innovación, que fija las normas de los métodos de investigación propios de cada ciencia. Es la manera sistémica de hacer cierta cosa.

En la presente tesis se comprenderá como modelamiento al componente del proceso que hace posible sistematizar métodos y técnicas para hacer posible el proceso de gestión de proyectos. "Los métodos -según Martínez Miguélez (1999)- son vías que facilitan los conocimientos seguros y confiables, para dar solución a problemas que la vida plantea". El método es el modo o técnica de llevar a cabo una acción, modo de proceder, el hábito o costumbre que cada cual tiene y observa. Un modo ordenado de proceder para llegar a un fin.

La necesidad de un modelamiento en todas las disciplinas, sean cual sea el área de su desarrollo, se trabajó con ciertos métodos. La calidad del software por medio de una gestión eficiente de los proyectos no es la excepción. El uso de un modelamiento facilita la colaboración en los procesos de pruebas de software, donde se requiere de habilidades especializadas. Nadie dedicado a la calidad de software como un probador o analista de calidad de software puede proceder sin una ruta de trabajo.

Un modo ordenado de proceder para llegar a un resultado o fin.

La necesidad de un modelamiento en todas las disciplinas, sean cual sea el área de su desarrollo, se trabajó con ciertos métodos. La calidad del software por medio de una gestión eficiente de los proyectos no es la excepción. El uso de un modelamiento facilita la colaboración en los procesos de pruebas de software, donde se requiere de habilidades especializadas. Ningún buen probador o analista de calidad de software puede proceder sin un modelamiento de trabajo.

Una forma ordenada de llevar a cabo un resultado o fin.

La necesidad de un modelamiento en las disciplinas, cualquiera sea el área de su desarrollo, se trabajó con ciertos métodos. La calidad del software por medio de una gestión eficiente de los proyectos no es la excepción. El uso de un modelamiento facilita la colaboración en los procesos de pruebas de software, donde se requiere de habilidades especializadas. Ningún buen probador o analista de calidad de software puede proceder sin un modelamiento de trabajo.

Un modo ordenado de proceder para llegar a un resultado o fin.

La necesidad de un modelamiento en todas las disciplinas, sean cual sea el área de su desarrollo, se trabajó con ciertos métodos. La calidad del software por medio de una gestión eficiente de los proyectos no es la excepción. El uso de un modelamiento facilita la colaboración en los procesos de pruebas de software, donde se necesita características especializadas. Ningún buen probador o analista de calidad de software puede proceder sin un modelamiento de trabajo.

Si un Líder de un Proyecto de Pruebas de software encomienda una tarea a un grupo de probadores y/o analistas de pruebas de software, sin darles una guía, objetivos o lineamientos, los resultados en la mayoría de los casos no serán satisfactorios. Los jefes no son irresponsables por solamente los resultados de su grupo de personas a cargo. La colaboración en proyectos, funciona mejor si existe un liderazgo definido, en donde se da una estructura y guía para la gestión.

Enfocado a la administración eficiente de proyectos de calidad de software, la esencia de un modelamiento es notoria en la evolución de un deseo en un primer intento y de este intento a la práctica. El intento es el punto de partida, y al convertir el intento en un hecho real, es donde el modelamiento ocupa el protagonismo más importante.

PROYECTO:

Según lo indicado por (Cleland & King, 1975) establece que “Un proyecto es la combinación de recursos humanos y no humanos reunidos en una organización temporal para conseguir un propósito determinado”.

Otra definición establecida por la Guía del PMBOK- Quinta Edición (2010), señala que:

“Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único.”

Sin embargo, se puede definir un proyecto como un grupo de actividades que se realizan para lograr objetivos específicos, en las cuales participan recursos humanos e implica costos, todo ello en determinado tiempo con inicio y fin establecidos.

GESTIÓN:

Entre las definiciones de “Gestión” se puede establecer:

“Gestión es hacer diligencias que conduzcan al logro de reglas de negocio de un deseo cualquiera”, esta definición según el sitio web de Real Academia de la Lengua Española.

Simplemente son fines de la presente tesis de investigación se puede definir qué gestión es el grupo de tareas que se llevan a cabo usando técnicas y herramientas que permitan obtener los resultados esperados.

GESTIÓN EFICIENTE:

El doctor Steve Mc Connell, en Rapid Development, elabora los principales aspectos que hacen posible que un proyecto software es eficiente. El autor define eficiente a estar adelante de la capacidad de finalizar en tiempo, en coste y la funcionalidad adecuada.

Los resultados mejores que tengan éxito sea posible de realizar en distintos proyectos que se aborden se debería de incluir 3 puntos fundamentales:

Evitar los errores clásicos: Mc Connell dividió: errores de las personas, a los procesos llevados a cabo al software que se desarrolle y las tecnologías utilizadas.

Seguir buenas prácticas: buenas prácticas de gestión, buenas prácticas técnicas y buenas prácticas de aseguramiento de la calidad.

Gestionar los riesgos del proyecto: No considerar riesgos en un proyecto es no cumplir los 3 aspectos fundamentales del proyecto (coste, tiempo, alcance).

PROCESO:

De acuerdo con lo señalado por A. Ljungberg, 1994, “Un proceso es un conjunto organizado de tareas enlazadas secuencialmente las que se llevan a cabo velozmente y que utilizan recursos e información para transformar insumos en productos abarcando desde el comienzo del proceso hacia la satisfacción de necesidades del cliente”

Según Jacobson, Booch, Rumbaugh, 2000; la metodología RUP es un conjunto de actividades, no de su ejecución.

En términos sencillos se puede indicar que un procedimiento o proceso es una de las tareas, actividades o acciones vinculadas entre ellas, a partir de una o varias entradas de información, dan lugar a una o varias salidas con algún valor añadido.

PRUEBAS DE SOFTWARE:

El testing de software es el conjunto de procesos utilizado para evaluar la calidad del software de la computadora. La calidad no se limita al proceso de la ejecución de un programa o aplicación con el objetivo de encontrar defectos de software.

Un problema con las pruebas de software es que probar todas las combinaciones de entradas y precondiciones no es factible cuando se prueba algo que no sea un producto simple. La cantidad de bugs en un software suele ser amplio y los bugs que ocurren a menudo con frecuencia son complejos de identificar en la prueba.

Un enfoque eficaz para probar los productos complejos implica "Investigación" o "Cuestionar" el producto con el fin de hacer su evaluación. Aquí "Preguntas" o "Investigaciones" son las actividades realizadas por el probador de software en el producto de software. Como resultado de tales operaciones de los probadores de software, el producto responde con su comportamiento.

Los tres objetivos más importantes del testing de software son:

1) El testing de software tienen un motivo principal para detectar errores en la aplicación, independientemente de la causa de los errores.

2) El punto de inicio de la prueba es la creación de un testcase eficiente el cual tiene probabilidad de errores que no se han detectado hasta el momento.

3) Se puede decir que una prueba es exitosa si es capaz de descubrir los errores que aún no se han detectado.

El esfuerzo de prueba es exitoso cuando es capaz de descubrir varios tipos de errores pertenecientes a diferentes clases en el menor tiempo posible y, por supuesto, siguiendo un enfoque científico y sistemático. Una ventaja adicional del buen esfuerzo de prueba es que

proporciona una demostración científica del funcionamiento perfecto de la aplicación de software en línea con las especificaciones definidas. Muchos datos se generan a partir del proceso de prueba, lo que refleja la fiabilidad y el nivel de calidad de la aplicación de software. Sin embargo, se debe tener en cuenta que las pruebas de software son un indicador de la presencia de defectos, a la vez que pueden proporcionar un certificado en blanco de ausencia de errores.

CALIDAD DE SOFTWARE:

La calidad no es un término absoluto; es valioso para alguna persona. Por lo tanto, debemos tener en mente que las pruebas no pueden garantizar por completo que la aplicación del software sea absolutamente correcta. Dimensiones funcionales de calidad como Usabilidad, Escalabilidad, Rendimiento, Compatibilidad, Confiabilidad son términos altamente subjetivos que tienen un valor significativo para una persona pero que pueden ser intolerables para la otra.

Las etapas que influyen sobre la calidad del software se dividen en dos grandes grupos:

Medición directa (defectos identificados en el testing).

Medición indirecta (facilidad de uso o de mantenimiento).

En cada uno de los casos deben realizar mediciones. Se debe hacer un match con software en un grupo de datos y conseguir una pista sobre calidad. McCall, Richards & Walters, dieron clasificación a las variables que se involucran en la calidad de software. Se describen 3 aspectos de todo software:

Características operacionales.

Capacidad para los cambios.

Capacidad de adaptación de diferentes entornos.

MODELO DE CALIDAD

Tabla 1:
Modelo de Calidad

	TMM	TMMi	TIM	TestPAI	Tutelkn	ISO 29119	TPI
Microempresa				X	X	X	
Entorno	X	X	X		X		
Planteamiento	X	X	X	X	X	X	X
Información Documentario		X			X		
Diseño de Pruebas	X		X		X		X
Testwares	X			X	X		X
Cultura organizacional			X		X		X
Control de Testing		X			X		
Ejecución de Testing	X	X	X	X	X		
Organización de personal	X			X	X		
Formación de personal	X	X	X				X
Integración con el ciclo de vida	X	X	X		X		X
Testing No Funcional	X			X		X	X
Mediciones	X	X		X		X	
Prevenir defectos	X			X	X	X	
Optimizar continuamente		X	X	X			
Reutilizar el testing		X	X		X	X	

Control de riesgos	X	X			X		X
Requerimientos Organizaciones		X		X	X		X
Testing de Bajo Nivel	X	X	X	X			X

Fuente: Elaboración propia.

MICROEMPRESA

Microempresa es una unidad económica formada por persona natural o jurídica, bajo cualquiera de las formas organizativas y/o empresariales que se contemplan en las leyes vigentes, las que tienen por objeto implementar tareas de producción, extracción, comercialización, transformación, de bienes o que brinda servicios.

En el Perú, al 30 de junio de 2017 existen 1 millón 998 mil 772 unidades empresariales, de los cuales el 99,6% son microempresas, considerando la más reciente agrupación establecida por la Ley N° 30056 “Ley que modifica diversas leyes para facilitar la inversión, impulsar el desarrollo productivo y el crecimiento empresarial”, dictada el 1 de julio de 2013.

Tabla 2:
La Ley N° 30056 - Julio del 2013

	Ley MYPE D.S. N° 007-2008-TR		Ley N° 30056	
	Ventas Anuales	Trabajadores	Ventas Anuales	Trabajadores
Microempresa	Hasta 150 UIT	1 a 10	Hasta 150 UIT	No hay límites
Pequeña Empresa	Hasta 1,700 UIT	1 a 100	Más de 150 UIT y hasta 1,700 UIT	No hay límites
Mediana Empresa			Más de 1700 UIT y hasta 2,300 UIT	No hay límites

Fuente: Congreso de la República del Perú

Esta Ley norma que las micro, pequeñas y medianas empresas se fundan según sus ventas anuales establecidas en la cantidad de Unidades Impositivas Tributarias (UIT). Por lo que microempresas son aquellas las que alcanzan ventas hasta un monto de 150 UIT, pequeñas empresas las que tienen ventas al año mayores a este valor y hasta 1700 UIT y son medianas empresas las que venden más de 1700 UIT hasta 2300 UIT. De esta manera el valor de la Unidad Impositiva Tributaria del 2019 es de 4200 soles, fijado por el Decreto

Supremo N° 233-2011-EF, los ingresos registrados en el 2018 por las micro, pequeñas y medianas empresas, significan el 25,7% de las ventas totales del país.

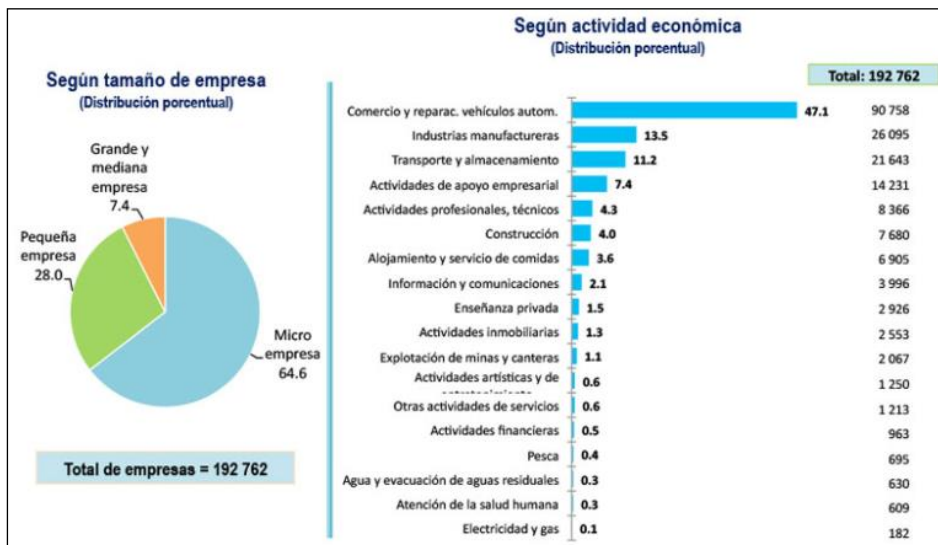


Figura 26. Empresas según tamaño y actividad económica

Fuente: Encuesta Nacional de Empresas (ENE) 2016 elaborada por el INEI y el Ministerio de la Producción (PRODUCE)

2.2 TEORÍAS GENERALES RELACIONADOS CON EL TEMA

2.2.1 GESTIÓN DE PROYECTOS

La administración en proyectos es una agrupación de procedimientos y de procesos perenne. El mismo que requiere de habilidades globales, soportadas por mecanismos de actividades que hacen crecer productividad. Los objetivos de planificación y control son de dar propuestas en común para la mejora y la gestión de proyectos. La planificación debe fundamentarse en fases estratégicas, tácticas y a niveles operativos de las empresas con el objetivo principal de lograr una misión en conjunto a plazos cortos, plazos medianos y plazo largo.

Por medio del ciclo de vida de proyectos, se establecen dos tipos de tareas a ejecutar y que ambas estén estrechamente interrelacionadas: tareas gestionables y tareas operativas del sistema-software.

Las tareas gestionables son las relacionadas con la gestión de las empresas, miembros del equipo, software y procesos planificar y desarrollo del software. La planificación y el control, es reiterada en cada etapa del proyecto y proveer de destrezas gerenciales con las

cuales las tareas de construcción del software son pronosticadas, planificadas y puestas en marcha.

Las tareas de construcción del software se ubican en la etapa de desarrollo.

Las metodologías de construcción de software son típicas y organizadas en diferentes etapas en conjuntos de áreas funcionales de investigación, de diseño y desarrollo, fundadas en estructuras de trabajo.

La gestión y la planificación de los proyectos necesitan integrarse con los dos modelos de trabajo: modelo gestionable y modelo de construcción.

Un modelo gestionable identifica las interacciones con la administración misma y los procedimientos y procesos de control.

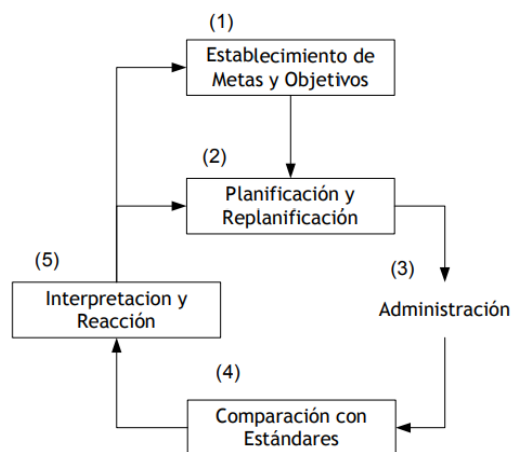


Figura 7: El Modelo de Gestión

Según Toro & Peláez (2017) un modelo de gestión de pruebas de software agrupa a microempresas, software y personas.

Los gestores de proyectos de software planifican la construcción, control de las actividades y asegurar que el trabajo es un estándar, está a tiempo y con el presupuesto. Una gestión buena no asegura el éxito, pero una mala conlleva a entregar un software tarde, el costo por encima del estimado y muy caro de dar mantenimiento.

La gestión de proyectos informáticos se diferencia de los proyectos tradicionales en que: El software es intangible. El Jefe de proyecto depende de la documentación a la mano

para evaluar su situación. No se cuenta con entendimiento total del proceso de construcción del Software. Los modelos que se aplicaron son parte de la ayuda para comprender.

La gerencia de proyectos software es llevada a cabo mayoritariamente en nuestro país mediante la guía PMBOK del Project Management Institute. En este trabajo de investigación se ha estudiado la manera de incluir en el modelamiento planteada además del PMBOK, el PRINCE2 y SCRUM, mientras que para la gestión de pruebas se ha tomado a TMMI versión 3.1, TPI-Next, ISO/IEC/IEEE 29119 y al “International Software Testing Qualifications Board” - ISTQB en sus enunciados y particularidades. Al totalizar dichos estándares, obtuvimos contar con el asunto de administración en proyectos de testing de software construido con el fin de ser utilizado por microempresas para alcanzar sus objetivos de los proyectos planificados.

El modelo de ciclos de mejora PHVA, organiza tareas en cuatro frentes: planear, hacer, verificar y actuar, con los cuales se plantean el mecanismo sistema de gestión de calidad para la gerencia de proyectos de testing en software.

El proceso revela esquemas generales en alto nivel de procedimientos y procesos tal es la guía para entender la comunicación de los procesos y encontrar los indicadores de todos ellos, para alcanzar las funciones trazadas.

2.2.2 MARCO NORMATIVO

A continuación, se describe el marco normativo que tiene alguna referencia a la presente tesis.

En el ámbito internacional

Norma ISO 9000:

“Normas para gestionar la calidad y el aseguramiento de calidad”

Un grupo de normas que brindan las especificaciones e indicaciones para el aseguramiento de la gestión de calidad en una organización.

Con las normas ISO 9000 se simplifica la relación mundial de tangibles, lo que está en mercados, un pedido para importar, a partir de esto el interés de exportadores por alinear a la norma. Para realizar la aplicación, se debe estudiar caso a caso, con apoyo experto.

Adicionalmente, el hecho de implementar un sistema gestionable para la calidad en la industria peruana con proporcionar mejores componentes de calidad para el mercado en su totalidad y los consumidores en general.

Norma ISO 9001:

“Modelo para el garantizar la calidad en el diseño, desarrollo, producción, instalación y post producción”

Muestra los requerimientos del software para gestionar calidad aplicable a toda la empresa que requiera mostrar suficiencia para entregar productos y servicios que cubran los requerimientos de usuarios finales y de aplicación y el objetivo de incrementar la aceptación.

ISO 9002:

” Modelo para la garantía de la calidad en producción e instalación, post producción”

El estándar ISO 9002 son medidas que admiten definir las exigencias mínimas a llevar a cabo por la organización que desee adquirir una certificación de clase mundial en el rubro de la calidad.

Norma ISO 9003:

” Modelo para asegurar la calidad en revisión final y testing”

Proporciona un manual para las compañías que deseen aplicar de la Norma ISO 9001:2000 en la adquisición, logística, construcción, operatividad y mantenimiento de software y soporte. Este estándar no adiciona requerimientos de la ISO 9001:2000.

En la actualidad "calidad" ha cobrado más importancia, los que consumen aguardan productos con una calidad capaces de satisfacer necesidades, solucionar problemas y ganar beneficios. En la industria del software, "calidad" aún no se consolida como el aspecto más importante del rubro.

Los 3 errores imponentes que existen dentro de la industria del software son el presupuesto para la depuración de un software, sobretiempo en la corrección del software y el fallo de entender los requerimientos del usuario final.

Actualmente la industria de la producción de software está implementando modelos para mejorar sus operaciones y corregir sus fallas. La esperanza es colocar el desarrollo de

software bajo un control estadístico para verificar cuáles son las tareas cíclicas que continuamente se tienen que programar, y que producen igual fin. Así, los procesos exitosos utilizados anteriormente pueden ser modelos base para la planeación de proyectos futuros, optimizando costos, incrementando la eficiencia y la productividad, desarrollando mejores productos de calidad y por consecuencia, haciendo beneficios tangibles para la organización.

Los modelos antes mencionados son los estándares de calidad ISO 9000 que han creado un grupal interés en la industria de software a raíz del acuerdo a un nivel mundial de muchas compañías.

Norma ISO 9004:

“Conjunto de Normas para la gestión de administración de calidad y lineamientos de sistemas de calidad”.

Dan eficacia y eficiencia del sistema de administración de calidad. La meta de este estándar es la mejora del rendimiento de la compañía y contentamiento de los usuarios y otros interesados.

Norma ISO/ IEC 14598

“Normas de Calidad de un Producto Software”

Proporciona un marco de trabajo para evaluar la calidad de todos los tipos de productos de software e indica los requisitos para los métodos de medición y para el proceso de evaluación.

Norma ISO/ IEC 9126

“Normas de Ingeniería de Software”

Establece un modelo de 2 componentes para calidad de software: a) Calidad interna y externa, y b) Calidad en utilidad. La primera parte del modelo trata de seis características para calidad interna y externa.

Norma UNE (Norma Europea)

Es un grupo de estándares de tecnologías de información y comunicaciones fundadas por el Comité Técnico de Normalización Español, para ser parte de los entes y dependencias involucradas en las investigaciones de este comité, este tipo de comités en la mayoría de los casos están hechos por AENOR (La Asociación Española de Normalización y Certificación), productores, clientelas y consumidores, dirección, recintos y sedes para investigación.

Es preciso mencionar que las Normas ISO son desarrollados por la ISO (International Organization for Standardization), Organización Internacional para la Estandarización, organización creada el 23 de febrero de 1947, es la organización dedicada de incentivar la producción de estándares mundiales de construcción, negocio y relaciones para todas las ramas industriales a excepción de la electrónica y eléctrica. Lo primordial es encontrar la normalización de productos y seguridad para las organizaciones a nivel mundial.

La ISO es la unión de institutos de normas nacionales de 157 países, sobre el fundamento de un representante por país, con una Secretaría Central en Suiza. La Organización Internacional de Normalización (ISO), con base en Ginebra está compuesta por delegaciones subdivididos en una serie de comités encargados de implementar las líneas guía que ayudaran al mejoramiento del ambiente.

Norma ISO 21500

De la creación de un comité técnico se encamino en conducir a la Norma ISO de Project Management para incluir las mejores prácticas, a los fines inclusivos de conocimiento en un estándar mundial mediante el consenso entre las partes interesadas y funcione como una línea guía en la cual se fundamente verdaderamente las mejores buenas prácticas gestionando proyectos comunes para todo el mundo.

La ISO 21500 nace como el estándar para la administración de proyectos. Se trata de un glosario de buenas prácticas que se unen en esta norma como resultado de la investigación en los años para integrar el estándar. La ISO 21500 no está en operatividad y apunta a que el entregable final será para el 2012, aunque hoy puede obtenerse la versión preliminar de la ISO 21500.

En el ámbito nacional

Normas Técnicas Peruanas 12207, NTP ISO/IEC12207:

“Procesos del Ciclo de Vida de un Software “

Estándar organizado en 3 categorías de procesos. Principales, de Apoyo y Organizativos. Sirve para encaminar toda una empresa para dar soporte en el logro de sus metas estratégicas. En todas las empresas los procesos se pueden organizar según la cadena del valor de Michael Porter.

Norma Técnica Peruana 10006, NTP ISO/IEC 10006:

“Lineamientos para la Gestión de la calidad en proyectos” Esta Norma Técnica Peruana encamina la aplicación de la gestión de la calidad en proyectos.

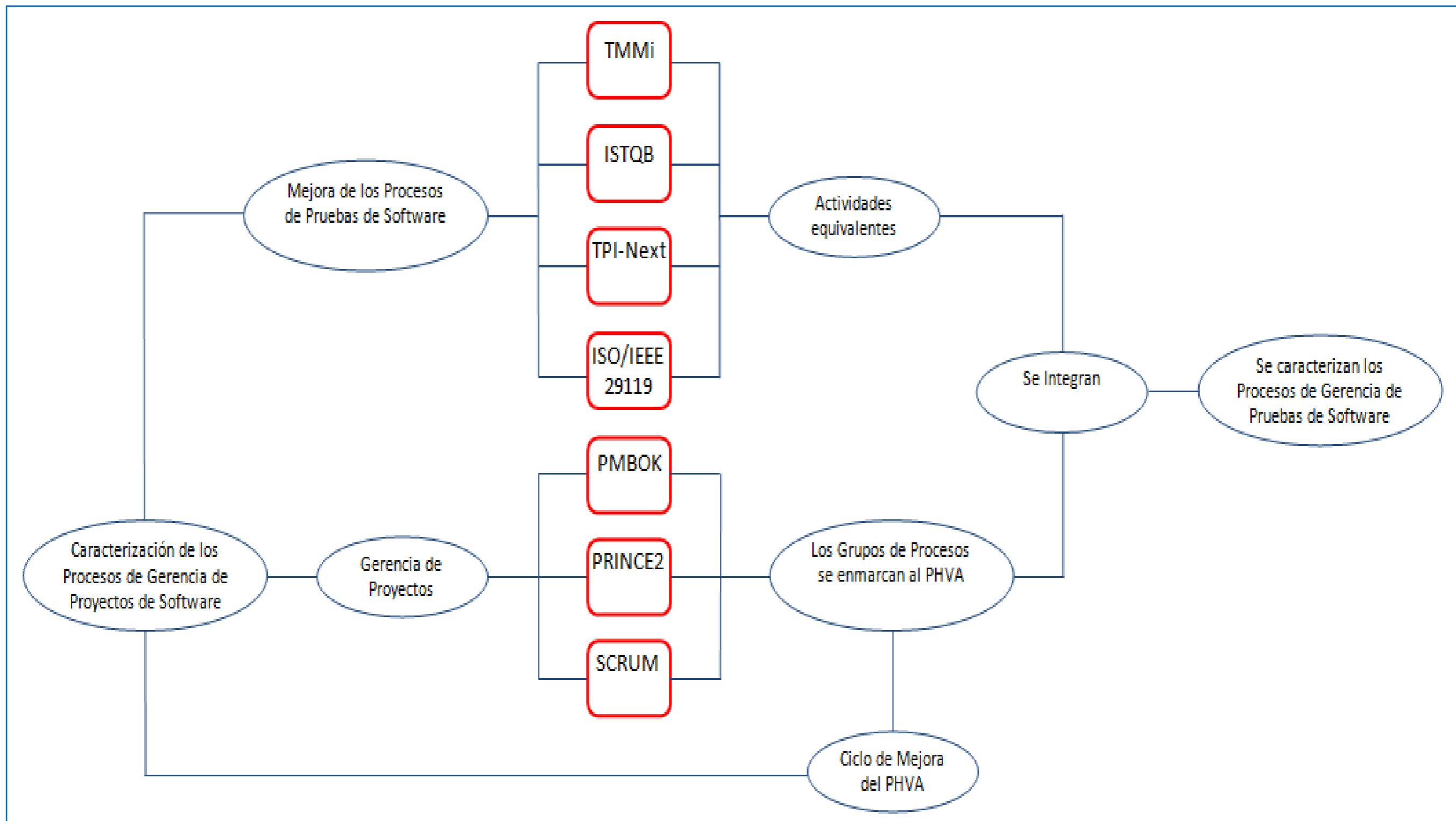


Figura 8: Mapa General de las bases teóricas especializadas sobre
Fuente: Elaboración propia

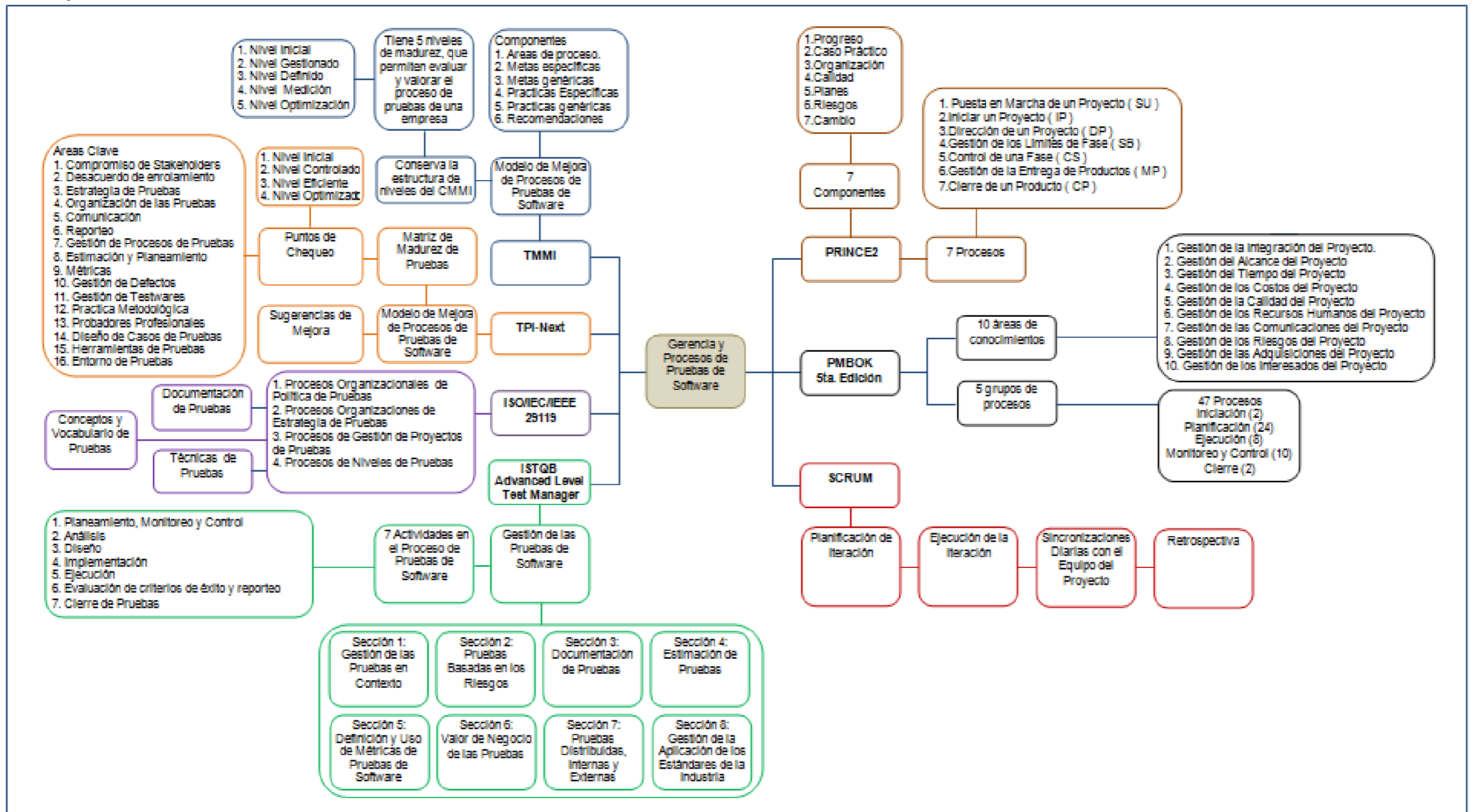


Figura 9: Mapa Detallado de las bases teóricas especializadas sobre el tema
Fuente: Esta investigación

Tabla 3:

Modelos de Calidad más usados por Pymes

Modelo	Año	Institución	País	Procesos
CMMI-DEV	2011	Instituto de ingeniería de software	Estados Unidos	22
ISO/IEC 12207	2008	Organización internacional para la estandarización	Internacional	43
MPS.BR	2011	Asociación para la promoción de la excelencia del software brasilero	Brasil	19
Competisoft	2008	Proyecto Competisoft	Latinoamérica	9
MoProSoft	2005	Asociación mexicana para la calidad en ingeniería de software	México	8

Fuente: Galvis (2013)

Moprosoft

El Modelo de Procesos para la Industria de Software (MoProSoft) en México, fue creado el año 2003 para fomentar la estandarización en las empresas de desarrollo de software

mexicanas a través de la incorporación de las mejores prácticas en gestión de ingeniería de software y fue elevado a nivel de norma el 2005 convirtiéndose en la norma mexicana NMXI-059/02-NYCE. Su principal objetivo es proporcionar un modelo basado en las mejores prácticas internacionales para la estandarización de los procesos de la industria de software en México que en su gran mayoría están compuestos por pequeña y mediana empresa (Oktaba et al. , 2005).

Según Oktaba et al. (2005), el modelo MoProSoft pretende apoyar a las organizaciones en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua a través de sus tres categorías: 1) Alta Dirección, 2) Gestión y 3) Operación (ver Figura 9).

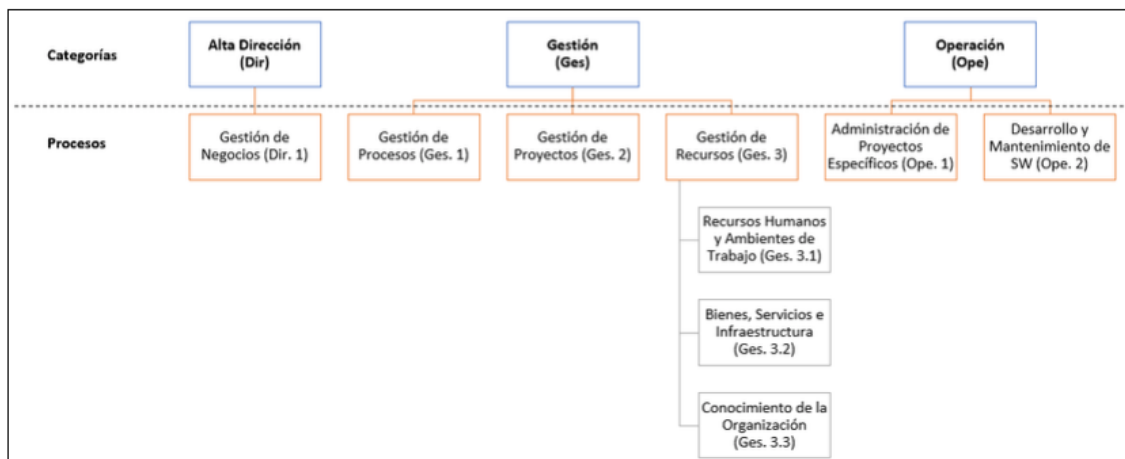


Figura 9: Categorías y procesos de MoProSoft
Fuente: Reyes et al. (2009)



Figura 8: Niveles MoProSoft
Fuente: <http://e-processmexico.com>

Según datos del último reporte NYCE (2018), en la actualidad se tiene un total de 478 empresas certificadas con MoProSoft, de las cuales el 96% se encuentran entre los niveles 1 y 2 (ver Figura 4). En el reporte de NYCE (2018) no se pudo determinar qué porcentaje de las empresas certificadas son PyME, pero si se pudo evidenciar que existen empresas multinacionales y empresas grandes en el listado reflejándose en las 21 empresas que se encuentran certificadas entre los niveles 3-5.

Modelo MPS.BR

MPS.BR es un programa para la mejora de los procesos de software de las PyME de desarrollo en el Brasil creado en diciembre de 2003 por la Asociación para Promoción de la Excelencia del Software Brasileño (SOFTEX, 2012).

Este programa se centra en mejorar la competitividad de las micro, pequeñas y medianas empresas de desarrollo de software, mejorando la calidad de los productos de software y sus servicios asociados, como en los procesos de producción y distribución de software. Siendo su objetivo la mejora del proceso de software en los países de Brasil, Colombia, México y Perú, con foco en las Micro, Pequeñas y Medianas empresas a un costo accesible.

El modelo MPS.BR está conformado por cuatro componentes: por el 1) Modelo de Referencia MPS para Software o “MR-MPS-SW”, 2) Modelo de Referencia MPS para Servicios o “MR-MPSSV”, 3) Método de Evaluación o “MA-MPS” y 4) Modelo de Negocio para Mejora de Proceso de Software y Servicios.

Cada componente del modelo está especificado en sus guías respectivas y basados en estándares internacionales como se muestra en la Figura 9

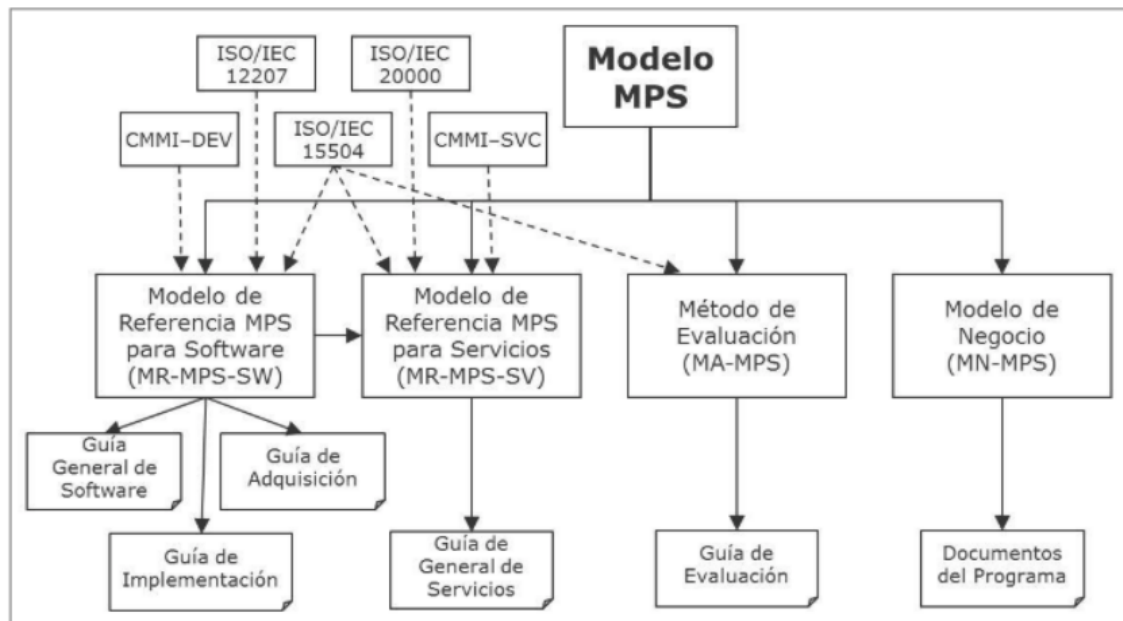


Figura 9: Componentes del modelo MPS
Fuente: SOFTEX (2012)

Este modelo define niveles de madurez que son una combinación entre procesos y

capacidades, estos niveles son etapas de mejora de la implementación de procesos en la organización. El MR-MPS-SW define siete niveles de madurez: A (En Optimización), B (Gestionado Cuantitativamente), C (Definido), D (Ampliamente Definido), E (Parcialmente Definido), F (Gestionado) y G (Parcialmente Gestionado). La escala de madurez se inicia en el nivel G y progresa hasta el nivel A. Para cada uno de estos siete niveles de madurez se atribuye un perfil de procesos que indican adonde la organización debe colocar el esfuerzo de mejora (SOFTEX, 2012).

Los procesos están definidos como el conjunto de propósitos y resultados. A su vez, los propósitos están definidos por atributos de procesos que especifican el grado de refinamiento con que el proceso es ejecutado en la organización. En el modelo MPS existen 9 atributos de procesos (AP) que se aplican a los niveles.

Según los datos de SOFTEX (s.f.) en la actualidad existen 110 empresas de desarrollo de software certificadas en el Brasil como MPS-SW: 48 en el nivel G, 40 en el nivel F, 2 en el nivel E, 19 en el nivel C y 1 en el nivel A. En Figura 10 podemos ver el porcentaje de empresas por nivel del modelo MPS.

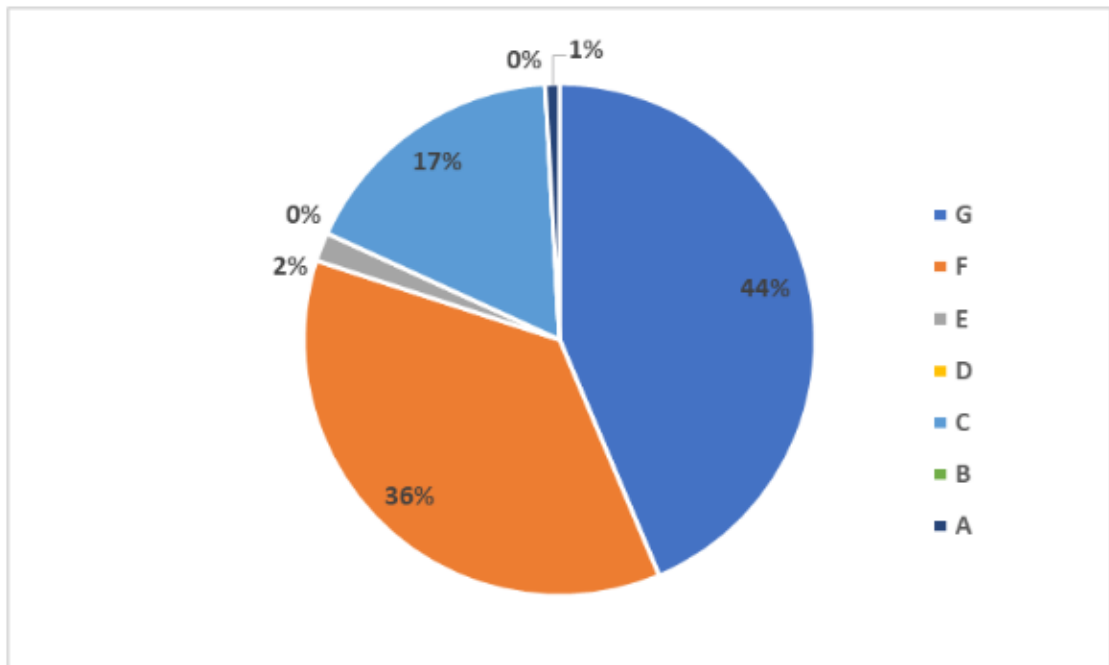


Figura 10: Porcentaje de empresas por niveles
Fuente: Elaboración propia

Modelo Competisoft

Competisoft es la denominación de un proyecto iberoamericano en el que participó la Pontificia Universidad Católica del Perú. El mismo que tenía el objetivo de incrementar la competitividad de la industria de software en la región ajustado a las necesidades y características principales de las PyME (costo, tamaño y personal) (Morillo et al., 2012; Maidana et al. , 2008; Competisoft, 2008).

(Dávila A. B., 2012) está dividido en tres modelos: 1) Modelo de procesos, 2) Modelo de mejora y 3) Modelo de evaluación. El modelo de procesos optó como base MoProSoft teniendo en cuenta las experiencias de la industria mexicana. El modelo de mejora de procesos se denominó PMCompetiSoft y está basado en AgilSPI que es un marco de mejora creado para la industria del software en Colombia. PMCompetiSoft está compuesto por 5 macro actividades: 1) instalación del programa, 2) diagnóstico, 3) formulación, 4) mejora y 5) revisión del programa. El modelo de evaluación se denominó EvalSoft y está basado en la ISO/IEC 15504. Este modelo evalúa la capacidad del proceso en una escala del 0 a 5, siendo 0 el valor de capacidad más bajo, lo que significa que no se alcanzó el propósito del proceso de mejora y el valor 5 se asocia al nivel de capacidad más alto significando que se lograron las metas establecidas (Morillo, 2012).

La implementación de un SPI en una organización tiene implícita la incertidumbre de su éxito o fracaso pues el problema no es por la falta de modelos sino por la falta de una estrategia efectiva de implementación. En iniciativas de mejora se precisa de conocimientos basados en la experiencia (Maidana, 2008). Es por esto que Competisoft establece un modelo de mejora y un modelo de evaluación.

Dado que Competisoft basa su modelo de procesos en MoProSoft, sus categorías y procesos son los mismos. Davila (2012) resume que después de las fases del proyecto Competisoft en el Perú se lograron certificar 5 empresas entre las regiones de Lima, Arequipa y Trujillo . Coque-Villegas et al. (2017) indica que Competisoft es un modelo aplicado en varios países de Iberoamérica como: España, Argentina y Colombia. Detalla que en Colombia se pudo implementar el modelo en 3 de las 5 empresas donde se procuró usar. Maidana (2008) desarrolla una recopilación de 15 categorías de factores críticos de éxito en la implementación de un proceso de mejora, pero hace énfasis en que los principales problemas son:

Una marcada resistencia al cambio y en algunos casos se tiene la percepción de que los nuevos procesos son una mera burocracia de documentos que ocasionarían pérdidas de tiempo.

Perdida de enfoque en la implementación del modelo, asociado al hecho que las PyME tienen que sobrevivir en un mercado complejo y por el que un proyecto de mejora pasa a un segundo plano. Coque-Villegas (2017) concluye que el modelo está concebido para ser aplicado a las PyME, pero su éxito depende del compromiso de la alta gerencia, de la actitud, la cultura empresarial y la predisposición del personal involucrado en la implementación. En este caso se está de acuerdo en que la implementación de cualquier modelo de mejora de proceso de software para cualquier PyME es dependiente de los actores involucrados. Aunque se cuenta con los casos de éxito de estudio que mencionan Coque-Villegas (2017) y Davila (2012), no se han podido encontrar más trabajos que mencionen implementaciones de Competisoft en Latinoamérica en fechas actuales y no se ha encontrado una lista oficial de empresas desarrolladoras de software certificadas.

ISO/IEC 29110

En el 2011, el nuevo estándar de ingeniería de software para PyME (ISO/IEC 29110) fue aprobada por la Organización Internacional de Estandarización (ISO por sus siglas en inglés) (Laporte, 2015). La principal característica del público objetivo de la ISO/IEC 29110 es el tamaño de la empresa, no debe pasar de 25 personas (O'Connor & Laporte, 2017; ISO, 2016). Sin embargo, también se toma en cuenta otras características como el modelo de negocio, factores situacionales y niveles de riesgos del negocio. (Coque-Villegas, 2017)

La ISO/IEC 29110, esta pensada para ser usada con cualquier ciclo de vida de desarrollo de software y hace una caracterización genérica de las empresas en cuatro perfiles: Inicial (Entry): Las PyME que están en este perfil, trabajan en pequeños proyectos con no más de seis personas o la empresa tiene no más de tres años de vida.

Básico (Basic): Las PyME que están en este perfil, trabajan en grupo en aplicaciones simples ya sea para uso interno o para terceros.

Intermedio (Intermediate): Las PyME que están en este perfil, trabajan en más de un proyecto en paralelo con más de un grupo de trabajo.

Avanzado (Advanced): Las PyME que están en este perfil, son empresas en crecimiento y competitivas en el mercado.

Coque-Villegas et al. (2017) especifica que este tipo de certificaciones internacionales pueden mejorar la credibilidad, la competitividad y el acceso a los mercados nacionales e internacionales. Esto se puede ver en Ecuador, donde en el año 2015 se adoptó la norma ITE INENISO/ IEC TR 29110-1 por el Comité Técnico de Normalización del INEN (Registro Oficial, 2015).

Dada la bibliografía consultada sobre la ISO/IEC 29110, vemos que es un modelo que ofrece a las PyME mayor facilidad de implementación por sus perfiles que son fáciles de asociar según el tipo y forma de trabajar de las empresas de desarrollo de software. Se tiene bastante expectativa sobre esta norma ya que al ser un subconjunto de la ISO/IEC 12207 es escalable con el crecimiento y transición de las PyME a empresas grandes.

Tabla 4:

Procesos relacionados con los modelos de calidad

Modelo	Proceso relacionado con la administración del conocimiento
MoProsoft	Administración de los procesos Administración de recursos humanos y de los ambientes de trabajo Conocimiento organizacional
MPS.BR	Administración de la configuración Definición de los procesos de la organización Administración de recursos humanos Desarrollo de la reutilización
Competisoft	Administración de los procesos Administración de recursos humanos y de los ambientes de trabajo Conocimiento organizacional
CMMI-DEV	Administración de la configuración Definición de los procesos de la organización Entrenamiento de la organización
ISO 12207	Administración de la configuración del software Proceso de resolución de problemas de software Administración de modelos de ciclo de vida Administración de recursos humanos Administración de reutilización de activos Dominio de ingeniería

Fuente: (Galvis-Lista, 2013)

2.3 BASES ESPECIALIZADAS TEORICAMENTE SOBRE EL TEMA

2.3.1 ESTÁNDARES EN GESTIÓN DE PROYECTOS

2.3.1.1 Cuerpo de Conocimiento de la Gestión de Proyectos (PMBoK)

El PMBoK (Project Management Body of Knowledge), es una norma en la gestión de proyectos del Project Management Institute (PMI). Se encuentra disponible en 11 lenguas, es una norma conocida mundialmente en la gestión de proyectos, la cual provee patrones y buenas prácticas, y que al aplicarlas elevan el horizonte de logro de una enorme diversidad de proyectos en distintas clases de industrias y que son útiles a una extensa categoría de proyectos, agregando construcción, software, ingeniería, educación, etc.

El PMBoK en su edición actual, la quinta, provee de 47 procesos y 10 áreas de conocimiento. Léxico alineado y una estructura consistente con las mejores prácticas que han madurado de edición.

La calidad se trata en 3 procesos, los cuales se menciona a continuación:

Fase de Planificación: Proceso 8.1 Planificación de la Gestión de Calidad

Fase de Ejecución: Proceso 8.2 Realización del Aseguramiento de Calidad

Fase de Monitoreo y Control: Proceso 8.3 Controlar la Calidad.

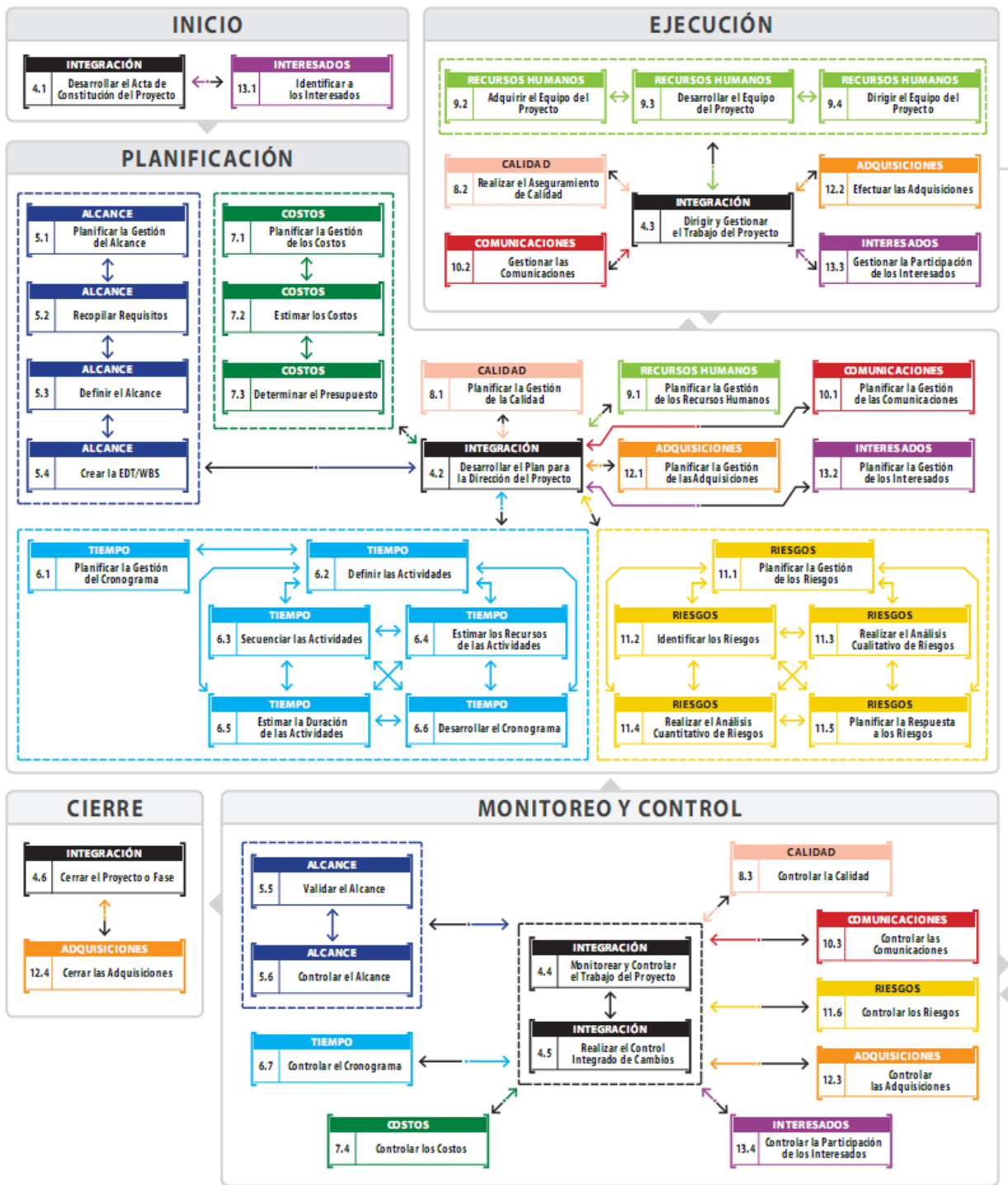


Figura 10: Procesos del PMBoK 5ta. Edición
Fuente: Project Management Institute

2.3.1.2 Proyectos Entornos Controlados (PRINCE2)

El PRINCE2 (Projects IN Controlled Environments (PRINCE2)), transformar proyectos, que manejan una carga vital y de incertidumbre, en entornos controlados, Más que un conjunto de buenas prácticas, PRINCE2 propone un modelamiento de gestión de proyectos que cubre, mediante lo que se conoce como temas, la Calidad, el Cambio, la estructura de roles del proyecto (Organización), los planes, el Riesgo del proyecto, justificado por un Business Case (o estudio de viabilidad) que debe ser revisado durante el ciclo de vida del proyecto y justificar en todo momento el proyecto como consecución de los beneficios estimados.

La utilización de PRINCE2 para proyectos, pudiendo aplicarse en proyectos de todo tema, como Producción de software.

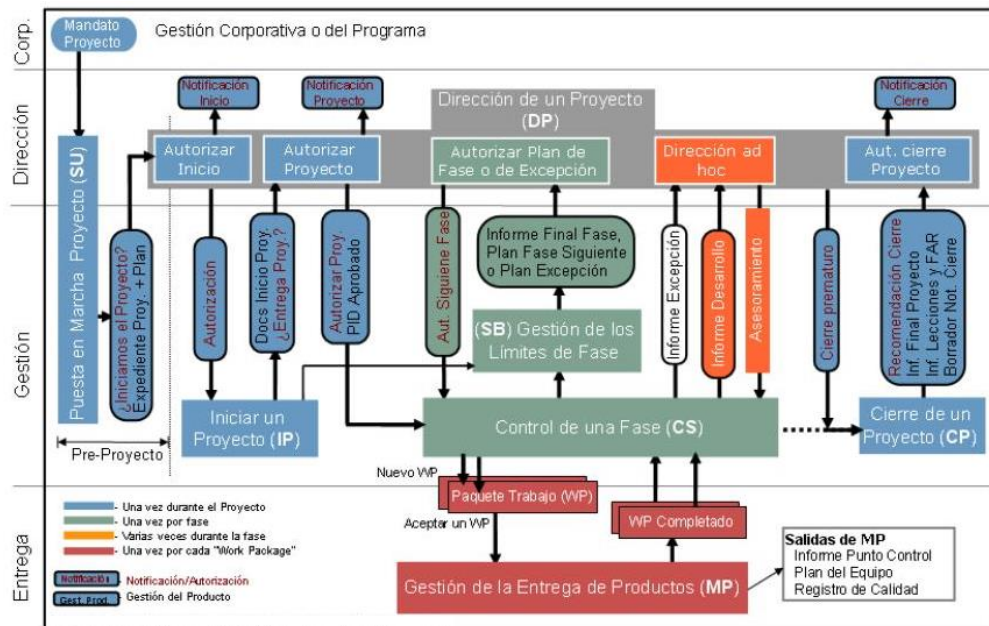


Figura 11. El Modelo de Procesos PRINCE2

Fuente: PRINCE 2

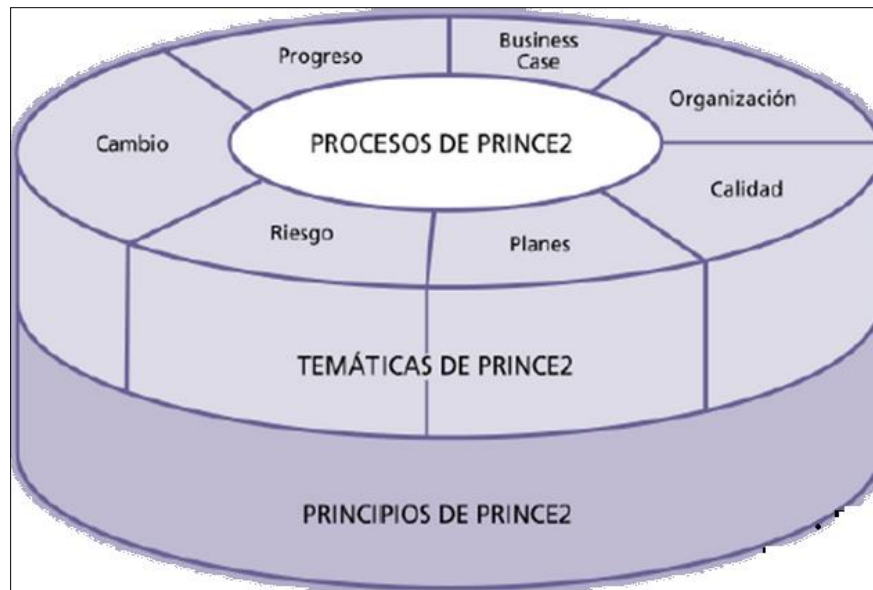


Figura 12. Los Procesos PRINCE 2

Fuente: PRINCE 2

2.3.1.3 GESTIÓN DE PROYECTOS MEDIANTE METODOLOGÍA ÁGIL (SCRUM)

SCRUM está orientada a la gestión de procesos de desarrollo de software, puede ser utilizado en otro tipo de proyectos.

Según González Yepes, Calvache Pardo, & Gómez Gómez (2015) , mencionan que el mundo cambiante en el que nos encontramos se ha tornado muy cambiante en cuanto a tecnología, lo cual es el motivo principal para optar por usar un método del agilísimo. La administración no se fundamenta en la trazabilidad de un plan único sin cambios sino en adaptarlo de manera frecuente a los escenarios según el transcurso del proyecto. De esta manera es útil entornos en los que se los requerimientos son dinámicos y que se requiere velocidad y adaptación, tales como la construcción del software. SCRUM se fundamenta a esta implicancia tan urgente desde ya varios años en la industria de la producción de software.

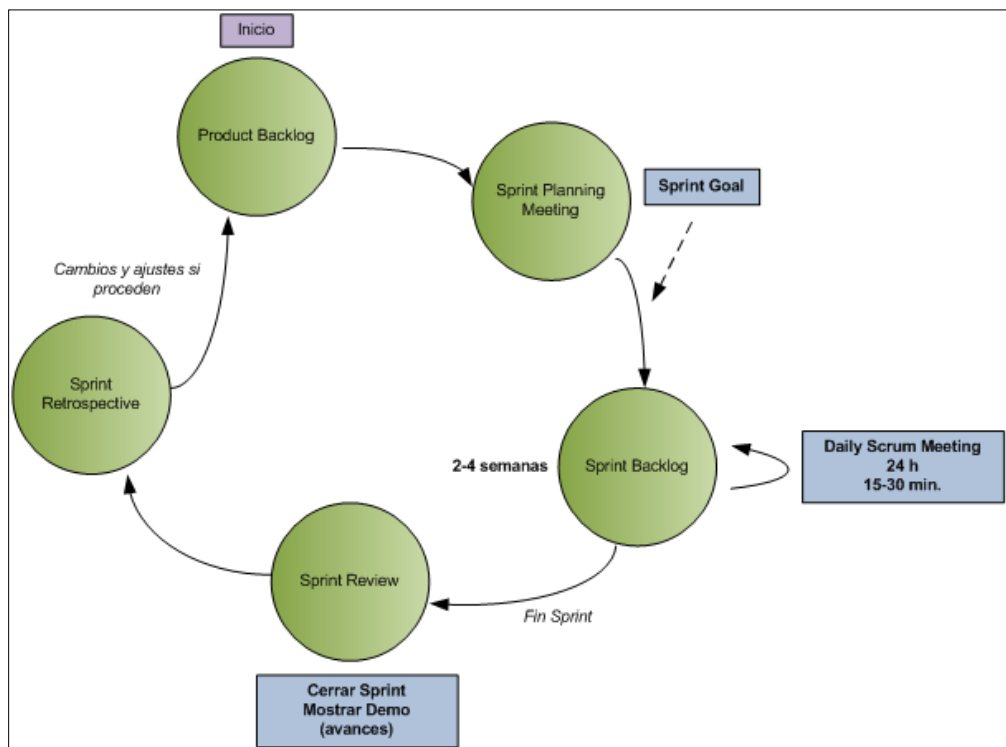


Figura 13. El Modelo de Gestión de Proyectos SCRUM

Fuente: SCRUM Alliance

A continuación se explica en que consiste la metodología SCRUM:

Esta metodología se basa en realizar iteraciones cuyo intervalo no sea mayor 4 semanas, dicho intervalo es considerado un Sprints.

En la duración del Sprint el objetivo del equipo es lograr una ampliación notable, usable, entregable del proyecto.

Al iniciar un Sprint se debe realizar una reunión de planificación conocido como “Sprint Planning Meeting” en la cual se examina el Product Backlog del proyecto (listado de los objetivos/requisitos priorizados). En dicha reunión el dueño del producto “Product Owner” identifica los objetivos que tiene planificado ejecutar y los sustenta al equipo scrum.

En el intervalo de ejecución del Sprint no se puede cambiar el alcance del sprint, se intenta no alterar los requisitos hasta que finalice la duración del Sprint.

A diario se lleva a cabo una reunión (Daily Meeting) que normalmente tiene una duración de 15 minutos o menos, en la cual los integrantes del equipo contestan concisamente estas interrogantes: ¿Qué hizo desde el último daily meeting? ¿Ejecutó todo lo planeado? ¿Qué actividades o tareas va a realizar a partir de este instante? ¿Cuáles son los problemas presentes o que problemas cree que va a presentarse en el cumplimiento de los objetivos del sprint?. Un soporte del equipo es contar con la lista de tareas de la iteración, en la cual se reformula el estado y el esfuerzo pendiente asociado a cada tarea, también se actualiza el gráfico de avance o “burndown” de horas pendientes en la iteración.

Al término del Sprint, se lleva a cabo una reunión de retrospectiva “retrospective meeting”, en esta reunión el equipo examina todo lo que aconteció durante la ejecución del Sprint considerando los hechos positivos así como también los hechos negativo. Con esto se logra mejorar la productividad, mejorar también las habilidades del equipo y finalmente mejorar la calidad del producto que se está creando.

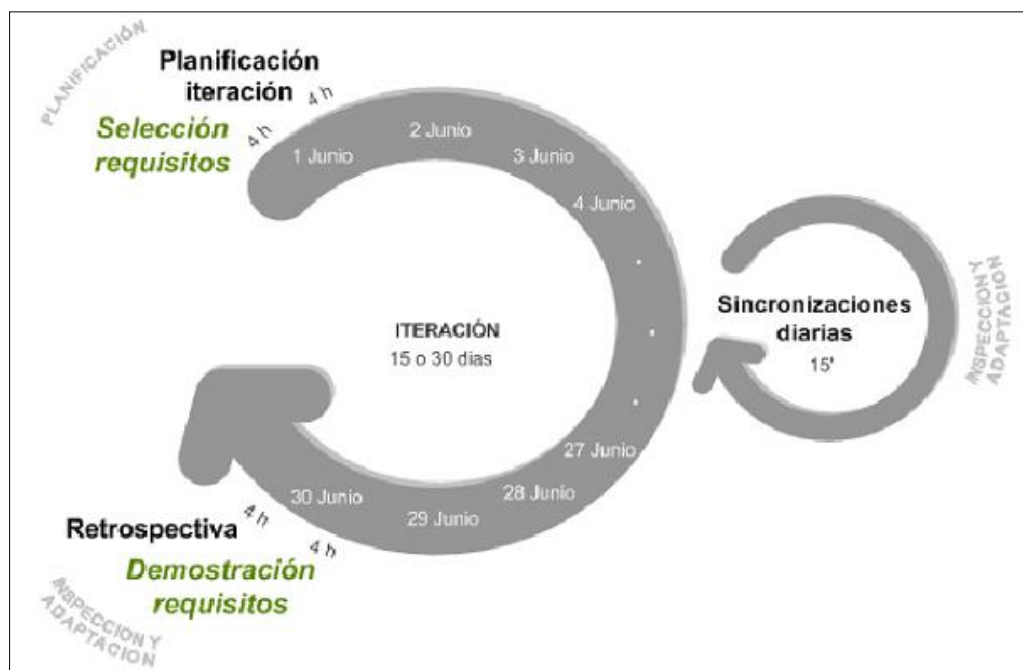


Figura 14. Procesos SCRUM

Fuente: SCRUM Alliance

2.3.2 ESTÁNDARES DE PRUEBAS DE SOFTWARE PARA LA CALIDAD DE SOFTWARE

2.3.2.1 Modelo de Madurez de Pruebas de Software Integrado (TMMi)

El Testing Maturity Model Integration (TMMi) es un método creado por el TMMi Foundation como un grupo de buenas prácticas para el mejoramiento de los procesos de testing de software, con el objetivo de contribuir a la estandarización que ofrece el CMMI-DEV, orientando en los inconvenientes para solucionarlos por parte de ingenieros, jefes y también especialistas en testing.

TMMi considera que las actividades de testing tratan desde un 35 a un 45% del presupuesto destinado para un proyecto de construcción de software, por esto es primordial superarlo, considerando buenas prácticas que engrandezcan y complementen a las normadas por el CMMI®.

TMMI adicionalmente propone conceptualmente niveles de madurez para superar procesos en problemas utilizando sus principios que lo sustentan. Asimismo, lleva consigo el interés de generar impacto en cuestión de calidad del software como producto, la productividad y la dedicación en el ciclo de vida de las tareas de testing de software. En el mismo sentido su marco de referencia TMMi, hace posible que el proceso de testing evolucione de procesos caóticos a procesos maduros, controlados, con objetivos preventivos frente a los defectos de software.

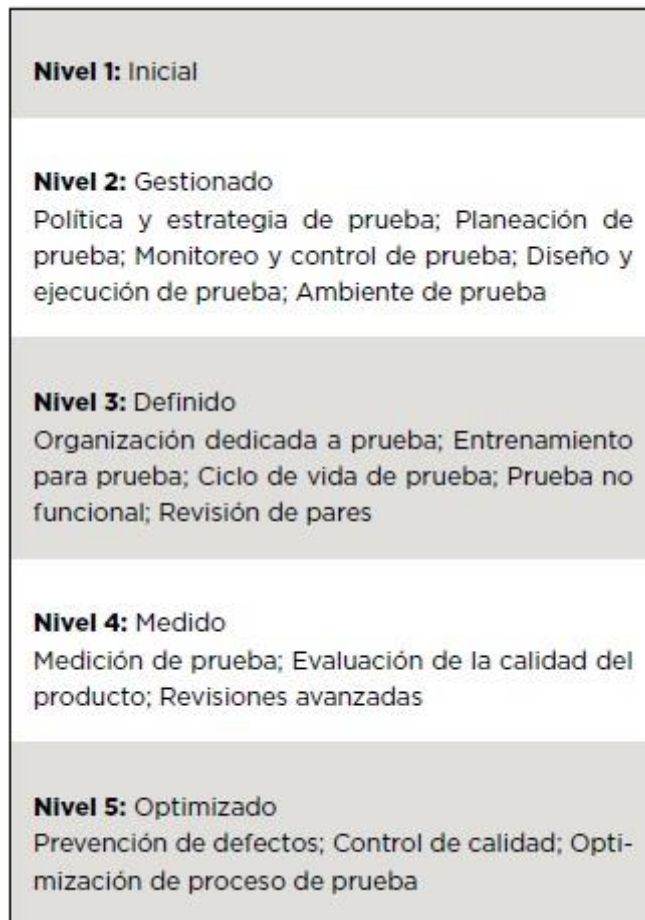


Figura 15. Estructura de los niveles y áreas de TMMi

Fuente: TMMi

2.3.2.2 Modelo de Madurez de Pruebas de Software Integrado (TPI- Next)

TPI NEXT® (Test Process Improvement) es un modelo creado por Sogeti en 1998 y convertido en un estándar internacional de facto por la comunidad de Calidad del Software.

El modelo hace posible conocer el grado de madurez del proceso de testing actual identificando aspectos más importantes y menos importantes.

Un valor inicial que podemos obtener del modelo es la estructuración de las áreas donde vamos a aplicar las mejoras, lo que nos permite evaluar todas siguiendo el mismo método y nos aporta una visión completa del escenario sobre el que estamos trabajando. TPI NEXT® agrupa dichas áreas en tres bloques: Relación con los Stakeholders (cada uno de los interesados en las actividades), Gestión de las Pruebas y Profesionalismo del Tester.

Para dotar de una mayor granularidad al proceso se desglosan los bloques anteriores en dieciséis áreas clave de las que algunos ejemplos representativos son: Compromiso de los Stakeholders, Comunicación, Informes, Métricas, Gestión de Defectos, Modelamiento o Herramientas de Test. La valoración de cada una de las áreas clave se materializa a través de un cuestionario con 157 checkpoints; este cuestionario se cumplimenta durante la realización de entrevistas a perfiles identificados como clave y pertenecientes a las áreas implicadas, directa o indirectamente, en el proceso de pruebas. Sus preguntas se encuentran agrupadas en clústeres lo que permite medir el estado actual y situarlo dentro de los niveles Inicial, Controlado, Eficiente y Optimizado que marca el proceso.



Figura 16. Estructura de TPI Next

Fuente: TPI Next

Una vez que hemos identificado la situación actual (As Is) es el momento de identificar las acciones necesarias para avanzar cada una de las áreas clave hacia el nivel marcado como objetivo. En este apartado es importante tener en cuenta las características de la compañía en la cual estamos aplicando TPI NEXT®, ya que no todas pueden tener como objetivo principal alcanzar el nivel optimizado en todas las áreas clave. Dependiendo del tipo de compañía y negocio puede que en determinadas áreas sea suficiente con mantenerse en niveles anteriores para lograr sus objetivos de negocio.

TPI NEXT® nos permite adquirir, en base a las entrevistas y al informe de situación actual, un gran conocimiento del entorno, que nos va a facilitar adaptar las medidas recomendadas por el método a la realidad de la compañía. Esto nos va a ayudar a que las propuestas que realicemos tengan un sentido y profundidad acorde a las necesidades individuales de cada uno de los clientes sobre los que apliquemos el método.

En general, TPI NEXT® nos aporta un marco de trabajo sobre el cual estructurar las necesidades y medidas a aplicar, fijando unos objetivos y un camino a seguir. Para ello, las propuestas se estructuran en el corto, medio y largo plazo con lo que en cualquier momento es posible obtener el estado actual y verificar si se están cumpliendo los objetivos marcados para cada una de las etapas. Así mismo, permite al cliente realizar una proyección de la situación de su proceso de prueba, conociendo en todo momento el nivel en el que se encuentra, la situación en la que quiere estar y los pasos que debe ir dando dentro de un escenario temporal.

Esta evaluación se recomienda realizarla cíclicamente, valorando diferentes niveles de profundidad, de modo que las acciones de mejora puedan adaptarse a los distintos escenarios y evoluciones que se producen en las empresas y así alcanzar los objetivos marcados inicialmente.

2.3.2.3 Estándar Internacional para Pruebas de Software ISO/IEC/IEEE 29119

El objetivo del estándar ISO/IEC/IEEE 29119 referente al testing de software es brindar un estándar para el testing definiendo el vocabulario, los procesos, la documentación, mecanismos técnicos, así como también un modelo evaluativo para el proceso de testing que se puede hacer uso en cualquier ciclo de vida de la construcción de un software.

El estándar está basado en el modelo de proceso en 3 niveles fundamentado en los riesgos para el testing que brinde guías para la construcción de estrategias de testing organizadas junto con políticas, administración de proyectos de testing considerando diseño estratégico de testing del proyecto / niveles y planes así como la trazabilidad, control de testing, y procesos de testing evolucionado como guía de análisis y diseño de testing, entornos para testing y su configuración así como también su mantenimiento, las ejecuciones de testing, informando los resultados. Hoy se construyen, probando y revisando en más de 30 países con grupos de trabajo encargados de mejorar la norma.

La ISO/IEC/IEEE 29119 está conformada por 4 secciones:

Parte 1: Definiciones y Vocabulario, la misma que tiene por objetivo brindar un panorama amplio de la norma y de la teoría general de testing de software y brindar un glosario de términos de testing de software abarcando el testing en el ciclo de vida de la producción de software.

Parte 2: Proceso de Prueba, la cual define un marco de testing como proceso general utilizable en toda construcción de software y ciclo de vida del testing. Este proceso se fundamenta en un proceso de testing de capas de alcance.

Parte 3: Documentación de prueba, la cual cubre la documentación de pruebas en el ciclo de vida de la producción del software. Lo cual incluye documentación personalizable que cubra las etapas del proceso de testing.

Parte 4: Técnicas de ensayo, la cual abarca un conjunto de mecanismo cambiantes en común de testing de software: Basados en Especificaciones Técnicas de testing y Técnicas basadas en la estructura de prueba.

La norma ISO/IEC/IEEE 29119 engloba a los actuales estándares IEEE BSI para el testing de software:

IEEE 829 Documentación de prueba

IEEE 1008 Unidad de Pruebas

BS 7925-1 Vocabulario de términos en Pruebas de Software

BS 7925-2 Componente de pruebas de software estándar



Figura 17. Partes de la ISO/IEC/IEEE 29119

Fuente: IEEE

2.3.2.4 El Cuadro Internacional de Cualificaciones de Pruebas de Software (ISTQB) Nivel Avanzado

La certificación International Software Testing Qualifications Board (ISTQB) fue formada por miles de especialistas en testing de software de calidad mundial. ISTQB fue fundada en noviembre de 2002 está registrada legalmente en Bélgica. La certificación International Software Testing Qualifications Board (ISTQB) Nivel Avanzado está para los expertos que alcanzaron un punto alto en su carrera de testing. Incluyendo expertos con funciones de testers, analistas de calidad, ingenieros en pruebas, consultores de testing, administradores de testing, testers de pruebas de aceptación y programadores de software. Advanced Level es apropiada para las personas que se interesan por un conocimiento más detallado y especializado de testing de software. La certificación Advanced Level es para las personas que han logrado un punto avanzado en el testing de software en base a su experiencia.

ISTQB * – ADVANCED LEVEL TEST MANAGER						
Testing Process	Test Management	Reviews	Defect Management	Improving the Testing Process	Test Tool and Automation	People Skills
Test Planning, Monitoring and Control	Risk-based Testing	Management Reviews and Audits	Defect Lifecycle	Test Improvement Process	Tool Selection	Individual Skills
Test Analysis and Design	Test Documentation	Managing Reviews	Defect Report Information	Improving the testing process with CMMi,TPI,CTP,STEP	Tool Lifecycle	Test Team Dynamics
Test Implementation and Execution	Test Estimation and Test Metrics	Metrics for reviews	Assessing Process Capability with Defect Report Info		Tool Metrics	Fitting Testing Within an Organization
Evaluating Exit Criteria and Reporting	Business Value of Testing	Managing Formal Reviews				Motivation
Test Closure Activities	Distributed, Outsourced and Insourced Testing					Communication
	Managing the Application of Industry Standards					

Figura 18. Estructura del ISTQB Nivel Avanzado

Fuente: ISTQB

2.3.3 ESTÁNDARES DE MODELOS DE EVALUACIÓN DE PROCESOS DE PRUEBAS DE SOFTWARE

2.3.3.1 Procesos de Evaluación de Pruebas Estandarizados (STEP)

STEP es principalmente un modelo de referencia de contenido que se basa en la idea de que la prueba es un ciclo de vida y que es una actividad que se inicia durante la formulación de requisitos y continúa hasta la implantación del sistema. STEP subraya "la prueba luego de código" mediante el uso de una estrategia de evaluación basado en los requisitos de garantizar que la creación temprana de casos de prueba, valida la especificación de requisitos antes de diseñar y codificar. STEP incluye premisas básicas, las cuales son:

- Una estrategia de evaluación basada en los requisitos.
- La prueba comienza al principio del ciclo de vida.
- Las pruebas se utilizan como requisitos y modelos de uso.
- El diseño de testware lleva el diseño de software.
- Los defectos se detectan antes para prevenirse por completo.
- Se analizan sistemáticamente los defectos.

- Los Testers y desarrolladores trabajan juntos.

Los procesos de evaluación STEP están definidos acorde a tres principales fases: (1) Alcance y Estrategia de Pruebas, (2) Preparación de Pruebas, (3) Pruebas, Resultados, y Reporte Final y una cuarta fase opcional (4) Integración y Desarrollo que es determinado por el sponsor. Cada fase de STEP tiene diferentes objetivos, acciones y documentos entregables asociados.

Los Checkpoints separan las fases, y cada fase debe ser completada antes que la siguiente se inicie. Estos Checkpoints ayudan asegurar la integridad de la evaluación.

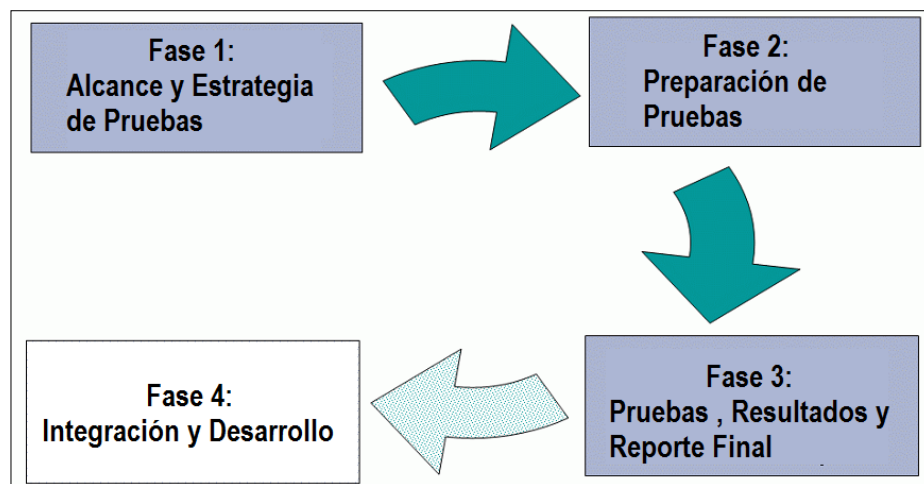


Figura 19. Fases de Evaluación de STEP

Fuente: STEP

2.3.3.2 Procesos de Pruebas Críticos (CTP)

La premisa básica del modelo de evaluación de los procesos de prueba críticos (CTP) es que ciertos procesos de prueba son críticos. Estos procesos críticos, si se realizan bien, apoyarán a los equipos de prueba con éxito. Por el contrario, si estas actividades se realizan mal, los probadores individuales incluso con talento y Gerentes tendrán pocas probabilidades de éxito. El modelo identifica doce procesos de prueba críticos. CTP es ante todo un modelo de referencia de contenido. El modelo de CTP es un enfoque contextual que permite adaptar el modelo que incluye:

Identificación de los retos específicos de reconocimiento de atributos de un buen proceso de selección de la orden y la importancia de la construcción del mejoramiento de los procesos de pruebas en software.

Estas son las mediciones cuantitativas se examinan comúnmente durante una evaluación CTP:

- Porcentaje de detección de defectos.
- El rendimiento sobre la inversión de pruebas.
- Requisitos cobertura y cobertura de riesgos.
- Sobrecarga de la liberación de pruebas.
- Tasa de rechazo del informe de defectos.

Los siguientes factores cualitativos se evalúan comúnmente durante una evaluación de la CTP:

- Papel en el equipo de prueba y la eficacia.
- Utilidad del plan de prueba.
- Habilidades de equipo de prueba en prueba, el conocimiento del dominio, y la tecnología.
- Utilidad del informe de defectos.
- Prueba de la utilidad informe de resultados.
- Utilidad del cambio de gestión y el balance.

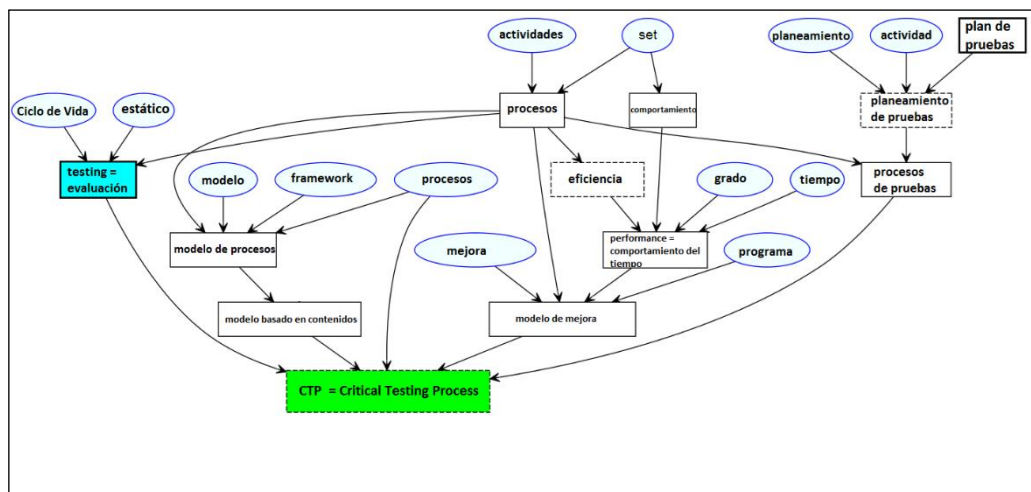


Figura 20. Procesos de Pruebas Críticas (CTP)

Fuente: Critical Testing Process

2.3.4 PRUEBAS A SOFTWARE EN TECNOLOGÍAS MODERNAS Y COMPLEJAS

2.3.4.1 Pruebas a Software en Arquitectura SOA

Los cimientos para el testing efectivo en una arquitectura de software SOA se mencionan a continuación:

- Equitativa asignación de esfuerzo del testing en el ciclo de vida de un proyecto. La mayoría de empresas todavía no aceptan los beneficios tangibles de técnicas estáticas de revisión y técnicas formales en las primeras etapas del proyecto. La gran mayoría del esfuerzo en el testing llega tarde casi al término del ciclo de vida de un proyecto. En SOA se requiere más esfuerzo debido al nivel de los servicios.

- El equipo de testing SOA es un equipo multidisciplinario de especialistas en negocio y tecnología.

- Diseñar las estrategias de testing de un proyecto además de las especificaciones técnicas y de negocio de esta misma.

- Presupuesto para el testing para que lo integren desde el comienzo del proyecto.

- Crear controles de calidad en el ciclo de vida del proyecto de software.

- El testing de seguridad no es una tarea al final de un proyecto.

Diseñar el testing de seguridad desde el inicio de un proyecto.

- Las herramientas para el testing de software SOA son indispensables.

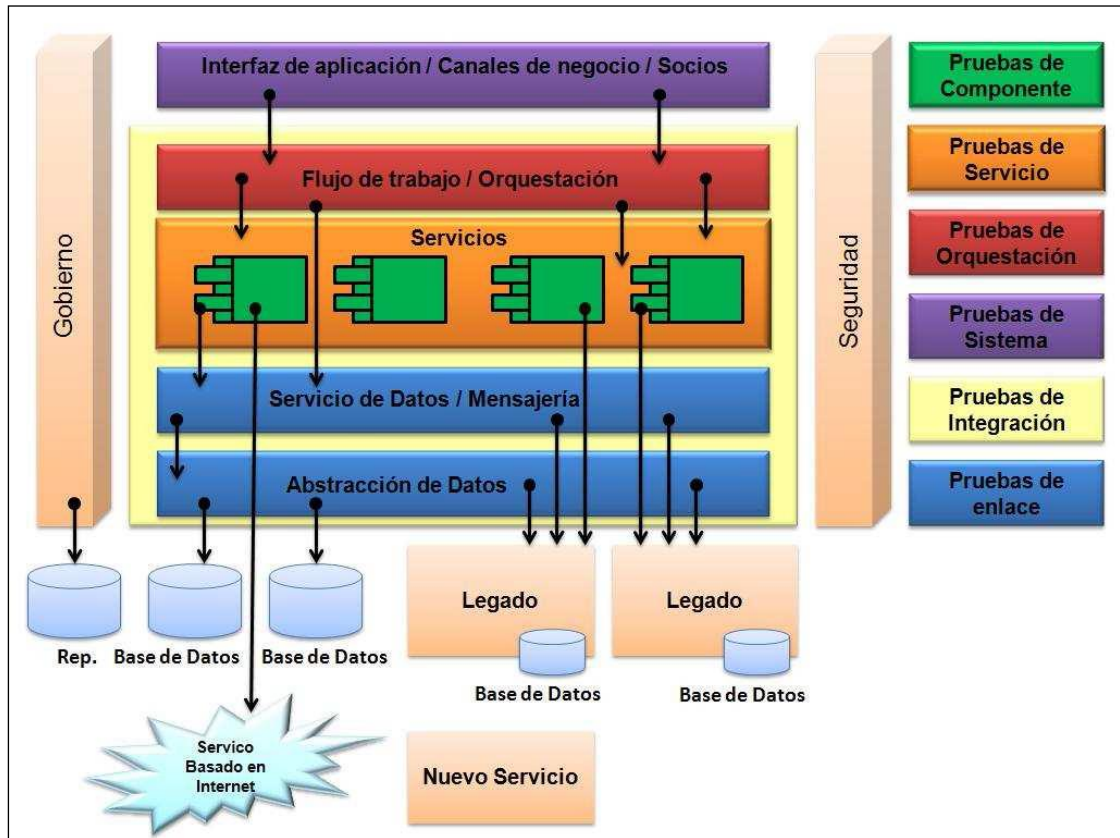


Figura 21. Procesos de Pruebas Críticas (CTP)

Fuente: Propuesta de un modelo de pruebas para una arquitectura orientada a servicios de Ramón Rivero Torres, Elizabeth Ochoa Luis y Leevan Abon Cepeda

2.3.4.2 Pruebas a Software para Dispositivos Móviles

La etapa más importante en el ciclo de vida de la construcción de un software móvil es la del testing. El testing cobra especial trascendencia puesto que la sensibilidad de las aplicaciones móviles en plataforma hardware de mucha división tecnológica tanto a nivel de pantallas y con sus diferentes resoluciones, así como también las diferentes versiones de sus sistemas operativos, esto se ilustra en el siguiente gráfico sobre la diversificación de Android en noviembre de 2016.

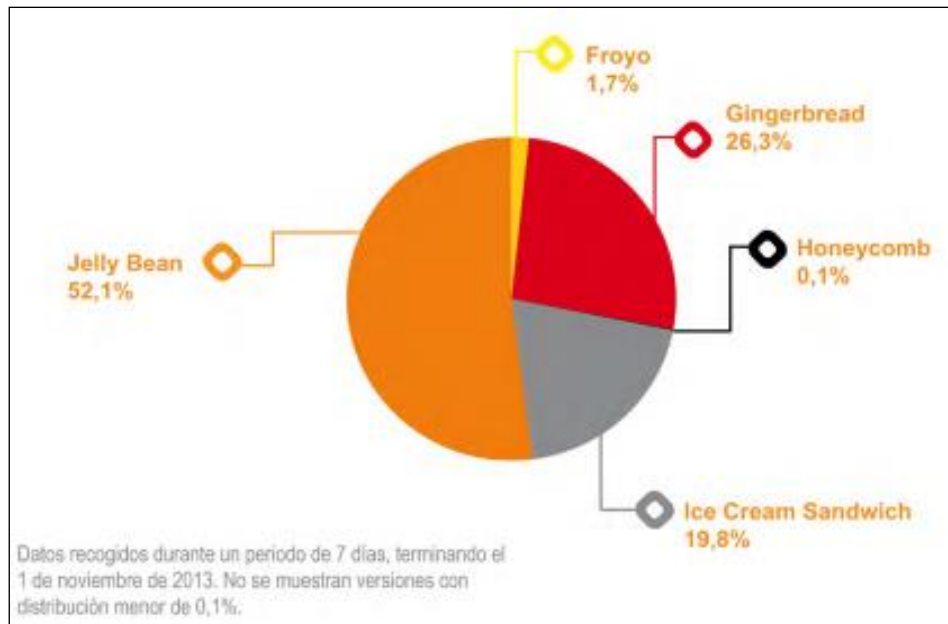


Figura 22. Complejidad de las Pruebas en Dispositivos Móviles por su Fragmentación de Sistema Operativo Android.

Fuente: At Sistemas

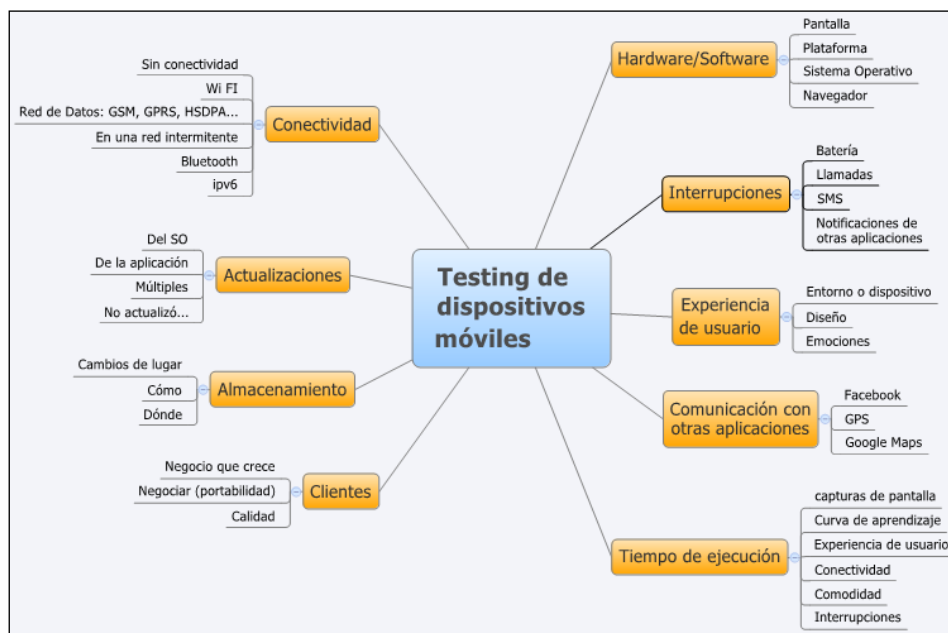


Figura 23. Complejidad de las Pruebas al Software para Dispositivos Móviles.

Fuente: Centro de Ensayos de Software del Uruguay.

2.3.4.3 Pruebas a Software en Cloud Computing

Se puede definir a las Pruebas a Software en Cloud Computing como Testing en la nube como el grupo de mecanismos y herramientas de testing que hace uso de recursos desde la nube. Entendiéndose por recursos a los elementos (hardware, software y plataforma tecnológica) necesaria para ejecutar testing.

El usuario aprecia al Cloud Testing como una capa SaaS (Software as Service), PaaS (Platform as service), IaaS (infraestructura as service) del modelo Cloud Testing, haciendo posible adquirir herramientas distribuidas en internet vía modalidad de pago por consumo.

El reto primero y fundamental de proporcionar servicios cloud es la disponibilidad de servicio. En lugar de mantener las instalaciones locales, se tiene que estar convencido que se puede tener acceso a los servicios y datos que necesita cada vez que necesita sin experimentar retrasos indebidos. El segundo reto es la garantía de servicio. ¿Cómo puede su proveedor de servicios cloud garantizar entrega puntual y disponibilidad de servicio incluso cuando no hay control de la conexión de comunicación de datos entre el servicio de cloud y los usuarios corporativos? ¿El proveedor de la comunicación de datos tiene la infraestructura de supervisión existente para garantizar acuerdos de nivel de servicio? ¿El proveedor de servicios cloud tiene la infraestructura de supervisión para asegurar los servicios prestados? Se tiene que plantear estas preguntas al proveedor en quien se esté pensando. El último reto es la eficacia del servicio. Abarca la eficacia en todos los aspectos, de ahorro de costos, espacio y eficiencia para la entrega de un servicio eficaz y escalable mediante virtualización, servidores de gama alta e interfaces de alta velocidad de alimentación. La infraestructura para supervisar que la eficiencia también debe seguir los mismos principios.

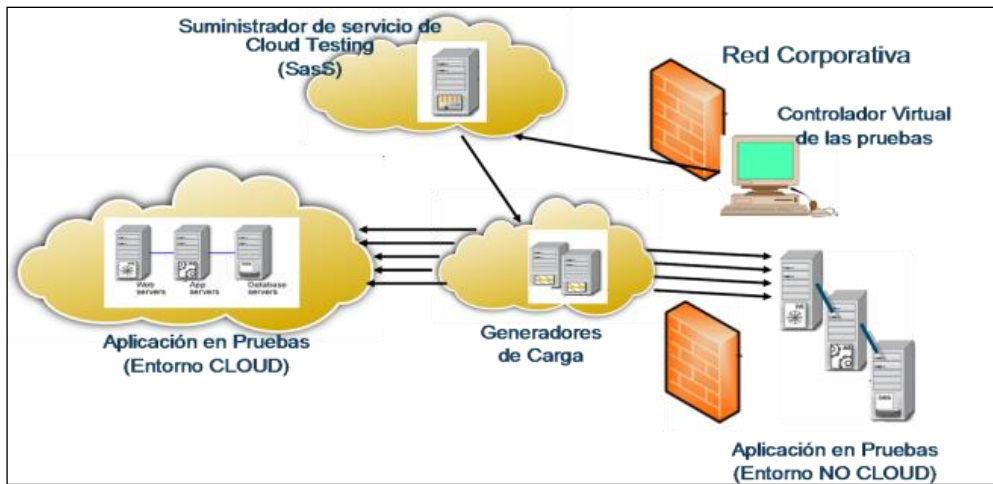


Figura 24. Complejidad de las Pruebas al Software en Cloud Computing.

Fuente: Cloud Testing. Teresa Díaz Yuiján, Jhonatan Hernández Ramos.

III. MÉTODO

3.1. TIPO DE INVESTIGACIÓN

3.1.1. BÁSICA

Si se propone como nuevo modelo es parte de la investigación básica, ya que se añadirá conocimiento existente según Callejas Cuervo, Alarcón Aldana, & Alvarez Carreno (2017) manifiestan que “Los modelos agregan conocimiento basados en la existencia de otros modelos antes que existieran estos”; basado en esto se puede deducir que es un conocimiento existente que genera otro conocimiento.

3.1.2. APLICADA

Se ajusta a una investigación aplicada considerando que el modelo teórico a implementar será aplicado a una herramienta tecnológica según La Ponencia de Investigación Científica y Desarrollo Tecnológico citado por Cegarra (2012) manifiesta que “La investigación aplicada , a veces llamada Investigación Técnica, tiende a la resolución de problemas o al desarrollo de ideas, a corto o mediano plazo , dirigidas a conseguir innovaciones, mejoras de procesos o productos, incrementos de calidad y productividad”.

3.2. POBLACIÓN Y MUESTRA

3.2.1. POBLACIÓN

La investigación está constituida por el total de profesionales que gestionan y participan en los proyectos microempresariales de calidad de software en la microempresa ARWEBSYSTEMS SAC ubicada en la Ciudad de Lima, así como también por los clientes que hacen uso del servicio que brinda ARWEBSYSTEMS SAC, ello va a permitir identificar la muestra poblacional con el cual se podrá desarrollar la investigación llegando a resultados en proyectos de calidad de software gestionados por la empresa en mención.

A continuación, el detalle de la población:

1. Los que gestionan con el uso del modelamiento

- a. A nivel Comercial

Gerentes Comerciales de la empresa Junior / Semi Senior / Senior

b. A nivel de Gestión Técnica

Jefe de Proyecto de Desarrollo de Software Junior / Semi Senior / Senior

Jefe de Proyecto de Calidad de Software Junior / Semi Senior / Senior

Especialistas en Desarrollo de Software Junior / Semi Senior / Senior

Especialistas en Calidad de Software Junior / Semi Senior / Senior

2. Los gestionados por medio del modelamiento

a. A nivel de Análisis y Diseño de software

Analistas de Sistemas Junior / Semi Senior / Senior

Analistas de Negocios Junior / Semi Senior / Senior

Analistas de Procesos Junior / Semi Senior / Senior

Analistas Funcionales Junior / Semi Senior / Senior

b. A nivel de Desarrollo de software

Programadores de Software Junior / Semi Senior / Senior

Analistas Programadores Junior / Semi Senior / Senior

DBAs Junior / Semi Senior / Senior

c. A nivel de Calidad de software

Probadores de Software Junior / Semi Senior / Senior

Analistas de Aseguramiento de la Calidad de Software Junior / Semi Senior / Senior

Analistas de Control de la Calidad de Software Junior / Semi Senior / Senior

Automatizadores de Pruebas de Software Junior / Semi Senior / Senior

3. Los que reciben servicios por medio del modelamiento

a. A nivel Comercial

Gerentes Comerciales de la empresa Cliente Junior / Semi Senior / Senior

b. A nivel Gestión Técnica

Jefe de Proyecto de Desarrollo de Software de la empresa Cliente Junior /
Semi Senior / Senior

Jefe de Proyecto de Calidad de Software de la empresa Cliente Junior /
Semi Senior / Senior

Especialistas en Desarrollo de Software de la empresa Cliente Junior / Semi
Senior / Senior

Especialistas en Calidad de Software de la empresa Cliente Junior / Semi
Senior / Senior

c. A nivel Técnico Operativo

i. A nivel de Análisis y Diseño de software

Analistas de Sistemas Junior / Semi Senior / Senior

Analistas de Negocios Junior / Semi Senior / Senior

Analistas de Procesos Junior / Semi Senior / Senior

Analistas Funcionales Junior / Semi Senior / Senior

ii. A nivel de Desarrollo de software

Programadores de Software Junior / Semi Senior / Senior

Analistas Programadores Junior / Semi Senior / Senior

DBAs Junior / Semi Senior / Senior

iii. A nivel de Calidad de software

Probadores de Software Junior / Semi Senior / Senior

Analistas de Aseguramiento de la Calidad de Software Junior /
Semi Senior / Senior

Analistas de Control de la Calidad de Software Junior / Semi Senior
/ Senior

Automatizadores de Pruebas de Software Junior / Semi Senior /
Senior

4. Expertos Ajenos en Calidad de Software (Opinión previo análisis)

i. Expertos Nacionales en Calidad de Software

ii. Expertos Internacionales en Calidad de Software

Junior menor a dos años de experiencia demostrada en la posición.

SemiSenior de 2 a 5 años de experiencia demostrada en el cargo

Senior mayor a 5 años de experiencia demostrada en el cargo con al menos una
certificación internacional en Calidad de Software

Experto mayor a 10 años de experiencia en el cargo con al menos dos certificaciones
internacionales en Calidad de Software

Tabla 5:
Número de preguntas del cuestionario dirigido a Jefes y Profesionales

GRUPOS	CANTIDAD
Jefes de Proyectos de Software, Lideres en Calidad de Software y Probadores de software Senior de la empresa ARWEBSYSTEMS.	17
Expertos Nacionales en Calidad de Software	27
Expertos Internacionales en Calidad de Software	75
Jefes de Calidad de Software, Lideres de Equipos de Pruebas de Software, Supervisores de Calidad de Software de las empresas clientes ¹ de ARWEBSYSTEMS	30
Gerentes Comerciales de las empresas clientes de ARWEBSYSTEMS	12
TOTALES	161

Fuente: Elaboración propia.

La población está conformada por dos grupos referentes de la microempresa ARWEBSYSTEMS SAC, que son las siguientes:

- Jefes y Profesionales que totalizan 507.
- Clientes que totalizan 12.

Dada que la población es finita, se tiene un total poblacional de 161 personas (ver tabla 3).

3.2.2. MUESTRA

En la muestra de estudio participaron 233 personas de los dos grupos referentes de la microempresa ARWEBSYSTEMS SAC. La muestra es probabilística, y se seleccionó utilizando un muestreo estratificado aleatorio con afijación proporcional, en el año 2017. Es decir, la muestra se seleccionó por estratificación de los grupos referentes y extracción posterior al azar de los participantes en cada estrato.

a) Tipo de muestreo

¹ Cantidad de clientes promedio que atiende la empresa ARWEBSYSTEMS SAC en un año.

Se utilizó un muestreo probabilístico, estratificado y de selección aleatoria proporcional al tamaño (con afijación proporcional), siendo los grupos referentes las unidades de estratificación. Se construyó el marco muestral estratificado según grupo referente (Jefes y Profesionales, y Clientes) de la microempresa ARWEBSYSTEMS SAC ubicada en la ciudad de Lima y en cada estrato se seleccionaron muestras independientes.

b) Tamaño de la muestra

Dado que no se tiene estudios anteriores, y la variable principal de estudio es cualitativa, se ha establecido que la proporción de satisfacción es de 50%. Con un margen de error del 5% y un nivel de confianza del 95%, y tamaño poblacional de 519 personas.

En ese sentido, considerando que el tamaño de nuestra población se encuentra entre 500 y 1000, lo cual correspondería un tamaño de muestra entre 222 y 286 (ver tabla 4), se determinó que el tamaño de muestra mínimo para el estudio es 250 personas, empleando la siguiente tabla de tamaño de muestra para estudios observacionales:

*Tabla 6:
Tamaño de la muestra para poblaciones finitas con intervalo de confianza del 95,5%*

Amplitud de la población	± 1%	± 2%	± 3%	± 4%	± 5%
500					222
1000				385	286
1500			638	441	316
2000			714	476	333
2500		1250	769	500	345
3000		1364	811	517	353
3500		1458	843	530	359
4000		1538	870	541	364
4500		1607	891	549	367
5000		1667	909	556	370
6000		1765	938	566	375
7000		1842	959	574	378
8000		1905	976	580	381
9000		1957	989	584	383
10000	5000	2000	1000	588	385
15000	6000	2143	1034	600	390
20000	6667	2222	1053	606	392
25000	7143	2273	1064	610	394

50000	8333	2381	1087	617	397
100000	9091	2439	1099	621	398

Fuente: Estadística aplicada a los negocios y la economía de Lind / McGraw-Hill / interamericana de México.2006

Luego, los grupos referentes se subdividieron en 2 estratos:

Tabla 7:
Tamaño de Muestra

GRUPOS	NH	WH
Jefes y Profesionales	507	507/519= 0.97
Clientes	12	12/519= 0.02
TOTAL	N=519	

Fuente: Elaboración propia.

Donde:

NH= Es la cantidad de la muestra por cada estrato

WH= Es el peso de la muestra por cada estrato

En base al tamaño de muestra y de acuerdo a lo establecido en la tabla 4 “Métodos y Aplicaciones de Muestreo” se consideró trabajar con 250 personas, procediendo a realizar una afijación proporcional, seleccionando de cada estrato un número de elementos para la muestra, proporcional al tamaño del estrato; es decir, el tamaño de muestra para cada estrato sería el siguiente:

$$n_1 = (507) * (250/519) = 244.2 = 244$$

$$n_3 = (12) * (250/519) = 5.7 = 6$$

Entonces se extraerán para la muestra, 244 unidades del primer estrato (Jefes y Profesionales), y 6 unidades del tercer estrato (Clientes).

Tabla 8:
Distribución de la muestra por Estratos

GRUPOS	MUESTRA N _H
Jefes y Profesionales	244
Clientes	6
TOTAL	n=250

Fuente: Elaboración propia.

Características de las muestras:

Respecto a la muestra de jefes y profesionales se ha encontrado un predominio del sexo masculino, con instrucción superior, cuya edad de la mayoría de participantes de este grupo de estudio es mayor a 40 años.

Respecto al grupo de clientes, también se ha encontrado un predominio del sexo masculino, donde un importante porcentaje son representantes de empresas financieras, inmobiliarias, estudios jurídicos, notarias, etc., y cuya edad promedio de este grupo de estudio es de 35 años.

3.3. OPERACIONALIZACIÓN DE LAS VARIABLES

3.3.1. DEFINICIÓN DE VARIABLES

A. VARIABLE INDEPENDIENTE

MODELAMIENTO PARA LA GESTIÓN EFICIENTE DE PROYECTOS MICROEMPRESARIALES

Definición conceptual:

Es la agrupación de normas, métodos, prácticas y procesos establecidas para gestionar el ciclo de vida de un proyecto para un ámbito microempresarial. Dicho modelamiento ha sido diseñado y adaptado en base a modelos y metodologías de gestión de proyectos de clase mundial como el PMBOK, PRINCE2, ISTQB Test Manager y SCRUM.

Definición operativa:

Es la cantidad de microempresas que utilizaron y se beneficiaron con el modelamiento propuesto.

Indicadores:

Aplicación en microempresas

Conocimiento

Nivel de Éxito

Índices:

Número de microempresas que se beneficiaron por medio de la aplicación del modelamiento.

Número de proyectos microempresariales de calidad de software gestionados con el modelamiento propuesto y que resultaron exitosos.

B. VARIABLE DEPENDIENTE

GESTIÓN EFICIENTE DE LOS PROYECTOS MICROEMPRESARIALES DE CALIDAD DE SOFTWARE

Definición conceptual:

Son las características que obtienen los proyectos microempresariales de calidad de software, como resultado de la aplicación del modelamiento propuesto.

Definición operativa:

Identificación del cumplimiento de los diversos aspectos del proyecto.

Indicadores:

Eficiencia

Índices:

Factibilidad Económica que brinda el modelamiento a los proyectos de pruebas de software.

Grado de Concordancia al presupuesto promedio asignado a proyectos pequeños, medianos o grandes por parte de la microempresa.

Factibilidad Operativa que brinda el modelamiento a los proyectos de pruebas de software.

Grado de Simplicidad para la aplicación del modelamiento por los líderes o integrantes del equipo del proyecto de pruebas de software de la microempresa, que trabajan desde el inicio o que ingresan durante el proyecto.

Nivel de Funcionamiento del modelamiento en Proyectos de Pruebas de Software de una microempresa.

Factibilidad Técnica que brinda el modelamiento a los proyectos de pruebas de software.

Cantidad y complejidad de capacidades técnicas necesarias para aplicar el modelamiento

Flexibilidad que brinda el modelamiento a los proyectos de pruebas de software.

Grado de flexibilidad del modelamiento ante un proceso de cambio simple, moderado o crítico.

Adaptabilidad que brinda el modelamiento a los proyectos de pruebas de software.

Grado de Adaptabilidad del modelamiento ante un cambio externo o interno.

Escalabilidad que brinda el modelamiento a los proyectos de pruebas de software.

Grado de Escalabilidad del modelamiento ante una ampliación razonable en términos de coste, tiempo o complejidad.

Utilidad para proyectos de pruebas a software de arquitectura orientada a servicios (SOA).

Grado de Utilidad del modelamiento respecto a las dependencias de otras aplicaciones y sistemas; servicios sin interfaz gráfica de usuario, integración de diferentes tecnologías y diferentes formatos de datos, planificación y estrategia para abordar los problemas de disponibilidad y seguridad en pruebas a software SOA.

Utilidad para proyectos de pruebas a software para dispositivos móviles.

Grado de Utilidad del modelamiento respecto a las pruebas de funcionalidad, rendimiento y seguridad en pruebas a software móvil.

Utilidad para proyectos de pruebas a software en Cloud Computing.

Grado de Utilidad del modelamiento respecto a pruebas a software con ayuda total, casi total o ayuda media de Cloud Computing.

3.4. INSTRUMENTOS

Los instrumentos de recolección fueron dos (02) cuestionarios, uno para cada uno de los estratos identificados (jefes y profesionales, y clientes). Los cuestionarios se construyeron en función a los objetivos del estudio.

El cuestionario dirigido a los jefes y profesionales que gestionan y/o participan en algún proyecto de tecnología de información de la ARWEBSYSTEMS SAC consta de un total de catorce (14) preguntas. A continuación, se describe las áreas del cuestionario.

Tabla 8:
Número de preguntas del cuestionario dirigido a Jefes y Profesionales

ÁREAS	Nº DE PREGUNTAS
Conocimiento de las metodologías según PMBOK, PRINCE2, ISTQB Test Manager y SCRUM.	4
Aplicación del modelamiento	1
Gestión de Proyectos de Calidad de Software	3
Trazabilidad y Control de Proyectos de Calidad de Software	4
Expectativas con la aplicación del modelamiento	2

Fuente: Elaboración propia.

Por otro lado, el cuestionario dirigido a los clientes que hacen uso del servicio que brinda ARWEBSYSTEMS SAC consta de un total de doce (12) preguntas. A continuación, se describe las áreas del cuestionario.

*Tabla 9:
Número de preguntas del cuestionario dirigido a Clientes*

ÁREAS	Nº DE PREGUNTAS
Calidad de Software	2
Calidad del Servicio	4
Infraestructura Tecnológica	2
Proyectos Informáticos	3
Expectativas	1

Fuente: Elaboración propia.

Los cuestionarios para cada uno de los estratos identificados, han sido validados a través de juicio de expertos. Para su validez de contenido, en este caso se consultó a 8 expertos en calidad de software obteniendo un nivel de concordancia en las opiniones altamente significativo.

▪ Validez y Confiabilidad del Cuestionario

Juicio de Expertos que le dan la validez de contenido

Se solicitó la intervención de 8 especialistas en el área de calidad de software (6) y en modelamientos (2) con la finalidad de establecer la validez de contenido del instrumento, utilizando un cuestionario de validación (ver anexo 1).

Seguidamente se muestra una tabla resumen de los resultados de la validación:

Tabla 10:
Resultados de la Validación de Contenido

INDICADOR	JUECES								RESULTADO
	1	2	3	4	5	6	7	8	
1	MB	EXC	MB	EXC	B	EXC	EXC	MB	Excelente
2	BUE	EXC	BU	MB	EXC	MB	MB	EXC	Excelente
3	EXC	EXC	MB	EXC	B	MB	MB	MB	Muy buena
4	MB	MB	EXC	EXC	EXC	EXC	EXC	EXC	Excelente
5	EXC	EXC	EXC	EXC	EXC	EXC	EXC	EXC	Excelente
6	MB	MB	MB	EXC	EXC	EXC	EXC	EXC	Excelente
7	MB	MB	EXC	EXC	EXC	EXC	EXC	EXC	Excelente
8	MB	MB	EXC	MB	EXC	EXC	EXC	MB	Excelente
9	EXC	EXC	EXC	EXC	EXC	EXC	MB	EXC	Muy buena
10	MB	EXC	MB	EXC	BU	EXC	MB	MB	Muy buena

Fuente: Elaboración propia.

Donde:

D: Deficiente 0-21%

R: Regular 22-40%

BUE: Buena 41-61%

MB: Muy Buena 62-80%

EXC: Excelente 81 -100%

El resultado en total nos indicó que el instrumento ha tenido una calificación de excelente para medir la variable percepción, teniendo en cuenta la respuesta modal. Luego se tomó las encuestas a los participantes en forma personal.

Estos cuestionarios están suscritos como fuente primaria, ya que es obtenida de manera directa por el investigador, por medio de sus respuestas transmitidas por los integrantes en un acontecimiento. Luego se procedió al vaciado de datos para determinar las diferencias en las percepciones vertidas por los participantes a través del paquete de análisis para la investigación en ciencias sociales del IBM SPSS (versión 23) con el fin de comprobar las hipótesis planteadas al inicio de la presente tesis, los nombres se han omitido debido a que no se tiene autorización para mostrar sus nombres. A continuación, se muestra la validación por juicios de expertos.

Tabla 11:
Resultados de Opinión del Modelo por Expertos

Nº	EXPERTO	PROCEDENCIA	GRADO ACADÉMICO	OPINION
1	Experto 1 – Computación y Sistemas.	Lima – Perú	Doctorado en Universidad Federal de Rio de Janeiro.	FAVORABLE
2	Experto 2 – Gestión de Proyectos	México -México DF	Doctorado en Proyectos de Integración y Desarrollo – Universidad de León – España.	FAVORABLE
3	Experto 3 – Industria de Producción de Software para Microempresas	Lima – Perú	Doctorado en Ingeniería de Sistemas – Universidad Nacional de Ingeniería	FAVORABLE
4	Experto 4 – Start Ups	Lima - Perú	Doctorado en Ingeniería de Sistemas – Universidad Nacional Federico Villareal	FAVORABLE
5	Experto 5 – Calidad de Software	Lima – Perú	Doctorado en Lenguajes y Sistemas Informáticos e Ingeniería de Software. Universidad Politécnica de Madrid. España.	FAVORABLE
6	Experto 6 – Calidad de Software	Lima – Perú	Doctorado en Informática, Especialidad Informática y aplicaciones - UCBN- Francia.	FAVORABLE
7	Experto 7 – Calidad de Software	Santa Catarina - Brasil	Doctorado en Ingeniería Eléctrica, mención Sistemas de Información. Universidad	FAVORABLE

			Federal de Santa Catarina – Brasil.	
			Promedio Final	87.73%

Fuente: Elaboración Propia

3.5. PROCEDIMIENTOS

3.5.1. ESTRATEGIA DE PRUEBA DE HIPÓTESIS

Para la contrastación o prueba de hipótesis formulada en la actualmente presentada tesis se ha hecho uso la prueba del Chi-cuadrado².

De acuerdo a lo indicado por Castañeda, la prueba de Chi-cuadrado establece la determinación si 2 variables cualitativas se relacionan o no. Para lo cual, se plantea una comparación de hipótesis, entre la hipótesis nula:

H0: No existe relación entre las variables.

E hipótesis alternativa:

Ha: Sí hay relación entre las variables.

Si culminando estudio se concluye que las variables no se encuentran vinculadas se puede decir con un alto nivel de confianza, previamente establecido, que ambas son independientes.

El resultado de la prueba de Chi-cuadrado se obtiene mediante una tabla de contingencia que expresa las frecuencias del comportamiento de las variables del objeto del estudio, donde las filas son las variables y las columnas son las posibles respuestas. A estas frecuencias se le denominadas frecuencias observadas.

Sin embargo, para realizar la prueba de Chi-cuadrado es importante hacer el cálculo de frecuencias esperadas y hacer la comparación con las frecuencias que se estén observando en la realidad. Siendo la formula a aplicar la siguiente:

$$X_t^2 = \sum \frac{(Fo - Fe)^2}{Fe},$$

Donde:

Fo, es la frecuencia observada y

² Prueba estadística denominada también como Test de Chi-cuadrado ó Prueba de Ji Cuadrado

F_e , es la frecuencia esperada

Si la hipótesis nula es cierta, el valor obtenido debería estar dentro del rango o zona de aceptación, es decir menor o igual al valor teórico del Chi cuadrado.

De lo contrario, si el resultado obtenido aplicando la prueba de Chi cuadrado es mayor que el valor teórico se dirá que la diferencia es significativa y se podrá concluir que las 2 variables no son independientes, sino que están relacionadas. Por lo que se rechazará la hipótesis nula (H_0) y se aceptará la hipótesis alternativa (H_a).

En ese sentido, en la presente tesis se realizará la prueba de Chi cuadrado a las tres (03) hipótesis específicas planteadas, considerando que si las tres hipótesis nulas específicas son rechazadas, se dará por válida la hipótesis general formulada para esta tesis.

3.5.2. DISEÑO DE INVESTIGACIÓN

La presente tesis es una investigación descriptiva, correlacional y evaluativa.

Es descriptivo, porque se expresan las principales características de la gestión de proyectos en todos sus aspectos, basándose en el enfoque de las metodologías PMBOK, PRINCE2, ISTQB Test Manager y SCRUM.

El estudio es de tipo correlacional, porque relaciona la variable del modelamiento para la Gestión Eficiente de Proyectos Microempresariales en Calidad de Software.

Es evaluativo, porque sobre la base de los indicadores propuestos, se ha hecho una apreciación global de la gestión eficiente de proyectos de calidad de software en el ámbito microempresarial. Desde este punto de vista, la investigación evaluativa se concibe como una investigación aplicada “cuasi experimental”, cuya finalidad primordial se orienta a la búsqueda de juicios de valor sobre la aplicación y resultados obtenidos con la utilización del modelamiento para la gestión eficiente de proyectos de calidad de software.

3.6. ANÁLISIS DE DATOS

Terminado el trabajo en campo, se implementó una base de datos para luego realizar el análisis estadístico en el paquete IBM SPSS versión 23.0, en base a lo siguiente:

- a. Obtener las frecuencias y porcentajes en variables cualitativas.

- b. Construir tablas para cada apreciación según grupos referentes.
- c. Elaborar gráficos para cada apreciación estudiada para la muestra de resultados.
- d. Analizar mediante la prueba de Chi cuadrado

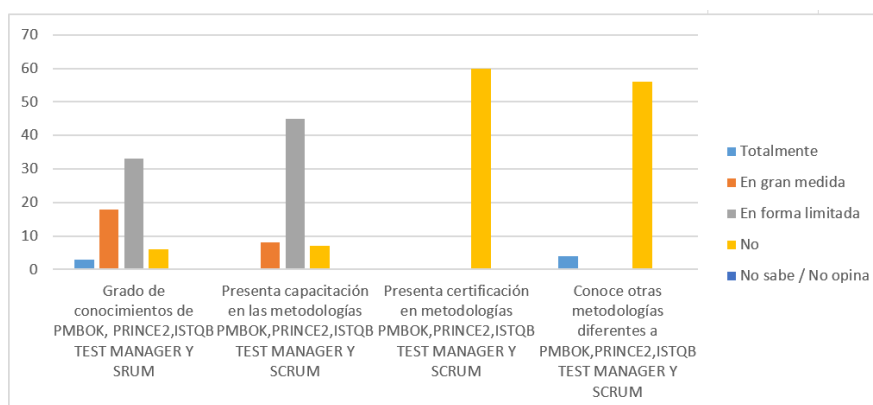
3.6.1. ANÁLISIS DESCRIPTIVO

a) Apreciación de los Jefes y Profesionales

Tabla 12:
Apreciación de los Jefes y Profesionales sobre el área: Conocimiento de las metodologías de PMBOK, PRINCE2,ISTQB Test Manager y SCRUM.

Indicador	Totalmente	En gran medida	En forma limitada	No	No sabe / No opina
Grado de conocimientos de PMBOK, PRINCE2,ISTQB TEST MANAGER Y SRUM	3	18	33	6	0
Presenta capacitación en las metodologías PMBOK,PRINCE2,ISTQB TEST MANAGER Y SCRUM	0	8	45	7	0
Presenta certificación en metodologías PMBOK,PRINCE2,ISTQB TEST MANAGER Y SCRUM	0	0	0	60	0
Conoce otras metodologías diferentes a PMBOK,PRINCE2,ISTQB TEST MANAGER Y SCRUM	4	0	0	56	0

Fuente: Elaboración Propia



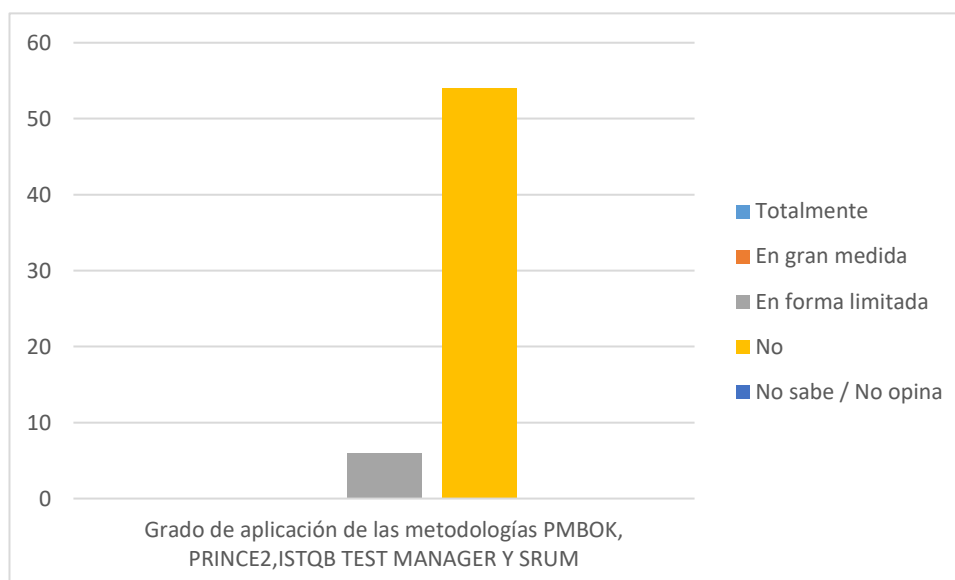
Se observa respecto al área: Conocimiento de las metodologías de Gestión de Proyectos que respecto a los indicadores de grado de conocimiento y capacitación la mayor cantidad de los jefes y profesionales encuestados manifiestan en un nivel de forma limitada el grado de conocimiento (33) y capacitación (45), respecto al indicador certificación en alguna metodología en Gestión de Proyectos el total de encuestados manifestó NO tener dicha certificación (60) y en referencia al indicador conocimiento de otra metodología la mayor cantidad señalo un nivel no conocer otra metodología distinta a la del PMI (56). En consecuencia, la apreciación de los jefes y

profesionales en el área: Conocimiento del modelamiento del PMI-PMBOK es de nivel negativo (NO).

Tabla 14:
Apreciación de los Jefes y Profesionales sobre el área: Aplicación de los modelamientos de Gestión de Proyectos.

Indicador	Totalmente	En gran medida	En forma limitada	No	No sabe / No opina
Grado de aplicación de las modelamientos PMBOK, PRINCE2,ISTQB TEST MANAGER Y SRUM	0	0	6	54	0

Fuente: Elaboración Propia



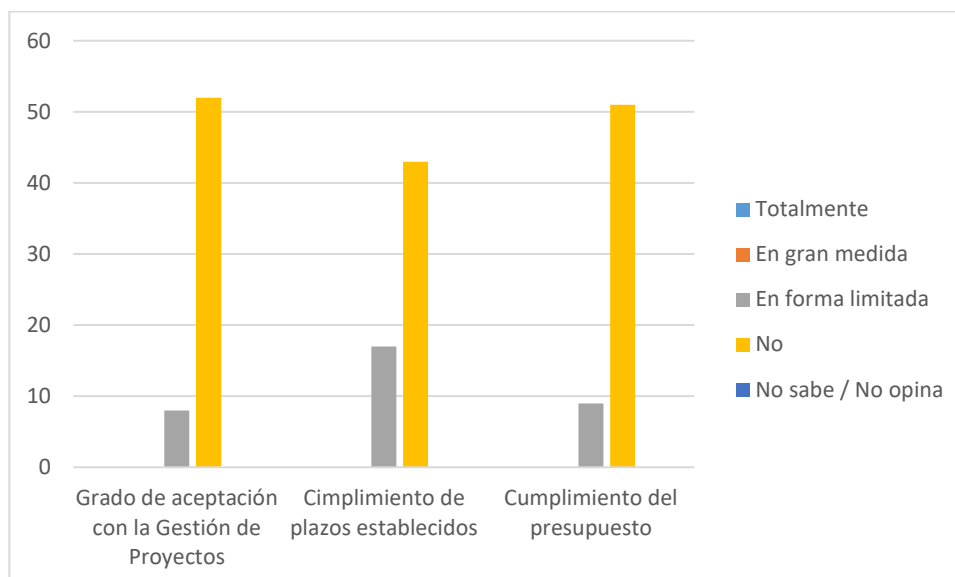
En lo que se refiere al área: Aplicación de las modelamientos de Gestión de Proyectos, la mayor cantidad de los jefes y profesionales encuestados señalan no aplicar dicha modelamiento (54); consiguientemente, la respuesta de los jefes y profesionales sobre el área: Aplicación del modelamiento del PMI-PMBOK es de nivel negativo (NO).

Tabla 15:
Apreciación de los Jefes y Profesionales sobre el área: Gestión de Proyectos.

Indicador	Total	En mayor medida	En manera limitada	No	No sabe/No opina
Grado de aceptación	0	0	7	48	0

Logro del cronograma	0	0	18	46	0
Logro del presupuesto	0	0	10	54	0

Fuente: Elaboración Propia



Se observa en el área: Gestión de proyectos que la mayor cantidad de jefes y profesionales califican con un valor NO los indicadores de aceptación con la Gestión de Proyectos (52), cumplimiento de plazos establecidos (43) y de igual manera respecto al indicado cumplimiento de presupuesto (51). En consecuencia, la apreciación de los profesionales en el área: Gestión de Proyectos es de nivel negativa (No)

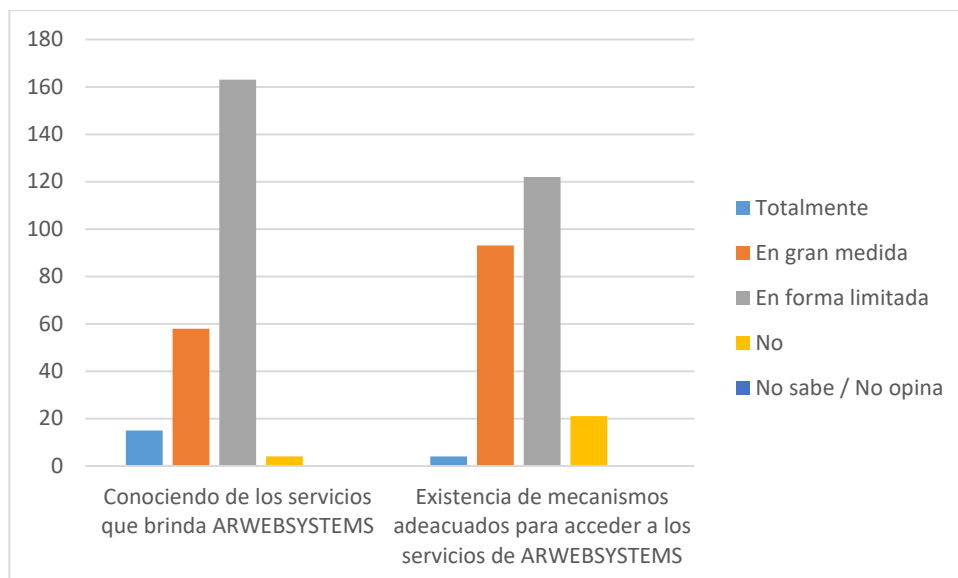
b) Apreciación de los Clientes

Tabla 16:

Apreciación de los Clientes sobre el área: Gestión de Proyectos.

Indicador	Totalmente	En gran medida	En forma limitada	No	No sabe / No opina
Conociendo de los servicios que brinda ARWEBSYSTEMS	15	58	163	4	0
Existencia de mecanismos adecuados para acceder a los servicios de ARWEBSYSTEMS	4	93	122	21	0

Fuente: Elaboración Propia

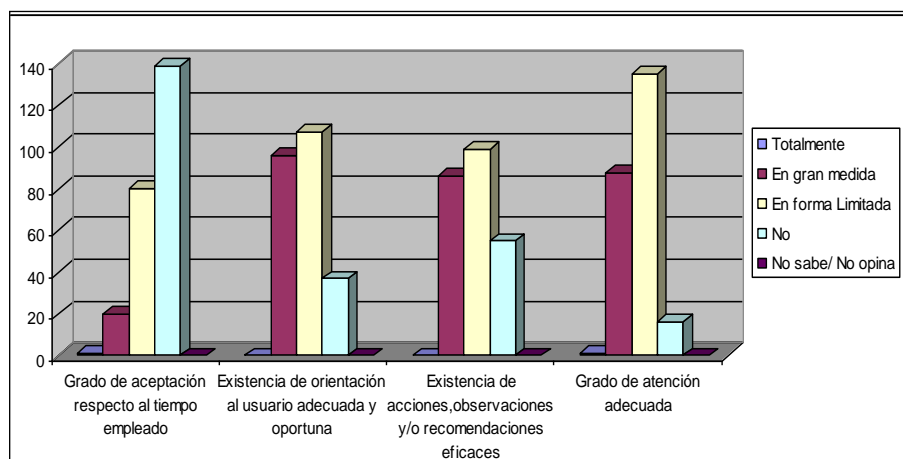


Se observa en el área: Calidad de Software, que la mayor cantidad de usuarios encuestados señalo un nivel en forma limitada a los indicadores de conocimiento de los servicios que brinda ARWEBSYSTEMS SAC (163), y existencia de mecanismos de acceso adecuados (122). En consecuencia, la apreciación de los usuarios en el área: Calidad de Software es de nivel en forma limitada.

Tabla 17:
Apreciación de los Usuarios sobre el área: Calidad de Software.

Indicador	Total	En mayor medida	En manera limitada	No	No sabe/No opina
Grado de aceptación	2	40	90	156	0
Orientación de usuario correcta y adecuada	1	198	109	98	0
Recomendaciones y Conclusiones exactas	1	167	178	99	0
Grado de atención oportuna	1	178	104	101	0

Fuente: Elaboración Propia

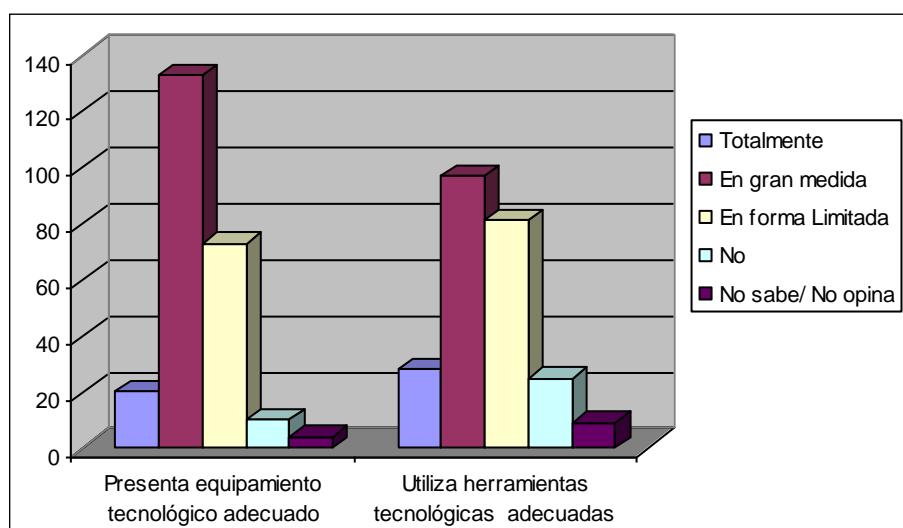


Respecto al área: Calidad del Software, los usuarios califican de manera negativa (NO) el indicador grado de aceptación respecto al tiempo empleado (139), mientras que señalan con un nivel en forma limitada a los indicadores existencia de orientación al usuario adecuada y oportuna (107), existencia de acciones, observaciones y/o recomendaciones eficaces (99), grado de atención adecuada (135). Por consiguiente, la apreciación de los usuarios en el área: Calidad de software es en forma limitada.

Tabla 18:
Apreciación de los Usuarios sobre el área: Infraestructura Tecnológica en Calidad de Software.

Indicador	Total	En mayor medida	En manera limitada	No	No sabe/No opina
Presenta equipos tecnológicos adecuados	22	132	74	12	4
Utiliza herramientas tecnológicas correctas	29	98	86	27	10

Fuente: Elaboración Propia

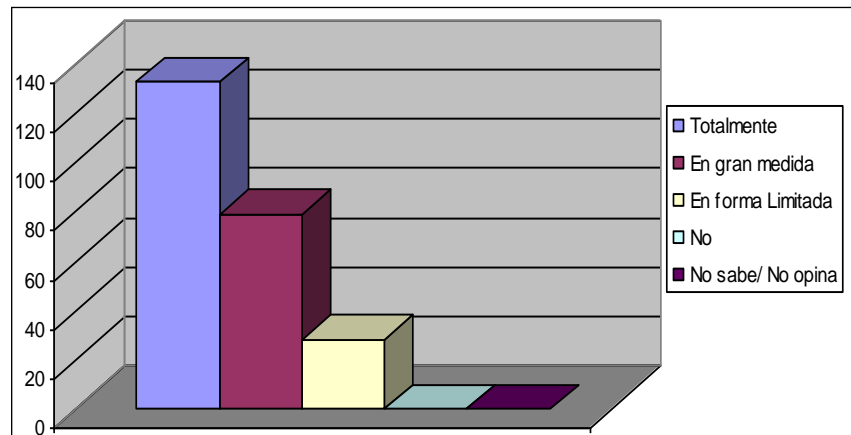


Se observa en el área: Infraestructura Tecnológica, la mayor cantidad de usuarios encuestados señaló un nivel en gran medida a los indicadores de presenta equipamiento tecnológico adecuado (133), y existencia de herramientas tecnológicas adecuadas para la calidad de software (97). En consecuencia, la apreciación de los usuarios en el área: Infraestructura Tecnológica es de nivel en gran medida.

Tabla 19:
Respuesta de los Usuarios sobre el área: Expectativas.

Indicador	Total	En mayor medida	En manera limitada	No	No sabe/No opina
La ejecución exitosa de proyectos pruebas de software contribuye a la calidad del servicio	134	80	32	0	0

Fuente: Elaboración Propia



Se observa en el área: Expectativas, que la mayor cantidad de usuarios encuestados señaló un nivel totalmente respecto al indicador la ejecución exitosa de proyectos de calidad de software contribuye a mejorar la calidad del software (133). En consecuencia, la respuesta de los usuarios en el área: Expectativas es de nivel muy buena (totalmente).

Por otro lado, considerando que parte de los indicadores del presente trabajo de investigación corresponden a la aplicación de metodologías de Gestión de Proyectos en las empresas clientes de ARWEBSYSTEMS, se ha aplicado un tipo de muestreo intencional u opinativo³, seleccionando de modo directo una cantidad representativa, determinando el tamaño de la muestra en una cantidad de siete (07) empresas, de los cuales se obtendrá la información relevante y relacionadas al objeto del estudio. Las empresas clientes seleccionadas son:

- BBVA Banco Continental
- Universidad Tecnológica del Perú (UTP)
- Registro Nacional de Identificación y Estado Civil (RENIEC)
- Caja de Beneficios y Seguridad Social del Pescador (CBBSP)
- Agencia Comercializadora de Publicidad Televisiva (MADUEÑO SAC)
- Cooperativa Americana para Remesas Exteriores (CARE-PERU)
- Concesionario Autorizado TOYOTA AUTOSPAR

³ Según (Latorre, Rincón y Arnal; 2003), este tipo de muestreo busca seleccionar intencionalmente los elementos de la muestra para que pueden facilitar una información más válida.

Para la obtención de data se utilizó cuestionario de nueve (09) preguntas dirigido a los Jefes de Informática de las empresas clientes anteriormente referidos.

Luego de realizar el trabajo en campo, se realizó una construcción de la base de datos para luego analizar estadísticamente en el paquete IBM SPSS versión 23.0, realizando lo siguiente:

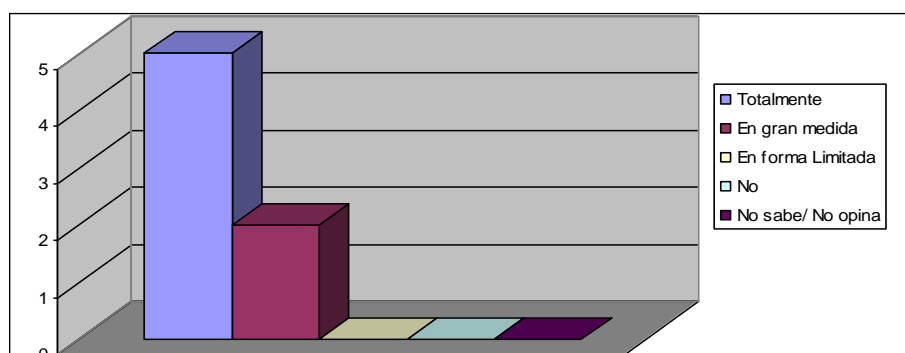
- a. Obtener frecuencias y porcentajes de las variables cualitativas.
- b. Construir tablas para cada una de las apreciaciones
- c. Elaborar gráficos por cada valoración estudiada para la muestra de resultados.

A continuación, el análisis descriptivo de los resultados obtenidos:

*Tabla 20:
Nivel de Éxito del uso de metodologías de Gestión de Proyectos de Calidad de software en las empresas clientes.*

Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe / No opina
Nivel de éxito con la aplicación de metodologías de Gestión de Proyectos	5	2	0	0	0

Fuente: Elaboración Propia



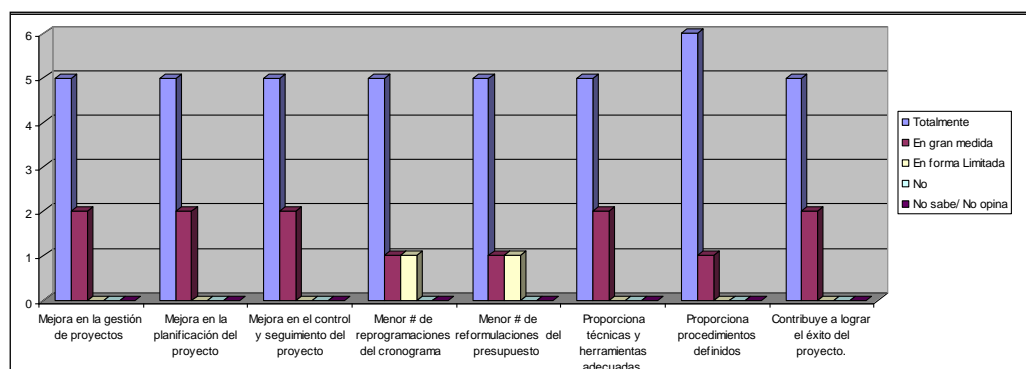
Se observa en el área: Nivel de éxito con la utilización de metodologías para Gestionar Proyectos de Calidad de Software que mayor cantidad en las empresas

clientes de ARWEBSYSTEMS calificó haber logrado un nivel totalmente de éxito (7).

Tabla 21:
Beneficios obtenidos en las Empresas Clientes

Indicador	Total	En mayor medida	En manera limitada	No	No sabe/No opina
Optimiza la gestión de proyectos de calidad de software	7	2	0	1	0
Optimiza la planificación de proyectos de calidad de software	7	2	0	0	0
Optimiza el control y seguimiento de proyectos de calidad de software	7	1	1	1	1
Reducción de reprogramaciones	7	2	0	1	0
Reducción de reformulaciones	7	1	1	1	1
Ayuda a alcanzar el éxito del proyecto	7	2	1	0	1

Fuente: Elaboración Propia



Como se puede apreciar en la tabla 21, la mayor cantidad de las empresas clientes de ARWEBSYSTEMS han calificado con el nivel totalmente respecto a los indicadores mejora en la gestión del proyecto (5), mejora en la planificación del proyecto (5), mejora en el control y seguimiento del proyecto (5), menor cantidad de reprogramaciones del cronograma (5), menor cantidad de reformulaciones del presupuesto (5), proporciona técnicas y herramientas adecuadas (5), proporciona procedimientos definidos (6) y contribuye a lograr el éxito de proyecto (5).

IV. RESULTADOS

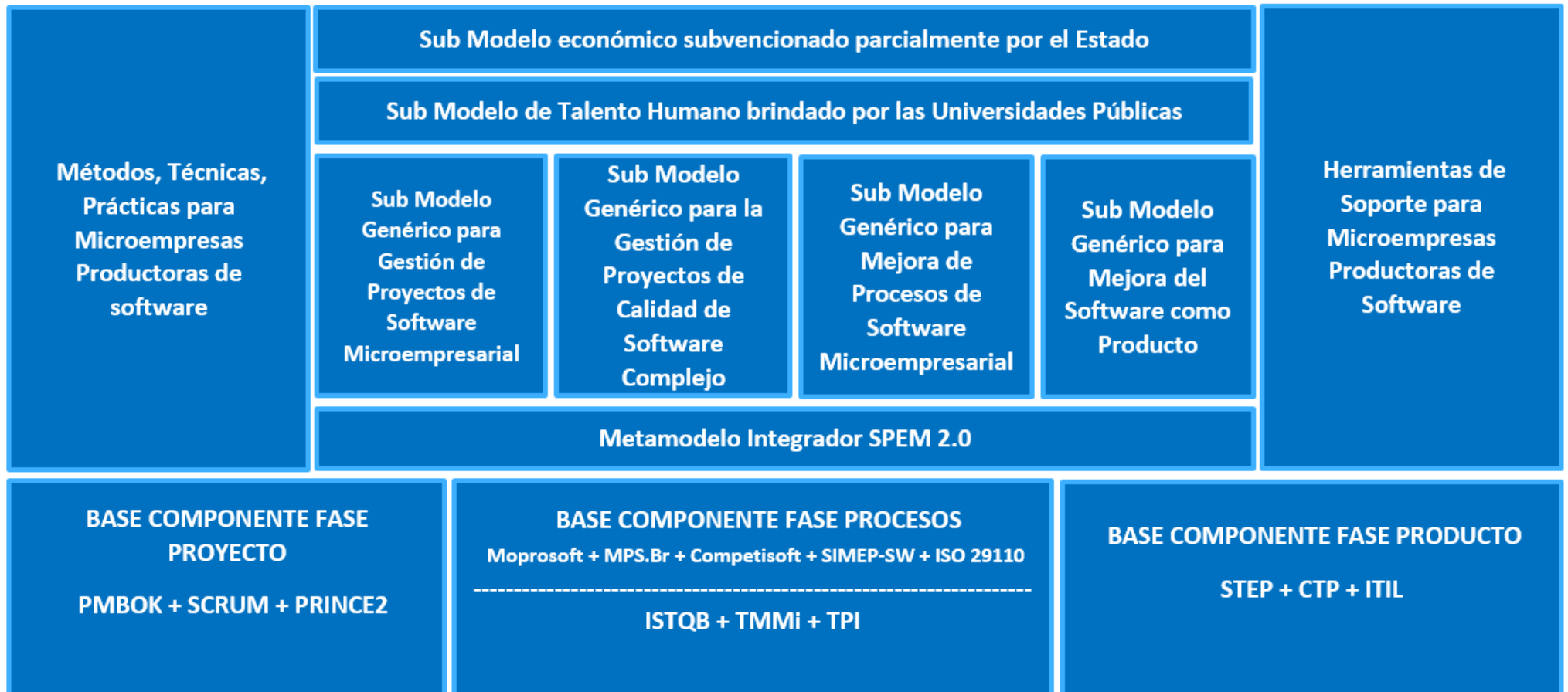
4.1. DESCRIPCIÓN DE SOFTMICRO.PE 1.0: MODELO DE CALIDAD PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN MICROEMPRESAS

Teniendo como premisa que las empresas productoras de software deben implementar proyectos de mejoramiento de procesos de producción y de calidad, existen en Perú y Latinoamérica diferentes esfuerzos que intentan fortalecer la industria de software de cada país, pero a nivel de pequeña y mediana empresa (PyMes), mas no se ha realizado esfuerzos equivalentes para el sector microempresarial. Ese esfuerzo ha ido enfocado hacia la mejora de los procesos de producción de software de calidad, de tal manera que les permita a estas empresas (Pymes) incrementar su competitividad. La certificación de calidad del proceso de producción de software y de los productos de él derivados es un paso que tarde o temprano las microempresas productoras de software deben dar como respuesta a dos situaciones: la primera, por imagen, para incursionar y mantenerse en un mercado global; la segunda, por necesidad, para poder hacer de sus proyectos unidades administrativas eficientes y eficaces. Amable (2015)

La mayoría de estos esfuerzos, están enfocados a trasladar los requisitos que imponen los modelos como el CMMI e ISO a empresas típicas en Latinoamérica: micro, pequeña y mediana empresa de software. Los más representativos de estos esfuerzos son mps Br , Mejoramiento del Proceso de Software en Brasil y en México, se ha desarrollado el modelo MoProSoft Modelo de Procesos para la Industria de Software en México que tiene como propósito fomentar la estandarización de su operación a través de la incorporación de las mejores prácticas en gestión e ingeniería de software, adicionalmente a estos esta COMPETISOFT. Sin embargo, se necesita conocer las diferencias entre una microempresa y una pequeña y mediana empresa, además de conocer los requisitos que estos modelos brindan, esfuerzos orientados a la implementación de estos requisitos dentro o fuera del marco de un proyecto de mejora.

En el marco del modelo propuesto SOFTMICRO.PE se ha definido una estrategia de mejora, la cual intenta cubrir dos esfuerzos: el de alivianar requisitos y guiar en el proceso de mejora, así como el de generar un conjunto de recomendaciones prácticas para la implementación de los requisitos del proceso software.

SoftMicro.Pe 1.0: Modelo de Calidad para la Gestión de Proyectos de Software en Microempresas



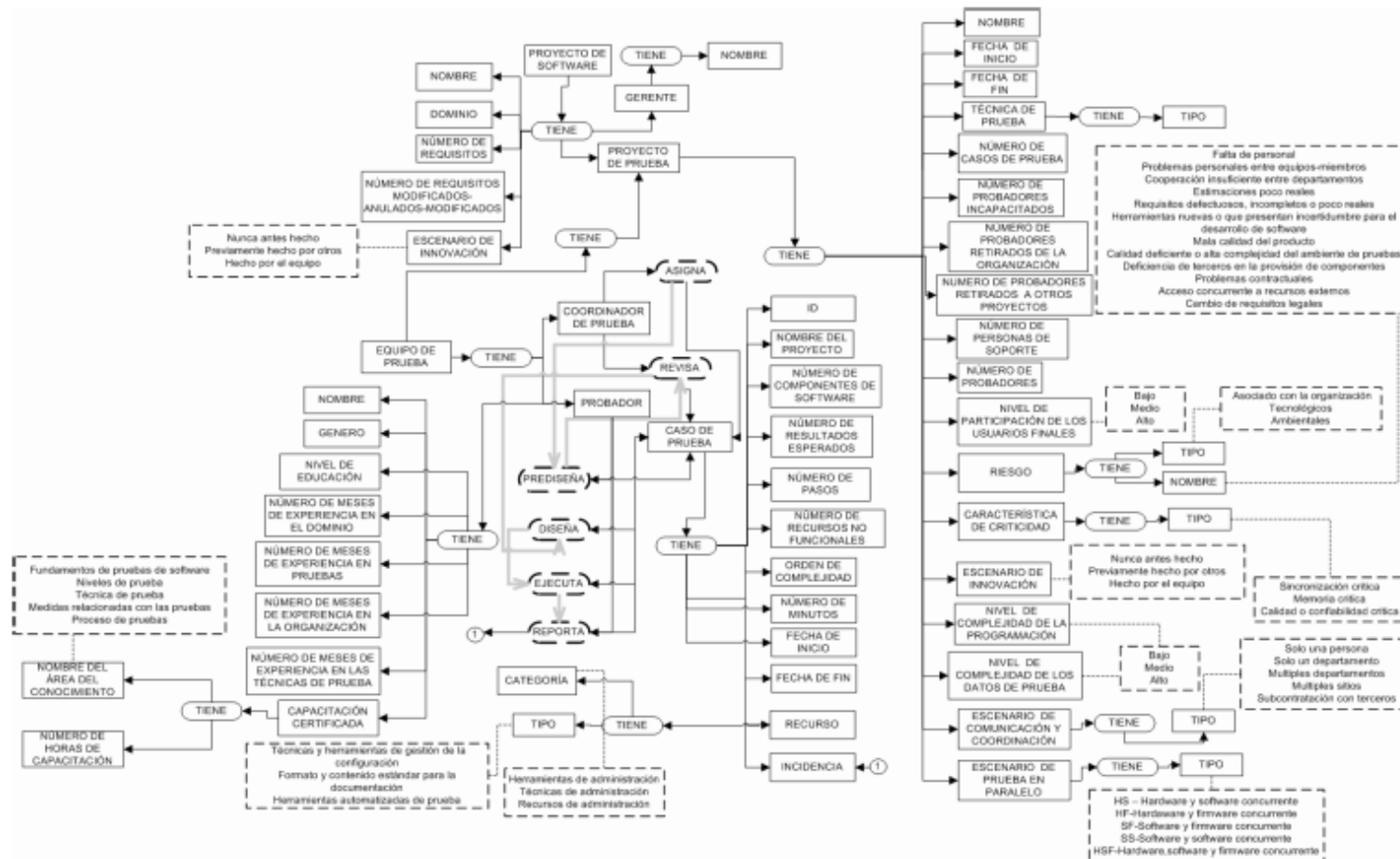


Figura 25. Submodelo de Genérico para la Mejora de Procesos de Software Microempresarial

Sub Modelo económico subvencionado parcialmente por el Estado

Este submodelo está conformado por la intervención de las siguientes entidades del estado:

Concytec y SUNAT mediante la ley 30309, y su decreto de urgencia que deduce impuestos hasta un 215% a las empresas que invierten proyectos de ciencia, tecnología e innovación tecnológica.

Fondecyt.

Pronabec, mano de obra calificada subvencionada por el Estado, obligados a trabajar para el estado.

Ministerio de la Producción, mediante sus proyectos adscritos como Innóvate Peru, Start Up Peru , Inacal y COFIDE

La secretaria de Gobierno Digital para gestionar los proyectos y establecer normas técnicas peruanas.

Congreso de la Republica para proyectos de Ley.

Universidades Nacionales mediante las instalaciones de sus Facultades de Ingeniería de Sistemas, Software, Informática y/o afines

Aportes de Empresas Ángeles: Empresas Privadas, Universidades Privadas, ONGs, Fundaciones Internacionales, BID , FMI , BM.

Sub Modelo de Talento Humano proporcionado por las Universidades Publicas

Las Universidades publicas proporcionan la lista de estudiantes de Ingeniería de Sistemas, Ingeniería de Software, Ingeniería Informática y/o afines.

Con las siguientes características:

1. Pertenecen al ranking de Excelencia Académica de su universidad promedio de todos los años de estudio perteneciendo al Tercio Superior, Quinto Superior, Decimo Superior y Primeros Puestos.
2. Estar cursando el Penúltimo o Ultimo ciclo o semestre de estudios universitarios.

3. Que obtengan en el examen nacional Software Fortune 500 (Evaluación provista en conjunto por Amazon, Google, Airbnb, Microsoft, Oracle), obteniendo un puntaje equivalente mayor a 15 puntos en sistema vigesimal.
4. Son puntos adicionales si cuentan con estudios técnicos relacionados a su carrera profesional, Experiencia Laboral Pre-Profesional, nivel de dominio del idioma inglés, Concursos Ganados relacionados a temas de Sistemas, Software y/o Informática, Primeros puestos en su educación primaria y secundaria.

Sub Modelo Genérico para Gestión de Proyectos de Software Microempresarial

Está basado en mayor porcentaje en 50% por SCRUM (Desarrollo Ágil de Software), 25% de PMBOK (Adquisiciones, Comunicaciones, Stakeholders) , 25% de PRINCE2 (Adquisiciones, Comunicaciones, Stakeholders)

Sub Modelo Genérico para Mejora de Procesos de Software Microempresarial

Está basado en la mejora de procesos,

Sub Modelo Genérico para Mejora del Software como Producto

Con el CPI, ITIL

Metamodelo Integrador SPEM 2.0

Las microempresas se formalizarían por las siguientes razones:

- Apoyo económico del Estado proporcionado mano de obra que ya subvenciona mediante becas de estudios universitarios, y estas tienen que ser retribuidas a la nación, por medio de la adopción del modelo propuesto.
- Deducción de impuestos hasta el 215% de la inversión en proyectos de ciencia, tecnología y/o innovación.
- Plataforma tecnológica del Estado al Servicio de las Microempresas.

El Estado estaría de acuerdo con este modelo de calidad por las siguientes razones:

- Aumento del PBI de nuestro país.
- Generación de empleo directo e indirecto.
- Incremento de la renta per cápita.
- Aumento de inversión privada (Google pondría en Lima su sede de desarrollo tecnológico al igual que en Belo Horizonte).
- Aumento de la actividad económica.
- Reducción de la Pobreza. Ejemplo Caso de la India.
- Reducción de la Deuda Externa.

Sub Modelo Genérico para la Gestión de Proyectos de Calidad de Software Complejo

Está basado en el desarrollo basado en Arquitecturas SOA, Cloud Computing, Aplicaciones Móviles y la integración entre estas.

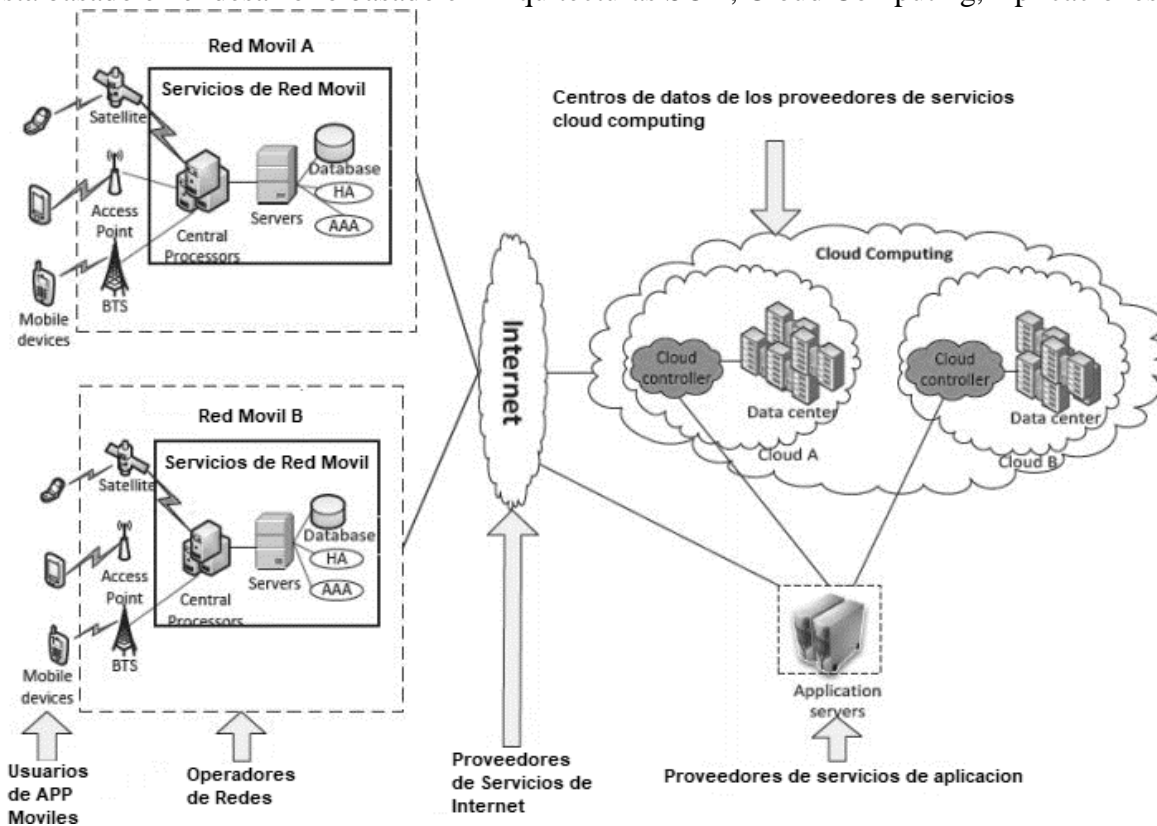


Figura 26. Submodelo de Genérico para la Gestión de Proyectos de Calidad de Software Complejo

4.2. CONTRASTACIÓN DE HIPÓTESIS

Para probar nuestra hipótesis nula general de la presente tesis procederemos a probar las hipótesis nulas específicas planteadas en el Capítulo II del presente documento, las cuales son:

La hipótesis específica 1 es:

El diseño de procesos flexibles, adaptables y escalables con sus respectivas actividades y herramientas de soporte a un grado gestionable y técnicos permitirá que proyectos en calidad en software estén alineados a las metodologías más modernas de desarrollo de software.

La hipótesis específica nula 1 será:

El no diseñar procesos flexibles, adaptables y escalables con sus respectivas actividades y herramientas de soporte a un grado gestionable y técnicos ocasionará que los proyectos en calidad en software no estén alineados a las metodologías más modernas de desarrollo de software.

Para probar esta hipótesis nula utilizaremos la Prueba del Chi Cuadrado, considerando los siguientes pasos:

Tomamos la muestra, siendo en nuestro caso un total de sesenta (60), la cantidad muestral de jefes y profesionales que gestionan y/o participan en gestionar proyectos de Calidad en Software de la microempresa ARWEBSYSTEMS SAC.

Elaboramos la tabla de contingencia en filas y columnas, y se distribuyeron en las áreas o criterios: Conocimiento de metodologías de Gestión de Proyectos , Aplicación de metodologías de Administración de Proyectos , Gestión de Proyectos de Calidad de Software Expectativas de aplicación de metodologías para Gestionar de proyectos en Calidad de Software.

Aplicamos la fórmula :

Fo = Frecuencia observada

Fe = Frecuencia esperada

$$X^2 = \sum \frac{(Fo - Fe)^2}{Fe},$$

Donde:

F_o= La frecuencia observada se obtendrá de las respuestas brindadas por los jefes y profesionales mediante el instrumento número 1.

F_e= La frecuencia esperada es una frecuencia probabilística (al azar) que se obtendrá aplicando la siguiente fórmula:

$$F_e = \frac{\text{Total de cada fila} \times \text{Total de cada columna}}{\text{Total General}}$$

Consideramos un nivel de significación de 0.05 (5%)

Consideramos un Grado de libertad igual a 8, a través de la fórmula:

$$G_L = (N^\circ \text{ filas} - 1) * (N^\circ \text{ columnas} - 1)$$

$$G_L = (4 - 1) * (3 - 1) = 6$$

Tabla 22:
Áreas de Conocimiento de la Investigación

ÁREAS	ALTO	MEDIO	BAJO	Total
Conocimiento de metodologías de Gestión de Proyectos	13	9	2	24
Aplicación de modelamientos de Gestión de Proyectos	2	3	11	16
Gestión de Proyectos de Calidad de Software	2	4	12	18
Expectativas de utilización de los modelamientos de Administración de proyectos de Calidad de Software	3	5	13	21
Total	20	21	38	79

Fuente: Elaboración Propia

El valor teórico del Chi-cuadrado con un nivel de libertad igual a 6 y con un nivel de significación igual a 0.05 (5%) corresponde en la tabla a un valor de $X_{02} = 15.50$, por lo tanto, la hipótesis nula fue rechazada ya que el valor obtenido del Chi-cuadrado con los datos de la tabla es mayor a 15.50, tal como se muestra a continuación:

Tabla 23:
Resultado aplicando Chi cuadrado a la Investigación

Fo	Fe	Fo-Fe	(Fo-Fe) ²	(Fo-Fe) ² /Fe
2	7	-5	25	3.57
0	7	-7	49	7.00
0	7	-7	49	7.00
0	7	-7	49	7.00
34	7	27	729	104.14
26	16	10	100	6.25
6	16	-10	100	6.25
11	16	-5	25	1.56
10	16	-6	36	2.25
25	16	9	81	5.06
32	37	-5	25	0.68
54	37	17	289	7.81
49	37	12	144	3.89
50	37	13	169	4.57
1	37	-36	1296	35.03
$\Sigma=$ 300	300		$\Sigma=$ 202.06	

Fuente: Elaboración Propia

La hipótesis específica 2 es:

El diseño de procesos, actividades y herramientas factibles de gestión permitirá a las microempresas productoras de software gestionar proyectos de calidad de software con factibilidad económica, operativa y técnica.

La hipótesis específica nula 2 será:

El no diseño de procesos, actividades y herramientas factibles de gestión no permitirá a las microempresas productoras de software gestionar proyectos de calidad de software con factibilidad económica, operativa y técnica.

Para probar esta hipótesis nula utilizaremos la Prueba del Chi Cuadrado, considerando los siguientes pasos:

Tomamos la muestra, siendo en nuestro caso un total de doscientos cuarenta (240), cantidad muestral de usuarios que utilizan el servicio calidad de software que brinda la ARWEBSYSTEMS SAC

Elaboramos la tabla de contingencia en filas y columnas, y se distribuyeron en las áreas o criterios: Servicios Registrales, Calidad del Servicio, Infraestructura Tecnológica, Proyectos tecnológicos y Expectativas.

- Aplicamos la fórmula:

Fo = Frecuencia observada

Fe = Frecuencia esperada

$$X^2_t = \sum \frac{(Fo - Fe)^2}{Fe}$$

Donde:

Fo= La frecuencia observada se obtendrá de las respuestas brindadas por los usuarios del servicio registral mediante el instrumento nro 2.

Fe= La frecuencia esperada es una frecuencia probabilística (al azar) que se obtendrá aplicando la siguiente fórmula:

$$Fe = \frac{\text{Total de cada fila} \times \text{Total de cada columna}}{\text{Total General}}$$

Consideramos un nivel de significación de 0.05 (5%)

Consideramos un Grado de libertad igual a 8, a través de la fórmula:

$$Gl = (N^\circ \text{ filas} - 1) * (N^\circ \text{ columnas} - 1)$$

$$Gl = (5 - 1) * (3 - 1) = 8$$

Tabla 23:

Resultado aplicando Chi cuadrado – Hipótesis específica 2

ÁREAS	BUENO	REGULAR	MALO
Conocimiento de modelamientos de Gestión de Proyectos	13	9	2
Aplicación de modelamientos de Gestión de Proyectos	2	3	11
Administración de Proyectos de Calidad de Software	2	4	12
Expectativas de utilización de modelamientos para Gestionar proyectos en Calidad en Software	3	5	13
Total	20	21	38

Fuente: Elaboración Propia

El valor teórico del Chi-cuadrado con un nivel de libertad igual a 8 y con un grado de significación igual a 0.05 (5%) corresponde en la tabla a un valor de $X_{02} = 15.50$, por lo tanto la hipótesis nula fue rechazada ya que el valor obtenido del Chi-cuadrado con los datos de la tabla es mayor a 15.50, tal como se muestra a continuación:

Tabla 24:
Resultado aplicando Chi cuadrado – Hipótesis específica 2

Fo	Fe	Fo-Fe	(Fo-Fe) ²	(Fo-Fe) ² /Fe
10	52	-42	1764	33.92
0	52	-52	2704	52.00
24	52	-28	784	15.08
95	52	43	1849	35.56
133	52	81	6561	126.17
218	139	79	6241	44.90
178	139	39	1521	10.94
192	139	53	2809	20.21
0	139	-139	19321	139.00
107	139	-32	1024	7.37
12	49	-37	1369	27.94
62	49	13	169	3.45
24	49	-25	625	12.76
145	49	96	9216	188.08
0	49	-49	2401	49.00
$\Sigma =$	1200	1200		$\Sigma =$ 766.37

Fuente: Elaboración Propia

La hipótesis específica 3 es:

El diseño de una arquitectura tecnológicamente moderna permitirá a las microempresas productoras de software gestionar proyectos de calidad de software complejo bajo plataformas SOA, Móviles, Cloud Computing o la integración entre éstas.

La hipótesis específica nula 3 será:

El no diseño de una arquitectura tecnológicamente moderna no permitirá a las microempresas productoras de software gestionar proyectos de calidad de software complejo bajo plataformas SOA, Móviles, Cloud Computing o la integración entre éstas.

Para probar esta hipótesis nula utilizaremos la Prueba del Chi Cuadrado, considerando los siguientes pasos:

Tomamos la muestra, siendo en nuestro caso un total de doscientos cuarenta (240), cantidad muestral de usuarios que utilizan el servicio calidad de software que brinda la ARWEBSYSTEMS SAC.

Elaboramos la tabla de contingencia en filas y columnas, y se distribuyeron en las áreas o criterios: Conocimiento de modelamientos de Gestionar Proyectos, Gestionar Proyectos en Calidad en Software y Expectativas para la utilización en modelamientos para Gestionar proyectos en Calidad en Software.

Aplicamos la fórmula :

Fo = Frecuencia observada

Fe = Frecuencia esperada

$$X^2 = \sum \frac{(Fo - Fe)^2}{Fe},$$

Donde:

Fo= La frecuencia observada se obtendrá de las respuestas brindadas por los clientes del servicio de calidad de software mediante el instrumento nro 2.

Fe= La frecuencia esperada es una frecuencia probabilística (al azar) que se obtendrá aplicando la siguiente fórmula:

$$Fe = \frac{\text{Total de cada fila} \times \text{Total de cada columna}}{\text{Total General}}$$

Consideramos un nivel de significación de 0.05 (5%)

Consideramos un Grado de libertad igual a 8, a través de la fórmula :

$$Gl = (N^{\circ} \text{ filas} - 1) * (N^{\circ} \text{ columnas} - 1)$$

$$Gl = (5 - 1) * (3 - 1) = 8$$

Tabla 23:
Resultado aplicando Chi cuadrado – Hipótesis específica 3

ÁREAS	BUENO	REGULAR	MALO
Conocimiento de modelamientos de Gestión de Proyectos	13	9	2
Aplicación de modelamientos de Gestión de Proyectos	2	3	11
Administración de Proyectos de Calidad de Software	2	4	12
Expectativas de aplicación de las modelamientos de Gestión de proyectos de Calidad de Software	3	5	13

Fuente: Elaboración Propia

El valor teórico del Chi-cuadrado con un nivel de libertad igual a 8 y con un grado de significación igual a 0.05 (5%) corresponde en la tabla a un valor de $X_{02} = 15.50$, por lo tanto, la hipótesis nula fue rechazada ya que el valor obtenido del Chi-cuadrado con los datos de la tabla es mayor a 15.50, tal como se muestra a continuación:

Tabla 24:

Resultado aplicando Chi cuadrado – Hipótesis específica 3

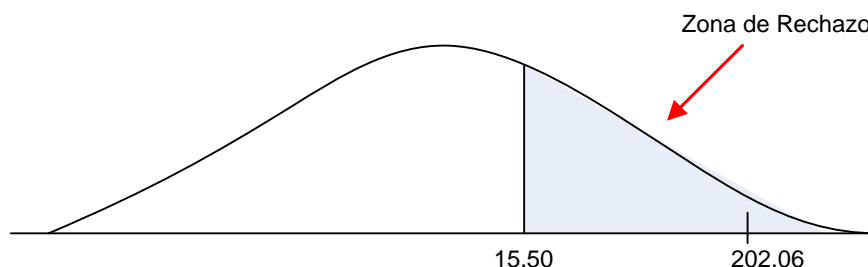
Fo	Fe	Fo-Fe	(Fo-Fe) ²	(Fo-Fe) ² /Fe
10	52	-42	1764	33.92
0	52	-52	2704	52.00
24	52	-28	784	15.08
95	52	43	1849	35.56
133	52	81	6561	126.17
218	139	79	6241	44.90
178	139	39	1521	10.94
192	139	53	2809	20.21
0	139	-139	19321	139.00
107	139	-32	1024	7.37
12	49	-37	1369	27.94
62	49	13	169	3.45
24	49	-25	625	12.76
145	49	96	9216	188.08
0	49	-49	2401	49.00
$\Sigma =$ 1200	1200		$\Sigma =$ 766.37	

Fuente: Elaboración Propia

4.3. ANÁLISIS E INTERPRETACIÓN

Hipótesis específica 1

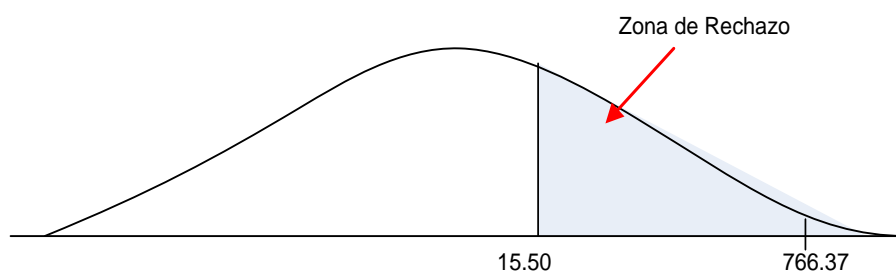
Aplicando Chi cuadrado en la hipótesis específica nula 1 se obtuvo el valor de 202.06, cayendo así en la zona de rechazo.



En tal sentido, se rechaza la hipótesis específica nula 1; concluyendo así que si aplicamos mecanismos eficaces y eficientes mejoraremos la administración de los proyectos tecnológicos de información de la ARWEBSYSTEMS SAC.

Hipótesis específica 2

Aplicando Chi cuadrado en la hipótesis específica nula 2 se obtuvo el valor de 766.37, cayendo así en la zona de rechazo.



En tal sentido, se rechaza la hipótesis específica nula 2; concluyendo así que el uso idóneo de herramientas de gestión en los proyectos tecnológicos de información permitirá hacer posible servicios con calidad al cliente interno y externo de ARWEBSYSTEMS SAC.

En consecuencia, al rechazar las hipótesis nulas específicas, se rechaza la hipótesis nula general, concluyendo que la aplicación del modelamiento del PMI - PMBOK mejorará la Administración de Proyectos Tecnológicos de Información de la ARWEBSYSTEMS SAC.

4.4. CASO DE APLICACIÓN DEL MODELO PROPUESTO

Contexto de aplicación

ARWEBSYSTEMS es la microempresa productora de software donde se utilizó el modelo, se formó hace 12 años para brindar servicios de producción de software a medida a empresas de cualquier rubro. Percibe utilidades menores a 150 UITs al año.

Entre sus clientes se encuentran entidades financieras, empresas del rubro retail y entidades públicas entre otros.

Como visión, tiene planeado brindar servicios de producción software a nivel sudamericano. Es por este motivo que requieren optimizar los procesos de sus proyectos de testing para mejorar la calidad del software producido.

Tabla 25:
Resultado cuestionario sección "Política Organizacional".

N°	Pregunta	Respuesta Afirmativas	Respuestas Negativas
1	¿Existen políticas y objetivos para el testing?	0	4
2	¿Existe política de planificación de pruebas?	0	4
3	¿Existen estándares definidos para testing?	0	4
4	¿La política de asignación de personal al testing es clara?	1	3
5	¿La política de asignación de otro tipo de recursos es conocida?	0	4
6	¿Existe un control para la efectividad de las pruebas?	1	3

Fuente: Elaboración Propia

Tabla 26
 Resultado cuestionario sección "Testing de Bajo Nivel"

N°	Pregunta	Respuesta Afirmativas	Respuestas Negativas
1	¿Existen capacitaciones para testing?	0	4
2	¿Existen instancias comparativas entre pares?	0	4
3	¿Se cuenta con herramientas (tecnológicas) para el testing?	3	1
4	¿Las pruebas se documentan?	2	2
5	¿Las pruebas se planean?	4	0

Fuente: Elaboración Propia

Tabla 27
 Resultado cuestionario sección "Diseño de Escenarios".

N°	Pregunta	Respuesta Afirmativas	Respuestas Negativas
1	¿Existe un diseño de pruebas?	0	4
2	¿Se conocen pruebas para determinados requerimientos?	1	3
3	¿Los procedimientos de ejecución de pruebas son conocidos?	0	4
4	¿Existe un control para el diseño?	0	4

Fuente: Elaboración Propia

Tabla 28
 Resultado cuestionario sección "Integración al Proyecto".

Nº	Pregunta	Respuesta Afirmativas	Respuestas Negativas
1	¿Existen instancias de encuentros entre testers y analistas/diseñadores?	0	4
2	¿Existen instancias de encuentro entre testers y programadores?	1	3
3	¿Las pruebas se basan en requerimientos?	3	1
4	¿Existe un documento cuantitativo de aprobación de pruebas?	0	4
5	¿Existen instancias comparativas de pares?	0	4

Fuente: Elaboración Propia

Tabla 29
 Resultado cuestionario sección "Prevención".

Nº	Pregunta	Respuesta Afirmativas	Respuestas Negativas
1	¿Los equipos (análisis, diseño, programación, testers) rotan sus roles?	0	4
2	¿Existen capacitaciones orientadas a la prevención?	0	4
3	¿Existe revisión entre pares?	0	4

4	¿Existe revisión entre los distintos roles?	1	3
5	¿Las pruebas detectan grandes fallas del sistema?	0	4

Fuente: Elaboración Propia

Tabla 30
Resultado cuestionario sección "Testing No Funcional".

N°	Pregunta	Respuesta Afirmativas	Respuestas Negativas
1	¿Existen pruebas no funcionales?	1	3
2	¿Existen métricas cuantitativas para el testing no funcional?	0	4
3	¿Existe mantenimiento preventivo para los proyectos?	0	4
4	¿La organización cuenta con capacitación?	0	4
5	¿Existen herramientas de evaluación de satisfacción de los clientes y usuarios finales?	0	4
6	¿Se busca entregar un valor agregado a los productos?	4	0

Fuente: Elaboración Propia

Tabla 31

Resultado cuestionario sección "Mejora Continua".

Nº	Pregunta	Respuesta Afirmativas	Respuestas Negativas
1	¿Existe rotación de roles/tareas entre los distintos actores?	0	4
2	¿Existen instancias de evaluación para el proceso de testing?	0	4
3	¿Las pruebas se basan en requerimientos?	0	4
4	¿Existe un documento cuantitativo de aprobación de pruebas?	0	4
5	¿Existen instancias comparativas de pares?	0	4

Fuente: Elaboración Propia

Tabla 32

Antes de la aplicación del Sub Modelo Genérico para Mejora de Software como Producto

Sección aplicación	de Respuestas positivas (%)	Respuestas negativas (%)
Niveles de Servicio	18,33%	81,67%
Mantenimiento Correctivo	35%	753%
Mantenimiento Evolutivo	16,45%	82,75%
Escalamiento	11.9%	88.1%
Acuerdos de Atención	4.3%	95.7%

Fuente: Elaboración Propia

Tabla 33

Después de la aplicación del Sub Modelo Genérico para Mejora de Software como Producto

Sección aplicación	de Respuestas positivas (%)	Respuestas negativas (%)
Niveles de Servicio	98,33%	2,67%
Mantenimiento Correctivo	75%	25%
Mantenimiento Evolutivo	60,45%	30,75%
Escalamiento	89%	11%
Acuerdos de Atención	83.6%	16.4%

Tabla 34
Comentarios de respuestas.

N° Tabla	N° Pregunta	Comentarios
5.1	2	NO. Pero existen períodos establecidos de marcha blanca planificados por contrato.
5.2	1	NO. Pero existe un interés en informarse en la medida que es posible.
5.2	2	NO. Cada uno es responsable de sus tareas.
5.2	4	NO. Existe un registro pobre a través de e-mail.
5.2	5	SÍ. Hay un programador que planifica las pruebas. SÍ. Al conocer los requerimientos se piensa en cómo probar.
5.3	2	SÍ. En desarrollo web al conocer los requerimientos ya se sabe qué probar, ya que muchos errores son de compatibilidad de browsers.
5.4	3	SÍ. Siempre
5.5	5	NO. Errores pequeños generalmente. NO. Errores de requerimientos principalmente.
5.6	1	SÍ. Usabilidad sobretodo. Fácil y rápido para todos.
5.6	6	SÍ. Usabilidad. SÍ. Estabilidad y cosas que convenzan al cliente de que somos la mejor opción.

Fuente: Elaboración Propia

Durante las visitas a la empresa se pudo conocer el estilo de trabajo que se aplica allí. Existen una secuencia de deficiencias que traban poder llevar un proceso de testing dedicado y de calidad.

El estilo de desarrollo es el siguiente: Los analistas se reúnen con los clientes para obtener los requerimientos del producto, luego estos mismos analistas se encargan de diseñar el sistema para luego, una vez que terminan, entregar las especificaciones a los programadores. Desarrolladores se ocupan de testear personalmente los módulos, para

hacer a los analistas testing de sistemas para ir en un rango de tiempo para el cliente que a menudo determina siendo los entregables el prototipado.

Tabla 35
Antes de la aplicación del Sub Modelo Genérico para Gestión de Proyectos de Software Microempresarial

Sección aplicación	de Respuestas positivas (%)	Respuestas negativas (%)
Política Organizacional	8,33%	91,67%
Testing de Bajo Nivel	55%	53%
Design	6,45%	92,75%
Integración de Proyectos	19%	81%
Prevenir	4%	93%
Pruebas No Funcionales	18,83%	77,17%
Mejora Continua	0%	100%

Fuente: Elaboración Propia

Tabla 36
Después de la aplicación del Sub Modelo Genérico para Gestión de Proyectos de Software Microempresarial

Sección aplicación	de Respuestas positivas (%)	Respuestas negativas (%)
Política Organizacional	90,33%	9,67%
Testing de Bajo Nivel	62%	38%
Design	96,45%	4,75%
Integración de Proyectos	85%	15%
Prevenir	94%	6%
Pruebas No Funcionales	78,83%	22,17%

Mejora Continua	70%	30%
-----------------	-----	-----

Tabla 37
Mejoras propuestas y aceptadas

Sección de aplicación	Mejoras
Política microempresarial	<p>Dar objetivos para el proceso de pruebas. ✓</p> <p>Dar políticas para planificar pruebas. ✓</p> <p>Estándares de testing. ✓</p> <p>Establecer recursos económicos para mejorarlas.</p> <p>Establecer recursos humanos para las mejoras.</p> <p>Establecer recursos tiempo para la mejoría. ✓</p> <p>Establecer un equipo de personas que participará en el cambio. ✓</p>
Testing de Bajo Nivel	<p>Generar instancias de revisiones entre distintos desarrolladores.</p> <p>Adquisición de herramientas para automatizar pruebas. ✓ (sólo herramientas libres de costos)</p> <p>Registrar y documentar pruebas. ✓</p>
Diseño	<p>Establecer una estructura para diseñar testing en función de los requerimientos. ✓</p> <p>Especificar los procesos de ejecución de pruebas.</p>
Integración al Proyecto	<p>Integrar el testing a todo el ciclo de desarrollo.</p> <p>Establecer un trabajo conjunto entre los distintos actores del ciclo.</p> <p>Fomentar instancias de evaluación entre pares.</p> <p>Establecer documentación evaluativa para aprobar los requerimientos. ✓</p>
Prevención	<p>Generar planificación de pruebas en fases de diseño. ✓</p>

Desarrollar modelos de pruebas que permitan encontrar errores críticos del sistema. ✓

Fuente: Elaboración Propia

Tabla 37
Mejoras propuestas y aceptadas

Sección de aplicación	Mejoras
Testing No Funcional	<p>Crear políticas en base a los requerimientos no funcionales a dar a los clientes. ✓</p> <p>Establecer métricas para el testing no funcional</p> <p>Adquisición de herramientas.</p> <p>Establecer planes de mantenimientos preventivos. ✓</p>
Mejora Continua	<p>Establecer un plan de mejoras continuas. ✓</p> <p>Crear procesos periódicos evaluativos para el testing. ✓</p>

Fuente: Elaboración Propia

Luego de entrevistar a los fundadores de la empresa para crear el plan de trabajo se ha concluido en un acuerdo de trabajo que, como se muestra en las tablas, no han aceptado algunas sugerencias, entre ellas destacan que:

No se dará presupuesto para la compra de herramientas ni contratación de empleados.

El testing de nivel bajo será mejorado de inicio, sin embargo, sólo en cuanto a diseño y planificación, pues no se considera gran interacción entre los distintos programadores para la mejora del testing. Esto, se debe principalmente a la forma de trabajo que hace complicado generar estas instancias de forma permanente.

El testing no se añadirá de forma formal al proceso de construcción, debido a la complicación de involucrar a todo el personal de forma permanente en el proceso, pero no existe el compromiso de que el testing esté vinculado a los requisitos del usuario final.

El testing no funcional se institucionalizará, pero no se invertirá en la mejora de pruebas ni en su forma de aprobación. No obstante, en un proceso de mantenimiento preventivo se añadirá la evaluación de estos puntos por parte de los usuarios.

Política Organizacional

Definición de metas para el testing

Los objetivos específicos que se definieron para cumplir el proceso de testing son los siguientes:

Verificar y validar los sistemas desarrollados por la organización.

Corregir los errores detectados durante los períodos de desarrollo de productos.

Prevenir defectos desde etapas tempranas del desarrollo de productos.

Definición de políticas para el testing

Las políticas definidas en la planificación si bien no son específicas y tienden a ser superficiales se deben a que estas políticas se formalizarán en la medida que el nuevo proceso de testing se implemente y se obtenga una claridad de los alcances que éste tendrá. Por lo tanto, inicialmente las políticas básicas para la planificación de las pruebas son:

Testing unitario implementado por propios programadores.

Testing de integración dirigido por jefes de proyecto.

Definición de condiciones especiales al inicio del proyecto.

Definición de estándares para el testing

Lo estándares definidos en esta primera aplicación, solo consideran aspectos relativos al alcance de las pruebas o la dedicación que éstas tendrán:

Las pruebas deben ser aprobadas en su 100%

Por cada 5 horas de codificación corresponde un mínimo de 1 hora de trabajo dedicado al proceso de testing

Establecimiento de recursos para el testing

Normalmente un trabajo de 2 horas semanales presencialmente en el lugar de trabajo por parte del encargado de implementar las mejoras.

Se definen tareas a cumplir por parte de la organización y del encargado de mejoras. Los cuales deberán cumplir los plazos que se establezcan para el correcto funcionamiento de las mejoras.

Establecimiento de los miembros del equipo de testing

Analistas/diseñadores (2 trabajadores).

Programadores (2 trabajadores).

Testing Técnico

Aprovisionamiento de herramientas open source para automatización de testing

PHP CodeSniffer

Si bien la idea era automatizar la ejecución de pruebas en sí, tras la recolección de información que permitiera encontrar las herramientas adecuadas para la mejora de las pruebas se llegó a la conclusión de que no es prioridad la automatización. Sin embargo, se implementó el uso de una herramienta que permita controlar de mejor forma los estándares en la sintaxis de programación (estándares en desarrollo). Dado el mayor orden obtenido se consiguió mayor facilidad en la comprensión de códigos lo que se traduce en beneficios a la hora de corrección de módulos ante defectos detectados en las pruebas.

PHPUnit

Se experimenta con la automatización de las pruebas de unitarias a través del uso de este framework. Considerando que el proceso formal de pruebas dentro de la organización aún es inmaduro, el uso de esta herramienta hasta este punto se usó de forma experimental, mientras no se definan completamente que componentes del proceso de pruebas es el correcto para el desarrollo obtenido.

Documentación del testing

Para mantener un correcto historial de las pruebas de bajo nivel que se realizan se decidió que las pruebas deben contener ciertos aspectos que se detallan a continuación:

Consideraciones sobre el plan de pruebas

Antes de la explicitar el plan de pruebas de un proyecto, es necesario conocer como está conformado el producto a testear para poder establecer una línea base que permita la correcta planificación, alcance, ejecución y control de las pruebas.

Un resumen de las funcionalidades del sistema puede ayudar para la comprensión de esta documentación en el futuro, sin tener que recurrir a todo el material destinado al proyecto total. Es importante considerar los módulos que contempla y su interacción con otros. Estas consideraciones deben incluir aspectos relevantes tanto para las inspecciones de código como para las pruebas en sí.

Formulario de inspección

La tabla 5.11 es el documento que permitirá tener información sobre las inspecciones que se hayan realizado a algún trozo de código, funcionalidad o módulo. Se describe a continuación.

Tabla 38
Formulario de inspección.

Responsable:	Nombre de quien inspecciona el código.
Sección:	Especificación del número de líneas del código fuente o funcionalidad o módulo a inspeccionar.
Fecha:	Fecha en que se realiza la inspección.
Resultados:	Conclusiones de la inspección y/o explicitación de los defectos detectados. Especificar incongruencias entre la codificación y documentos de diseño.

Fuente: Elaboración Propia

Ambiente del sistema

El ambiente de pruebas debe ser, en lo posible, idéntico al ambiente en donde el sistema será explotado finalmente. De no ser así, se debe especificar las diferencias y dejar constancia las posibles consecuencias en la ejecución del sistema. Sin embargo, lo importante es registrar las condiciones del ambiente de prueba.

Calendario de pruebas

El ideal es que exista un calendario a seguir desde el inicio del proyecto. No obstante, cualquier cambio que se produzca debe ser documentado, con mayor motivo si el período de pruebas se extiende debido a fallas de implementación detectadas en el proceso de pruebas. El calendario debe cumplir con el mínimo establecido en las políticas organizacionales.

Formulario de pruebas

La tabla 5.12 es el documento que permitirá tener información sobre las pruebas que se hayan realizado en algún módulo. Entendiendo la gran cantidad de pruebas a realizar y apelando a la libertad propuesta para testear de cada programador, debe ser usado con el fin de registrar los casos de prueba más relevante.

Tabla 39
Formulario de descripción/resultados de la prueba.

Nº Prueba: Número de la prueba.	
Encargado: Persona que realiza la prueba.	
Fecha: Fecha cuando se realiza la prueba.	
Sección: Función, módulo o conjunto de módulos a probar.	
Descripción: Breve descripción de la prueba.	
Resultados esperados: Resultados que el usuario espera del elemento a probar.	
Entradas: Datos que fueron ingresados.	
Salidas: Datos que se entregaron como resultado a las entradas.	
Actividades: pasos que se realizaron durante la prueba.	
Observaciones: explicitación de defectos, situaciones anómalas, suspensiones o simples comentarios de la prueba y/o elemento probado.	
Nombre	Firma

Fuente: Elaboración Propia

Diseño

Definición de un diseño estructural de testing en base a las especificaciones

Para poder diseñar las pruebas en base a los requerimientos es necesario que las especificaciones de cada proyecto se definan, entendidos y convenidos tanto por los clientes como por la organización desarrolladora. Para esto, se ha debido generar una

estructura que lleve a cabo la forma en que se captan los requerimientos de los clientes para acordar el alcance de los proyectos.

La estructura considera los siguientes aspectos:

Definición del Problema y Objetivos.

Descripción del Problema.

Objetivo General.

Objetivos Específicos.

Especificación de Requerimientos.

o Requerimientos Generales.

o Requerimientos Específicos del usuario.

o Requerimientos No Funcionales.

o Casos de Uso Gráfico.

Una vez formalizados los requerimientos y expectativas del usuario se establece un esquema para diseñar las pruebas que permitan probar lo que corresponda a cada proyecto. La tabla 5.13 explica los contenidos mínimos que debe cumplir los casos de pruebas según el requerimiento otorgado. Cabe destacar que este esquema se aplica a los requerimientos específicos de cada proyecto (funcionalidades propias del software), por lo que para las pruebas de bajo nivel que buscan la verificación de un determinado módulo genérico (como el validar una fecha, nombre, u otro dato) se aplica el criterio establecido inicialmente para las pruebas de bajo nivel.

Tabla 40

Esquema para diseño de pruebas.

Requerimiento: Se especifica el requerimiento especificado por el usuario
Tipo de prueba: Definir si será de tipo caja blanca o caja negra, en base a la funcionalidad/requisito a testear.
Especificaciones de la prueba: En qué consiste la prueba.
Requerimientos para la prueba: Necesidades de las prueba
Proceso de obtención de datos: Especificación de cómo se obtendrán la data y qué tipo de data son necesarios.

Fuente: Elaboración Propia

Inclusión al Proyecto

Elaboración de un documento de métricas para la aceptación de las especificaciones

Este documento tiene por objetivo involucrar en forma directa el proceso de planeación del testing en etapas tempranas del proyecto, logrando una integración del proceso de testing a lo largo del ciclo de vida. Específicamente, lo que pretende es que el analista, al momento de captar los requerimientos de los clientes, pueda definir un esbozo de las pruebas que serán necesarias para conseguir la aprobación de los requerimientos. Este documento debe definir en parámetros medibles ciertas condiciones que han de cumplir las pruebas para considerarlo satisfactorias.

Esta modalidad solo se utilizó en uno de los proyectos en que se implementaron mejoras de testing. Contextualizando se trataba de un sitio web, que debía exhibir a los clientes.

Tabla 42
Métricas para aprobación de requerimientos.

Requisito	Pruebas	Métrica
Recomendar propiedades vía e-mail a conocidos	Enviar a múltiples usuarios Evitar filtro de spam	Lograr llegar a la bandeja de entrada a 5 dominios relevantes (Hotmail.com (y sus variaciones), gmail.com, yahoo.es, terra.cl, vtr.net)
Solicitar cita para visita de propiedad	Recepción de correo electrónico al corredor Solicitar citas a una misma hora y día	Confirmar 1 cita, tras la solicitud de 2 (o más) concurrentes.

Enviar propiedad para evaluación	Subir múltiples imágenes digitales.	Subida exitosa de 3 formatos: jpg, npg, bmp.
	Indicar características de propiedad	Comprobar en búsqueda con resultado 100% de coincidencia
Buscador de propiedades por características	Establecer características para propiedades	Obtener resultados precisos en un 100%
	Obtener resultados de búsqueda correctos	Sugerir máximo 2 propiedades que no cumplan características. (rango de precios siguiente o sector habitacional cercano)

Fuente: Elaboración Propia

Tabla 42
Antes de la aplicación del Sub Modelo Genérico para Gestión de Proyectos de Calidad de Software Complejo

Sección aplicación	de Respuestas positivas (%)	Respuestas negativas (%)
SOA	5,33%	94,67%
Cloud Computing	10%	90%
APP Móviles	6,45%	92,75%
Integración de Tecnologías	11.5%	88.5%

Fuente: Elaboración Propia

Tabla 43
Después de la aplicación del Sub Modelo Genérico para Gestión de Proyectos de Calidad de Software Complejo

Sección aplicación	de Respuestas positivas (%)	Respuestas negativas (%)
SOA	98,33%	1,67%
Cloud Computing	73.5%	26.5%
APP Móviles	96,45%	2,75%

Integración de Tecnologías	89.6%	10.4%
----------------------------	-------	-------

Prevención

Planificar las pruebas en la fase de diseño

Con el fin de prevenir futuras complicaciones en los productos que sean detectadas durante la fase de pruebas, se debe comenzar a planificar las pruebas mientras se esté diseñando el sistema. Esto, tendrá un impacto relevante en la medida que más proyectos documentados existan, pues en la medida que los diseños dispongan de mayor documentación pasada para planear las pruebas resultará más fácil el predecir ciertos errores típicos del pasado.

Dado que esta es la primera construcción de la mejora en el testing, no existe la documentación, durante esta primera aplicación se realizó el ejercicio de discutir experiencias pasadas que permitieran predecir errores que pudieran parecer típicos en ciertos tipos de proyectos. Esto, trajo como resultado el evitar ciertos trabajos innecesarios y decisiones más rápidas en cuanto a tipos de tecnologías utilizadas en los desarrollos, lo que permitió ahorrar tiempo y esfuerzo por parte de los programadores.

Implementar el modelo de testing que encuentre errores de alta criticidad en el software bajo prueba

A diferencia de los tipos de pruebas aplicados normalmente, que solo buscan probar las funcionalidades de los distintos módulos, lo que se pretende ahora es desarrollar pruebas con la intención de “botar” el sistema desarrollado, de modo de encontrar errores críticos que no permitan la operación del producto testeado. Esta etapa del testing, se da como consecuencia de un proceso de testing correctamente gestionado en los otros aspectos.

Este aspecto del testing no fue posible implementarse durante la aplicación del modelo de mejoras, principalmente a causa de falta de tiempos debido a atrasos inesperados

(aunque comunes) en los proyectos que participaron en las mejoras. Sin embargo, se declaró como forma de trabajo la siguiente metodología:

El sistema debe ser testeado como un todo durante esta etapa.

El sistema en lo posible según las capacidades organizacionales debe ser testeado por un programador ajeno al proyecto.

Dentro de la ejecución de pruebas se pueden premeditar situaciones que ponga en riesgo la consistencia del sistema (por ejemplo: manipulación errónea en la base de datos, manipulación directa de los datos).

Cualquier falla detectada debe ser reparada antes de dar de alta el proyecto.

Los errores detectados en un proyecto que puedan estar replicados en proyectos pasados, implica la revisión obligatoria de dichos trabajos pasados (mantención preventiva).

Pruebas No Funcionales

Establecimiento de políticas basadas en las especificaciones no funcionales para hacer entrega a los clientes propietarios del software bajo prueba

Las políticas pueden variar según la naturaleza de cada proyecto, sin embargo con el fin de establecer un sello característico en los productos elaborados por la organización se definen ciertas características no funcionales que deben considerarse en los desarrollos. Estas políticas, para cumplirse, implican que deben implementarse cambios en las etapas de desarrollo para así como en el diseño de pruebas para comprobar su buena aplicación.

Especificaciones de entrega

No se requieren pruebas específicas para comprobar el cumplimiento de este aspecto. Basta con verificar si se cumplen plazos, condiciones, formas y otros detalles que han de establecerse con el cliente.

Usabilidad

En vista de que gran parte de los desarrollos son con interfaz de un sitio web, lo que implican interacción con usuarios que no estarán familiarizados con el sistema, se hace muy necesario entregar sistemas con un gran nivel de usabilidad.

Para la verificación de este aspecto, mediante el uso de pruebas, se deben añadir evaluaciones heurísticas en los procesos que permitan detectar problemas en las interfaces. Para esto, el encargado de las mejoras ha debido instruir con casos prácticos sobre el cómo abordar las heurísticas implicadas en este proceso.

La tabla 5.15 es un resumen de problemas detectados en un proyecto de sitio web promocional de un hotel luego de realizar una evaluación heurística con cuatro evaluadores (personal de la organización que fue capacitado). En la columna del “principio no cumplido” se utiliza la letra “S” representa el promedio de severidad del problema (en una escala de 0 a 4) y la letra “F” el promedio de la frecuencia del problema (en una escala de 0 a 4).

Tabla 40
Resumen de evaluación heurística 01

N°	Explicación	Ejemplo	Principio no cumplido
1	La página principal sobrecargada de información. Lo que dificulta la ubicación de lo que se busca.	Si se busca el e-mail de contacto cuesta mucho encontrarlo.	Diseño estético y minimalista. S:3,25 F: 2
5	Inconsistencia en el uso de menús en distintos navegadores.	El menú de accesos directos en Mozilla Firefox lleva al link seleccionado al hacerle click. En Internet Explorer se elige el link y luego se clickea el botón Ir para ir a la página seleccionada.	Consistencia y estándares. S: 2 F :3

Fuente: Elaboración Propia

Tabla 41
Resumen de evaluación heurística 02

N°	Explicación	Ejemplo	Principio no cumplido
7	Distinta vista en navegadores.	En Mozilla Firefox se muestran 4 links en la parte inferior izquierda de la página principal. En Internet Explorer no se ven.	Consistencia y estándares. S: 2,25 F :2
10	Menú izquierdo muy extenso (11 opciones)	Muestra demasiadas opciones de una sola vez	Diseño estético y minimalista S:2,25 F :2
11	Menú izquierdo finaliza con opción inexistente	El menú izquierdo pareciera que no tiene fin, no está bien delimitado	Consistencia y Estándares S: 1,5 F :1
17	Imagen “Bienvenidos” del menú izquierdo no conduce a nada	La frase se confunde con un link	Compatibilidad entre el sistema y el mundo real. Consistencia y Estándares S: 3,75 F :1

Fuente: Elaboración Propia

Tabla 42
Resumen de evaluación heurística 03

N°	Explicación	Ejemplo	Principio no cumplido
----	-------------	---------	-----------------------

19	Links se abre en ventana nueva	Los menú abren en nuevas ventanas, lo que podría complicar la navegación por el sitio web	El usuario debe tener el control y libertad, S: 2 F :3
23	Espacio perdido, desaprovechado. En blanco.	A la derecha de la página, queda un espacio bastante grande que no se utiliza	Otro S: 2,25 F :2
24	No existe mapa del sitio		Consistencia y Estándares S: 3,25 F :1

Fuente: Elaboración Propia

*Tabla 34
Antes de la aplicación del Sub Modelo Genérico para Mejora de Procesos de Software Microempresarial*

Sección aplicación	de Respuestas positivas (%)	Respuestas negativas (%)
Comportamiento de equipo.	5,33%	94,67%
Compromiso de mejora constante.	14%	86%
Establecimiento de objetivos locales.	6,45%	92,75%
Establecimiento de mecanismos de medición.	11.5%	88.5%
Verificación de resultados.	32.7%	67.3%
Aplicación de medidas correctivas o preventivas	12.4%	87.6%

Fuente: Elaboración Propia

Tabla 35
Después de la aplicación del Sub Modelo Genérico para Mejora de Procesos de Software
Microempresarial

Sección aplicación	de Respuestas positivas (%)	Respuestas negativas (%)
Comportamiento de equipo.	80,33%	19,67%
Compromiso de mejora constante.	90%	10%
Establecimiento de objetivos locales.	62,45%	38,55%
Establecimiento de mecanismos de medición.	91.5%	8.5%
Verificación de resultados.	78.89%	21.11%
Aplicación de medidas correctivas o preventivas	97.89%	2.11%

Eficiencia

Para cumplir con estos requerimientos los sistemas deben ser sometidos a pruebas de “estrés”. De esta forma, se pretende sobrecargar de datos con el fin de determinar si los tiempos de respuesta del sistema, mediante las bases de datos, responden con los requisitos planteados para el proyecto. De esta forma, se medirán la capacidad de respuestas ante consultas, ingresos y modificaciones en los datos.

Además de las pruebas mencionadas, dado el interés de asegurar la eficiencia, adicionalmente el desarrollo de los sistemas deberá incluir una especificación sobre el hardware necesario para mantener su correcto funcionamiento. Esto, implica nuevas pruebas que permitan establecer cuáles son las características que deben cumplir los equipos en que se instalarán los sistemas.

Fiabilidad

En la medida que el testing detecte errores que no sean de codificación, se deben establecer proporciones de fiabilidad de los sistemas, con el fin de determinar si se

alcanzan los requerimientos del cliente en este aspecto. Por otro lado, se deben establecer los criterios de respaldo y resguardo de los sistemas, y probar las capacidades de recuperación de los sistemas antes una caída o pérdida de datos. Estas pruebas se ejecutan de forma manual y en base a casos específicos que puedan presentarse más casos poco probables.

Portabilidad

Si bien los sistemas son normalmente especificados para ciertos tipos de equipos para ser instalados y ejecutados, se realizarán pruebas en entornos similares (principalmente otros sistemas operativos a través de virtualización) para establecer capacidad de funcionamiento en entornos similares. Esto, con el fin de disminuir potenciales reclamos en caso de cambios no notificados en los equipos por parte de los usuarios.

Planes de mantenimiento de prevención

Para adelantarse a reportes de errores detectados por los usuarios una vez que ya ha sido entregado con el sistema se establecen planes de mantenimientos de tipo preventivos para los productos en funcionamiento.

Los plazos y ciclos para estos mantenimientos dependen de cada proyecto, sin embargo a modo general se establecen los aspectos mínimos que evaluarán estas tareas:

Pruebas de estrés y consistencia. Orientado a las bases de datos y su integridad

Testing a nivel de integración en base a conocimientos adquiridos por otros proyectos. Por ejemplo: errores detectados en proyectos similares de posterior desarrollo

Actualizaciones de tecnologías (bases de datos principalmente) lo que conlleva a nuevas pruebas de compatibilidad

Comprobación de copias de respaldo. Pruebas de restauración de sistema

Mejora continua

Establecimiento de un procedimiento evaluativo para el testing

Para verificar los alcances del proceso de testing y proponer modificaciones se establece que cada año, es decir, a partir de este momento, cada agosto de cada año

evaluar el proceso de testing. Así, se debe decidir cuales aspectos no están funcionando como se espera que lo hagan y qué acciones se pueden tomar para corregirlos.

Este trabajo no se resume simplemente a una revisión anual, pues se entiende que por cada desarrollo que es sometido al proceso de testing se pueden obtener lecciones y propuestas que podrán ser evaluadas año a año para generalizarlas y hacerlas partes del proceso de testing.

La evaluación no contará con el encargado de mejoras de testing, por lo que requiere de un compromiso de parte de la organización de valorar las mejoras alcanzadas con el fin seguir mejorando los procesos en general y en específico las pruebas.

En concreto, se acuerda que la evaluación debe realizarse con actores de los distintos niveles del ciclo de desarrollo y que debe contar con un mínimo de dos semanas en donde se dediquen espacios idóneos para los debates y profundización de propuestas que hagan.

Resultados

Al analizar el resultado obtenido con la construcción del modelo de mejora se debería de implementar un resumen histórico del trabajo. Primeramente, fue el área de cultura organizacional, no duró más de una semana en la cual se tomaron decisiones obligadas, luego se mejoró el testing de bajo nivel usando las tareas planificadas en los proyectos que ya se encontraban en construcción, y luego añadiéndose nuevos proyectos implementados en la implementación de mejoras. En paralelo se elaboraron 2 áreas del modelo: El Testing No Funcional y el Diseño, debido a que las características no funcionales se podían avanzar sin estar en un proyecto exclusivamente, asimismo las normas de diseño tampoco requerían afrontar un único proyecto por lo que se pudo culminar sin inconvenientes estas 2 áreas. Seguidamente se administraron las áreas restantes: Integración de Proyectos, Preventivas y Mejora Continua.

Posterior a todas las mejoras implementadas se establecieron reuniones con la organización de modo de obtener la retroalimentación necesaria para comprender y dimensionar los alcances que tuvo la aplicación del modelo. De estos encuentros se establecen las siguientes conclusiones:

- **Ciclo definido de testing**

Si se evalúa la situación al inicio del proceso de testing existe la situación alcanzada tras la utilización del modelo, se encontrarán diferencias determinantes en cuanto al nivel de procesos de desarrollos de los productos. La organización no poseía un proceso de testing definido dentro de su ciclo de desarrollo, ya que solo se dedicaban tiempos a testing, pero no se debería seguir algún estándar, políticas o expectativas planificadas.

Luego de la aplicación del modelo, se ha alcanzado un proceso de testing definido en tiempo, cobertura y otros aspectos. La microempresa ya cuenta con un proceso formal y los productos construidos son sometidos al modelo, según Sigcha Moya & Utreras Zapata(2018) lo que permite la fabricación de software con un mejor control de calidad pudiendo satisfacer el cumplimiento de requerimientos funcionales y no funcionales de parte de los clientes.

La organización se declara satisfecha con este primer paso, que, si bien aún puede ser mejorado, ya representa una mejora a nivel global del funcionamiento de su negocio. Tanto así, que los socios fundadores plantean la disposición a formalizar todo el proceso de desarrollado utilizado en su empresa.

- **Reducción del retrabajo**

Al elaborar la documentación del testing, tanto en su planificación, diseño y ejecución, se cuenta con información de valor para proyectos siguientes. Esto se pudo verificar en los proyectos producto de la implementación de las mejoras, ya que hizo posible la reutilización de casos de prueba en componentes de software enlazadas; y la carencia de pruebas en componentes idénticos de código fuente reutilizado.

Un ejemplo claro de esta situación se da con la creación de un sitio web que hubo que desarrollar posteriormente al proyecto con un hotel. Las pruebas heurísticas realizadas en el primer proyecto y resumidas en 5.3.6.1 permitieron un significativo ahorro de trabajo tras descubrir errores en el primer proyecto, que se tuvo gran cuidado de no volver a cometer en los proyectos siguientes.

Por otro lado, las pruebas realizadas a bajo nivel han permitido hoy un aumento significativo en la producción por parte de los programadores en base a dos factores: el primero como es evidente es la no repetición de los mismo errores de codificación y la

segunda y más interesante se da al cometer errores de programación y no recordar cómo resolverlos (situación que siempre se ha dado en la organización), sin embargo hoy existe una documentación que permite chequear la solución y solucionar problemas que antes incluso tardaban días, sobre todo cuando es otro programador el que se ve enfrentado a la situación.

En este aspecto quedan en absoluta evidencia las mejoras entregadas a nivel organizacional la aplicación correcta de un proceso de testing. En conversaciones con programadores, manifiestan que este aspecto es altamente relevante para ellos y sienten que su trabajo es mucho más eficiente, puesto que normalmente no es parte de sus funciones lidiar con los conflictos que manifiestan los usuarios tras la entrega, pero sí es cotidiano lidiar con sus propios errores y frustraciones a la hora de codificar los sistemas.

- Reducción de costos

Según la utilización de testing riguroso, mejor planificadas y diseñadas. Se han podido hallar errores de construcción con mayor itinerancia, si bien esto implica un costo de re-implementación que no ayuda a la agilidad de los desarrollos, es un costo por debajo del mercado al tener que enfrentar con un usuario no satisfecho o que descubra fallas una vez que le es dado el producto.

La organización dentro de su historia, nunca ha tenido que enfrentarse a situaciones críticas que comprometan el funcionamiento de sus productos tras la entrega final a sus clientes, sin embargo, se han dado ocasiones en que los usuarios son los que denuncian el hallazgo de pequeñas fallas y piden su corrección. El costo de tiempo, de personal y de oportunidad que implica el tener que “retomar” el desarrollo de un proyecto finalizado es tremendo para la organización, es por esto que el poder estar cada vez más seguros de que el cliente no manifestará problemas tras la entrega de su producto es tremendamente relevante para el buen funcionamiento de la empresa.

De los tres proyectos que fueron evaluados en la implementación del modelo de mejoras, tras casi dos meses de entrega del producto con más tiempo de funcionamiento no han existido reclamos posteriores por parte de los usuarios, al menos en el aspecto de funcionalidades. Esto marca un hito, históricamente el mayor número de solicitudes por parte de los clientes se da durante el primer mes de uso de su producto.

Toda esta detección temprana y su ahorro en costos han permitido a la organización poder continuar avanzando en nuevos proyectos sin correr el peligro de retomar viejos proyectos que pudieran atentar con la correcta atención que merece un cliente. Este estado sin “clientes insatisfechos” debe ser chequeado periódicamente a través de las mantenciones preventivas con el fin seguir consolidando su buena relación con los clientes.

- **Reducción de la insatisfacción de los clientes**

De lo anterior la disminución en la insatisfacción de los usuarios no solo se ha visto por el no descubrimiento de errores por su parte. Si el software cumple con sus especificaciones de los usuarios estarán relativamente satisfechos, pero si se hacen los esfuerzos por validar que se cumplan las mínimas características no funcionales, los clientes se manifiestan mejor.

Acá las pruebas de estrés fueron muy valoradas por la organización, ya que a pesar de siempre estar consideradas nunca se habían realizado debido a que no existía un procedimiento claro sobre cómo abordarlas. Su realización trajo como consecuencia bases de datos más estables y mejores tiempos, o al menos suficientes, de espera ante las distintas transacciones de los sistemas. Esto da como resultado usuario más a gusto con los nuevos sistemas y una mayor satisfacción de parte del cliente.

Durante los últimos encuentros de evaluaciones con la organización han manifestado los logros alcanzados en cuanto a la mejora de usabilidad en sus proyectos de desarrollo web. Ya que al ser sistemas que interactúan con usuarios sin preparación, éste aspecto se torna importante. Estas nuevas características integrados en sus desarrollos y comprobadas a través de pruebas especialmente diseñadas con ese fin, han permitido disminuir los plazos de entrega, en hasta un 10%, ya que los usuarios quedan satisfechos con los primeros prototipos o solo hacen pequeñas observaciones, lo cual en comparación a experiencias pasadas era totalmente distinto.

Otro factor que ha influido en la disminución de insatisfacción, a criterio de los dueños de la organización, es la formalización a través de documentos de la toma de requerimientos de los clientes. La estructura mencionada en 5.3.3.1 ha permitido mejorar la comunicación entre analistas y clientes, ya que existe un método más claro de hacer entender al cliente que se ha interpretado. Además, este documento evita de cierta forma al usuario al contradecir sus requerimientos y a los desarrolladores les

permite hacer mejor su trabajo y a los testers planear de manera mejor y con más tiempo las pruebas a la que los sistemas serán sometidos.

- **Mejoramiento del testing**

La mejora de testing dentro de la organización ha traído ventajas evidentes en muchos aspectos del desarrollo de software, sin embargo, para convencer a la microempresa, especialmente a los programadores. Esto se debe esencialmente a la burocrático que se adopta la mejora del proceso del testing. Al inicio se tiene que escribir los formatos de la documentación, conocer el reglamento, la documentación, retrabajo por no cumplir lo que se acordó y otros aspectos que significan desmotivación de los trabajadores.

Por otro lado, los resultados obtenidos hasta ahora, y en la medida que se han ejecutado nuevos proyectos, siguiendo las indicaciones del proceso de testing, han hecho valorar la inversión de trabajo realizado, ya que se entregan productos mejores y, gracias a la reutilización, en algunos casos con un menor esfuerzo.

Haciendo el balance entre ambas perspectivas se determina que el testing es valorado en cuanto a sus resultados casi totalmente, pero sus métodos no son del todo apreciados. Esta situación debería cambiar en función de que el proceso de testing sea totalmente integrado y deje de ser visto como nuevas exigencias organizacionales. Para esto, habrá que esperar a que se desarrollen más proyectos y a que se realicen las primeras evaluaciones del proceso para ver de qué manera se puede mejorar y tal vez desburocratizar las actividades.

Estas aprensiones en contra del testing son originadas principalmente por los programadores, no así por los analistas quienes son los dueños de la empresa y de alguna forma son capaces de ver a nivel global las consecuencias de estas implementaciones.

- **Demora en la construcción de productos software**

Se aborda de manera aparte a la sección anterior, pues independientemente de los proyectos se vieron pospuestos por la implementación de una etapa formal de testing en el ciclo de construcción. Este aumento en el ciclo de la producción solo es protagónico en las primeras actividades de testing, pues a la medida que se muestren

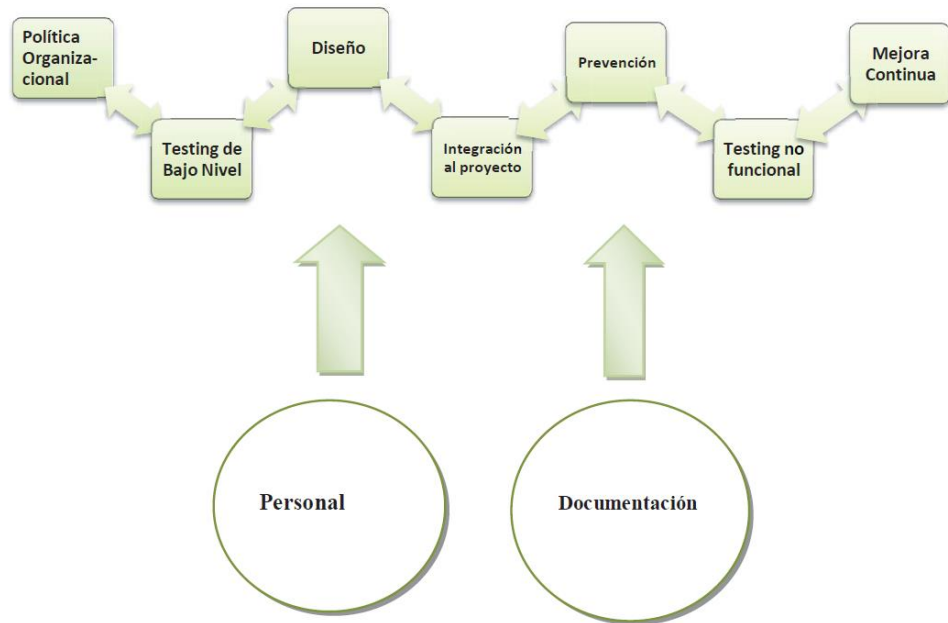
más temas previos al testing los tiempos se van haciendo de nuevo, como ya ha sucedido en otros proyectos.

Con el tiempo, los tiempos deberían recuperarse y la organización deberá decidir si estos recursos son destinados a desarrollo más rápido de sistemas, y por lo tanto abordar más proyectos o se destina a evaluar y seguir mejorando el proceso de testing con el fin de alcanzar productos de mayor calidad para sus clientes.

Si se contrastan los tiempos de desarrollo antes de la aplicación del modelo con los actuales, es evidente la diferencia, sin embargo, se deben valorar los beneficios obtenidos tanto a un nivel de procesos de construcción como a nivel de entrega final a los clientes. Por lo tanto, y en pleno consenso a los encargados de proyectos, se considera que el retraso (o inversión en tiempo) es necesaria y es rentable para los objetivos del negocio, ya que permite mejorar a largo plazo los proyectos y en el corto plazo la satisfacción de los clientes.

V. DISCUSIÓN DE RESULTADOS

5.1. DISCUSIÓN



Aaron J. Shenhar, Dov Dvir (2007) señala que los proyectos de software fracasan porque las empresas se concentran en la operatividad y que la gerencia suele utilizar las mismas métricas para medir proyectos diferentes. El problema es que el conocimiento respecto a la gestión de proyectos de software convencional no es adaptable a un entorno comercial en dinámico cambio.

Por otro lado autores como Dennis Cohen, Robert Granham (2000) señalan que los mejores proyectos no son al azar sino son bien planificados y ejecutados, y deben desarrollar casos de negocios que obligue a concentrarse en el resultado ansiado y ayude a clarificar por qué es una inversión excelente para la microempresa. Así mismo indican que acostumbradamente un gestor de proyectos exitoso tenía que completar su proyecto a tiempo y con el presupuesto asignado. En el presente, según estos autores, el que gestiona un proyecto también es responsable por la contribución cualitativa y/o cuantitativa que puede brindar su proyecto a su organización.

Las publicaciones recientes sobre gestión de proyectos tienen cada vez más acogida por

las organizaciones debido al éxito que están obteniendo muchas empresas nacionales e internacionales al aplicar el modelamiento del PMI, esto lo podemos notar en la bibliografía de D´ Labra (2007) en la cual dice que en la actualidad la utilización de las buenas prácticas (aquellas que han sido probadas por varias organizaciones y que han mostrado que su aplicación da un resultado positivo), es una tendencia que facilita la asimilación y rápida puesta en uso de procesos que ayuden a una microempresa, y en especial una de las prácticas más recomendadas están enfocadas al tema de Gestión de Proyectos, ya que como es de entenderse, una parte de las inversiones de las organizaciones se destina al desarrollo de proyectos, de los cuales se espera brinden un beneficio a la organización.

Según Nuñez A. Morales Ch, (2006) mencionan que los proyectos se definen como exitosos si a) Culminan en el tiempo estimado b) Con el presupuesto asignado, y c) Implementando requerimientos establecidos durante su concepción.

También exponen que para lograr que un proyecto tenga éxito no solo depende de las herramientas software y del modelamiento, sino que también se necesita de personas idóneas para la administración, y otros elementos. Sin embargo, para resolver carencias que se encuentran en la ejecución en áreas como el cronograma, alcance, calidad, integración y comunicación, sería necesario aplicar prácticas recomendadas por PMI.

Concepción S. (2007) en su tesis doctoral sostiene que las Gestiones Públicas, por intermedio de sus mandatarios, requieren a menudo de modelos de administración multidimensional que permita integrar todos los intereses de los diversos interesados, con el talento de la administración pública de crear valor para alcanzar necesidades. Los líderes públicos, deben de administrar, medir y evaluar esta complejidad, para así poder traducir este equilibrio en las metas estratégicas que se hayan proyectado.

Este elevado nivel de autonomía genera dificultades de colaboración eficiente en proyectos en común, haciendo uso de herramientas y técnicas de administración de proyectos que garanticen la cooperación entre las diferentes organizaciones para una gestión más eficiente de recursos públicos.

Autores concuerdan que, en el futuro, las microempresas tendrán éxito en proyectos que ejecuten, es por esto que Graham, y Englund (1999) mencionan que “la mayoría del crecimiento futuro en las compañías serán el resultado de proyectos de implementación con éxito que crean productos y servicios”. El gran auge que ha alcanzado esta disciplina, la mayoría de empresas a nivel internacional están perfilando esfuerzos a la gestión de proyectos. Estos proyectos alrededor del mundo se deben a la manera cómo funcionan, sus modelamientos y la perspectiva en los procesos y procedimientos fundamentales escritos por la gerencia general, respecto a esto Graham y Englund plantean:

“...los modelamientos de proyectos y la formación en Gestión de Proyectos hace posible que otros aspectos de los directivos establecen secuencias de evaluación de logro, enfoquen sus tareas en clientes, que cuantifiquen sus fortalezas de una empresa, mejoren la utilización de recursos, incorporando principios de calidad, implementen planificación estratégica, mejoren el desempeño y respuesta del mercado, innoven con nuevos productos, mejoren el conocimiento y formación de sus trabajadores.”

Por lo manifestado en los párrafos anteriores es vital que la alta dirección de las instituciones públicas y privadas ponga énfasis en la gestión de sus proyectos considerando que anualmente se invierte millones en la ejecución de proyectos no logrando necesariamente las metas planteadas.

Tal es el caso particular de la ARWEBSYSTEMS SAC, de acuerdo a los resultados obtenidos de la presente tesis de investigación se puede señalar dentro los hallazgos relevantes lo siguiente:

Según Grados Aguirre (2015) el incumplimiento del cronograma de actividades en la ejecución de proyectos de TI, en consecuencia, los proyectos no concluyen en el tiempo estimado inicialmente.

Necesidad de reformular el presupuesto del proyecto, debido a que, como existe ampliaciones o prorrogas de tiempo para el cumplimiento de actividades, éstas implican asignación de recursos, materiales, etc.

De acuerdo al organigrama institucional, la ARWEBSYSTEMS SAC es una organización estructuralmente funcional, donde las gestiones de los proyectos son lideradas por Gerentes o jefes de área en adición a sus funciones quien trabaja con un equipo humano que depende de otras áreas.

La capacitación respecto a la gestión de proyectos que han recibido algunos de los profesionales de la ARWEBSYSTEMS SAC ha sido netamente teórica, esto en razón que aún no se aplica dicho modelamiento para gestionar un proyecto.

Bajo grado de aceptación por parte de los profesionales de ARWEBSYSTEMS SAC respecto a la gestión de proyectos.

Según lo indicado en el Plan Estratégico de la ARWEBSYSTEMS SAC periodo 2007-2011 el presupuesto planificado para la ejecución de proyectos tecnológicos es de 1'042, 900,000 nuevos soles, considerando que para el presente año ya debería estar ejecutado el 70%.

VI. CONCLUSIONES

1. Se diseñaron más de 50 procesos flexibles, adaptables y escalables con sus respectivas actividades y herramientas de soporte a un nivel de gestión y técnico para gestionar eficientemente los proyectos microempresariales de calidad de software alineados a las metodologías de desarrollo de software más modernas en la actualidad. Los procesos se aplicaron en proyectos de ARWEBSYSTEMS SAC en sus clientes. En este aspecto, el 39.4%(ver Tabla 34) del total de encuestados manifestaron utilizar dichos servicios.
2. Se diseñaron más de 50 procesos, actividades y herramientas que permitan la factibilidad económica, operativa y técnica para la gestión eficiente de los proyectos de calidad de software de las MPS. El 100% (Ver Tabla 35) de las encuestas señaló que la ejecución exitosa de proyectos de calidad de software contribuye a mejorar la calidad de servicio.
3. Se diseñó una arquitectura tecnológicamente moderna que hace posible gestionar proyectos de la calidad de software complejo bajo plataformas SOA, Móviles, Cloud Computing y/o la integración entre éstas. La microempresa ARWEBSYSTEMS SAC utilizó directamente la arquitectura en sus clientes de más envergadura tecnológica como lo fue la Entidad Financiera.

VII. RECOMENDACIONES

A partir de la información obtenida de los elementos de estudio, me permito hacer las siguientes recomendaciones:

1. Aplicar en la microempresa ARWEBSYSTEMS las guías modelamientos PMBOK, PRINCE2, ISTQB Test Manager y SCRUM, para la administración de proyectos de calidad de software, por el resultado obtenido en la prueba de hipótesis del presente trabajo de investigación.
2. Difundir en ARWEBSYSTEMS los beneficios por la utilización de los Modelamientos de gestión de proyectos PMBOK, PRINCE2, ISTQB Test Manager y SCRUM.
3. Incorporar en la gestión de proyectos conceptos claves como gestión del tiempo, costo y calidad.
4. Crear grupos de trabajo que permitan analizar las deficiencias actuales en la gestión de proyectos, lo cual permita plantear y ejecutar alternativas de solución para revertirlas.
5. Establecer técnicas y herramientas para el trabajo que realice una gestión total inclusiva de los proyectos, con la simplicidad para la retroalimentación de lecciones que se aprendieron por parte de todos los profesionales que participan en algún proyecto en la microempresa ARWEBSYSTEMS SAC.
6. Involucrar en la ejecución de cualquier proyecto a los usuarios finales y demás interesados.
7. Brindar capacitación especializada en gestión de proyectos a los jefes de área y profesionales que son responsables de grupos de trabajo. Dicha capacitación debe ser aplicada y orientada a los proyectos de la microempresa.

8. Difundir a la ciudadanía los proyectos tecnológicos desarrollados por la microempresa ARWEBSYSTEMS SAC a fin de incrementar la cantidad de clientes que se beneficien con dichos servicios.

VIII. REFERENCIAS

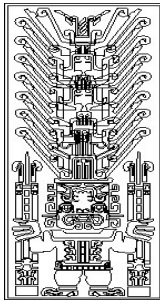
8.1 REFERENCIAS BIBLIOGRÁFICAS

- Aguirre, G. (2015). Desarrollo de un marco metodológico del proceso de verificación y validación de software para pequeñas y medianas empresas. *Revista de la Facultad de Ingeniería Industrial de la Universidad de Atacama*.
- Amable, M. ., (2015). Propuesta de Modelo Calidad de software en las mypes productoras de Software.
- Apesoft y Prompex Perú. (2015). *Situación de la Industria Nacional de Software en el Perú*. Lima: Instituto Cuanto.
- Arciniega Alemán, A. J., Flores Rosales, C. L., & García Paucar, L. H. (2014). *Integrando la oferta de las microempresas peruanas con la demanda de soluciones de software*. Tesis de Maestría, Lima.
- Bertolino, A. (2007). oftware Testing Research: Achievements, Challenges, Dreams, Future of Software Engineering. *IEEE Software*.
- Calá, C. D. (2018). *Buenas prácticas de inserción internacional de PyMEs: el caso de las empresas productoras de software y servicios informáticos de la ciudad de Mar del Plata*. Secretaría de Comercio de la Nación, Buenos Aires.
- Callejas Cuervo, M., Alarcón Aldana, A. C., & Alvarez Carreno, A. M. (2017). Modelos de calidad de software, un estado del arte. *Scielo*.
- Coque Villegas, S., Jurado Vite, V., Avendaño Sudario, A., & Pizarro, G. (2016). Análisis de experiencias de mejora de procesos de desarrollo de software en pymes. *Ciencia Unemi*.
- Coque-Villegas, S. J.-V.-S. (2017). Análisis de experiencias de mejora de procesos de desarrollo de software en PYMEs. *Ciencia Unemi*.
- Dávila, A. B. (2012). The peruvian component of Competisoft project: Lesson learned from academic perspective.
- Dávila, A., García, C., & Córdor, S. (2017). Análisis exploratorio en la adopción de prácticas de pruebas de software de la ISO/IEC 29119-2 en organizaciones de Lima, Perú. *Scielo*.
- Escobar Sánchez, M. E. (2014). *Modelo de pruebas de software para el segundo nivel de TMMI en la organización Sirecom, Quito Ecuador*. Tesis de Maestría, Quito.
- Galvis-Lista, E. &.-T. (2013). A critical review of knowledge management in software process reference models. *Journal of Information Systems and Technology Management*.

- García Paucar, L., Laporte, C. Y., Arteaga, J., & Bruggmann, M. (2015). *Implementation and Certification of ISO/IEC 29110 in an IT Startup in Peru. Software Quality Professional*. Lima: American Society for Quality (ASQ).
- González Yepes, D. J., Calvache Pardo, J. C., & Gómez Gómez, S. O. (2015). Revisión sistemática acerca de la implementación de metodologías ágiles y otros modelos en micro, pequeñas y medianas empresas de software. *Revista Tecnológica ESPOL – RTE*.
- Grados Aguirre, L. (2015). *Marco metodológico del proceso de verificación y validación de software para pequeñas y medianas empresas*. Lima.
- Larrucea, X., O'Connor, R. V., Colomo-Palacios, R., & Laporte, C. Y. (2016). Software process improvement in very small organizations. *IEEE Software*.
- Loya Páez, A. (2018). Industria de software en Perú. Software Gurú. . *Recuperado de <https://sg.com.mx/revista/53/industria-software>*.
- Quintana, E. D. (2017). *La arquitectura orientada a servicios aplicada a micro, pequeñas y medianas empresas*. Buenos Aires.
- Rojas Montes, M., Pino-Correa, F., & Martínez, J. (2015). Proceso de pruebas para pequeñas organizaciones desarrolladoras de software. *Revista Facultad de Ingeniería Universidad del Cauca*.
- Salazar Montes, D. P., Cárdenas Gaviria, L. A., & Franco Bedoya, O. H. (2018). Gestión del conocimiento en procesos de desarrollo de software un marco de trabajo para apoyar a las MiPyMEs. *Scientia et Technica*.
- Sigcha Moya, W. L., & Utreras Zapata, E. V. (2018). *Fábrica de software para pequeñas y mediana empresas*. Tesis de Maestría, Quito.
- SOFTEX. (2012). MPS.BR - Mejora de Proceso del Software Brasileño. Guía General MPS.
- SUÁREZ Urresti, D. R. (2019). Las PyME de desarrollo de software. Modelos de mejora de sus procesos en Latinoamérica. *Espacios*.
- Toro, A., & Peláez, L. E. (2017). Validación de un modelo para el aseguramiento de la calidad del software en MIPYMES que desarrollan software en el Eje Cafetero. *Entre Ciencia e Ingeniería*.
- van Zyl, J. (2010). *Software testing en una pequeña compañía: un caso de estudio*. Pretoria. Obtenido de <https://www.iitpsa.org.za/wp-content/uploads/2012/06/Testing-in-a-Small-Company1.pdf>

ANEXOS

ANEXO N° 01: ENCUESTAS



INSTRUMENTO Nro 01

UNIVERSIDAD NACIONAL FEDERICO VILLAREAL

ESCUELA UNIVERSITARIA DE POST GRADO

ENCUESTA DE OPINIÓN: (Dirigido a los profesionales de ARWEBSYSTEMS SAC que participan en algún proyecto de calidad de software)

Área:

Cargo:

1. ¿Conoce Ud. el modelamiento de Gestión Eficiente de Proyectos de Calidad de Software de la empresa ARWEBSYSTEMS?

- a) Totalmente
- b) En gran medida
- c) Poco o en forma limitada
- d) No
- e) No sabe/ no opina

2. ¿Ha recibido alguna capacitación referente a el modelamiento de Gestión Eficiente de Proyectos de Calidad de Software de la empresa ARWEBSYSTEMS?

- a) Sí
- b) No

De ser la respuesta SI, indicar que tipo de capacitación:

- i) Curso o seminario ii) Diplomado iii) Maestría iv) Otros

3. ¿Tiene Ud. Alguna Certificación en gestión de proyectos?

- a) Si
- b) No

4. ¿Conoce Ud. algun modelamiento de gestión de proyectos distinta a el modelamientos basadas en PMBOK, PRINCE2, ISTQB Test Manager o SCRUM?

- a) Sí
- b) No

De ser la respuesta SI, favor de indicar cuál o cuáles:

-
5. ¿Cuál es el grado de aplicación de las modelamientos basadas en PMBOK, PRINCE2, ISTQB Test Manager o SCRUM que Ud. ha realizado, en los proyectos que ha participado?
 - a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina

 6. Desde su percepción, en los proyectos que Ud. ha participado cuál es el grado de aceptación respecto a la gestión de proyectos en sus diversos componentes tales como alcance, tiempo, costo, calidad, RRHH, etc?
 - a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/opina

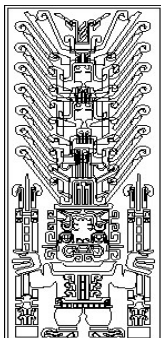
Favor de detallar el (los) problema (s) identificado (s):

-
7. En los proyectos que Ud. ha participado se ha cumplido los plazos establecido en el cronograma del proyecto?
 - a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina

 8. En los proyectos que Ud. ha participado, se ha cumplido con el presupuesto formulado y asignado inicialmente para la ejecución del proyecto?
 - a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina

 9. Desde su punto de vista, en los proyectos que Ud ha participado considera que se ha brindado la debida importancia al seguimiento y control que se debe realizar en un proyecto?
 - a) Totalmente

- b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina
10. Desde su experiencia profesional, cree Ud que ha sido eficaz el procedimiento utilizado para llevar acabo el seguimiento y control del proyecto?
- a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/opina
11. ¿En los proyectos que Ud. ha participado, se ha establecido periodos de tiempo fijo (día, semana, mes, etc) para la entrega de informes y/o reportes para el seguimiento y control del proyecto?
- a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/opina
12. ¿Cuál es el grado de aceptación respecto al formato y contenido de los informes y/o reportes que se manejan en el seguimiento y control del proyecto?
- a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/no opina
13. ¿Estaría Ud de acuerdo con la aplicación de las modelamientos basadas en PMBOK, PRINCE2, ISTQB Test Manager o SCRUM en la gestión de los proyectos de calidad de software de ARWEBSYSTEMS SAC?
- a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina
14. ¿Cree Ud que la aplicación de las modelamientos basadas en PMBOK, PRINCE2, ISTQB Test Manager o SCRUM en la gestión de los proyectos de calidad de software de ARWEBSYSTEMS SAC traerá algún beneficio a la empresa?
- a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/opina



INSTRUMENTO N 02
UNIVERSIDAD NACIONAL FEDERICO VILLAREAL
ESCUELA UNIVERSITARIA DE POST GRADO

ENCUESTA DE OPINIÓN: (Dirigido a los usuarios del servicio de calidad de software que brinda la empresa ARWEBSYSTEMS SAC)

Estimado Usuario: Con la finalidad de cubrir la mayor parte de sus expectativas respecto a los servicios que brinda la empresa ARWEBSYSTEMS SAC, agradeceremos mucho unos minutos de su tiempo para que conteste la siguiente encuesta.

Instrucciones: Por favor marque con una X la respuesta que usted considere pertinente.

1. Conoce Ud. los servicios que brinda la ARWEBSYSTEMS SAC?

<input type="checkbox"/> Totalmente	<input type="checkbox"/> En gran medida	<input type="checkbox"/> En forma Limitada	<input type="checkbox"/> No	<input type="checkbox"/> No sabe/ No opina
-------------------------------------	---	--	-----------------------------	---

2. ¿Considera Ud. adecuados los mecanismos de acceso que tiene la empresa ARWEBSYSTEMS SAC (Presencial, Vía Telefónica, Internet) para atender los servicios de calidad de software que brinda?

<input type="checkbox"/> Totalmente	<input type="checkbox"/> En gran medida	<input type="checkbox"/> En forma Limitada	<input type="checkbox"/> No	<input type="checkbox"/> No sabe/ No opina
-------------------------------------	---	--	-----------------------------	---

3. ¿Cuál es el grado de aceptación respecto al tiempo que emplea la ARWEBSYSTEMS SAC para la prestación del servicio que solicitó?

<input type="checkbox"/> Totalmente	<input type="checkbox"/> En gran medida	<input type="checkbox"/> En forma Limitada	<input type="checkbox"/> No	<input type="checkbox"/> No sabe/ No opina
-------------------------------------	---	--	-----------------------------	---

4. ¿Considera adecuada y oportuna la orientación que recibe Ud. antes, durante y después de la atención del servicio de calidad de software solicitado?

<input type="checkbox"/> Totalmente	<input type="checkbox"/> En gran medida	<input type="checkbox"/> En forma Limitada	<input type="checkbox"/> No	<input type="checkbox"/> No sabe/ No opina
-------------------------------------	---	--	-----------------------------	---

5. ¿Considera eficaces las acciones, observaciones y/o recomendaciones realizadas por la empresa ARWEBSYSTEMS SAC para atender el servicio de calidad de software que solicitó?

<input type="checkbox"/> Totalmente	<input type="checkbox"/> En gran medida	<input type="checkbox"/> En forma Limitada	<input type="checkbox"/> No	<input type="checkbox"/> No sabe/ No opina
-------------------------------------	---	--	-----------------------------	---

6. ¿En términos generales, considera adecuada Ud. la atención que brinda la empresa ARWEBSYSTEMS SAC?

<input type="checkbox"/> Totalmente	<input type="checkbox"/> En gran medida	<input type="checkbox"/> En forma Limitada	<input type="checkbox"/> No	<input type="checkbox"/> No sabe/ No opina
-------------------------------------	---	--	-----------------------------	---

7. ¿Considera adecuado el equipamiento tecnológico que presenta la empresa ARWEBSYSTEMS SAC (computadoras, impresoras, etc.) para atender al usuario?

<input type="checkbox"/> Totalmente	<input type="checkbox"/> En gran medida	<input type="checkbox"/> En forma Limitada	<input type="checkbox"/> No	<input type="checkbox"/> No sabe/ No opina
-------------------------------------	---	--	-----------------------------	---

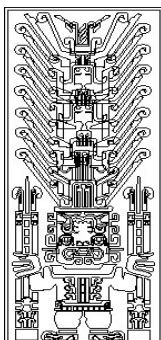
8. ¿Considera adecuadas las herramientas tecnológicas (página web, sistemas informáticos, etc.) que utiliza la empresa ARWEBSYSTEMS SAC para atender al usuario?

<input type="checkbox"/> Totalmente	<input type="checkbox"/> En gran medida	<input type="checkbox"/> En forma Limitada	<input type="checkbox"/> No	<input type="checkbox"/> No sabe/ No opina
-------------------------------------	---	--	-----------------------------	---

9. Desde su punto de vista, cree Ud. que la ejecución exitosa de proyectos de calidad de software contribuiría favorablemente en mejorar la calidad del servicio de calidad de software que brinda la empresa ARWEBSYSTEMS SAC?

<input type="checkbox"/> Totalmente	<input type="checkbox"/> En gran medida	<input type="checkbox"/> En forma Limitada	<input type="checkbox"/> No	<input type="checkbox"/> No sabe/ No opina
-------------------------------------	---	--	-----------------------------	---

Observaciones y/o sugerencias



INSTRUMENTO N 03
UNIVERSIDAD NACIONAL FEDERICO VILLAREAL
ESCUELA UNIVERSITARIA DE POST GRADO

ENCUESTA DE OPINIÓN: (Dirigido a Jefes de Informática de empresas clientes de ARWEBSYSTEMS SAC)

Nombre de la Empresa:

1. ¿Qué nivel de éxito se ha logrado en su empresa con la aplicación del modelamiento de Gestión Eficiente de Proyectos de Calidad de Software de la empresa ARWEBSYSTEMS basada en PMBOK, PRINCE2, ISTQB Test Manager y SCRUM en la gestión de los proyectos de calidad de software?
 - a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina

2. ¿Cree Ud. que la aplicación del modelamiento de Gestión Eficiente de Proyectos de Calidad de Software de la empresa ARWEBSYSTEMS basada en PMBOK, PRINCE2, ISTQB Test Manager y SCRUM ha contribuido favorablemente en la gestión de los proyectos de calidad de software en su empresa?
 - a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina

3. ¿Cree Ud. que la aplicación del modelamiento de Gestión Eficiente de Proyectos de Calidad de Software de la empresa ARWEBSYSTEMS basada en PMBOK, PRINCE2, ISTQB Test Manager y SCRUM ha contribuido en mejorar la planificación de los proyectos de calidad de software en su empresa?
 - a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina

4. ¿Cree Ud. que la aplicación del modelamiento de Gestión Eficiente de Proyectos de Calidad de Software de la empresa ARWEBSYSTEMS basada en PMBOK, PRINCE2, ISTQB Test Manager y SCRUM ha contribuido en mejorar el control y seguimiento de los proyectos de calidad de software en su empresa?
 - a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina

5. ¿Cree Ud. que la aplicación del modelamiento de Gestión Eficiente de Proyectos de Calidad de Software de la empresa ARWEBSYSTEMS basada en PMBOK, PRINCE2, ISTQB Test Manager y SCRUM ha contribuido en disminuir la cantidad de reprogramaciones de los plazos establecidos en el cronograma de los proyectos de calidad de software en su empresa?
 - a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina

6. ¿Cree Ud. que la aplicación del modelamiento de Gestión Eficiente de Proyectos de Calidad de Software de la empresa ARWEBSYSTEMS basada en PMBOK, PRINCE2, ISTQB Test Manager y SCRUM ha contribuido en disminuir la cantidad de reformulaciones del presupuesto asignado inicialmente a los proyectos de calidad de software en su empresa?
 - a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina

7. ¿Cree Ud. que la aplicación del modelamiento de Gestión Eficiente de Proyectos de Calidad de Software de la empresa ARWEBSYSTEMS basada en PMBOK, PRINCE2, ISTQB Test Manager y SCRUM le proporciona técnicas y herramientas adecuadas para la gestión de los diversos componentes del proyecto de calidad de software?
 - a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina

8. Cree Ud. que la aplicación del modelamiento de Gestión Eficiente de Proyectos de Calidad de Software de la empresa ARWEBSYSTEMS basada en PMBOK, PRINCE2, ISTQB Test Manager y SCRUM permite que todo el personal involucrado en el proyecto trabaje bajo procedimientos definidos y estandarizados?
- a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina
9. En términos generales, cree Ud. que la aplicación del modelamiento de Gestión Eficiente de Proyectos de Calidad de Software de la empresa ARWEBSYSTEMS basada en PMBOK, PRINCE2, ISTQB Test Manager y SCRUM contribuye a lograr el éxito del proyecto?
- a) Totalmente
 - b) En gran medida
 - c) Poco o en forma limitada
 - d) No
 - e) No sabe/ no opina

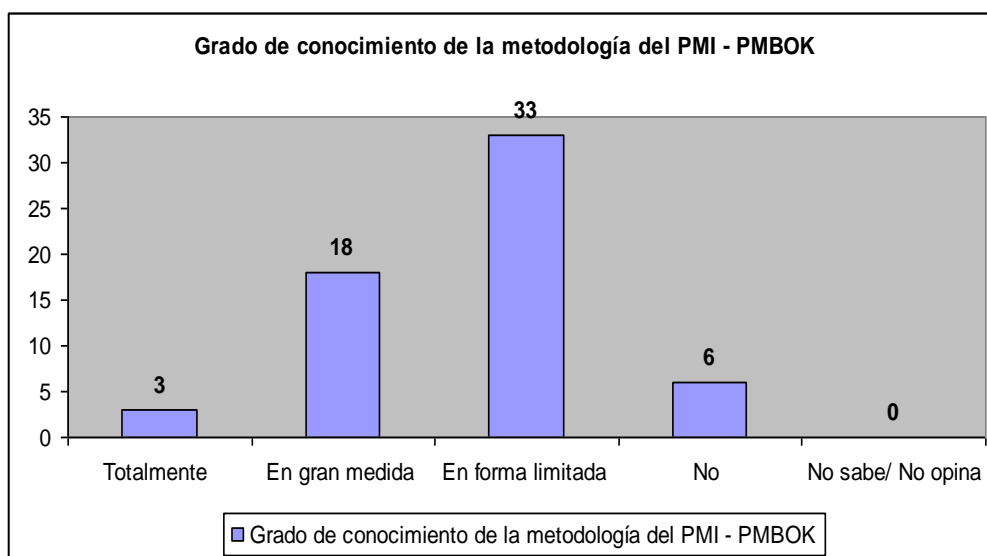
ANEXO N° 02: ANÁLISIS DESCRIPTIVO

Resultado de la encuesta

Jefes y Profesionales

1. ¿Conoce Ud. el modelamiento del PMI - PMBOK?

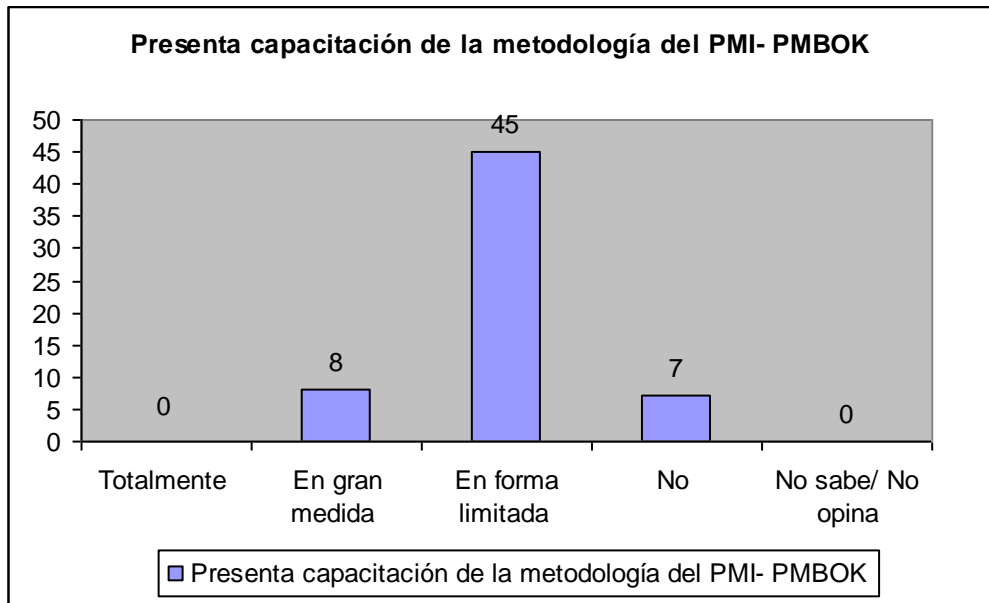
Indicador	Totalmente	En gran medida	En forma limitada	No	No sabe/ No opina
Grado de conocimiento de la metodología del PMI - PMBOK	3	18	33	6	0



El cuadro muestra que de una muestra de 60 jefes y profesionales: 33 señalan conocer en forma limitada el modelamiento del PMI – PMBOK, 18 lo conocen en gran medida, 3 conocen el modelamiento totalmente; mientras que 6 profesionales no conocen dicha modelamiento.

2. ¿Ha recibido alguna capacitación referente a el modelamiento del PMI- PMBOK?

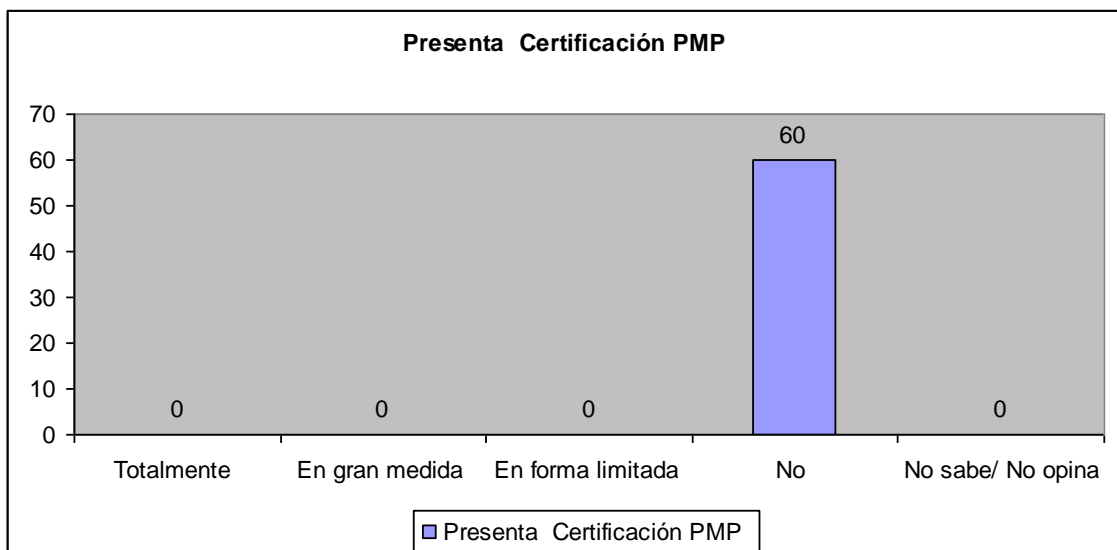
Indicador	Totalmente	En gran medida	En forma limitada	No	No sabe/ No opina
Presenta capacitación de la metodología del PMI- PMBOK	0	8	45	0	0



El gráfico indica que de una muestra de 60 jefes y profesionales: 45 señalan haber sido capacitados en forma limitada, 8 recibieron capacitación en gran medida, 7 no recibieron ninguna capacitación.

3. ¿Tiene Ud. Certificación PMP (Project Management Professional)?

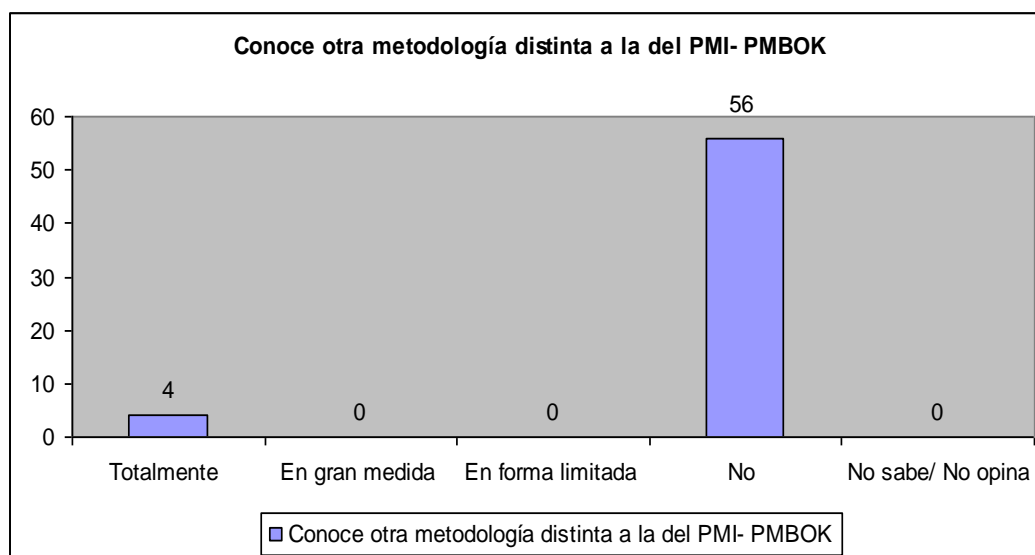
Indicador	Totalmente	En gran medida	En forma limitada	No	No sabe/ No opina
Presenta Certificación PMP	0	0	0	60	0



El gráfico indica que de una muestra de 60 jefes y profesionales: No cuentan con certificación PMP.

4. ¿Conoce Ud. algún modelamiento de gestión de proyectos distinta a el modelamiento del PMI- PMBOK?

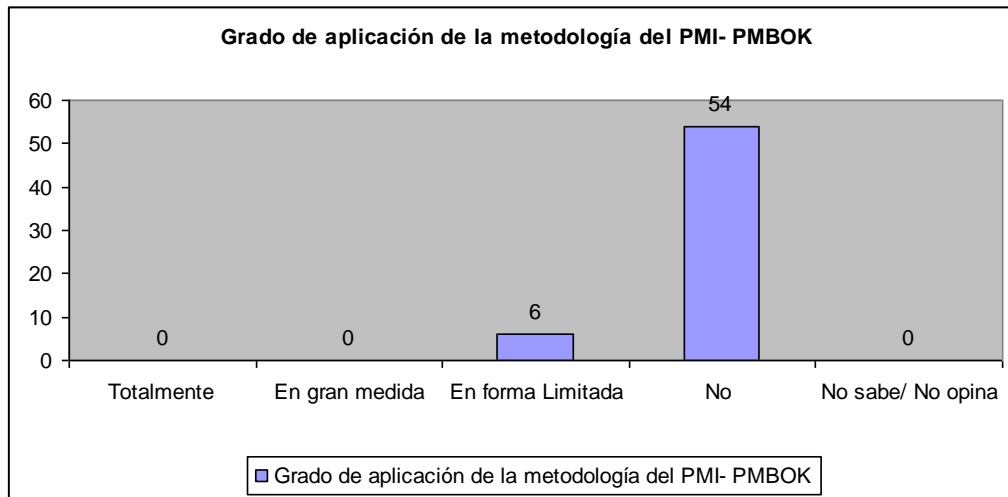
Indicador	Totalmente	En gran medida	En forma limitada	No	No sabe/ No opina
Conoce otra metodología distinta a la del PMI- PMBOK	4	0	0	56	0



El gráfico indica que de una muestra de 60 jefes y profesionales: 56 no conocen otra modelamiento distinta ala del PMI-PMBOK y 4 tienen conocimiento de otra modelamiento.

5. ¿Cuál es el grado de aplicación del modelamiento del PMI- PMBOK que Ud. ha realizado, en los proyectos que ha participado?

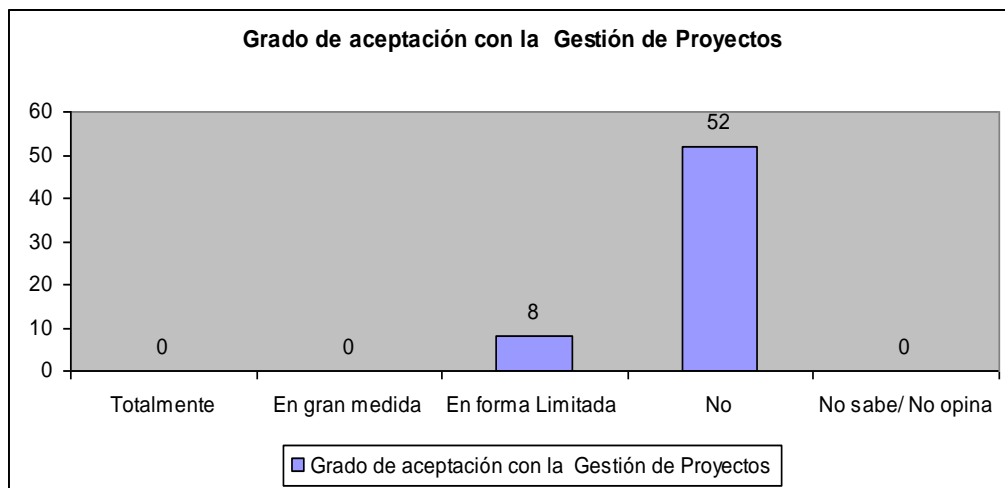
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Grado de aplicación de la metodología del PMI- PMBOK	0	0	6	54	0



El gráfico indica que de una muestra de 60 jefes y profesionales: 54 no aplicaron el modelamiento en la gestión de proyecto y 6 profesionales aplicaron en forma limitada.

6. Desde su percepción, en los proyectos que Ud. ha participado cuál es el grado de aceptación respecto a la gestión de proyectos en sus diversos componentes tales como alcance, tiempo, costo, calidad, RRHH, etc?

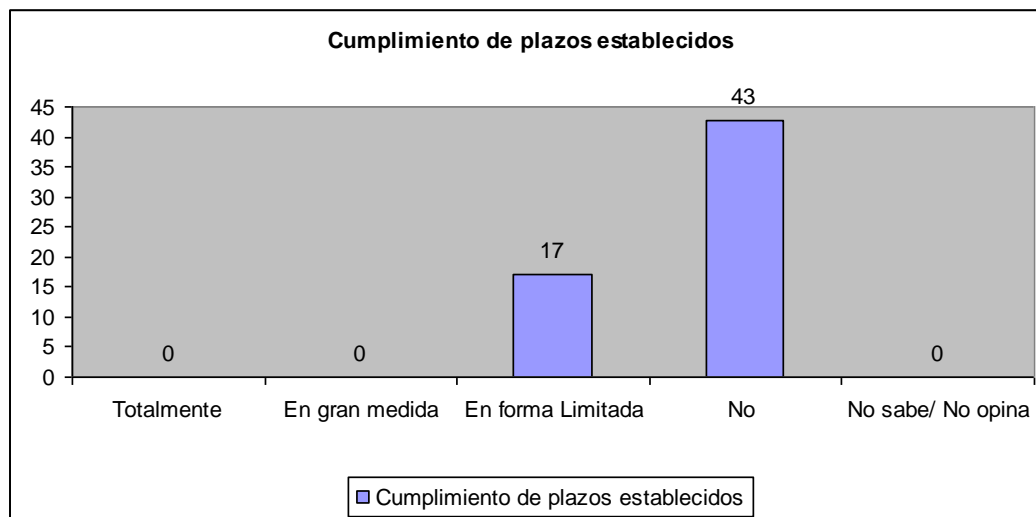
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Grado de aceptación con la Gestión de Proyectos	0	0	8	52	0



El gráfico indica que de una muestra de 60 jefes y profesionales: 52 califican con un NO los indicadores de aceptación con la Gestión de proyectos y 8 profesionales calificaron en forma limitada la aceptación.

7. En los proyectos que Ud. ha participado se ha cumplido los plazos establecido en el cronograma del proyecto?

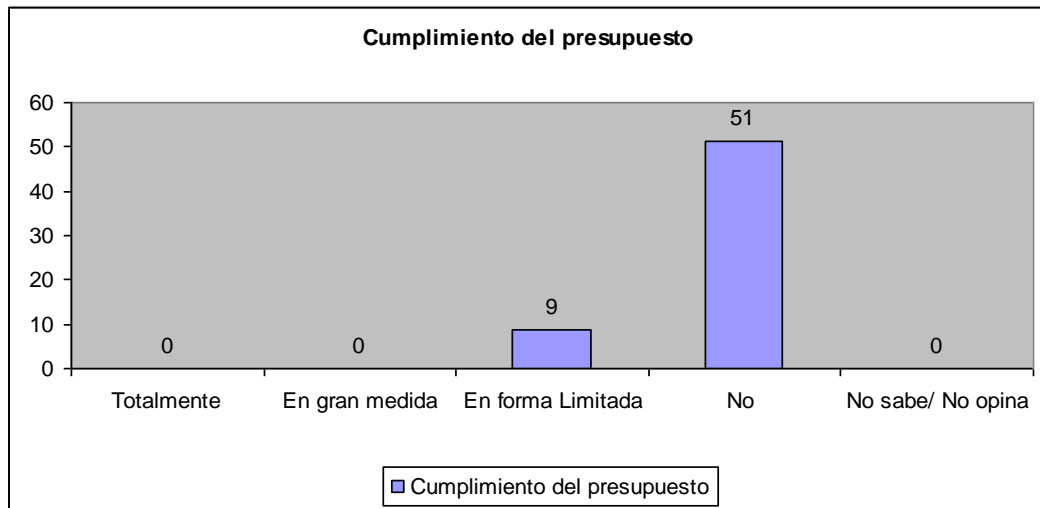
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Cumplimiento de plazos establecidos	0	0	17	43	0



El gráfico indica que de una muestra de 60 jefes y profesionales: 43 califican con un NO el cumplimiento de plazos establecidos para la ejecución de las actividades del proyecto y 17 profesionales calificaron en formal limitada el cumplimiento de los plazos.

8. En los proyectos que Ud. ha participado, se ha cumplido con el presupuesto formulado y asignado inicialmente para la ejecución del proyecto?

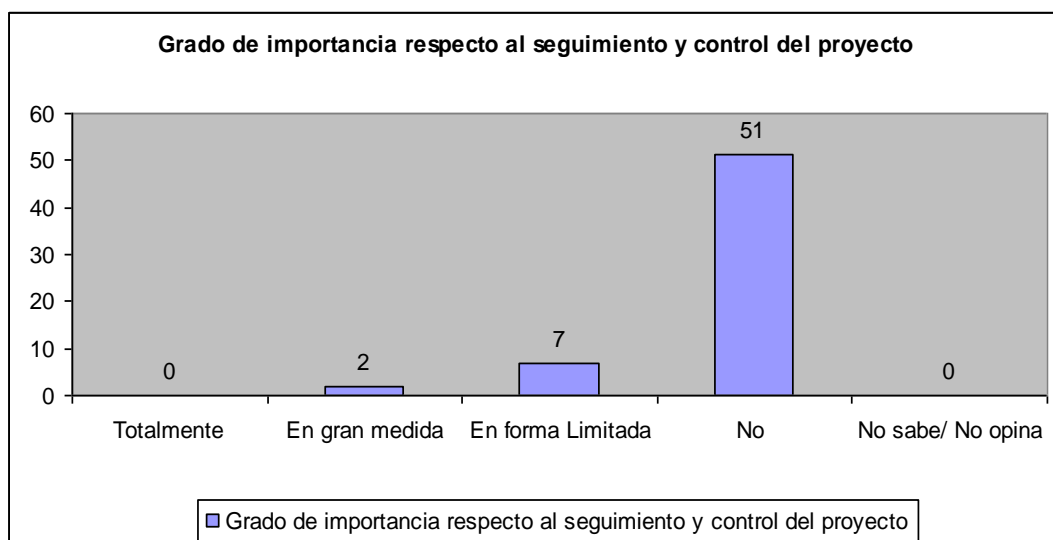
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Cumplimiento del presupuesto	0	0	9	51	0



El gráfico indica que de una muestra de 60 jefes y profesionales: 51 califican con un NO el cumplimiento del presupuesto establecidos para la ejecución del proyecto y 9 profesionales calificaron en formal limitada el cumplimiento del presupuesto.

9. Desde su punto de vista, en los proyectos que Ud ha participado considera que se ha brindado la debida importancia al seguimiento y control que se debe realizar en un proyecto?

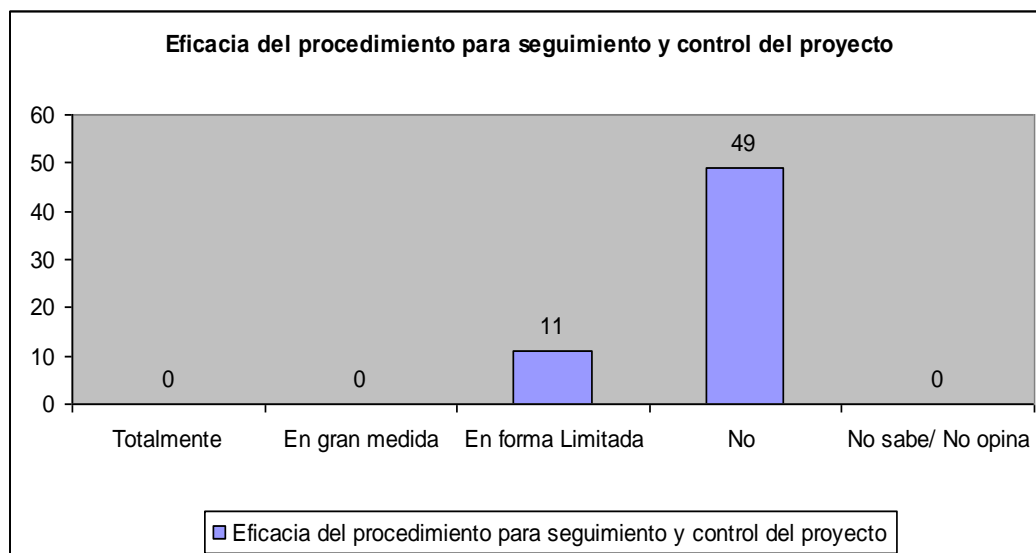
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Grado de importancia respecto al seguimiento y control del proyecto	0	2	7	51	0



El gráfico indica que de una muestra de 60 jefes y profesionales: 51 califican con un NO el grado de importancia respecto al seguimiento y control del proyecto, 7 calificaron en forma limitada y 2 calificaron en gran medida el grado de importancia.

10. Desde su experiencia profesional, cree Ud que ha sido eficaz el procedimiento utilizado para llevar acabo el seguimiento y control del proyecto?

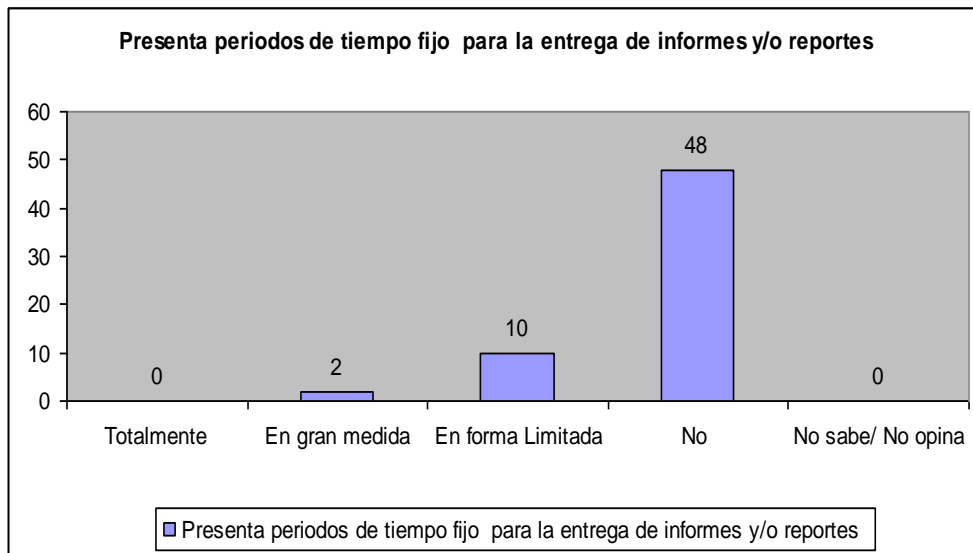
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Eficacia del procedimiento para seguimiento y control del proyecto	0	0	11	49	0



El gráfico indica que de una muestra de 60 jefes y profesionales: 49 califican con un NO la eficacia del procedimiento para seguimiento y control del proyecto, 11 calificaron en forma limitada la eficacia del procedimiento.

11. ¿En los proyectos que Ud. ha participado, se ha establecido periodos de tiempo fijo (día, semana, mes, etc) para la entrega de informes y/o reportes para el seguimiento y control del proyecto?

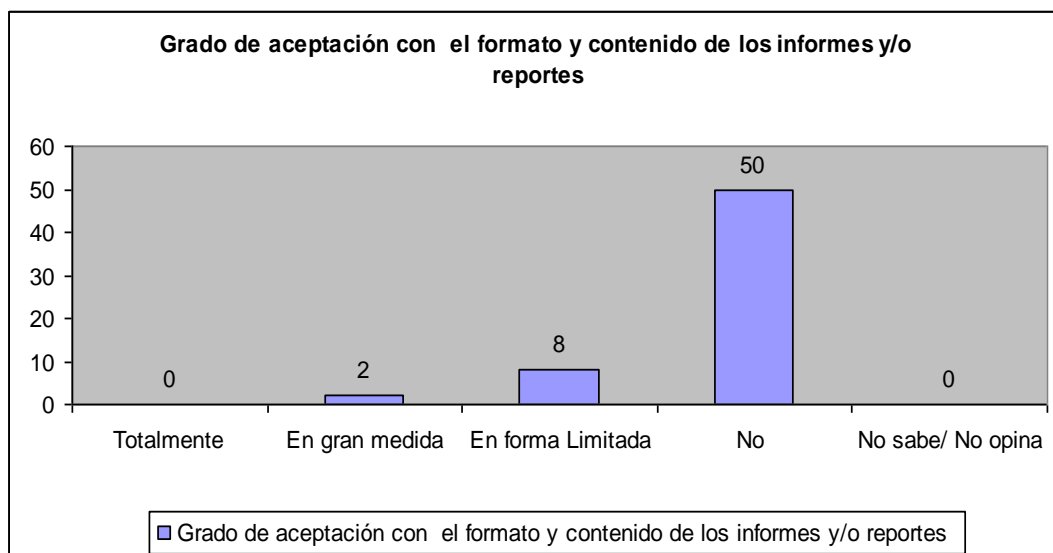
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Presenta periodos de tiempo fijo para la entrega de informes y/o reportes	0	2	10	48	0



El gráfico indica que de una muestra de 60 jefes y profesionales: 48 califican con un NO la presentación de periodos de tiempo fijo para la entrega de informes y/o reportes, 10 califican de forma limitada y 2 en gran medida la presentación de periodos.

12. Cual es el grado de aceptación respecto al formato y contenido de los informes y/o reportes que se manejan en el seguimiento y control del proyecto?

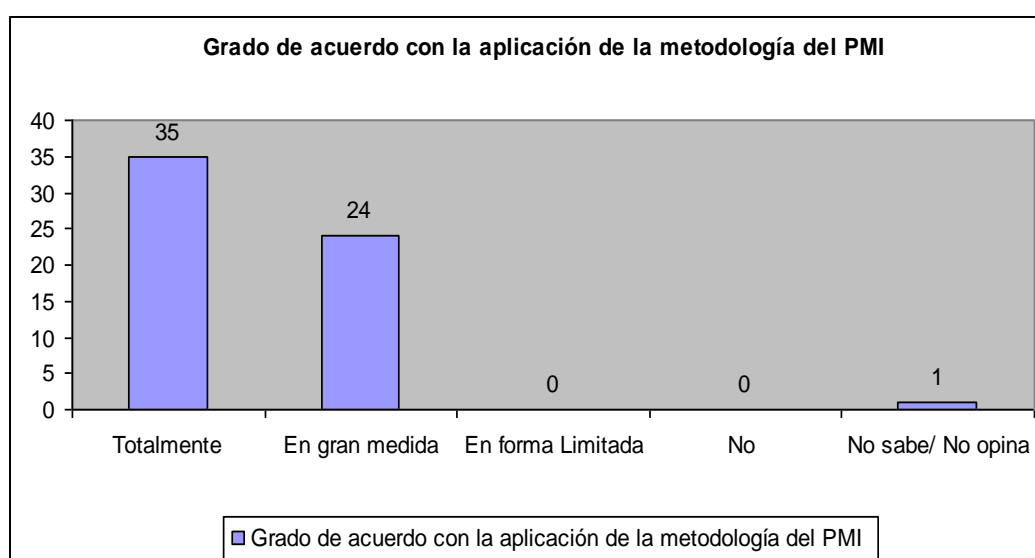
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Grado de aceptación con el formato y contenido de los informes y/o reportes	0	2	8	50	0



El gráfico indica que de una muestra de 60 jefes y profesionales: 50 califican con un No el grado de aceptación con el formato y contenido de los informes y/o reportes, 8 califican de forma limitada y 2 califican en gran medida el grado de aceptación.

13. ¿Estaría Ud de acuerdo con la aplicación del modelamiento del PMI en la gestión de los proyectos de tecnología de la información de la ARWEBSYSTEMS SAC?

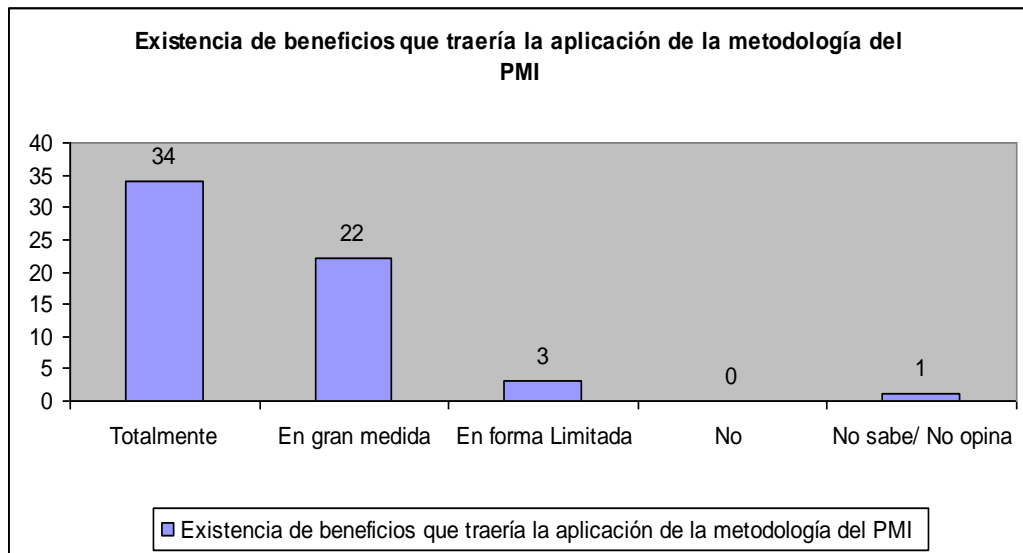
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Grado de acuerdo con la aplicación de la metodología del PMI	35	24	0	0	1



El gráfico indica que de una muestra de 60 jefes y profesionales: 35 califican que están totalmente de acuerdo con la aplicación del modelamiento del PMI, 24 en gran medida y 1 persona no sabe/ no opina.

14. ¿Cree Ud que la aplicación del modelamiento del PMI en la gestión de los proyectos de tecnología de la información de la ARWEBSYSTEMS SAC traerá algún beneficio a la institución?

Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Existencia de beneficios que traería la aplicación de la metodología del PMI	34	22	3	0	1



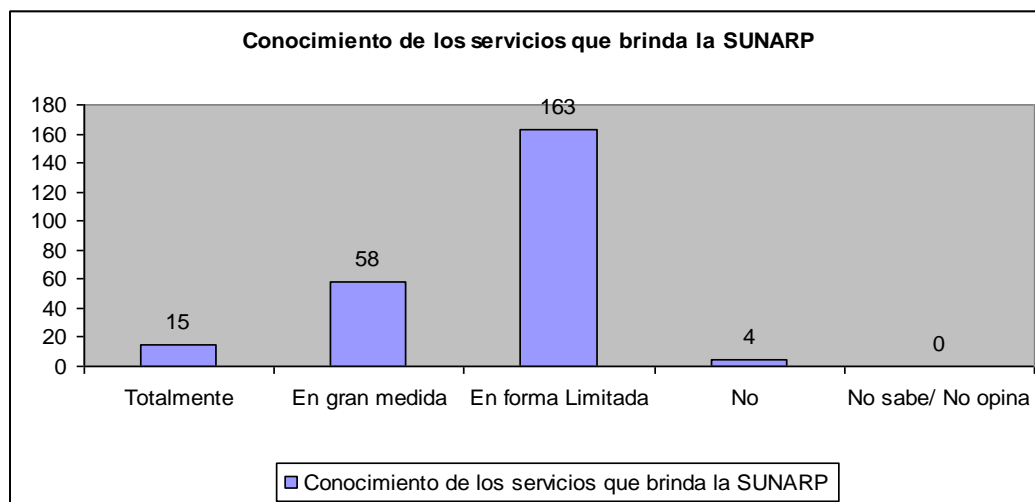
El gráfico indica que de una muestra de 60 jefes y profesionales: 34 indican que la aplicación del modelamiento del PMI traería beneficios totalmente, 22 en gran medida, 3 en forma limitada la existencia de beneficios y 1 persona no sabe/ no opina.

Encuesta 2:

Usuarios

1. Conoce Ud. los servicios que brinda la ARWEBSYSTEMS SAC?

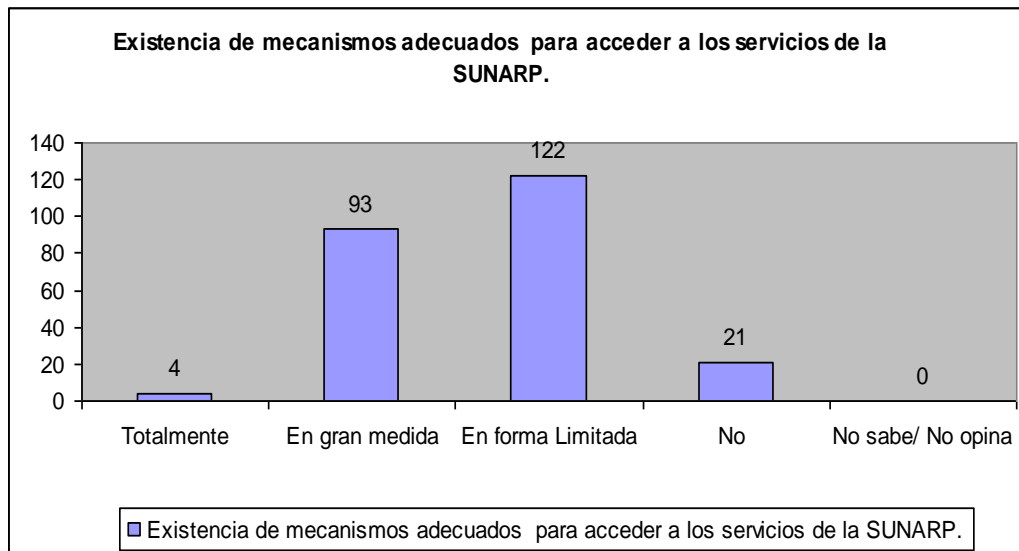
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Conocimiento de los servicios que brinda la SUNARP	15	58	163	4	0



El gráfico indica que de una muestra de 240 usuarios: 163 tienen conocimientos de los servicios que brinda la ARWEBSYSTEMS SAC, 58 en gran medida, 15 conocen totalmente y 4 usuarios no conocen los servicios.

2. Considera Ud. adecuados los mecanismos de acceso que tiene la ARWEBSYSTEMS SAC (Presencial, Vía Telefónica, Internet) para atender los servicios registrales que brinda?

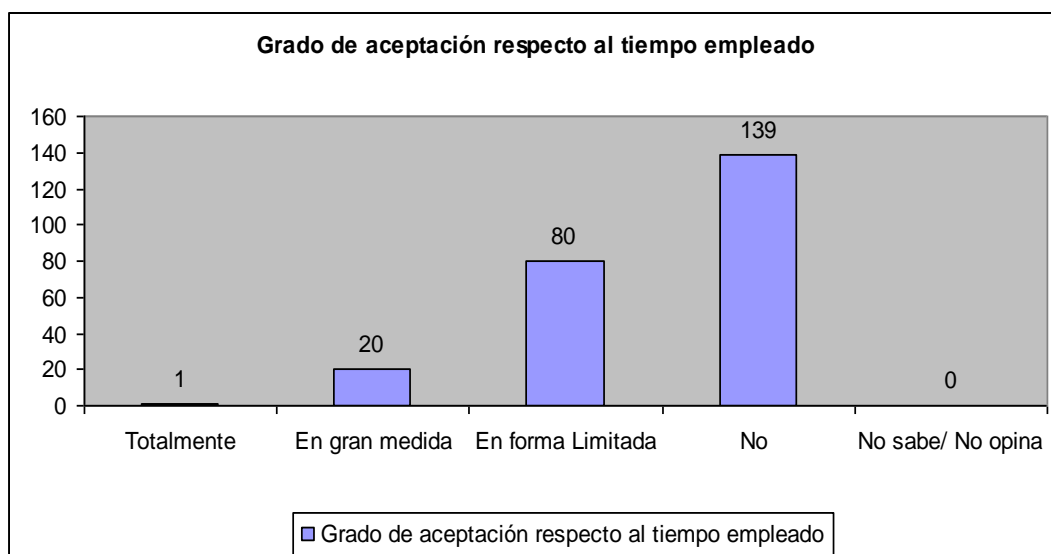
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Existencia de mecanismos adecuados para acceder a los servicios de la SUNARP.	4	93	122	21	0



El gráfico indica que de una muestra de 240 usuarios: 122 consideran en forma limitada los mecanismos de acceso que tiene la ARWEBSYSTEMS SAC para atender los servicios, 93 en gran medida, 21 no consideran adecuados los mecanismos de acceso y 4 usuarios consideran totalmente adecuados los mecanismos.

3.Cuál es el grado de aceptación respecto al tiempo que emplea la ARWEBSYSTEMS SAC para la prestación del servicio que solicitó?

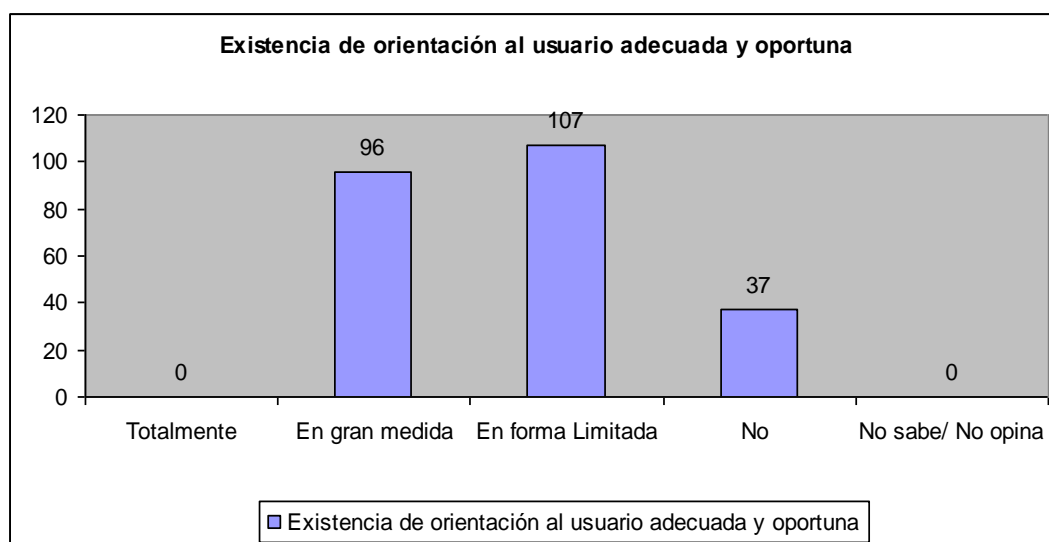
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Grado de aceptación respecto al tiempo empleado	1	20	80	139	0



El gráfico indica que de una muestra de 240 usuarios: 139 no acepta el tiempo que emplea la ARWEBSYSTEMS SAC para la prestación del servicio que solicita, 80 consideran de forma limitada, 20 consideran que en gran medida aceptan el tiempo empleado por la ARWEBSYSTEMS SAC.

4. Considera adecuada y oportuna la orientación que recibe Ud. antes, durante y después de la atención del servicio solicitado?

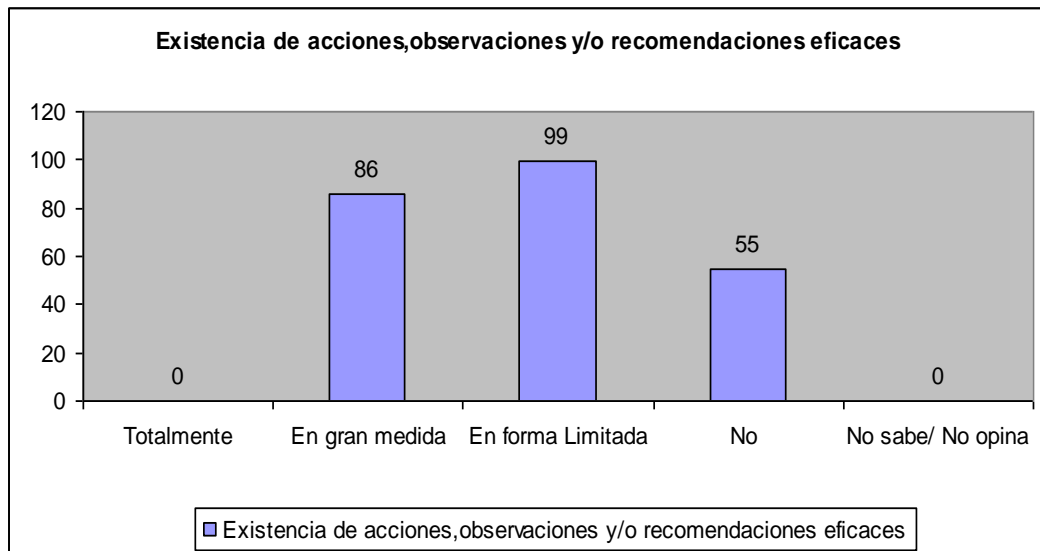
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Existencia de orientación al usuario adecuada y oportuna	0	96	107	37	0



El gráfico indica que de una muestra de 240 usuarios: 107 consideran que la orientación se da de forma limitada, 96 en gran medida y 37 consideran que no existe ninguna orientación para el servicio solicitado.

5. Considera eficaces las acciones, observaciones y/o recomendaciones realizadas por la ARWEBSYSTEMS SAC para atender el servicio que solicitó?

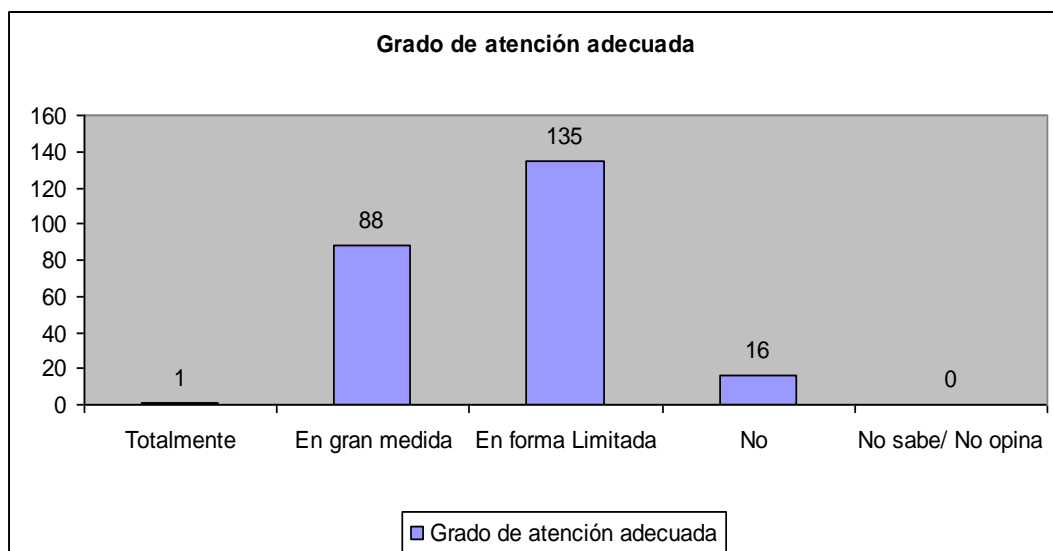
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Existencia de acciones, observaciones y/o recomendaciones eficaces	0	86	99	55	0



El gráfico indica que de una muestra de 240 usuarios: 99 consideran que es limitada la eficacia referente a las acciones, observaciones y recomendaciones realizadas por la ARWEBSYSTEMS SAC, 86 en gran medida y 55 usuarios consideran que no son eficaces las acciones, observaciones y recomendaciones realizadas por la ARWEBSYSTEMS SAC.

6. En términos generales, considera adecuada Ud la atención que brinda la ARWEBSYSTEMS SAC?

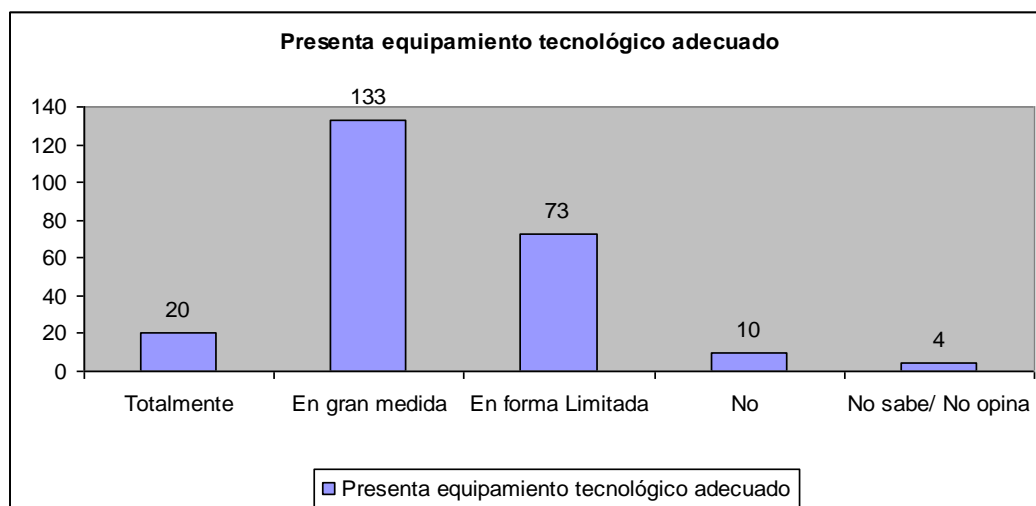
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Grado de atención adecuada	1	88	135	16	0



El gráfico indica que de una muestra de 240 usuarios: 135 consideran que la adecuada atención se da en forma limitada, 88 en gran medida, 16 no consideran adecuada la atención y 1 usuario considera adecuada la atención brindada por la ARWEBSYSTEMS SAC.

7. Considera adecuada el equipamiento tecnológico que presenta la ARWEBSYSTEMS SAC (computadoras, impresoras, etc) para atender al usuario?

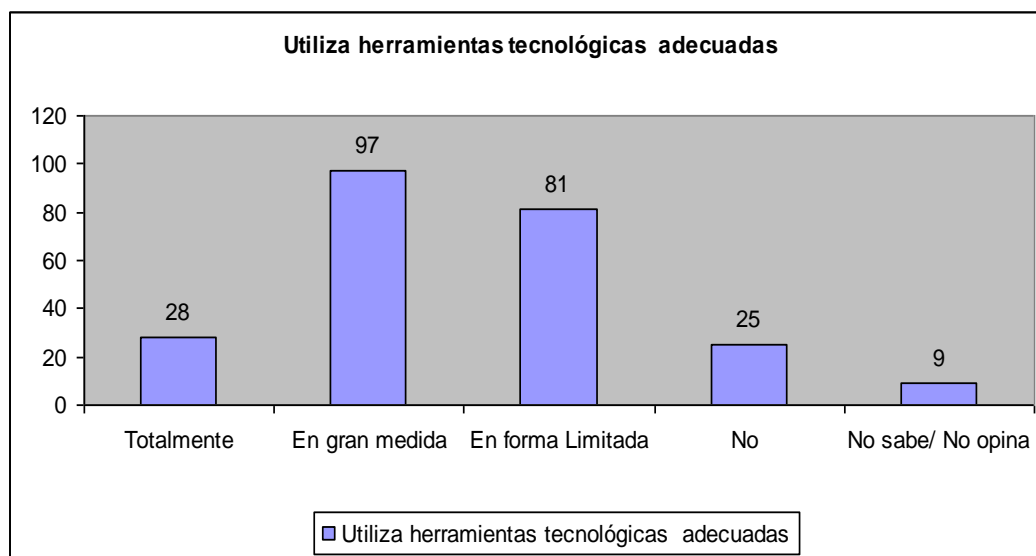
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Presenta equipamiento tecnológico adecuado	20	133	73	10	4



El gráfico indica que de una muestra de 240 usuarios: 133 consideran que en gran medida cuentan con un adecuado equipamiento tecnológico, 73 en forma limitada, 20 consideran que cuenta con un adecuado equipamiento, 10 usuarios consideran que no es el adecuado y 4 usuarios No sabe/No opina.

8. Considera adecuadas las herramientas tecnológicas (página web, sistemas informáticos, etc) que utiliza la ARWEBSYSTEMS SAC para atender al usuario?

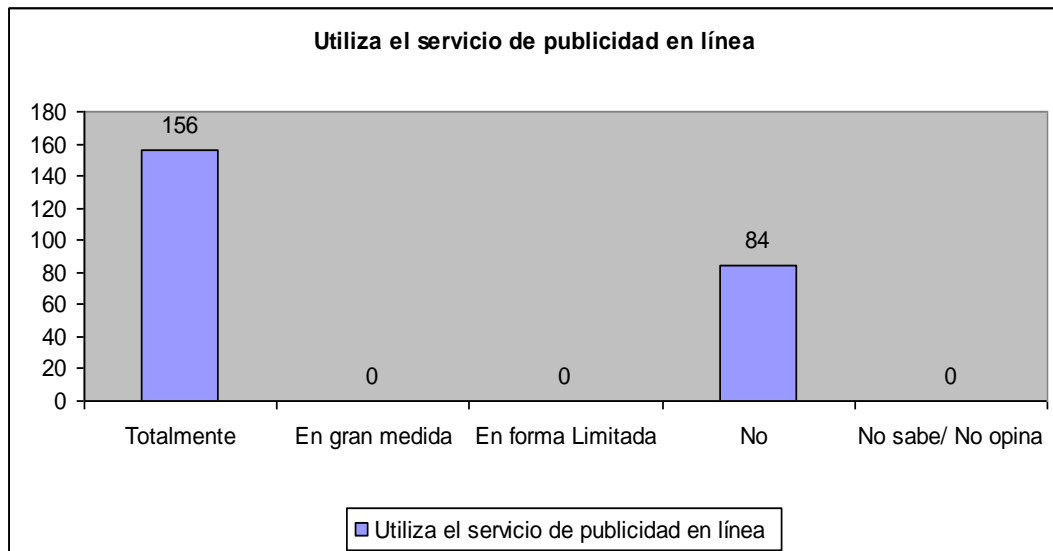
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Utiliza herramientas tecnológicas adecuadas	28	97	81	25	9



El gráfico indica que de una muestra de 240 usuarios: 97 consideran que en gran medida utilizan herramientas tecnológicas adecuadas, 81 en gran medida, 28 manifiestan que son adecuadas las herramientas utilizadas y 25 consideran que las herramientas tecnológicas no son las adecuadas.

9. Hace uso del servicio de publicidad en línea que brinda la ARWEBSYSTEMS SAC a través de internet?

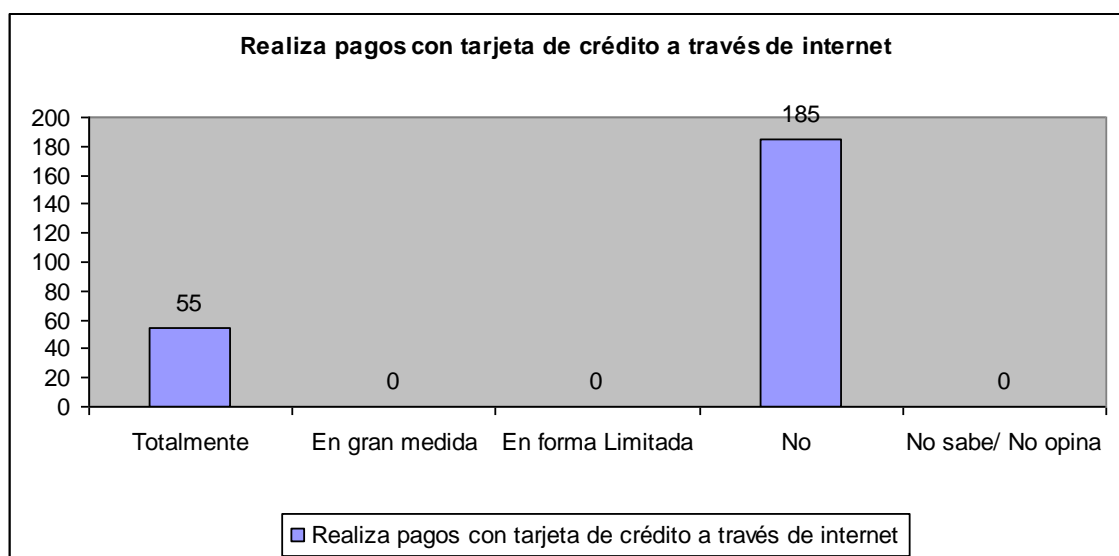
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Utiliza el servicio de publicidad en línea	156	0	0	84	0



El gráfico indica que de una muestra de 240 usuarios: 156 consideran que utilizan los servicios de publicidad que brinda la ARWEBSYSTEMS SAC por Internet y 84 no utilizan dichos servicios.

10. Realiza pagos con tarjeta de crédito para cancelar algún servicio que brinda la ARWEBSYSTEMS SAC a través de internet?

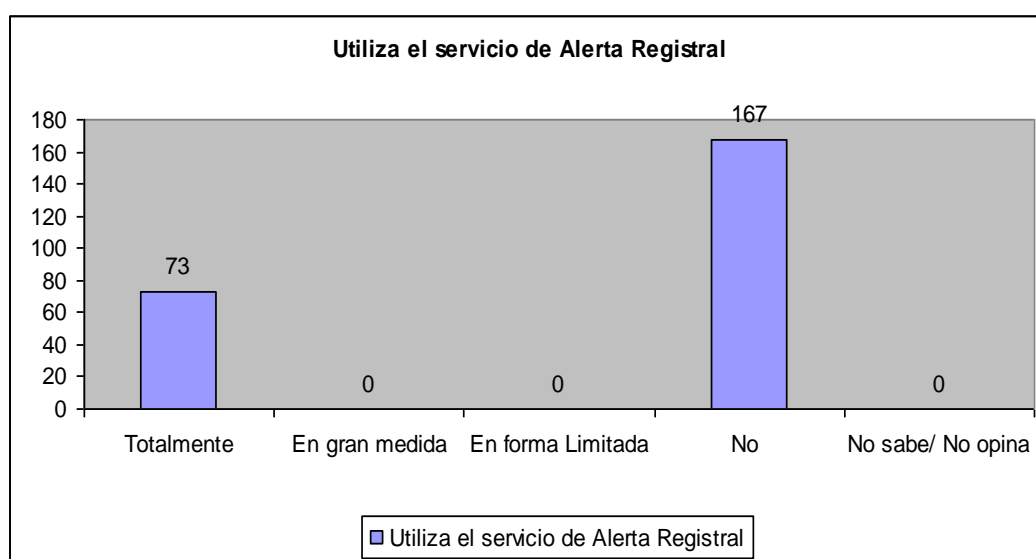
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Realiza pagos con tarjeta de crédito a través de internet	55	0	0	185	0



El gráfico indica que de una muestra de 240 usuarios: 185 no realizan pagos con tarjeta de crédito a través de internet de los servicios que brinda la ARWEBSYSTEMS SAC y 55 si realizan los pagos de los servicios con tarjeta de crédito.

11. Hace uso del servicio de Alerta Registral que brinda ARWEBSYSTEMS SAC?

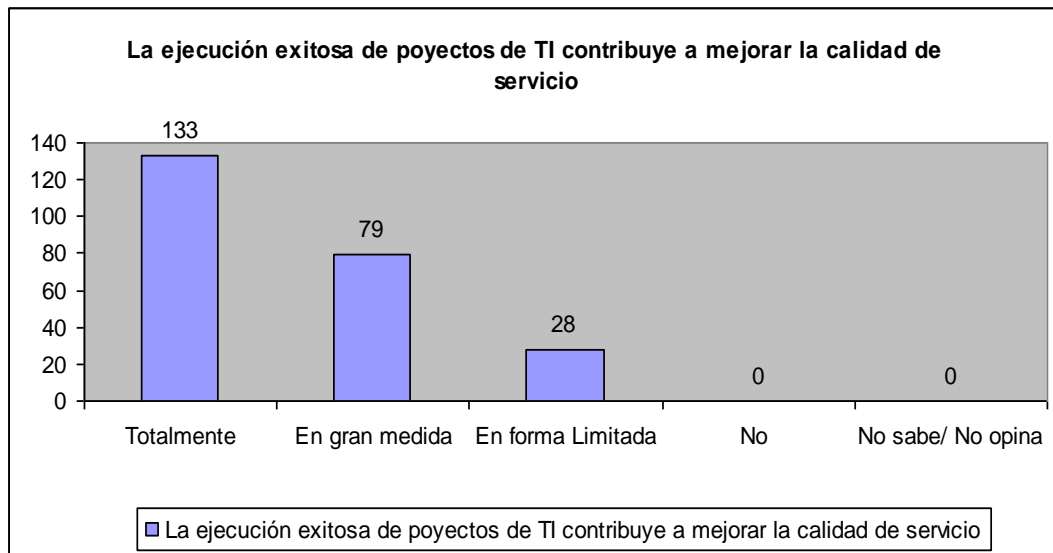
Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
Utiliza el servicio de Alerta Registral	73	0	0	167	0



El gráfico indica que de una muestra de 240 usuarios: 167 no utilizan el servicio de Alerta Registral y 73 utilizan dicho servicio que brinda la ARWEBSYSTEMS SAC.

12. Desde su punto de vista, cree Ud. que la ejecución exitosa de proyectos de tecnología de la información contribuiría favorablemente en mejorar la calidad del servicio que brinda la ARWEBSYSTEMS SAC?

Indicador	Totalmente	En gran medida	En forma Limitada	No	No sabe/ No opina
La ejecución exitosa de proyectos de TI contribuye a mejorar la calidad de servicio	133	79	28	0	0



El gráfico indica que de una muestra de 240 usuarios: 133 consideran que la ejecución exitosa del proyecto contribuiría a mejorar la calidad de servicio que brinda la ARWEBSYSTEMS SAC, 79 en gran medida y 28 consideran que contribuiría en forma limitada la calidad del servicio brindado por la ARWEBSYSTEMS SAC la ejecución de proyectos de tecnología de información.

ANEXO N° 03

GLOSARIO DE TÉRMINOS

- a) **Actividad:** Un componente del trabajo realizado en el transcurso del proyecto.
- b) **Acta de Constitución:** Denominada también acta de autorización del proyecto, con el cual se da por iniciado el proyecto. Documento emitido por el iniciador o patrocinador del proyecto que autoriza formalmente la existencia de un proyecto.
- c) **Calidad:** De acuerdo a lo señalado por *Drucker (2000)* se define a calidad como “es lo que el cliente está dispuesto a pagar en función de lo que obtiene y valora. Según lo señalado por la Norma Internacional ISO 8402: define a calidad como “totalidad de las características de un entidad que le confiere la aptitud para satisfacer las necesidades establecidas e implícitas” [ISO8402:1995]

Además se puede definir como calidad al logro de los objetivos cumpliendo tres enfoques: tiempo, costo y recursos humanos.

- d) **Conocimiento:** En el año 2001, *Davenport & Prusak* señalaron que el conocimiento no es ni datos ni información, aunque se relaciona con ambos y a menudo las diferencias entre estos términos es una cuestión de grado. Conocimiento es una mezcla fluida de experiencia estructurada, valores, información contextual e internalización experta que proporciona un marco para la evaluación e incorporación de nuevas experiencias e información.

Así mismo, *Sarah Becker (2005)*, señala que el conocimiento es el proceso en virtud del cual la realidad se refleja y reproduce en el pensamiento humano. Dicho proceso está condicionado por las leyes del devenir social y se halla indisolublemente unido a la actividad práctica.

Sin embargo, se puede definir también como conocimiento al conjunto de información almacenada en una persona mediante la experiencia o el aprendizaje,

es decir se trata de poseer múltiples datos interrelacionados que tomados en su conjunto poseen un valor agregado.

- e) **Controlar:** Es comparar el desempeño real con el desempeño planificado.
- f) **Criterios:** Normas, reglas o pruebas sobre las que se puede basar una opinión o decisión, o por medio de la cual se puede evaluar un producto, servicio, resultado o proceso.
- g) **Entregable:** Cualquier producto, resultado o capacidad de prestar un servicio único y verificable que debe producirse para terminar un proceso, una fase o un proyecto.
- h) **Juicio de Expertos:** Un juicio que se brinda sobre la bases de la experiencia en un área de aplicación, área de conocimiento, disciplina, industria, etc.
- i) **Monitorear:** Consiste en recolectar datos de desempeño del proyecto con respecto a un plan, producir medidas de desempeño, e informar y difundir la información sobre el desempeño.
- j) **Objetivo:** Una meta hacia la cual se debe dirigir el trabajo.
- k) **Presupuesto:** Es la estimación aprobada para el proyecto o cualquier otro componente de la estructura de desglose del trabajo u otra actividad del cronograma.
- l) **Proceso de Cierre:** Comprende aquellos procesos realizados para finalizar todas las actividades a través de todos los grupos de procesos de la gestión de proyecto.
- m) **Requisito:** Una condición o capacidad que un sistema, producto, servicio, resultado o componente debe satisfacer o poseer para cumplir con un contrato, norma, especificación u otros documentos formalmente impuestos.

Informe de viabilidad y alcance.

Informe de viabilidad y alcance
Resumen de la documentación analizada
<i>Contenido de la documentación recibida</i>
Viabilidad de las funcionalidades a probar
<i>Viabilidad de las diferentes pruebas</i>
Alcance de las pruebas
<i>Tipos de prueba a realizar según su funcionalidad</i>
Definición entorno de pruebas
<i>Recursos hardware y software necesarios</i>
Roles y responsabilidades
<i>Diferentes roles existentes y sus responsabilidades</i>

Informe de planificación.

Informe de planificación
Actividades
<i>Detalle de todas las actividades a realizar en las siguientes etapas del proceso de pruebas.</i>
Alcance de las pruebas
<i>Que se va a probar y tipo de pruebas a realizar.</i>
Definición entorno de pruebas
<i>A imagen del entorno final de producción.</i>
Estimaciones de tiempo
<i>El tiempo para realizar todas las actividades y responsabilidades.</i>
Recursos
<i>Para ejecutar cada una de las actividades planificadas.</i>
Herramienta gestión de pruebas
<i>Definición de herramienta para gestión de pruebas</i>

Plan de pruebas.

Plan de pruebas	
Identificador	
Alcance	
<i>Estructura de la prueba</i>	
Estrategia	
Prioridades	
<i>Por características de prioridades o por prioridad de casos de uso</i>	
Criterios de aceptación / fallo	
<i>Porcentaje mínimo de éxito de caso de prueba</i>	
Criterios de suspensión	
<i>Problemas y/o limitaciones que puedan darse a lo largo del proyecto</i>	
Complejidad	
<i>Alta, media y baja</i>	
Funciones y responsabilidades	
Recursos necesarios	
Entorno	Personal

Métricas
<i>Se aplican a lo largo del ciclo de vida del proyecto</i>
Herramientas de gestión
Aprobaciones

Casos de prueba.

Caso de prueba		
Identificador		
Responsable		
Elementos de Software		
<i>Definir los elementos que se van a probar y las características que ejercerá este caso.</i>		
Necesidades de entorno		
<i>Hardware, software, personal</i>		
Entradas	Salidas	Resultados
<i>Especificaciones de cada entrada requerida para ejecutar cada caso de prueba, incluyendo las relaciones entre las diversas entradas</i>	<i>Especificaciones de todas las salidas y las características requeridas para los elementos que se van a probar</i>	<i>Resultados esperados</i>
Observaciones		

Matriz de trazabilidad.

Caso de prueba Caso de uso	ID caso de prueba 1	ID caso de prueba 2	ID caso de prueba 3	...
Id Caso de uso 1				
Id Caso de uso 2				
Id Caso de uso3				
Id Caso de uso 4				
.				
.				
.				

Informe de incidencias.

Informe de incidencias	
Identificador	
Resumen del incidencias.	
Descripción de datos objetivos	
<i>Fecha/hora, entradas, resultados.</i>	
Impacto que tendrá sobre las pruebas	



Los olivos, Lima Perú
ARWEBSYSTEMS Tu Presencia en Internet

AUTORIZACIÓN DE PUBLICACIÓN EN TESIS

Por medio del presente, yo, Alcira Molina Neyra, con DNI 40207050 y CIP 74724, en calidad de Gerente General de la microempresa ARWEBSYSTEMS SAC, autorizo al Ingeniero Rembrandt Ubalde Enriquez, con DNI 01345428, hacer uso del nombre de la empresa, para publicarla en su Tesis para obtener su grado de Maestro en Ingeniería de Sistemas, mas no autorizo se publiquen los nombres de nuestros clientes.

Se expide el presente a solicitud del interesado para los fines que estime conveniente.

Los Olivos, 02 de enero del 2017.

Ing. Alcira Molina Neyra
Gerente General
CIP 74724



INFORME DE OPINIÓN DE EXPERTO RESPECTO AL MODELO DE CALIDAD PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN MICROEMPRESAS

I. DATOS GENERALES:

- 1.1. Apellidos y nombres del informante (Experto): Yalmar Ponce Atencio.....
- 1.2. Grado Académico. Doctor en Ciencias de la Computación en la Universidad Federal de Rio de Janeiro.....

II. OPINIÓN

INDICADORES DE EVALUACIÓN	CRITERIOS Sobre el modelo de calidad de software propuesto	Muy Malo	Malo	Regular	Bueno	Muy Bueno
		1	2	3	4	5
1. CLARIDAD	El modelo se muestra simple, fácil de entender y aplicar por una microempresa				X	
2. OBJETIVIDAD	El modelo se plantea con componentes Observables y medibles				X	
3. CONSISTENCIA	Existe una organización lógica en los contenidos y relación con la teoría a nivel de gestión de proyectos y mejora de procesos de software				X	
4. COHERENCIA	Existe relación de los contenidos de los objetivos planteados				X	
5. PERTINENCIA	El modelo está acorde a la realidad microempresarial de producción de software				X	
6. SUFICIENCIA	Son suficientes la cantidad y calidad de teorías , estándares , practicas , métodos , técnicas y herramientas					X

III. RESULTADOS DE LA OPINIÓN

- 3.1. Valoración total cuantitativa: 4.5
- 3.2. Opinión: FAVORABLE x
- 3.3. Observaciones: _____

INFORME DE OPINIÓN DE EXPERTO RESPECTO AL MODELO DE CALIDAD PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN MICROEMPRESAS

I. DATOS GENERALES:

- 1.1. Apellidos y nombres del informante (Experto): Alberto Gaspar Vera.
 1.2. Grado Académico. Doctor en Proyectos de Integración y Desarrollo – Universidad de León – España.

II. OPINIÓN

INDICADORES DE EVALUACIÓN	CRITERIOS Sobre el modelo de calidad de software propuesto	Muy Malo	Malo	Regular	Bueno	Muy Bueno
		1	2	3	4	5
1. CLARIDAD	El modelo se muestra simple, fácil de entender y aplicar por una microempresa				X	
2. OBJETIVIDAD	El modelo se plantea con componentes Observables y medibles				X	
3. CONSISTENCIA	Existe una organización lógica en los contenidos y relación con la teoría a nivel de gestión de proyectos y mejora de procesos de software					X
4. COHERENCIA	Existe relación de los contenidos de los objetivos planteados				X	
5. PERTINENCIA	El modelo está acorde a la realidad microempresarial de producción de software				X	
6. SUFICIENCIA	Son suficientes la cantidad y calidad de teorías , estándares , practicas , métodos , técnicas y herramientas				X	

III. RESULTADOS DE LA OPINIÓN

- 3.1. Valoración total cuantitativa: _____
 3.2. Opinión: FAVORABLE X DEBE MEJORAR NO FAVORABLE
 3.3. Observaciones:

INFORME DE OPINIÓN DE EXPERTO RESPECTO AL MODELO DE CALIDAD PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN MICROEMPRESAS

I. DATOS GENERALES:

1.1. Apellidos y nombres del informante (Experto): Javier C. Sánchez Espinoza

1.2. Grado Académico. Doctor en Ingeniería de Sistemas – Universidad Nacional de Ingeniería

II. OPINIÓN

INDICADORES DE EVALUACIÓN	CRITERIOS Sobre el modelo de calidad de software propuesto	Muy Malo	Malo	Regular	Bueno	Muy Bueno
		1	2	3	4	5
1. CLARIDAD	El modelo se muestra simple, fácil de entender y aplicar por una microempresa				X	
2. OBJETIVIDAD	El modelo se plantea con componentes Observables y medibles			X		
3. CONSISTENCIA	Existe una organización lógica en los contenidos y relación con la teoría a nivel de gestión de proyectos y mejora de procesos de software				X	
4. COHERENCIA	Existe relación de los contenidos de los objetivos planteados				X	
5. PERTINENCIA	El modelo está acorde a la realidad microempresarial de producción de software				X	
6. SUFICIENCIA	Son suficientes la cantidad y calidad de teorías , estándares , practicas , métodos , técnicas y herramientas			X		

III. RESULTADOS DE LA OPINIÓN

3.1. Valoración total cuantitativa: _____

3.2. Opinión: FAVORABLE DEBE MEJORAR NO FAVORABLE

3.3. Observaciones: Mejora la manera de medir los procesos con CPI

-

INFORME DE OPINIÓN DE EXPERTO RESPECTO AL MODELO DE CALIDAD PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN MICROEMPRESAS

I. DATOS GENERALES:

1.1. Apellidos y nombres del informante (Experto): Ivan Petrlik Azabache.....

1.2. Grado Académico. Doctor en Ingeniería de Sistemas – Universidad Federico Villareal

II. OPINIÓN

INDICADORES DE EVALUACIÓN	CRITERIOS Sobre el modelo de calidad de software propuesto	Muy Malo	Malo	Regular	Bueno	Muy Bueno
		1	2	3	4	5
1. CLARIDAD	El modelo se muestra simple, fácil de entender y aplicar por una microempresa					X
2. OBJETIVIDAD	El modelo se plantea con componentes Observables y medibles				X	
3. CONSISTENCIA	Existe una organización lógica en los contenidos y relación con la teoría a nivel de gestión de proyectos y mejora de procesos de software					X
4. COHERENCIA	Existe relación de los contenidos de los objetivos planteados					X
5. PERTINENCIA	El modelo está acorde a la realidad microempresarial de producción de software				X	
6. SUFICIENCIA	Son suficientes la cantidad y calidad de teorías , estándares , practicas , métodos , técnicas y herramientas				X	

III. RESULTADOS DE LA OPINIÓN

- 3.1. Valoración total cuantitativa: 4
- 3.2. Opinión: FAVORABLE __X
- 3.3. Observaciones: Ninguna.

INFORME DE OPINIÓN DE EXPERTO RESPECTO AL MODELO DE CALIDAD PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN MICROEMPRESAS

I. DATOS GENERALES:

1.1. Apellidos y nombres del informante (Experto): Luz Sussy Bayona Oré

1.2. Grado Académico. Doctora en Lenguajes y Sistemas Informáticos e Ingeniería de Software. Universidad Politécnica de Madrid. España.

II. OPINIÓN

INDICADORES DE EVALUACIÓN	CRITERIOS Sobre el modelo de calidad de software propuesto	Muy Malo	Malo	Regular	Bueno	Muy Bueno
		1	2	3	4	5
1. CLARIDAD	El modelo se muestra simple, fácil de entender y aplicar por una microempresa					X
2. OBJETIVIDAD	El modelo se plantea con componentes Observables y medibles					X
3. CONSISTENCIA	Existe una organización lógica en los contenidos y relación con la teoría a nivel de gestión de proyectos y mejora de procesos de software					X
4. COHERENCIA	Existe relación de los contenidos de los objetivos planteados					X
5. PERTINENCIA	El modelo está acorde a la realidad microempresarial de producción de software				X	
6. SUFICIENCIA	Son suficientes la cantidad y calidad de teorías , estándares , practicas , métodos , técnicas y herramientas				X	

III. RESULTADOS DE LA OPINIÓN

3.1. Valoración total cuantitativa: ___5

3.2. Opinión: FAVORABLE X

3.3. Observaciones:

INFORME DE OPINIÓN DE EXPERTO RESPECTO AL MODELO DE CALIDAD PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN MICROEMPRESAS

I. DATOS GENERALES:

1.1. Apellidos y nombres del informante (Experto): Willy Gustavo Ugarte Rojas

1.2. Grado Académico. Doctor en Informática, Especialidad Informática y aplicaciones - UCBN-Francia.

II. OPINIÓN

INDICADORES DE EVALUACIÓN	CRITERIOS Sobre el modelo de calidad de software propuesto	Muy Malo	Malo	Regular	Bueno	Muy Bueno
		1	2	3	4	5
1. CLARIDAD	El modelo se muestra simple, fácil de entender y aplicar por una microempresa					X
2. OBJETIVIDAD	El modelo se plantea con componentes Observables y medibles				X	
3. CONSISTENCIA	Existe una organización lógica en los contenidos y relación con la teoría a nivel de gestión de proyectos y mejora de procesos de software					X
4. COHERENCIA	Existe relación de los contenidos de los objetivos planteados				X	
5. PERTINENCIA	El modelo está acorde a la realidad microempresarial de producción de software					X
6. SUFICIENCIA	Son suficientes la cantidad y calidad de teorías , estándares , practicas , métodos , técnicas y herramientas				X	

III.RESULTADOS DE LA OPINIÓN

- 3.1. Valoración total cuantitativa: _____
- 3.2. Opinión: FAVORABLE **X** DEBE MEJORAR _____ NO FAVORABLE ____
- 3.3. Observaciones:

INFORME DE OPINIÓN DE EXPERTO RESPECTO AL MODELO DE CALIDAD PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE EN MICROEMPRESAS

I. DATOS GENERALES:

1.1. Apellidos y nombres del informante (Experto): Millaray Curilem.....

1.2. Grado Académico. Doctora en Ingeniería Eléctrica, mención Sistemas de Información. Universidad Federal de Santa Catarina – Brasil.

II. OPINIÓN

INDICADORES DE EVALUACIÓN	CRITERIOS Sobre el modelo de calidad de software propuesto	Muy Malo	Malo	Regular	Bueno	Muy Bueno
		1	2	3	4	5
1. CLARIDAD	El modelo se muestra simple, fácil de entender y aplicar por una microempresa					X
2. OBJETIVIDAD	El modelo se plantea con componentes Observables y medibles					X
3. CONSISTENCIA	Existe una organización lógica en los contenidos y relación con la teoría a nivel de gestión de proyectos y mejora de procesos de software					X
4. COHERENCIA	Existe relación de los contenidos de los objetivos planteados					X
5. PERTINENCIA	El modelo está acorde a la realidad microempresarial de producción de software					X
6. SUFICIENCIA	Son suficientes la cantidad y calidad de teorías , estándares , practicas , métodos , técnicas y herramientas					X

III. RESULTADOS DE LA OPINIÓN

3.1. Valoración total cuantitativa: **5**

3.2. Opinión: **FAVORABLE** DEBE MEJORAR _____ NO FAVORABLE ___

3.3. Observaciones: No tengo Observaciones, es un gran aporte a su país.

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	INDICADORES	INSTRUMENTOS
----------	-----------	-----------	-----------	-------------	--------------

<p>PROBLEMA GENERAL: ¿ Qué efecto tendrá la implementación de un modelo para mejorar la eficiencia en la gestión de los proyectos de calidad de software en las microempresas?</p>	<p>OBJETIVO GENERAL: Diseñar un modelo de calidad para la gestión de proyectos de software factible y con características alineadas a las necesidades de las microempresas .</p>	<p>HIPOTESIS GENERAL. La propuesta de un modelo para la gestión de proyectos software; factible y con características alineadas a las necesidades de las microempresas, permite a estas microempresas ser más competentes en su rubro.</p>	<p>Independiente: Modelo de Calidad para la Gestión de Proyectos de Software en Microempresas</p>	<p>Número de microempresas seleccionadas que se beneficiaron por medio de la aplicación de la modelo.</p> <p>Número de proyectos exitosos de pruebas de software gestionados con el modelo propuesta.</p> <p>Número de microempresas seleccionadas que no se beneficiaron por medio de la aplicación del modelo.</p>	<p>Casos de aplicación de la Modelo.</p> <p>Encuesta de Satisfacción del uso de la Modelo.</p> <p>Formato de Calificación a la Modelo por Expertos.</p> <p>Comparación de microempresas que no usan el Modelo versus microempresas que aplicaron la metodología.</p>
<p>201</p>					

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	INDICADORES	INSTRUMENTOS
----------	-----------	-----------	-----------	-------------	--------------

<p>PROB. ESPECIFICO 1): ¿Cómo se diseñaría un modelo de calidad económica, operativa y técnicamente factible para la gestión de los proyectos de software de las microempresas?</p>	<p>OBJETIVO ESPECIFICO 1: Diseñar un modelo flexible, adaptable y escalable con sus respectivos procesos, actividades y herramientas de soporte a un nivel de gestión y técnico para la gestión de los proyectos de software alineadas a las diferentes metodologías de desarrollo de software utilizadas por los clientes de las microempresas.</p>	<p>HIPOTESIS ESPECIFICO 1: El Diseño de una metodología flexible, adaptable y escalable para la gestión de los proyectos de pruebas de software alineadas a las diferentes metodologías de desarrollo de software utilizadas por los clientes de las microempresas peruanas especializadas en servicios de calidad de software, permite a estas microempresas ser más competentes en su rubro respecto a ampliar su cobertura de proyectos con sus clientes y mayor posibilidad de captación de nuevos clientes.</p>	<p>Dependiente: Factibilidad Económica Factibilidad Operativa Factibilidad Técnica</p> <p style="text-align: center;">203</p>	<p>Grado de Concordancia al presupuesto promedio asignado a proyectos pequeños por parte de microempresas.</p> <p>Grado de Concordancia al presupuesto promedio asignado a proyectos medianos por parte de microempresas.</p> <p>Grado de Concordancia al presupuesto promedio asignado a proyectos grandes por parte de microempresas.</p> <p>Grado de Simplicidad para la aplicación de la metodología por los líderes de equipos de proyectos de pruebas de software de microempresas.</p> <p>Grado de Simplicidad para la aplicación de la metodología por los miembros de equipos que trabajan desde el inicio del proyecto de pruebas de software de microempresas.</p> <p>Grado de Simplicidad para la aplicación de la metodología por nuevos miembros de equipos que se añaden durante el transcurso del proyecto de pruebas de software de microempresas.</p> <p>Nivel de Funcionamiento de la Metodología en Proyectos de Pruebas de Software de una microempresa.</p> <p>Cantidad de capacidades técnicas necesarias para aplicar la metodología</p> <p>Complejidad de capacidades técnicas necesarias para aplicar la metodología</p> <p>Grado de existencia o alcance de la tecnología necesaria para aplicar la metodología por parte de una microempresa.</p>	<p>Formato a nivel Gerencial para la Medición de la Curva de Aprendizaje de la Metodología.</p> <p>Formato a nivel Subordinado para la Medición de la Curva de Aprendizaje de la Metodología.</p> <p>Formato a nivel Subordinado Nuevo para la Medición de la Curva de Aprendizaje de la Metodología.</p> <p>Formato de evaluación del funcionamiento de la Metodología.</p> <p>Formato de evaluación de la Factibilidad Técnica de la Metodología.</p>
---	--	--	--	---	---

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	INDICADORES	INSTRUMENTOS
----------	-----------	-----------	-----------	-------------	--------------

<p>PROB. ESPECIFICO 2): ¿Cómo se diseñaría un modelo flexible, adaptable y escalable para la gestión de los proyectos de calidad de software en los diferentes modelos de ciclo de vida de desarrollo de software aplicados por las microempresas?</p>	<p>OBJETIVO ESPECIFICO 2: Diseñar un modelo con procesos, actividades y herramientas que permitan la factibilidad económica, operativa y técnica para la gestión de los proyectos de software de las microempresas .</p>	<p>HIPOTESIS ESPECIFICO 2: El Diseño de una metodología económica, operativa y técnicamente factible, permite a las microempresas especializadas contar con un modelo para la gestión de sus proyectos de software acorde a su bajo presupuesto, de simple aplicación y de clase mundial.</p>	<p>Dependiente:</p> <p>Flexibilidad e la Metodología</p> <p>Adaptabilidad de la Metodología</p> <p>Escalabilidad de la Metodología</p> <p style="text-align: center;">205</p>	<p>Grado de flexibilidad de la metodología ante un proceso de cambio simple.</p> <p>Grado de flexibilidad de la metodología ante un proceso de cambio moderado.</p> <p>Grado de flexibilidad de la metodología ante un proceso de cambio crítico.</p> <p>Grado de Adaptabilidad de la metodología ante un cambio externo.</p> <p>Grado de Adaptabilidad de la metodología ante un cambio interno.</p> <p>Grado de Escalabilidad de la metodología ante una ampliación razonable en términos de coste.</p> <p>Grado de Escalabilidad de la metodología ante una ampliación razonable en términos de tiempo.</p> <p>Grado de Escalabilidad de la metodología ante una ampliación razonable en términos de complejidad.</p>	<p>Formato de evaluación de la Flexibilidad de la Metodología.</p> <p>Formato de evaluación de la Adaptabilidad de la Metodología a cambios externos.</p> <p>Formato de evaluación de la Adaptabilidad de la Metodología a cambios internos.</p> <p>Formato de evaluación de la Escalabilidad de la Metodología ante ampliaciones en coste del proyecto.</p> <p>Formato de evaluación de la Escalabilidad de la Metodología ante ampliaciones en tiempo del proyecto.</p> <p>Formato de evaluación de la Escalabilidad de la Metodología ante ampliaciones de complejidad del proyecto.</p>
--	--	---	--	--	---

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	INDICADORES	INSTRUMENTOS
<p>PROB. ESPECIFICO 3): ¿Cómo se diseñaría un modelo de calidad para la gestión de proyectos de software que pueda ser aplicable en proyectos de desarrollo de software que funcionen en plataformas SOA , Mobile o Cloud Computing?</p>	<p>OBJETIVO ESPECIFICO 3: Diseñar un modelo considerando una arquitectura moderna con procesos, actividades y herramientas que hagan posible la gestión de proyectos de pruebas a software en plataformas SOA, Mobile, Cloud Computing o la integración entre éstas</p>	<p>HIPOTESIS ESPECIFICO 3: El Diseño de un modelo considerando una arquitectura moderna con procesos, actividades y herramientas que hagan posible la gestión de proyectos de pruebas de software en plataformas SOA, Mobile , Cloud Computing o la integración de estas, permite que las MPS puedan gestionar proyectos de pruebas de software complejo</p>	<p>Dependiente: Utilidad de la Metodología en Pruebas a Software en SOA</p> <p>Utilidad de la Metodología en Pruebas a Software para dispositivos Móviles</p> <p>Utilidad de la Metodología en Pruebas a Software en Cloud Computing</p>	<p>Grado de Utilidad del Modelo respecto a las dependencias de otras aplicaciones y sistemas en pruebas a software SOA.</p> <p>Grado de Utilidad del Modelo respecto a servicios sin interfaz gráfica de usuario en pruebas a software SOA.</p> <p>Grado de Utilidad del Modelo respecto a la integración de diferentes tecnologías y diferentes formatos de datos en pruebas a software SOA.</p> <p>Grado de Utilidad del Modelo respecto a la planificación y estrategia para abordar los problemas de disponibilidad en pruebas a software SOA.</p> <p>Grado de Utilidad del Modelo respecto a la seguridad en pruebas a software SOA.</p> <p>Grado de Utilidad del Modelo respecto a las pruebas de funcionalidad en pruebas a software móvil.</p> <p>Grado de Utilidad del Modelo respecto a las pruebas de rendimiento en pruebas a software móvil.</p> <p>Grado de Utilidad del Modelo respecto a las pruebas de seguridad en pruebas a software móvil</p> <p>Grado de Utilidad del Modelo respecto a pruebas a software con ayuda total de Cloud Computing</p> <p>Grado de Utilidad del Modelo respecto a pruebas a software con ayuda casi total de Cloud Computing</p>	<p>Formato de evaluación de la Utilidad del modelo para la gestión de proyectos de pruebas de software SOA para su característica de dependencias.</p> <p>Formato de evaluación de la Utilidad del Modelo para la gestión de proyectos de pruebas de software SOA para su característica de servicios sin GUI.</p> <p>Formato de evaluación de la Utilidad del Modelo para la gestión de proyectos de pruebas de software SOA para su característica de integración.</p> <p>Formato de evaluación de la Utilidad de la Metodología para la gestión de proyectos de pruebas de software SOA para su característica de planificación y estrategia.</p> <p>Formato de evaluación de la Utilidad del Modelo para la gestión de proyectos de pruebas de software SOA para su característica de seguridad.</p> <p>Formato de evaluación de la Utilidad del Modelo para la gestión de proyectos de pruebas de funcionalidad de software para dispositivos móviles.</p> <p>Formato de evaluación de la Utilidad del Modelo para la gestión de proyectos de pruebas de rendimiento de software para dispositivos móviles.</p>

				<p>Grado de Utilidad del Modelo respecto a pruebas a software con ayuda Media de Cloud Computing</p>	<p>Formato de evaluación de la Utilidad del Modelo para la gestión de proyectos de pruebas de seguridad de software para dispositivos móviles. Formato de evaluación de la Utilidad del Modelo para la gestión de proyectos de pruebas a software con ayuda total de Cloud Computing (SaaS)</p> <p>Formato de evaluación de la Utilidad del Modelo para la gestión de proyectos de pruebas a software con ayuda casi total de Cloud Computing (PaaS)</p> <p>Formato de evaluación de la Utilidad del Modelo para la gestión de proyectos de pruebas a software con ayuda Media de Cloud Computing (IaaS)</p>
--	--	--	--	--	---