



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

**DISEÑO DE INVESTIGACIÓN DEL MODELO DE MEJORA BASADO EN ISO/IEC/IEEE 29119  
Y PRUEBAS TMMI PARA AUMENTAR LA EFECTIVIDAD EN LA GESTIÓN DE ERRORES Y  
REDUCIR LOS COSTOS EN UNA EMPRESA DE DESARROLLO DE SOFTWARE**

**Alan Santiago Ruyán Borrayo**

Asesorado por la Msc. Inga. Isiris Yasmín Córdova Grajeda

Guatemala, abril de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE INVESTIGACIÓN DEL MODELO DE MEJORA BASADO EN ISO/IEC/IEEE 29119  
Y PRUEBAS TMMI PARA AUMENTAR LA EFECTIVIDAD EN LA GESTIÓN DE ERRORES Y  
REDUCIR LOS COSTOS EN UNA EMPRESA DE DESARROLLO DE SOFTWARE**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**ALAN SANTIAGO RUYÁN BORRAYO**

ASESORADO POR LA MSC. INGA. ISIRIS YASSMÍN CÓRDOVA GRAJEDA

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, ABRIL DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Francisco Javier Guevara Castillo
EXAMINADOR	Ing. Herman Igor Véliz Linares
EXAMINADOR	Ing. César Augusto Fernández Cáceres
SECRETARIA	Inga. Lesbia Magalí Herrera López

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**DISEÑO DE INVESTIGACIÓN DEL MODELO DE MEJORA BASADO EN ISO/IEC/IEEE 29119  
Y PRUEBAS TMMI PARA AUMENTAR LA EFECTIVIDAD EN LA GESTIÓN DE ERRORES Y  
REDUCIR LOS COSTOS EN UNA EMPRESA DE DESARROLLO DE SOFTWARE**

Tema que me fuera asignado por la Dirección de la Escuela de Estudios de Postgrado, con fecha 1 de septiembre de 2020.

**Alan Santiago Ruyán Borrayo**

Ref. EEPFI-0237-2021  
Guatemala, 16 de febrero de 2021

Director  
Carlos Gustavo Alonzo  
Escuela de Ciencias y Sistemas  
Presente.

Estimado Ing. Alonzo:

Reciba un cordial saludo de la Escuela de Estudios de Postgrado. El propósito de la presente es para informarle que se ha revisado y aprobado el **DISEÑO DE INVESTIGACIÓN: MODELO DE MEJORA BASADO EN ISO/IEC/IEEE 29119 Y PRUEBAS TMMI PARA AUMENTAR LA EFECTIVIDAD EN LA GESTIÓN DE ERRORES Y REDUCIR LOS COSTOS EN UNA EMPRESA DE DESARROLLO DE SOFTWARE**, presentado por el estudiante **Alan Santiago Ruyán Borrayo** carné número 200614944, quien optó por la modalidad del "PROCESO DE GRADUACIÓN DE LOS ESTUDIANTES DE LA FACULTAD DE INGENIERÍA OPCIÓN ESTUDIOS DE POSTGRADO". Previo a culminar sus estudios en la Maestría en Artes en Gestión Industrial.

Y habiendo cumplido y aprobado con los requisitos establecidos en el normativo de este Proceso de Graduación en el Punto 6.2, aprobado por la Junta Directiva de la Facultad de Ingeniería en el Punto Décimo, Inciso 10.2 del Acta 28-2011 de fecha 19 de septiembre de 2011, firmo y sello la presente para el trámite correspondiente de graduación de Pregrado.

Sin otro particular,

Atentamente,



Isiris Y. Córdova Grajeda  
INGENIERA EN CIENCIAS Y SISTEMAS  
Colegiado No.13753

Mtra. Isiris Yasmín Córdova Grajeda  
AsesorA

*"Id y Enseñad a Todos"*



Mtro. Carlos Humberto Aroche  
Coordinador de Maestría Gestión  
Industrial – Fin de Semana



Mtro. Ing. Edgar Darío Álvarez Cotí  
Director

Escuela de Estudios de Postgrado  
Facultad de Ingeniería



El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del Asesor, el visto bueno del Coordinador y Director de la Escuela de Estudios de Postgrado, del Diseño de Investigación en la modalidad Estudios de Pregrado y Postgrado titulado: **MODELO DE MEJORA BASADO EN ISO/IEC/IEEE 29119 Y PRUEBAS TMMI PARA AUMENTAR LA EFECTIVIDAD EN LA GESTIÓN DE ERRORES Y REDUCIR LOS COSTOS EN UNA EMPRESA DE DESARROLLO DE SOFTWARE**, presentado por el estudiante universitaria **Alan Santiago Ruyán Borrayo**, procedo con el Aval del mismo, ya que cumple con los requisitos normados por la Facultad de Ingeniería en esta modalidad.

ID Y ENSEÑAD A TODOS

Ing. Carlos Gustavo Alonzo  
Director

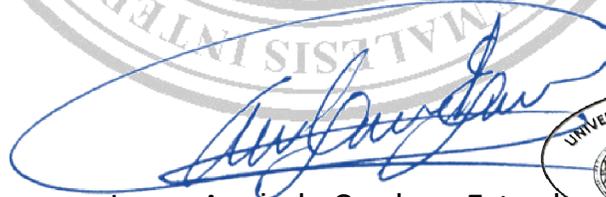
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, febrero de 2021

DTG. 186.2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **DISEÑO DE INVESTIGACIÓN DEL MODELO DE MEJORA BASADO EN ISO/IEC/IEEE 29119 Y PRUEBAS TMMI PARA AUMENTAR LA EFECTIVIDAD EN LA GESTIÓN DE ERRORES Y REDUCIR LOS COSTOS EN UNA EMPRESA DE DESARROLLO DE SOFTWARE**, presentado por el estudiante universitario: **Alan Santiago Ruyán Borrayo**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Anabela Cordova Estrada  
Decana



Guatemala, abril de 2021

AACE/asga

## **ACTO QUE DEDICO A:**

### **Dios Todopoderoso**

Por la gracia abundante que concede en mi caminar, por medio de la cual tengo la oportunidad de seguir adelante y ser una mejor persona día con día.

### **Mis padres**

Maira Borrayo y Lester Ruyán, con todo mi amor y agradecimiento, por todo lo recibido de su mano, por darme la libertad de caminar mi propio camino, depositando confianza y sabiduría en mi persona. Esto es parte de los muchos frutos que de su esfuerzo y dedicación hoy cosechamos juntos.

### **Mis hermanos**

Pamela, Lester y Edwin Ruyán, con todo mi corazón, pues han sido parte importante de esta meta, al brindarme su apoyo en todo momento.

### **Mis amigos y compañeros**

Por su apoyo en todo momento, ofreciendo amistad, conocimiento, consejo, y que me abrieron las puertas de su casa para llevar a cabo el trabajo duro.

## **AGRADECIMIENTOS A:**

- Mi Señor Jesucristo** Fuente de vida, gracia y sabiduría, por su compañía incondicional, dirección en cada etapa de mi vida y de mi andar académico y profesional.
- Mis padres** Maira Borrayo y Lester Ruyán, quienes con su amor, paciencia y trabajo duro siempre han sido fuente de bienestar, proveyendo la guía y formación adecuada para mi crecimiento personal.
- Mis hermanos** Pamela, Lester y Edwin Ruyán, por su amor y apoyo incondicional, que con su ejemplo me inspiran a ser cada día una mejor persona y profesional, sirviendo a otros con dedicación y entrega.
- Amigos y compañeros** Que, desde el colegio, pasando por la universidad, hasta mi carrera profesional me han apoyado de forma desinteresada. Ellos han sido parte fundamental de mi éxito, pues han soportado todo lo que soy y ofrezco, puliendo mis asperezas y celebrando mis logros.

**Mi casa de estudios**

Universidad de San Carlos de Guatemala, mi Facultad de Ingeniería y mis catedráticos y auxiliares, que al compartir sus conocimientos y orientar mi interés académico y profesional han formado a un ser humano dispuesto a devolver multiplicado lo recibido.

**Guatemala**

Tierra en donde nací, crecí y me formé como profesional, que ha hecho posible este logro a través de su gente de fe, que es esforzada, valiente y trabajadora.

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS.....	IX
GLOSARIO.....	XI
RESUMEN.....	XIII
1. INTRODUCCIÓN.....	1
2. ANTECEDENTES.....	5
3. PLANTEAMIENTO DEL PROBLEMA.....	11
3.1. Definición del problema.....	11
3.2. Descripción del problema.....	11
3.3. Formulación de preguntas.....	13
3.3.1. Pregunta central.....	13
3.3.2. Preguntas auxiliares.....	13
3.4. Delimitación.....	14
3.5. Viabilidad.....	14
3.6. Consecuencias de la investigación.....	14
4. JUSTIFICACIÓN.....	17
5. OBJETIVOS.....	19
5.1. Objetivo general.....	19
5.2. Objetivos específicos.....	19

6.	NECESIDADES POR CUBRIR Y ESQUEMA DE SOLUCIÓN.....	21
7.	MARCO TEÓRICO .....	23
7.1.	Industria de desarrollo de software.....	23
7.1.1.	Historia .....	23
7.1.2.	Ingeniería de software .....	25
7.1.3.	El valor de la ingeniería de software .....	25
7.1.4.	Procesos y ciclos de vida de software .....	26
7.1.5.	Actividades en el ciclo de vida en cascada.....	27
7.1.6.	Gestión de proyectos de software .....	28
7.1.7.	Mejora de procesos de software.....	30
7.1.8.	Mercado de software en Guatemala .....	30
7.1.8.1.	Historia .....	30
7.1.8.2.	Geografía.....	35
7.1.8.3.	Economía .....	35
7.1.8.4.	Política.....	37
7.1.8.5.	Marco jurídico .....	38
7.1.9.	Empresa de desarrollo de software analizada .....	39
7.2.	Calidad y mejora continua .....	39
7.2.1.	Calidad .....	39
7.2.2.	Herramientas.....	40
7.2.3.	Medición de la calidad .....	43
7.2.4.	Modelo de mejora continua de Deming .....	46
7.2.4.1.	Planificar.....	48
7.2.4.2.	Hacer .....	49
7.2.4.3.	Verificar .....	49
7.2.4.4.	Actuar .....	50
7.2.4.5.	Implementación del modelo de mejora continua .....	51

7.3.	Modelo de pruebas de software.....	53
7.3.1.	Modelo TMMI.....	59
7.3.1.1.	Nivel Inicial.....	59
7.3.1.2.	Nivel gestionado.....	60
7.3.1.3.	Nivel definido.....	60
7.3.1.4.	Nivel medido.....	60
7.3.1.5.	Nivel optimizado.....	60
7.3.2.	ISO/IEC/IEEE 29119.....	62
7.4.	Efectividad en el desarrollo de software.....	65
7.4.1.	Procesos.....	65
7.4.2.	Eficiencia en los procesos.....	72
7.4.3.	Calidad en procesos y productos de software.....	75
7.4.4.	Gestión de errores del software.....	80
7.4.5.	Efectividad en la gestión de errores.....	82
7.4.6.	Importancia en el cumplimiento de los tiempos de entrega.....	83
7.4.7.	Reducción de costos.....	83
7.4.7.1.	Técnicas y procedimientos para reducir costos.....	84
7.5.	Competitividad de las empresas de desarrollo de software.....	85
7.5.1.	Estrategias.....	85
7.5.2.	Herramientas para la competitividad.....	86
8.	PROPUESTA DE ÍNDICE DE CONTENIDOS.....	91
9.	METODOLOGÍA DE INVESTIGACIÓN.....	95
9.1.	Enfoque de la investigación.....	95
9.2.	Diseño de la investigación.....	96
9.3.	Tipo de estudio.....	96

9.4.	Variables e indicadores .....	97
9.5.	Fases de la investigación .....	100
9.5.1.	Fase 1: revisión documental de la teoría y bibliografía existente.....	100
9.5.2.	Fase 2: análisis de las actividades del área de pruebas que reducen la efectividad en la gestión .	100
9.5.3.	Fase 3: elaboración del modelo de gestión del área de pruebas para garantizar la efectividad y reducir los costos de desarrollo .....	104
9.5.4.	Fase 4: propuesta de evaluación y ajuste del proceso de pruebas para cumplir los objetivos .....	105
9.5.5.	Fase 5: análisis comparativo de las metodologías de gestión para proyectar los beneficios del modelo propuesto.....	106
10.	TÉCNICAS DE ANÁLISIS DE INFORMACIÓN.....	107
11.	CRONOGRAMA DE ACTIVIDADES.....	111
12.	FACTIBILIDAD DEL ESTUDIO.....	113
13.	REFERENCIAS .....	115
14.	APÉNDICES.....	129

# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Investigación sobre proyectos de tecnologías de la información (TI).....	26
2.	El retorno de la madurez.....	29
3.	Diagrama de Causa-Efecto.....	41
4.	Histograma.....	42
5.	Diagrama de Pareto.....	43
6.	Espiral de mejora continua.....	47
7.	Herramientas de la fase “Planificar”.....	48
8.	Herramientas de la fase “Hacer”.....	49
9.	Herramientas de la fase “Verificar”.....	50
10.	Herramientas de la fase “Actuar”.....	51
11.	El Ciclo de Vida Analítico.....	52
12.	IBM DevOps.....	53
13.	Etapa de corrección de errores.....	54
14.	Etapa de diseño de pruebas funcionales.....	55
15.	Etapa de detección de errores (parte 1).....	55
16.	Etapa de detección de errores (parte 2).....	56
17.	Etapa de evaluación de los productos de software.....	56
18.	Etapa de prevención de errores (parte 1).....	57
19.	Etapa de prevención de errores (parte 2).....	57
20.	Etapas de la arquitectura y áreas de proceso.....	61
21.	Estructura de la Norma ISO/IEC/IEEE 29119.....	63

22.	Correspondencia de los Grupos de Procesos de Dirección de Proyectos al ciclo PHVA .....	67
23.	Gestión de la integración del proyecto .....	68
24.	Gestión del alcance del proyecto .....	68
25.	Gestión del tiempo del proyecto.....	69
26.	Gestión de los costos del proyecto .....	69
27.	Gestión de la calidad del proyecto .....	69
28.	Gestión de los recursos humanos del proyecto .....	70
29.	Gestión de las comunicaciones del proyecto .....	70
30.	Gestión de los riesgos del proyecto .....	70
31.	Gestión de las adquisiciones del proyecto .....	71
32.	Gestión de los interesados del proyecto .....	71
33.	Rendimiento del proyecto .....	75
34.	Ciclo de vida del producto y su integración con el ciclo de vida del proyecto.....	76
35.	Mapa estratégico, base del Balanced ScoreCard .....	78
36.	Resultados iniciales de desempeño al adoptar CMMI V2.0, según el Instituto CMMI .....	79
37.	Diagrama de control .....	81
38.	Propiedades de la matriz FODA .....	87
39.	Estrategias que derivan de la matriz FODA .....	88
40.	Gráfico de PESTEL .....	89
41.	Cronograma.....	111

## TABLAS

I.	Modelos de pruebas de software más utilizados .....	58
II.	Normas internacionales que influenciaron la ISO/IEC/IEEE 29119 .....	62
III.	Correspondencia de los grupos de procesos del PMBOK con el ciclo PHVA .....	66
IV.	Factores del análisis estratégico .....	89
V.	Cuadro de Variables e Indicadores .....	99
VI.	Cuadro de la muestra de proyectos a evaluar .....	101
VII.	Cálculo de la muestra de personal del departamento de TI.....	103
VIII.	Recursos financieros.....	114



## LISTA DE SÍMBOLOS

Símbolo	Significado
$\sigma$	Desviación estándar
$e$	Error de muestreo aceptable
$Z$	Intervalo de confianza
$>$	Mayor que
$<$	Menor que
$\%$	Porcentaje
$\bar{x}$	Promedio de la muestra
$n$	Tamaño de la muestra
$N$	Tamaño de la población



## GLOSARIO

<b>Desarrollo de software</b>	Creación de programas de computadoras.
<b>En línea/On line</b>	Estado de conexión a una red de computadoras (generalmente, internet).
<b>Error</b>	Problema en un programa de computadora que produce resultados no deseados.
<b>Gestión de errores</b>	Tratamiento de los problemas que presenta un programa de computadora.
<b>Software</b>	Conjunto de aplicaciones o programas que realizan tareas dentro de un computador.
<b>Tecnologías de la información</b>	Servicios que abarcan el dominio de la información, empleando recursos tecnológicos.



## RESUMEN

El presente trabajo de graduación explora un conjunto de herramientas metodológicas que abordan el tratamiento de fallos en los programas de computadoras. La creación de aplicaciones informáticas lleva más de medio siglo en auge y es el mismo tiempo que las fallas en dichas aplicaciones tienen de existir al ser elaboradas por seres humanos.

Dado que no hay ninguna persona que desee utilizar un programa que no garantice un comportamiento planificado, es necesario abordar las posibles fallas de las aplicaciones previo a su liberación al cliente final. Muchos avances se han llevado a cabo, en términos del perfeccionamiento de las aplicaciones, desde el momento en que apareció el primer error de software, pero las fallas continúan siendo una parte ineludible dentro del desarrollo de programas de computadoras.

Por ello, se hace estrictamente necesario para las empresas que crean aplicaciones, implementar métodos que mitiguen al máximo las fallas. Los modelos de trabajo que propone el presente trabajo de investigación son una evolución de los modelos que se han empleado en las últimas décadas. Toda la experiencia ganada y las lecciones aprendidas en términos de calidad en la industria de desarrollo de software fortalecen los métodos que se examinan con este trabajo.

Los procesos de fabricación de diversas industrias han servido de modelo para la industria del software y cada vez más las aplicaciones se realizan bajo la idea de una línea de producción que debe ser precisa, automatizada, organizada, mejorada y optimizada. De esta manera, a la producción de software le aplican

prácticas de reducción y corrección de desperfectos, haciendo cada vez más efectivos los procesos de desarrollo. El enfoque principal de esta investigación se encuentra en los procesos de pruebas que se aplican al ciclo de vida de los proyectos de software. El modelo para pruebas de software ISO/IEC/IEEE 29119 así como el modelo de madurez de pruebas TMMI comparten características del ciclo PHVA, el cual busca la mejora continua de los procesos.

# 1. INTRODUCCIÓN

La calidad del software es un conjunto de cualidades que determinan su utilidad y existencia y es sinónimo de eficiencia, flexibilidad, corrección, fiabilidad y robustez. El usuario necesita que los productos sean funcionales, sin inconvenientes, tolerantes a fallos y seguros. El conjunto de actividades y tareas que resuelve un problema es trascendental y de este pueden depender vidas humanas. De igual manera, existen repercusiones económicas que supone un fallo tecnológico a lo interno y externo de la organización. Ejemplo de ello es el efecto 2000, que tuvo repercusiones económicas en todo el mundo.

Asimismo, a lo interno, se puede ahorrar dinero al evitar reproceso e invertirlo en incremento de valor del resultado. Para evitar la propagación de errores por detección tardía, los productores necesitan contar con estructuras basadas en normas industriales. El ciclo de vida de proyectos informáticos debe incorporar elementos que aborden las necesidades de producción. Es importante entender que, lo que no se invierte en calidad durante el ciclo de vida, representa costos adicionales en los procesos de puesta en marcha y mantenimiento.

Esta investigación se enfoca en una corporación financiera guatemalteca, que para el año 2020 se encuentra a la vanguardia en la prestación de servicios financieros en línea a nivel nacional. El problema que se busca solucionar en esta investigación se basa en proponer un modelo de mejora para aumentar la efectividad al tratar las fallas de software, incrementando así la calidad de los productos, la competitividad y reputación de la organización. Haciendo mejor uso de los recursos y aumentando el desempeño de los grupos operativos, se reducirán los costos de producción dando paso a la innovación y a la satisfacción

del cliente. La finalidad del presente trabajo de graduación es proponer un modelo de mejora continua para el área de pruebas que produzca resultados de interés a la empresa y la industria tecnológica. Al adoptar las buenas prácticas el departamento de TI se encaminará a perfeccionar la producción, el mantenimiento y la calidad.

Para la solución del problema planteado se propone desarrollar un instrumento que aumente la efectividad de la gestión de errores que realiza el área de pruebas del departamento de Tecnologías de la Información (TI). La propuesta empleará el Ciclo de Deming, la norma de pruebas de software ISO/IEC/IEEE 29119 y el Modelo de Madurez de Pruebas Integrado (TMMI, por sus siglas en inglés) para dotar el estudio con elementos que se pueden comparar e incorporar a los requisitos de producción de calidad.

Con la presente investigación se afrontará la necesidad de diseñar una metodología que aumentará la efectividad de los procesos y el rendimiento de los equipos de desarrollo y pruebas. Este estudio tratará sobre la calidad de los productos con el uso correcto de los recursos para evitar costos adicionales por reproceso. Asimismo, se enfocará el incremento de la satisfacción del cliente, la competitividad de la organización y su reputación en el mercado. Es importante demostrar con esta investigación que un modelo de mejora continua perfeccionará la gestión, afrontando las repercusiones económicas. Este estudio ayudará a garantizar la calidad y a elevar la imagen corporativa.

El presente trabajo de investigación es viable porque aborda necesidades puntuales dentro del departamento de TI, tiene la aceptación de la organización y es aplicable al ciclo de desarrollo de proyectos de software. Cuenta con los recursos humanos, físicos, financieros, materiales y la información para realizar el estudio y analizar los datos.

El resultado esperado es el modelo de mejora continua que defina los procesos, los recursos, la matriz de funciones y la estructura organizacional. Se crearán esquemas operativos de desarrollo y pruebas con instrumentos de verificación.

La propuesta será beneficiosa porque entregará un modelo capaz de perfeccionar la atención de fallas en los programas informáticos, la calidad de los productos y el desempeño de los equipos de desarrollo y pruebas. Ayudará en la detección de errores en etapas tempranas evitando su propagación a todo el ciclo de vida, reduciendo los tiempos de respuesta y los costos por reproceso. Los beneficiarios de la investigación serán los clientes porque obtendrán productos de calidad cumpliendo con el tiempo planificado. La organización se beneficiará con la rentabilidad de los proyectos. El personal de desarrollo y pruebas evitará reproceso e incrementará su desempeño. Por su parte, el investigador perfeccionará sus capacidades estratégicas y de gestión al aportar experiencia y conocimiento a la industria.

Como esquema de solución se propone trabajar bajo cinco fases: revisión documental de la teoría y bibliografía existente. Análisis de las actividades del departamento de TI que afectan la dirección. Elaboración del modelo de gestión del área que garantice la efectividad y reduzca los costos de desarrollo. Propuesta de evaluación y ajuste del proceso de pruebas que permita cumplir los objetivos y análisis comparativo de las metodologías de gestión para proyectar los beneficios del modelo propuesto. Dichas fases se encuentran relacionadas con los capítulos a exponer.

Con el fin de sustentar la investigación, en el primer capítulo se analizará la teoría sobre la industria tecnológica, así como el desarrollo y pruebas de aplicaciones informáticas. El segundo capítulo describirá los procesos previos a

la propuesta enlistando las actividades, procedimientos y prácticas que se llevan a cabo. Tratará sobre el nivel de productividad, servicio y cumplimiento de los objetivos, así como las oportunidades de mejora y los costos asociados. De esta forma se comprenderá la importancia de perfeccionar la gestión y de incrementar la calidad de los productos.

El tercer capítulo establecerá los elementos necesarios del modelo que garanticen la efectividad en la gestión y reduzcan los costos de proyectos. Aquí se definirán los procesos, los recursos y la estructura organizacional; de igual forma las funciones y planes operativos. En el cuarto capítulo se realizará una comparativa de la metodología tradicional contra la propuesta con el fin de determinar la efectividad en la gestión de errores y la reducción de los costos de producción con el nuevo régimen.

## 2. ANTECEDENTES

Mardomingo (2012) en su trabajo de investigación aplicó un modelo de gestión de calidad basado en el ciclo de mejora continua de Deming (PHVA) para el desarrollo de una oficina de control de los servicios de tecnologías de la información. Este estudio, que se realizó en el Centro de TI de Venezuela (CNTI), concluyó que el entorno organizacional estaba consciente de la existencia de mejores prácticas de TI y apoyaban su aplicación. Estas técnicas ayudan a la organización a adaptarse a procesos estandarizados alineados con el negocio permitiendo mejorar la calidad y los tiempos de respuesta de los servicios lo que aumenta la satisfacción del cliente. Por lo tanto, el presente estudio tomará en cuenta el rol del modelo de calidad en la dirección de un departamento de TI, con respecto de mejores prácticas industriales. Esto se hará con el fin de garantizar que sus productos y servicios satisfacen los requisitos de calidad de los clientes y cumplen todas las normas de referencia.

Van Veenendaal y Cannegieter (2013) analizaron cincuenta resultados derivados de la primera evaluación TMMI para obtener un índice de madurez de pruebas. Entre varias conclusiones a las que arribaron, se rescatan algunas de las más relevantes. El 14 % eran evaluaciones TMMI formales y el resto informal (poco rigurosas, más rápidas y baratas, pero faltas de precisión). El 84 % de las entidades estudiadas estaban aún en el primer nivel de madurez, un 10 % se encontraba un paso al frente y 6 % en el tercer plano. Ninguna de las compañías evaluadas llenaba los requisitos de las etapas 4 y 5. Las organizaciones en la fase inicial presentaban diferencias entre ellas: en algunas ocasiones las pruebas eran muy caóticas y sin un proceso definido, en tanto que otras estaban casi en el nivel 2. Inclusive en la etapa inicial, un proyecto podía ser exitoso; sin embargo,

esto se lograba gracias a la dedicación y esfuerzo de los “héroes de pruebas” y no por un proceso gestionado y respetable. Por lo tanto, la industria informática tiene aún muchos pasos que tomar hacia la mejora y madurez. En este sentido, la propuesta a realizar en el presente trabajo de investigación diseñará una metodología con el soporte de la norma de software ISO/IEC/IEEE 29119 y el modelo TMMI. La aplicación de sus criterios mejorará el proceso de pruebas e impactará la calidad, productividad y los tiempos de entrega.

Idrovo (2016) a través del método de encuestas al personal de la empresa, analizó la fase de pruebas de la organización con el fin de conocer los problemas y posibles mejoras a emplear. De igual forma encuestó a los clientes con el ánimo de entender su opinión sobre el producto que la organización les ofrecía. Su investigación descubrió que las principales causas de deficiencias del software eran la inexistencia de métodos de control e incumplimiento de tiempos internos. Falta de guías en tareas diarias, papeles y funciones confusas, poco análisis de riesgos y requisitos. Omisión de pasos, falta de documentación y métricas, nuevo personal mal entrenado o realizando el proceso de forma empírica y escasez de control. El efecto era negativo en el ciclo de vida del software y la calidad de este. Cada uno de los errores mencionados requería reproceso, lo cual elevaba los costos no planificados, tiempo y trabajo. Se necesitaba controlar la fase de pruebas que era un punto crítico del proceso.

El autor identificó la necesidad de abordar las deficiencias de software a través de procedimientos y normas de calidad que sirvan a la mejora continua del ciclo de vida. Por lo cual la presente investigación propondrá un manual basado en normas aprobadas por la industria que guíe la gestión, con mecanismos que permitan regular y controlar tanto el proceso como la calidad del software generado.

Herrera (2016) a través del método de encuestas al personal de la empresa, analizó el área de desarrollo informático y el proceso de control de pruebas y calidad. En su investigación determinó que el personal no tiene un sistema definido que incluya las etapas de planificación, análisis, diseño y ejecución de pruebas. Asimismo, parte del tiempo asignado a estas tareas se utilizó para finalizar el desarrollo de software e implicó postergar las mismas. También encontró falta de control y documentación de errores lo cual produjo reproceso en el desarrollo. A raíz de sus hallazgos el autor determinó que se puede poner en funcionamiento un ciclo de aseguramiento de la calidad y madurez para la mejora de procesos y productos. Por tanto, el modelo que se propondrá con la presente investigación tomará en cuenta la norma de pruebas de software ISO/IEC/IEEE 29119 y el modelo TMMI. Basándose en estas guías se establecerán políticas, procedimientos y métricas con el fin de mejorar la efectividad del proceso de control de pruebas y calidad de software. De esta manera, se obtendrán productos acordes con los requerimientos.

Mera-Paz (2016) con base en su experiencia laboral en empresas de desarrollo de software, docencia y otras, recopiló y seleccionó información para determinar el porqué de la importancia de implementar procesos de pruebas de software de calidad. Analizó literatura sobre esta temática en el contexto local, nacional y mundial. Estudió la información de importantes bases de datos y describió los conceptos relevantes, las propiedades, las metodologías y los marcos de referencia enfocados en la puesta en marcha correcta. En su investigación indicó que el accionamiento de un proceso de pruebas de software de valor en las organizaciones constituye un avance significativo. Esta activación garantiza márgenes de calidad en los programas informáticos y es un ingrediente fundamental para la imagen de las instituciones. El autor concluye que las empresas necesitan mejorar el rendimiento y la efectividad, así como optimizar los procesos de pruebas de calidad. Gracias a este aporte, para la empresa de

desarrollo de software bajo estudio, se diseñará un modelo basado en un ciclo de gestión de calidad que ayude a mejorar el rendimiento y optimice las etapas de los proyectos informáticos con prestancia.

Ponce, Domínguez, Gutiérrez y Escalona (2014), investigaron sobre las pruebas de aceptación orientadas al usuario, en el marco de un proyecto de innovación de gestión documental. Presentaron los contextos tecnológicos de referencia, una visión general de las metodologías ágiles y ofrecieron una revisión sobre las normas en la industria. En su investigación determinaron que las pruebas de usuario se suelen abordar en la etapa posterior a la codificación y esto produce un rastreo tardío, lo que añade costos al proyecto. Su solución es una ingeniería guiada por paradigmas ágiles para mejorar la realización de estas. Propusieron que las mismas se realicen de manera alterna con el desarrollo, que se generen desde los requisitos funcionales y por un equipo independiente, siguiendo las normas de referencia. Los costos de realización de pruebas se deben mantener dentro de los parámetros del proyecto, con el soporte de sistemas informáticos para automatizar el proceso, haciendo que dichos costos sean asumibles por el proyecto. Por lo tanto, en la presente investigación se diseñarán mecanismos con base en metodologías aprobadas por la industria. Se introducirán elementos sobre el proceso de pruebas de software de manera que el producto cumpla con las expectativas del cliente.

Fernández (2015), al investigar acerca de la aplicación de métodos de pruebas automáticas empleó metodologías y técnicas de referencia e ilustró su desarrollo a través del software VoDKA TV. Automatizó una serie de pruebas utilizando el lenguaje de programación funcional Erlang. En este diseño las condiciones usadas no dependían del bosquejo preciso de la aplicación. Concluyó que el uso de propiedades abstractas combinadas con el desarrollo dirigido por las pruebas apoya a idear las funciones del software con anticipación.

Esto permite codificar con mayor claridad y orden, dando lugar a mejores programas, fáciles de leer, modulares, estructurados, con soporte accesible y de baja complejidad. Determinó que las técnicas basadas en propiedades son eficaces y requieren menos líneas de código que las que especifican los casos uno a la vez y de forma manual. Estas permiten incrementar la eficiencia y eficacia en su construcción, mejorando el desempeño. Observó la necesidad de métodos concretos para situaciones puntuales. Que las pruebas basadas en propiedades abstractas no son una técnica de verificación formal, puesto que no comprueban todos los escenarios posibles y utilizan casos automáticos para comprobar las definiciones de entrada.

Esta investigación pone de manifiesto que no basta la aplicación de una única metodología o técnica de pruebas para resolver o prevenir los errores de software. Por lo tanto, con la propuesta de solución presente se abordará esta situación orientada a una combinación de diversos instrumentos dentro del proceso. Teniendo como objetivo la obtención de pruebas más efectivas, realizadas en menor tiempo, hallando más defectos en etapas tempranas del desarrollo, aumentando así, la prestancia de los programas.

El diseño de un modelo de gestión para el área de pruebas de la organización, basado en un ciclo de mejora continua de calidad, necesita considerar cada uno de los aspectos mencionados con anterioridad. Debe tomar en cuenta la norma ISO/IEC/IEEE 29119 y el modelo de madurez TMMI para mejorar los procesos de pruebas durante el ciclo de vida del software. Se espera que defina políticas, procedimientos y métricas; así como metodologías y técnicas de pruebas para garantizar una solución holística a la situación que se plantea.



### **3. PLANTEAMIENTO DEL PROBLEMA**

#### **3.1. Definición del problema**

Inefectividad en la gestión de errores de software del área de pruebas de una empresa de desarrollo de software, ocasionada por la falta de un proceso de gestión sistemático y auditable, lo cual impacta en los costos y tiempos de despliegue de proyectos en producción, así como en la satisfacción de los clientes por productos y servicios de mediana calidad.

#### **3.2. Descripción del problema**

Una empresa de desarrollo de software guatemalteca, que forma parte de una corporación financiera se enfoca en la industria Fintech. Para el año 2020 atiende a nueve empresas que integran el grupo financiero y hasta 400,000 clientes cada mes siendo reconocida como líder del sector a nivel nacional. Durante los más de 20 años que lleva de existencia, la empresa de desarrollo de software ha contribuido a la innovación de los productos y servicios de la organización financiera.

Con el cambio de milenio, la corporación expandió sus operaciones en la región a través de fusiones, absorciones y adquisiciones de los activos de otras entidades financieras. Esto ha aumentado y diversificado su portafolio de productos y servicios, así como su participación en espacios estratégicos. Con la expansión adquirió sistemas y procesos de tecnológicos y de conocimiento. Estas integraciones se llevaron a cabo bajo un enfoque económico, informático, cultural y de recursos humanos.

Los procesos de la organización financiera se apoyan en la tecnología, autoservicio, la atención personalizada y el diseño para asegurar la satisfacción de los clientes. En este sentido, las tareas informáticas son de vital importancia. A pesar de su compromiso con la calidad e innovación, la empresa de desarrollo de software presenta ineffectividad en la gestión de errores y su registro digital, así como en los tiempos de entrega. Necesita notificar con precisión a las áreas respectivas sobre la gestión evitando sobrecargar al personal con reproceso.

Asimismo, la definición de requerimientos de software presenta deficiencias dando lugar a la introducción de normas, controles y buenas prácticas. De igual manera, es posible optimizar la estructura organizacional, con el fin de hacerla más efectiva. En algunas ocasiones se han presentado retrasos en entregas, con tiempos que van desde días o semanas, hasta meses y en el peor escenario, años. Esto depende, en algunos casos, de la envergadura de los proyectos o del trabajo de terceros para su finalización.

A raíz de la situación planteada, la organización incurre en costos adicionales por reproceso, falta de personal que codifique soluciones, e insatisfacción del cliente. Se pierden oportunidades de nuevos negocios. El compromiso de la empresa con la calidad e innovación le permite realizar esfuerzos para mitigar cualquier problema de gestión en los procesos de todo el ciclo de vida poniendo en funcionamiento sistemas para aumentar el control y supervisión del desarrollo de proyectos. Estas herramientas permiten dar seguimiento, registrar los errores, detectar las prioridades y automatizar el despliegue. Sin embargo, las fallas de software continúan impactando en ambientes de producción.

La presente investigación planteará un modelo de mejora que apoye al área de pruebas de la empresa en la consecución de los objetivos estratégicos a

través de aumentar la efectividad de la gestión de errores de software. Asimismo, abordará la reducción de los costos de proyectos.

### **3.3. Formulación de preguntas**

A continuación, se describen las preguntas necesarias para la realización de esta investigación.

#### **3.3.1. Pregunta central**

- ¿Qué modelo aumenta la efectividad en la gestión de errores de la organización y reduce los costos de desarrollo de software?

#### **3.3.2. Preguntas auxiliares**

- ¿Cuáles son las actividades del área de pruebas que reducen la efectividad en la gestión e incrementan los costos de desarrollo de software?
- ¿Qué elementos son necesarios en el modelo para garantizar la efectividad en la gestión y reducir los costos de desarrollo de software?
- ¿Cómo se debe evaluar y ajustar el proceso de pruebas de software para garantizar el cumplimiento de los objetivos de la organización?
- ¿Cómo se reducen los errores y costos de desarrollo de software con el modelo diseñado?

### **3.4. Delimitación**

La investigación se llevará a cabo en una empresa de desarrollo de software, ubicada en Ciudad de Guatemala. Se realizará durante el período de diciembre del año 2020 a junio del año 2021. En ella se analizará el flujo de trabajo de las actividades de pruebas de software, los niveles del proceso, las prácticas, los documentos y las herramientas que se involucran en esta fase de los proyectos informáticos de dicha empresa.

### **3.5. Viabilidad**

La investigación que se propone está ambientada en el departamento de TI de una corporación financiera guatemalteca y cubrirá los retos que plantea el desarrollo de proyectos de software. Se enfocará en las necesidades de las etapas de pruebas, certificación e implantación de programas informáticos. Con la apertura de la organización hacia la calidad e innovación es posible realizar el estudio pertinente y analizar los datos que de este deriven. Por lo tanto, se cuenta con acceso a la información, los recursos humanos, financieros y materiales necesarios. La financiación del proyecto será asumida en su totalidad por el investigador. Por lo anterior, el presente trabajo es viable y puede ser llevado a buen término.

### **3.6. Consecuencias de la investigación**

Al concluir con éxito el presente trabajo de investigación, los beneficios recaerán en los socios inversionistas, la organización financiera y en sus clientes. El departamento de TI contará con un modelo para la mejora continua de la gestión capaz de aumentar las entregas en tiempo y calidad con ahorro de recursos monetarios y de trabajo. El cliente obtendrá productos cada vez más

acorde con sus expectativas, robustos y funcionales. El socio inversionista dispondrán de indicadores más acertados y precisos que le permitan tomar mejores decisiones de negocios futuros. Contar con un instrumento técnico que dirija y acompañe todo el ciclo de desarrollo impactará en la consecución de los objetivos del departamento de TI, mejorando los productos de software y los servicios informáticos que la empresa presta a sus clientes.

Por el contrario, en caso de que la investigación no pueda ser llevada a cabo, el socio inversionista perderá una oportunidad de contar con un instrumento capaz de mejorar el rendimiento de sus negocios por proyectos con mantenimientos no planificados y reproceso. Sin el presente estudio no se contará con el conocimiento que evite al cliente algunas frustraciones innecesarias por soluciones buenas, pero no excelentes. Por lo cual es importante afrontar la necesidad de mejora que cada vez entregue productos más precisos y acorde a los requerimientos con el fin de que el cliente pueda abordar los retos que le presenta su industria.

Los equipos de desarrollo y pruebas de la organización perderán oportunidades de aprendizaje y crecimiento. Sin una investigación como la que se propone, la empresa se verá afectada al no contar con un instrumento de conocimiento que evite incurrir en costos no planificados en desarrollo de software con afecciones a su rentabilidad. Asimismo, el investigador perderá la ocasión de contribuir con un instrumento para la competitividad de la organización e invertir los recursos en otros proyectos de mejora.



## 4. JUSTIFICACIÓN

El trabajo de investigación presentado se circunscribe en la línea de investigación de sistemas de control de calidad de la Maestría de Gestión Industrial, debido a que propone un modelo para el área de pruebas de una empresa de desarrollo de software que busca garantizar la efectividad de la gestión de errores a través un ciclo de mejora continua.

La principal necesidad que se abordará a través de la investigación es la de descubrir una nueva forma de gestión de errores del área de pruebas que carece de efectividad. Con la propuesta se pretende mejorar la calidad de los productos y la productividad de los equipos de trabajo. Asimismo, se cubrirán necesidades adicionales como la de aumentar la competitividad de la organización en el mercado mejorando así su reputación frente a los competidores, hacer un mejor uso de los recursos disponibles, incrementar la satisfacción del cliente y reducir los costos por reproceso, permitiéndole de esta manera incursionar en nuevos proyectos y continuar innovando en su industria.

La importancia de la investigación es que representa una oportunidad de solución a un problema presente en el giro diario de la organización, dado que las fallas en el software se traducen en reproceso en el ciclo de desarrollo, proyectos cancelados o entregados fuera de plazo, insatisfacción, pérdida de confianza y reputación frente a sus clientes, en los que el factor económico desempeña un papel fundamental. Asimismo, al ensayar la solución y alcanzar los objetivos propuestos, el investigador prueba su valía como un profesional en la industria tecnológica.

La motivación al realizar la investigación es que el área de pruebas de la organización plantea un problema que afecta las entregas de software en tiempo y calidad, adicionando costos no planificados. Por estas razones se busca diseñar una nueva forma de gestión basada en los conocimientos, metodologías y herramientas adquiridas en la Maestría de Gestión Industrial con el fin de realizar un aporte valioso a la organización y a la industria de desarrollo de software.

Los beneficios de la investigación son una herramienta de gestión para el área de pruebas que mejore la atención de errores, la calidad de los productos y la productividad de los equipos de desarrollo y pruebas. La contención de errores en la fase de pruebas evitando su propagación en fases posteriores del ciclo de vida del software, la reducción de los tiempos de respuesta y los costos por reproceso.

Los beneficiarios de la investigación son, principalmente, el cliente, al obtener productos en tiempo y de calidad. La organización, que obtiene una capacidad de gestión de errores más efectiva, para entregar los proyectos en el tiempo y bajo los requerimientos planteados, mejorando la utilización de los recursos tecnológicos y humanos disponibles, reduciendo los costos de desarrollo de software. El personal de desarrollo y pruebas, quienes simplifican sus operaciones al evitar reproceso, aumentando así la productividad y las posibilidades de compensaciones debido al aumento del rendimiento.

## **5. OBJETIVOS**

### **5.1. Objetivo general**

Diseñar un modelo de mejora basado en ISO/IEC/IEEE 29119 y pruebas TMMI para aumentar la efectividad en la gestión de errores y reducir los costos en una empresa de desarrollo de software.

### **5.2. Objetivos específicos**

- Identificar las actividades del área de pruebas que reducen la efectividad en la gestión e incrementan los costos de desarrollo de software.
- Establecer los elementos necesarios en el modelo para garantizar la efectividad en la gestión y reducir los costos de desarrollo de software.
- Establecer la metodología para evaluar y ajustar el proceso de pruebas de software para garantizar el cumplimiento de los objetivos de la organización.
- Determinar cómo se reducen los errores y costos de desarrollo de software con el modelo diseñado.



## **6. NECESIDADES POR CUBRIR Y ESQUEMA DE SOLUCIÓN**

La necesidad principal que se pretende solucionar en la organización es la de aumentar la efectividad en la gestión de errores del área de pruebas de software. Es necesario alinear el área y el proceso de pruebas con los objetivos de la organización a través de la creación de un modelo de mejora para garantizar la calidad y la reducción de los costos de producción.

Para obtener márgenes de calidad y mejora se propondrá un modelo basado en el ciclo de Deming de mejora continua de la calidad, la norma de pruebas de software ISO/IEC/IEEE 29119 y el modelo de madurez de pruebas TMMI. Asimismo, se establecerá una metodología para evaluar el cumplimiento de los objetivos de la organización y ajustar el proceso en función del modelo propuesto.

A continuación, se describe el esquema de solución del trabajo de investigación para brindar solución al problema:

- Revisión documental de la teoría y bibliografía existente.
- Análisis de las actividades del área de pruebas que reducen la efectividad en la gestión e incrementan los costos de desarrollo de software.
- Elaboración del modelo de gestión del área de pruebas, estableciendo los elementos necesarios con base en el ciclo de mejora continua de Deming, la norma de pruebas de software ISO/IEC/IEEE 29119 y el modelo de madurez de pruebas TMMI.

- Propuesta de evaluación y ajuste del proceso de pruebas de software para garantizar el cumplimiento de los objetivos de la organización.
- Preparación del informe final. Análisis comparativo de la metodología de gestión tradicional contra la metodología propuesta para proyectar los beneficios que obtiene la organización con el modelo diseñado.

## **7. MARCO TEÓRICO**

Para llevar a cabo el siguiente trabajo es indispensable analizar estudios previos vinculados al tema en cuestión, con el fin de crear una base contextual y teórica de sustento. Este conjunto de conocimientos demostrará los elementos singulares que el proyecto aporta a la organización en estudio.

### **7.1. Industria de desarrollo de software**

Los seres humanos siempre han necesitado formas de preservar datos para su posterior estudio, efectuando cálculos y analizando la información. El procesamiento automático resulta de muchos sucesos históricos acontecidos a lo largo de varios siglos. Dicha automatización se ha visto influenciada y dirigida por las necesidades humanas y el avance tecnológico.

#### **7.1.1. Historia**

El primer procesamiento automático de datos se registró en el año 1886 cuando Herman Hollerith diseñó una máquina que a través de una tarjeta perforada almacenaba el censo poblacional en Estados Unidos. En el año 1896 Hollerith fundó la empresa Tabulating Machine Cor., que más tarde cambió su nombre por la Computing Tabulating Recording en 1911 (mismo año en que se estableció la Powers Accounting Machine Cor., precursora de la Remington-Rand Corporation que tiempo después construiría el ordenador Univac). Luego, bajo el mando de Thomas John Watson cambió su nombre a International Business Machines Corporation (IBM), el cual le venía bien dada la expansión que experimentaba en todo el mundo (Menéndez-Barzanallana, 2013).

Los ordenadores de segunda generación hicieron su aparición muchos años más tarde, en 1958, sustituyendo los tubos de vacío de la versión anterior por transistores. El lenguaje de programación FORTRAN se creó en 1954 para IBM y más tarde aparecieron COBOL, ALGOL y LISP entre 1959 y 1962. La tercera generación de computadoras inició gracias a la introducción de los circuitos integrados a mediados de los años 60. En la época de 1955 a 1965, con el surgimiento de los lenguajes de programación (como producto) al igual que las empresas de desarrollo y distribución de software se pensaba que todo era posible de ser realizado por medio de una aplicación de computadora. Sin embargo, esto desencadenó la “crisis del software” marcada por el desarrollo de programas extensos que nunca veían fin, que no cumplían con los requerimientos, con una gran cantidad de errores, costos impredecibles e ineficacia en las operaciones (Menéndez-Barzanallana, 2014).

En los años 1967 y 1968 surge el término Ingeniería de Software para la creación y mantenimiento, con costos razonables y satisfacción de los usuarios; con un uso inteligente de técnicas, principios y herramientas. En la década de 1970 los sistemas informáticos se tornaron más complejos y emergieron las redes de computadoras. Se hicieron avances conceptuales para fortalecer los fundamentos de la programación, la verificación del software y las metodologías de diseño. Las estructuras de procesamiento de equipos y sistemas informáticos (con instrucciones para realizar tareas) se volvieron más complejas (Universidad Politécnica de Valencia [UPV], 2011).

La ingeniería de software abordó la crisis dando resultados positivos y para el año de 1991 los números mejoraban. Había menos proyectos abandonados, se redujo la cantidad de los que terminaban con modificaciones a los requerimientos iniciales y se elevó el porcentaje de éxito.

La industria informática aprendió lecciones fundamentales durante décadas. Para el año 2020 se han hecho avances importantes en cuanto a conocimiento, metodologías y tecnologías que atienden los problemas de la industria. Sin embargo, en algunos contextos los inconvenientes persisten. Tal es el caso de la empresa guatemalteca.

### **7.1.2. Ingeniería de software**

La Ingeniería de Software, según el Instituto de Ingeniería Eléctrica y Electrónica (IEEE, 1995), busca que el desarrollo se realice bajo un enfoque sistemático, disciplinado y cuantificable. Se trata de emplear la ciencia en los programas y estudiar su comportamiento desde distintas perspectivas.

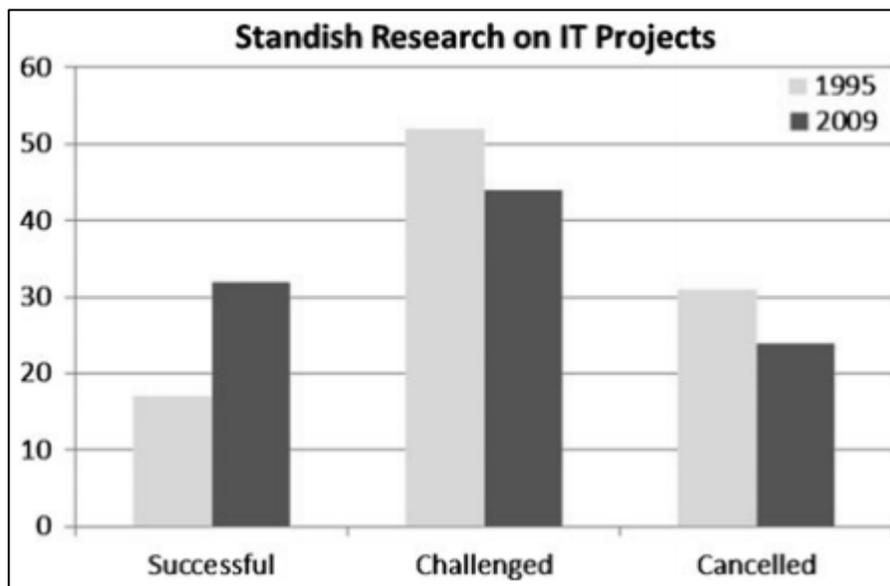
Esta disciplina va más allá de la codificación y las fases que componen el ciclo de vida de un producto de software. Debe ser aplicada por personas que han sido educadas como ingenieros y que cuentan con un conjunto de competencias científicas y de experiencia. Son capaces de crear modelos para abordar problemas de su industria empleando las mejores prácticas de manera profesional.

### **7.1.3. El valor de la ingeniería de software**

La ingeniería de software perfecciona las aplicaciones, cumpliendo con los requerimientos, el plan y el presupuesto asignado. En la gestión de proyectos es importante abordar el riesgo, la calidad, el tiempo, los recursos y los costos asociados. La administración deficiente impacta la organización, cuya percepción puede verse comprometida, llevándole a debilitar la relación con sus clientes y a la posterior pérdida de cuota de mercado. Es necesario establecer objetivos claros y realizar estimaciones basadas en métricas para cumplir con los

requerimientos, evaluando el desempeño y mejorando el proceso continuamente (O'Regan, 2017).

Figura 1. **Investigación sobre proyectos de Tecnologías de la Información (TI)**



Fuente: O'Regan. (2017). *Concise Guide to Software Engineering*. Consulta: 28 de agosto de 2020: Recuperado de <https://link.springer.com/book/10.1007/978-3-319-57750-0>

Esta investigación pone de manifiesto la relevancia de la ingeniería de software. Durante los años de 1995 a 2009, como sugiere el estudio, la cantidad de proyectos entregados con éxito aumentó, al mismo tiempo que disminuyó el número de los cancelados (figura 1).

#### **7.1.4. Procesos y ciclos de vida de software**

Los ciclos de vida de desarrollo de software más populares son:

- Ciclo de vida en cascada
- Ciclo de vida en espiral
- Proceso racional unificado
- Desarrollo ágil

Cada uno de los modelos aportan ventajas y desventajas a la gestión de proyectos de software y se emplean según la naturaleza de estos. Dependiendo de las necesidades que se presentan se puede elegir entre una opción y otra. Para tomar esta decisión es posible evaluar el conocimiento, las herramientas, el equipo de trabajo, el plan y demás artefactos con los que se cuenta para abordar el proyecto. Sin embargo, en las últimas décadas los ciclos de vida ágiles se han popularizado dado que aportan eficiencia y rapidez entregando pequeñas versiones funcionales. Responden a la incertidumbre y variabilidad minimizando los riesgos.

#### **7.1.5. Actividades en el ciclo de vida en cascada**

Las etapas relevantes del modelo de desarrollo en cascada, en términos generales son las siguientes.

- Definición de requerimientos de usuario
- Especificación de requerimientos del sistema
- Diseño
- Implementación
- Pruebas de software
  - Pruebas unitarias
  - Pruebas de integración
  - Pruebas de sistema
  - Pruebas de rendimiento

- Pruebas de aceptación del usuario
- Soporte y mantenimiento

Este modelo ha sido la base para algunas de las metodologías de desarrollo de software que se encuentran en uso, como por ejemplo la Rational Unified Process (RUP).

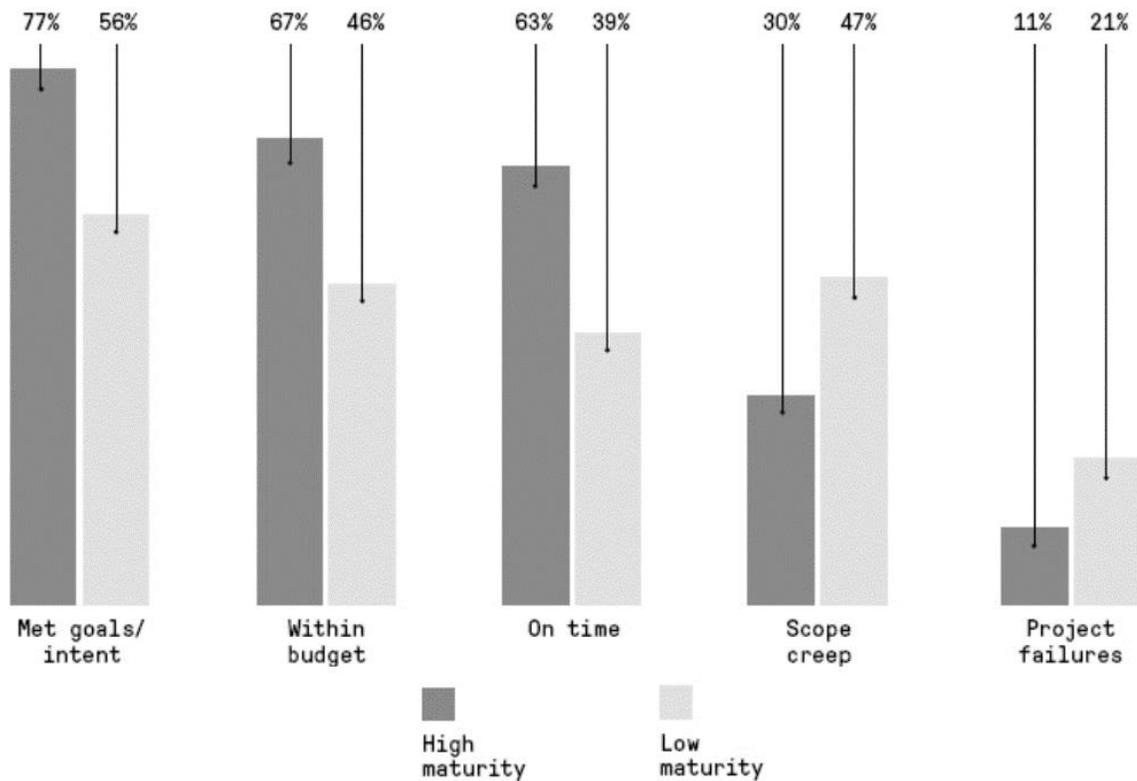
### **7.1.6. Gestión de proyectos de software**

Las entregas de software de calidad se logran a través de una buena gestión e ingeniería de procesos. Los directores de proyectos que se encargan de esta tarea desempeñan funciones importantes para alcanzar los objetivos trazados por la gerencia de Tecnologías de la Información (TI). Estos se aseguran de cumplir los requerimientos de costo, alcance y tiempo. Entre las funciones principales que desempeñan se pueden destacar las siguientes:

- Definición de proyectos
- Planeación y asignación de recursos
- Trazar los objetivos del proyecto
- Supervisar y controlar las actividades
- Buscar soluciones e introducir mejoras

Según el Project Management Institute (PMI, 2016), el 71 % de los proyectos que se realizan llena las expectativas de negocios al dar alto valor a la gestión. A diferencia de un 52 % cuando la prioridad es baja. Asimismo, al invertir en dirección, las entidades ahorrarán hasta 13 veces más dinero pues sus esfuerzos estratégicos tendrán más éxito (Pulse, 2020). El mismo estudio de Pulse muestra que al entregar valor, las entidades maduras superan al resto en puntos fundamentales (figura 2).

Figura 2. El retorno de la madurez



Fuente: PMI. (2016). *Ahead of the Curve*. Consulta: 3 de agosto de 2020. Recuperado de [https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2016.pdf?sc\\_lang\\_temp=es-ES](https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2016.pdf?sc_lang_temp=es-ES)

Los proyectos de las organizaciones que menosprecian la gestión como una competencia estratégica para maniobrar a través de la incertidumbre fallan hasta en un 67 % en promedio. Por lo tanto, se puede apreciar que la gestión de proyectos es relevante en la consecución de los objetivos de desarrollo de software.

### **7.1.7. Mejora de procesos de software**

Los procesos de alta calidad son fundamentales para la entrega de productos de valor. Por esto es necesario prestar especial atención a la forma en la que se desarrollan los productos. La calidad de las entregas mejora en la medida en la que las variaciones en el rendimiento del proceso se reducen. Estos argumentos fueron propuestos por Walter Shewhart, Edwards Deming y Joseph Juran y han revolucionado diversas industrias. La aplicación de estos principios en las organizaciones apoya a entregar productos de valor, reduciendo los costos de producción y aumentando la productividad, invirtiendo menor tiempo en reproceso. Como se estudiará más adelante en esta investigación, Humphreys, Snyder y Willis aplicaron los principios de Shewhart, Deming y Juran al desarrollo de software con el ánimo de incrementar la calidad de los productos informáticos (O'Regan, 2017).

### **7.1.8. Mercado de software en Guatemala**

La industria informática en Guatemala se encuentra en continuo desarrollo e incremento; cada día son más los servicios electrónicos que se ofrecen por internet y su calidad mejora con el tiempo. Día con día se suman a la ola tecnológica nuevos sectores e industrias y se alcanzan mercados y clientes que en otro momento eran imposibles de acceder, sin embargo, las bondades de los sistemas tecnológicos le han acercado y puesto a disposición. Entonces, ¿cuál es la naturaleza de la informática en el contexto guatemalteco?

#### **7.1.8.1. Historia**

El desarrollo de software evoluciona en el país desde hace más de medio siglo. Las computadoras en su forma más rudimentaria existieron en Guatemala

a partir de los años 30 y los primeros trabajos de programación se realizaron en la dirección General de Estadística. Pero no fue sino hasta finales de los años 50 y principios de los 60 cuando se conocieron en el país los ordenadores de segunda generación, basados en la arquitectura de von Neumann. Estas sustituían los tubos de vacío por transistores (Menéndez-Barzanallana, 2013).

A principios de los años 60 FIUSAC adquirió la computadora IBM 1620 para la enseñanza de métodos numéricos la cual operaba en Fortran II (De León, 1993). De igual manera, el Ministerio de Finanzas adquirió en el año 60 una computadora IBM en la que trabajó quien se considera como el primer programador guatemalteco, José Ordóñez, para el centro de cálculo de esa entidad. A principios de los años 70 la Universidad de San Carlos de Guatemala (USAC) inició a ofrecer la primera carrera de Ingeniería en Sistemas para impulsar el uso de equipos de cómputo en el país. A mediados de los años 70 el Centro de Cálculo de la Facultad de Ingeniería contó por primera vez con un ordenador para apoyar las actividades administrativas de la USAC. Asimismo, se hizo de la oportunidad de servir como un instrumento educativo y de investigación (Facultad de Ingeniería de la Universidad de San Carlos de Guatemala [FIUSAC], 2020).

En los años 70 las primeras universidades del país comenzaban a poner en funcionamiento estudios superiores en relación con la ingeniería e informática. La Facultad de Ciencias y Humanidades de la universidad del Valle creaba en el 76 la carrera de Computación. Asimismo, en el año 77 estableció el instituto de investigación. Aunque no fue sino hasta el año 2004 que inicia labores la Facultad de ingeniería (Universidad del Valle de Guatemala [UVG], 2020). Por su parte la Universidad Francisco Marroquín, en el año 77 puso en funcionamiento el proyecto de creación de una escuela universitaria en Ciencias de la computación, denominado Instituto de Informática y Ciencias de la Computación.

En enero del 78 la escuela comenzó a inscribir a las primeras personas en la carrera de ingeniería de sistemas, informática y ciencias de la computación. En el mes de julio de ese mismo año comienza a funcionar la carrera técnica de Programador y Analista de Sistemas con 100 estudiantes. El 30 de julio del 82 se publicó en el Diario Oficial el decreto mediante el cual se autorizaba el funcionamiento del Instituto de Informática y Ciencias de la Computación (IICC) como Facultad de Ingeniería de Sistemas, Informática y Ciencias de la Computación (FISICC). Más tarde se transformaría en la Universidad Galileo (Galileo, 2020). Por su parte, la universidad Rafael Landívar establecía su Centro de Cómputo en el año 81 (Medrano, 2001).

En los años 80 surgen los primeros creadores de aplicaciones a la medida para empresas que buscaban automatizar sus procesos industriales. Con esto conseguían agilizar y hacer más eficientes sus operaciones. Para este entonces aún no se tenía un pensamiento de crear programas normalizados que fuesen requeridos por diferentes organizaciones (Revista Investigación Científica Pensamiento del Sur [Eumednet], 2020). Fue en este periodo que se introdujeron al país los primeros cajeros automáticos (ATM) los cuales demandaban mantenimiento de software. Para finales de la década de los 80 y principios de los años 90 empresas como GBM, Byte, Ingesis y otras comenzaban a ofrecer servicios de desarrollo e ingeniería de software (Velásquez, 2010; Global Business Machines [GBM], 2020; Ingesis, 2020). El sector financiero y la banca era el que más empleaba sus servicios, aunque aún no se tenían conexiones a internet, pero si redes de intranet y las que mantenían en funcionamiento a los ATM.

En 1991 comienza a funcionar la primera conexión de internet en Guatemala, a través del Centro de Estudios en Informática Aplicada (CEIA) la cual evolucionó a MayaNet en el 95, cuando un acuerdo entre el Consejo

Nacional de Ciencia y Tecnología (CONCYT) y la Empresa Guatemalteca de Telecomunicaciones (GUATEL) permitió ofrecer todos los servicios de internet en todo el territorio nacional (UVG, 2020). Cybernet de Guatemala fue el primer Proveedor de Servicios de Internet (ISP) al cual le siguieron TikalNet, Infovia, Prodata, Corpotel y GBM (Country Code Top-Level Domain [ccTLD], 2020).

Para finales de los años 90 las empresas grandes y el sector financiero y la banca eran quienes más demandaban servicios de internet, aunque la asistencia que brindaban a sus clientes en línea era básica. Con la introducción de las conexiones por fibra óptica en el 98 se mejora en rapidez, confidencialidad, ahorro de costos y mayor desarrollo de aplicaciones para ofrecer servicios por internet (Lewis, 2001; Velásquez, 2010).

En el año 99 FIUSAC puso en funcionamiento el laboratorio de computación para las carreras de Ciencias de Ingeniería y Cursos Profesionales. Al mismo tiempo instaló la red de ingeniería que comunicaba a lo interno las escuelas, los centros, las coordinaciones y las unidades ejecutoras, así como con internet.

En los años 2000 nuevas empresas de servicios de desarrollo e ingeniería de software comenzaron a emerger, entre ellas Asesores en Informática (ASEINFO), Business Development Group (BDG), Canella, Gyssa, Icon, Blanco Silva, Coinsa Software Consulting, Grupo Segá y muchas otras. Los conocimientos y las herramientas de software continuaron evolucionando y apoyaron en el aumento de la productividad de diversas industrias (Chacón, 2007).

De acuerdo con Eumednet (2005) un grupo de empresarios de la industria tecnológica se agremian y conforman la Comisión de Software de Exportación (SOFEX), adscrita a la Asociación Guatemalteca de Exportadores (AGEXPORT).

Gracias a la creciente demanda de servicios electrónicos y canales de comunicación con el cliente final, el desarrollo de software ha evolucionado a lo largo de las últimas dos décadas. Y la evolución de la tecnología es quizás, la que marca el rumbo y acelera la demanda de nuevas formas de prestación de servicios en todas las esferas industriales. En la última década se atestiguó la llegada de la industria 4.0. Se observó el progreso de la tecnología celular, conexiones a internet de alta velocidad y mejoras en las tarifas de datos. Los servicios por en línea aumentaron, así como las tecnologías basadas en inteligencia artificial y el manejo de grandes volúmenes de datos. Estos y otros muchos avances tecnológicos, han permitido el crecimiento y mejora de la oferta de servicios de desarrollo e ingeniería de software. En el año 2014 el sector TI y Business Process Outsourcing (BPO) de AGEXPORT, integrado por empresas de desarrollo de software, creación de contenidos digitales, centros de contacto y tercerización de procesos de negocio cerraba el año con un crecimiento del 15%, posicionando al país como mercado emergente en la industria de tercerización de servicios (Vides, 2015).

Los centros de capacitación públicos y privados como, por ejemplo, el Instituto Técnico de Capacitación y Productividad (INTECAP) han aportado competitividad y productividad al sector. De igual manera lo han hecho las carreras de nivel medio y universitarias (algunas con posgrados o proyectos de emprendimiento). La cooperación técnica y financiera de países como Taiwán, India, Japón, Estados Unidos, La Unión Europea han hecho aportes importantes. Asimismo, los países amigos de América Latina han jugado un papel fundamental.

Existen proyectos tecnológicos como Campus Tec, SPLASH, AJE Guatemala, ENACTUS y Guatemala Emprende, entre otros. Se encuentran iniciativas del gobierno provenientes del CONCYT y el Ministerio de Economía

(MINECO) como el Programa Nacional de Competitividad (PRONACOM) o Emprende GT. De igual manera, en el año 2019 nace la iniciativa Asociación Fintech Guatemala agremiando a las organizaciones, a los empresarios y los emprendedores con interés en la innovación financiera. Un año más tarde la Superintendencia de Bancos de Guatemala (SIB) lanza el proyecto SIB Innovation Hub para regular el sector Fintech (Flores, 2016; Aspen, 2015; Ministerio de Economía [MINECO], 2018).

Como se puede apreciar, la historia del desarrollo e ingeniería de software en Guatemala lleva más de medio siglo y continúa en auge. Esta es una pequeña reseña histórica y no pretende ser un estudio exhaustivo de la historia tecnológica del país. Con ella se busca contextualizar al lector sobre la evolución de los servicios de desarrollo de software en Guatemala.

#### **7.1.8.2. Geografía**

El presente estudio se enfoca en una empresa de desarrollo de software del sector financiero en la Ciudad de Guatemala. Sin embargo, cabe resaltar que departamentos como Sacatepéquez, Chimaltenango, Quetzaltenango, Retalhuleu, Alta Verapaz y Escuintla también aportan talento al sector. Otros departamentos lo hacen en menor medida (Vides, 2019).

#### **7.1.8.3. Economía**

Guatemala es un país agricultor. La agricultura representa un cuarto del Producto Interno Bruto (PIB), dos tercios de las exportaciones y la mitad de la fuerza laboral. Otros sectores relevantes son el manufacturero, minero y constructor, así como el envío de fondos extranjeros y el turismo. En la última década la industria de servicios crece en importancia.

Gustavo Tasner, director de BPO Center de Capgemini Guatemala en el año 2015 afirmaba que las condiciones de infraestructura, productividad y competitividad económica eran buenas y estables. Las condiciones de tipo de cambio, la inflación, un desempleo razonable (no muy bajo ni muy alto), la mano de obra tecnificada, así como una cultura de servicio nata eran aceptables para su negocio (ConnectAmericas, 2015).

En el 2017 la exportación al extranjero de desarrollo y promoción de contenidos de software sumó \$400 millones (Central America Data, 2018). De igual forma en el año 2018 las ventas del sector de desarrollo digital sumaron \$37 millones, 4 % más comparado con el año 2017 (Central America Data, 2019). Asimismo, la calidad del internet continúa mejorando. Para el año 2018 Guatemala se posicionaba entre los países de la región con mayor cantidad de usuarios de internet (7.2 millones) y 6.8 millones de usuarios de internet móvil, con las tarifas de los planes móviles como principal barrera para su acceso (Jiguan, 2018). Durante el primer trimestre del año 2019 la industria digital exportó \$38 millones empatando en número de divisas de cierre del 2018. Asimismo, el sector de tecnología generaba alrededor de 3,500 empleos directos y 27,000 indirectos (Vásquez, 2019).

Mucho se puede decir sobre la economía guatemalteca en sus diferentes sectores, pero cabe resaltar que los servicios de desarrollo e ingeniería de software continúan en auge para el año 2020. La crisis generada por la pandemia de Coronavirus Disease 19 (Covid-19) en el mundo y que afectó al país desde el 13 de marzo de 2020, demandó mejoras de tecnología para atender a los diversos sectores de que depende la economía. Si bien otras industrias han sufrido un impacto negativo dada la crisis, el sector tecnológico se ha visto beneficiado gracias al acceso remoto y la automatización.

#### **7.1.8.4. Política**

En el año 2011, Edgar Santos, el entonces presidente de SOFEX, hacía hincapié en la necesidad y desafío del país de impulsar e incentivar, por medio de políticas públicas, el crecimiento económico. Santos proponía emplear para este fin el sector de TI. Ponía de manifiesto la necesidad de una visión integrada de largo plazo de estado. En aquel entonces, al comparar Guatemala con otros países de la región en términos de políticas públicas orientadas a la conectividad y TI, el país se encontraba en el puesto 11 (Asociación de Exportadores de Guatemala [Agexport], 2011).

Una política es un plan para alcanzar metas de interés público. Existen algunas que consiguen sus objetivos por medio del servicio público y requieren recursos económicos y de personal. Hay otras que lo hacen a través de la regulación y el sistema tributario; estas emplean menos recursos. Una política pública involucra a los partidos políticos, a la administración pública, a los actores productivos y sociales. Parte fundamental del quehacer del gobierno debería ser el diseño, gestión y evaluación de las políticas públicas. Estas representan soluciones puntuales para manejar asuntos públicos y derivan de la agenda pública. En este sentido, el gobierno de Guatemala no se ha caracterizado como uno que dé continuidad e innove en esta materia. Esto se debe a que existen alianzas político-partidistas débiles. Asimismo, no se cumplen las agendas, la planificación y los acuerdos políticos de interés nacional. Los servidores públicos son incapaces de alcanzar las metas de carácter económico, social e institucional. Su gestión es ineficaz y burocrática, con falta de probidad en su accionar, poca transparencia y rendición de cuentas, incumplimiento de la ley y altos índices de corrupción (Comisión Económica para América Latina y el Caribe [Cepal], 2011).

#### **7.1.8.5. Marco jurídico**

La falta de acuerdos políticos deriva en la falta de leyes y regulaciones en aspectos diversos que impiden el avance certero de la economía del país. Durante las últimas dos décadas se han llevado a cabo diversas iniciativas con el fin de modificar el marco legal para mejorar el clima de negocios en términos de desempeño y productividad. Los grandes actores económicos como los sectores financieros y de telecomunicaciones son los que generalmente realizan esfuerzos para mejorar las condiciones de su industria en materia jurídica. Pero un marco legal que garantice el crecimiento integral de la sociedad debe involucrar y beneficiar a todos los actores de dicha sociedad y no solamente a unos cuantos. La falta de certeza jurídica limita la inversión privada local y extranjera pues esta demanda la existencia de reglas claras y la garantía de que serán cumplidas y que los órganos jurisdiccionales las harán respetar. El imperio de la ley y la soberanía del derecho debe imperar sobre cualquier otro poder, como la impunidad, la corrupción, el dinero y el tráfico de influencias. (La Hora, 2019)

Por otro lado, los productores y las agrupaciones civiles, en ocasiones apoyados por entes del gobierno, constantemente impulsan programas y proyectos de desarrollo. Estos proponen su adopción por parte del gobierno y los partidos políticos en su agenda pública con el fin de generar leyes y reglamentos que integren a los productores y garanticen el crecimiento económico. Un ejemplo de este esfuerzo es el programa PRONACOM, cuyo plan busca hacer de Guatemala un país más competitivo para el 2032 (La Prensa, 2019). Asimismo, la SIB ha impulsado el SIB Innovation Hub que busca regular al sector Fintech encausando así la competitividad y productividad financiera del país con base en la tecnología y los servicios electrónicos (Superintendencia de Bancos de Guatemala [SIB], 2020).

### **7.1.9. Empresa de desarrollo de software analizada**

La organización bajo estudio está enfocada en el desarrollo de software a la medida para el sector Fintech en Guatemala, puntualmente, para una corporación financiera local. A continuación, se estudiarán sus características.

## **7.2. Calidad y mejora continua**

No se puede hablar de mejora continua sin antes definir calidad pues ambos conceptos guardan una estrecha relación. La mejora continua tiene como fin ulterior la calidad. Pero ¿de qué se trata?

### **7.2.1. Calidad**

La Real Academia Española (RAE, 2020) define calidad como el conjunto de propiedades esenciales a algo que ayuda a determinar su valor. Le asigna los atributos de superioridad y excelencia. La calidad ha evolucionado desde que hiciera su aparición el concepto de administración científica en 1911, de la mano de F. W. Taylor. Para aumentar la productividad, Taylor proponía que la planeación y ejecución del trabajo que se realizaba en los sistemas de producción en serie debían separarse. Si bien, mucho antes a este período, el valor en los productos y procesos comerciales era importante, no fue sino hasta ese momento que se comenzó a teorizar y conceptualizar la calidad como se le conoce en nuestros días. La combinación de los métodos de trabajo ha dado lugar a normalizar las tareas y procesos. A su vez, la estandarización es la base de la mejora continua (Cubillos y Rozo, 2009).

### **7.2.2. Herramientas**

Cuando se desea implementar controles y mejoras de calidad, se puede abordar un problema como un proyecto a través de métodos y técnicas de análisis y resolución. Sin embargo, los pasos principales para solventar un problema son:

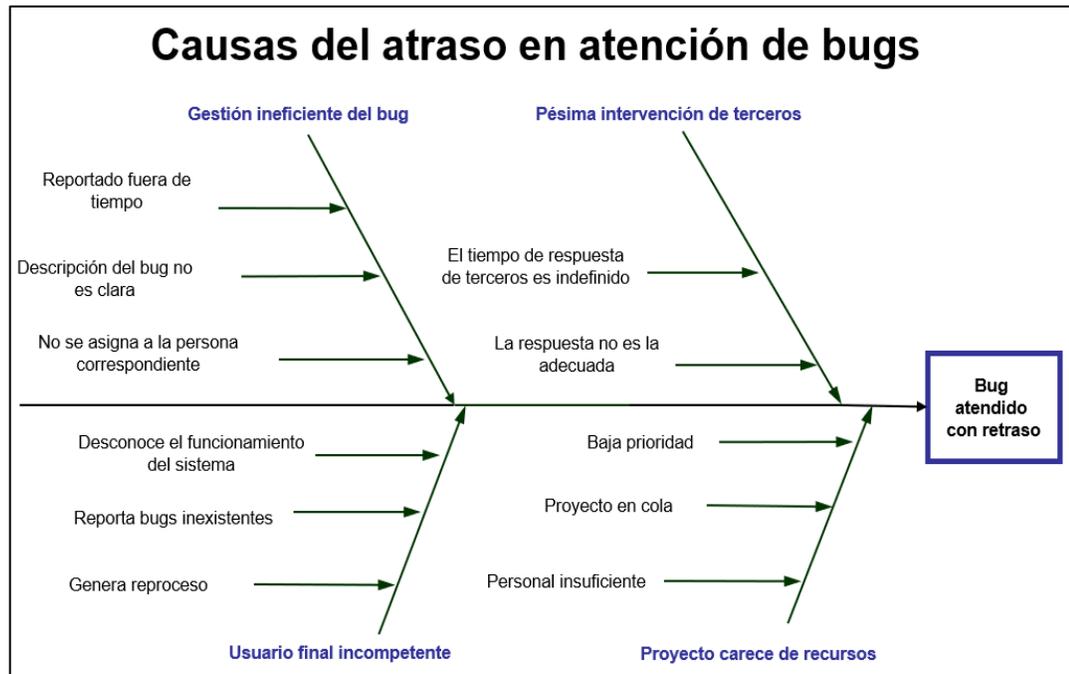
- Definición del problema
- Representación del problema
- Diseñar la solución
- Implementar la solución
- Verificar los resultados y sacar conclusiones

En cada una de las etapas mencionadas, se puede utilizar una herramienta de calidad con el fin de ejecutar las tareas y lograr los objetivos (Cuatrecasas & González, 2017). Entre las herramientas conocidas para estudiar, medir y aumentar la calidad se encuentran las siguientes:

- Diagrama de causa – efecto

Es una herramienta para el análisis de un problema y su solución (figura 3).

Figura 3. Diagrama de Causa-Efecto

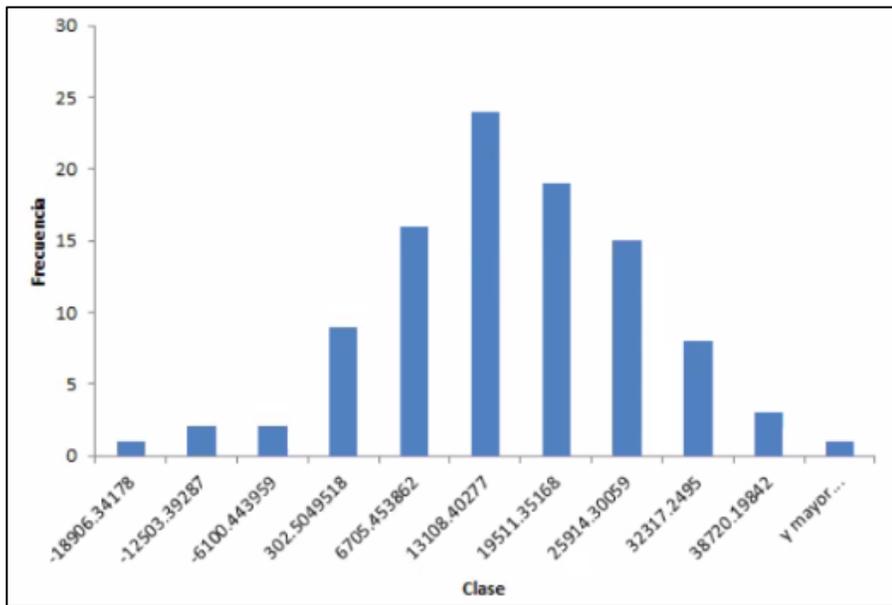


Fuente: elaboración: propia.

- Lluvia de ideas: es una técnica grupal que facilita la generación de nuevas ideas.
- Diagrama de árbol: es una técnica que define un problema y busca una solución analizando las causas y los efectos de este.
- Hoja de verificación: es un instrumento de recolección de datos para su posterior análisis.
- Gráficos de control: es una herramienta que permite describir el comportamiento de un proceso en orden cronológico para detectar variaciones.

- Histograma: es un gráfico que representa datos numéricos continuos y su distribución de frecuencias (figura 4).

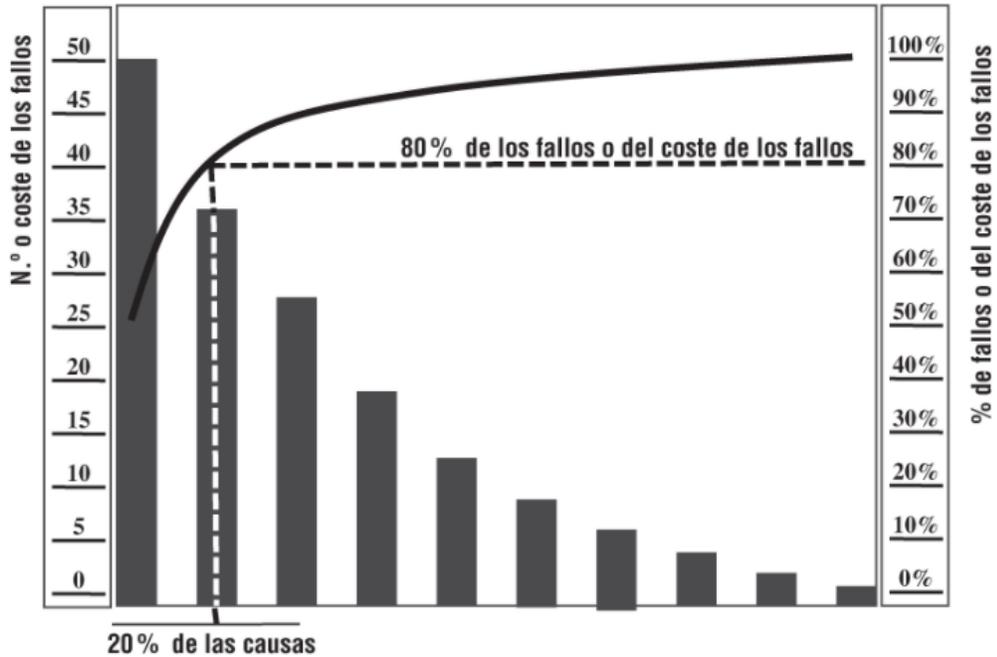
Figura 4. **Histograma**



Fuente de elaboración: propia.

- Diagrama de Pareto: basado en el principio de Pareto, es un gráfico que muestra el porcentaje de defectos que afectan un proceso (figura 5).

Figura 5. **Diagrama de Pareto**



Fuente: Cuatrecasas y González. (2017). *Gestión Integral de la Calidad*. Consulta: 3 de agosto de 2020. Recuperado de <http://reader.digitalbooks.pro/book/preview/101085/cover?1603912353967>

- Diagrama de dispersión: es un gráfico que permite determinar la relación entre dos variables numéricas.

### 7.2.3. **Medición de la calidad**

Las herramientas que se mencionaron en el apartado previo permiten registrar, analizar e interpretar datos sobre la gestión. Pero si se quiere determinar la calidad de los productos y procesos hace falta algo más que solo las herramientas de estudio de datos. Es necesario trazar planes con objetivos

de calidad. Los pasos propuestos en la resolución de problemas son una buena guía que seguir a la hora de trazar un plan de gestión de la calidad.

Cuando ya se ha detectado y representado el problema, diseñado y echado a andar la solución, corresponde evaluar los resultados. Pero no sirve de mucha ayuda obtener las salidas del proceso y encajonarlas. Se hace necesario verificar, validar y extraer información relevante a la gestión, con el fin de detectar las fallas en el proceso e introducir cambios. Para realizar las tareas de verificación hay que definir indicadores de calidad. Los indicadores son datos que permiten valorar la magnitud de las propiedades de un suceso. Por lo cual, en el control de calidad se vuelven de vital importancia pues apoyan en la medición del grado de precisión y perfección de los atributos de los procesos y productos (Escuela de Administración de Empresas [EAE], 2020).

En un proyecto de calidad interesará medir variables que sean relevantes para la mejora de los procesos y productos. Algunas de estas variables son:

- Satisfacción del cliente: para estudiar las necesidades del cliente y sus expectativas.
- Demanda: permite realizar planes de producción, distribución y ventas.
- Ventas: apoyan en conocer los ingresos y utilidades.
- Competencia: los competidores afectan las ventas y restan cuota de mercado
- Garantía: se trata del cumplimiento de los compromisos ante el cliente.

- Capacidad de respuesta: trata sobre la inmediatez y efectividad para atender al cliente.
- Accesibilidad: habla sobre la facilidad con que se accede a un producto o servicio que ofrece la empresa.

Luego de haber definido los atributos de estudio, es necesario asignar medidores para conocer su comportamiento. Aquí es donde los indicadores hacen su aparición. Estos se crean en función de las variables establecidas con anterioridad. Cada una de ellas puede tener uno o más indicadores para evaluar su comportamiento. Estos serán de carácter cualitativo o cuantitativo. Algunos ejemplos de indicadores, según las variables listadas con anterioridad serían:

- Volumen de ventas: es un indicador de carácter cuantitativo continuo.
- Efectividad: puede ser tratado como un indicador cuantitativo o cualitativo.
- Nivel de satisfacción: puede ser tratado como un indicador cuantitativo o cualitativo.
- Nivel de demanda: es un indicador de carácter cuantitativo continuo.
- Grado de accesibilidad: puede ser tratado como un indicador cuantitativo o cualitativo.
- Nivel de cobertura: es un indicador de carácter cuantitativo continuo para medir la disponibilidad de productos y servicios.

#### **7.2.4. Modelo de mejora continua de Deming**

Hasta este momento se ha mencionado un poco de historia sobre calidad según se conoce en nuestros días y de cómo esta es la base para la mejora continua. Se ha mencionado a Frederick Taylor como el precursor de la teoría de administración científica. Asimismo, se mencionó a Walter Shewhart como el padre del control estadístico de procesos (1924) y del control de calidad (1939). Fue en la década de 1930 a 1940 cuando W. Shewhart trata por primera vez el papel de la calidad en la administración. De sus estudios en este periodo nace la serie Planear, Hacer, Verificar y Actuar (PHVA).

Empleando el trabajo de Shewhart, Edwards Deming ayudó a perfeccionar la industria militar de los Estados Unidos, durante la última etapa de la segunda guerra mundial. Años después de finalizado el conflicto bélico, en 1950, el profesor comenzó a impartir a los japoneses lo aprendido sobre la calidad. Tanto a Deming como a J. Juran y A. V. Feigenbaum se les atribuye el despertar de la calidad en el país nipón y que el Doctor Kaoru Ishikawa ayudó a consolidar. El crecimiento económico y productivo japonés es un claro ejemplo de lo que un cambio en la administración, con fuertes bases en la ciencia e ingeniería es capaz de lograr (Cubillos y Rozo, 2009).

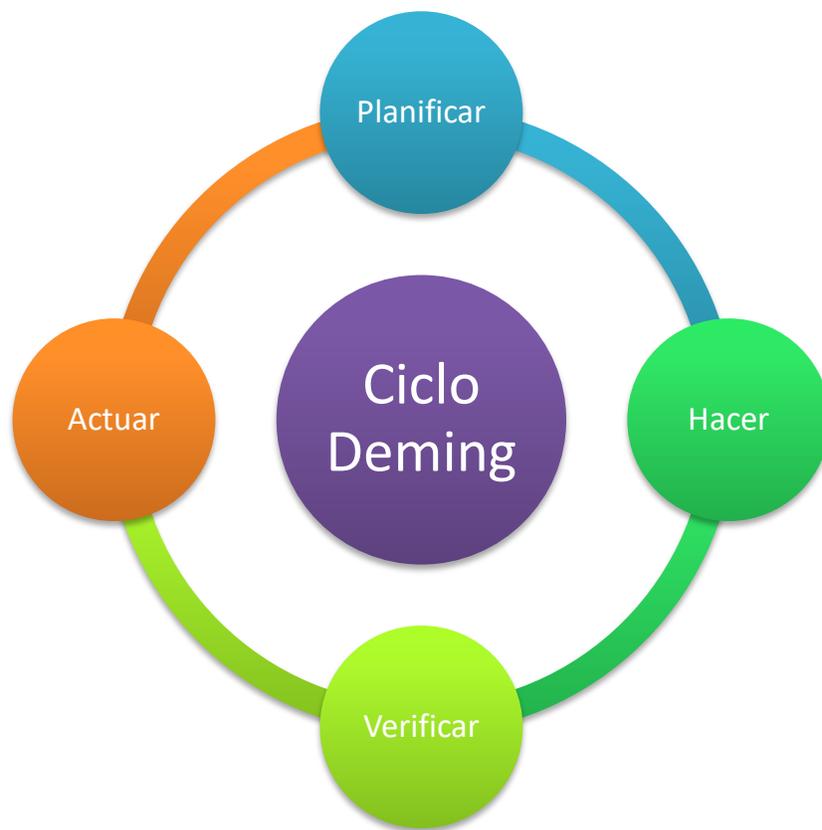
Como ya se ha estudiado, el ciclo PHVA es, en buena medida, el responsable del aumento de la calidad en la industria. El Ciclo de Deming o Ciclo de Shewhart como también se le conoce, es un modelo de mejora continua de procesos, como se puede apreciar en la figura 6 (Carro y González, 2012). Los cuatro pasos de la espiral son los siguientes.

- Planificar
- Hacer

- Verificar
- Actuar

A continuación, se estudiará cada una de las fases.

Figura 6. **Espiral de mejora continua**



Fuente: elaboración propia.

### 7.2.4.1. Planificar

En la primera etapa se define el problema, se determina su importancia y se trazan instrumentos de control. Luego se recopilan datos, se identifican variables de relevancia y se crean hojas de verificación. A continuación, se analizan las posibles causas del problema, se esbozan ideas de solución tomando en cuenta el conocimiento y experiencia que se posee. Para finalizar, se proponen soluciones, el plan de operaciones con un listado de prioridades (figura 7).

Las herramientas que se pueden emplear en esta etapa son:

Figura 7. **Herramientas de la fase “Planificar”**



Fuente: elaboración propia.

#### 7.2.4.2. Hacer

Luego de definir la solución es necesario implementarla. Y para ello se deben aplicar las modificaciones que se trazaron con el plan, en la fase anterior (figura 8).

Las herramientas propuestas para esta fase son:

Figura 8. **Herramientas de la fase “Hacer”**



Fuente: elaboración propia.

#### 7.2.4.3. Verificar

Después de implementar la solución se debe medir lo ganado y normalizar las mejoras. Para ello se recogen datos y se evalúan los resultados. También se

efectúan cambios, capacitaciones y se redefinen las obligaciones, las actividades y las instrucciones (figura 9).

En esta etapa se pueden utilizar las herramientas siguientes:

Figura 9. **Herramientas de la fase “Verificar”**



Fuente: elaboración propia.

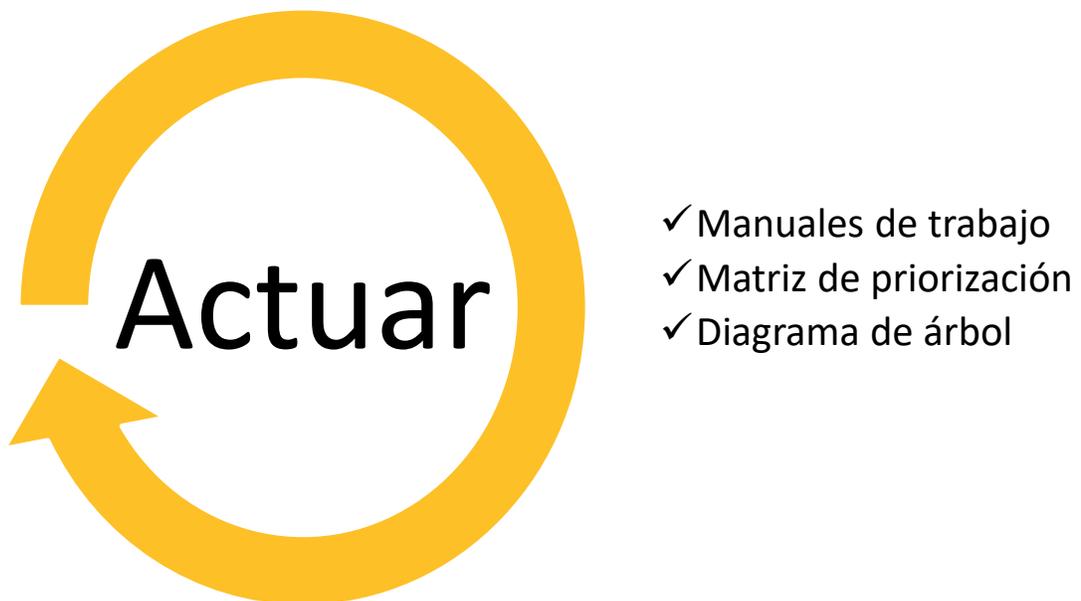
#### **7.2.4.4. Actuar**

Cuando se han verificado los resultados es momento de sintetizar y documentar el aprendizaje. Aquí se pueden listar las acciones que se realizaron y las salidas obtenidas, las lecciones aprendidas y las recomendaciones para

futuras iteraciones. A esta fase también se le denomina “Ajustar” pues es el paso de cierre y reinicio del ciclo (figura 10).

Las herramientas que se utilizarán en la fase final son:

Figura 10. **Herramientas de la fase “Actuar”**



Fuente: elaboración propia.

#### **7.2.4.5. Implementación del modelo de mejora continua**

Habiendo identificado un problema administrativo, se plantea una solución a través del ciclo de mejora continua. Definiendo el plan, este se aplica a la necesidad para conducir el sistema a un punto aceptable de calidad y a partir de ahí continuar ajustándolo.

La espiral de Deming se ha llevado a la práctica en diferentes industrias durante muchos años, probando su valía. En cuanto a la informática, el enfoque de control de calidad ha significado aportes fundamentales en la búsqueda de la mejora de los procesos de pruebas y los productos de software. Un ejemplo puntual de la implementación del ciclo de Deming se encuentra en los laboratorios de IBM en el año 1991, en el trabajo de Humphreys, Snyder y Willis. Ellos propusieron mejoras al proceso de pruebas a través de implementar la espiral de Deming (Escobar, 2014). En las imágenes siguientes se aprecian algunos ejemplos de cómo IBM emplea el modelo de mejora continua en sus creaciones.

Figura 11. **El Ciclo de Vida Analítico**

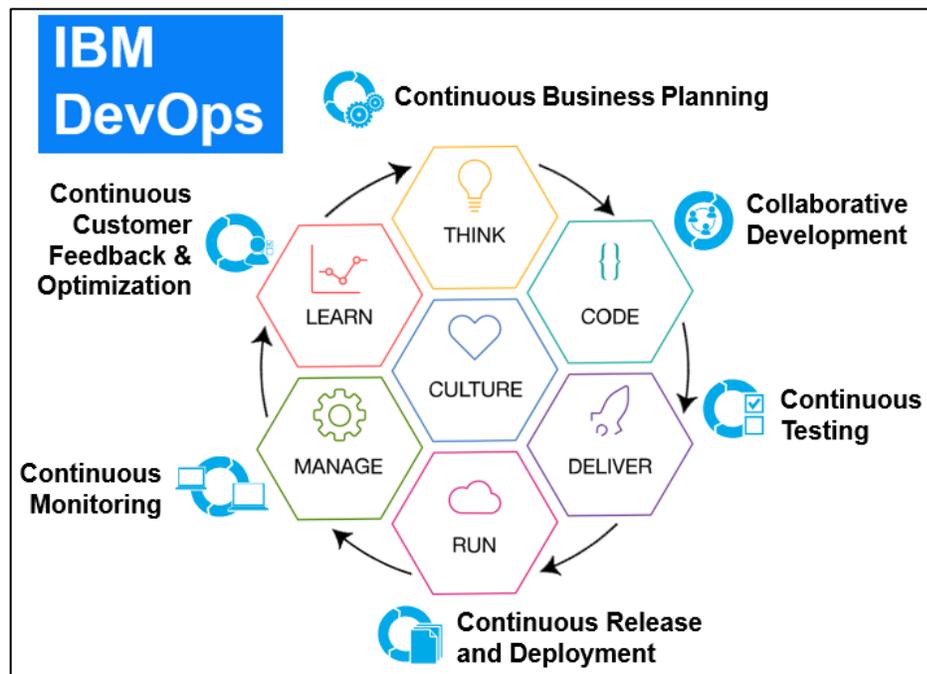


Fuente: IBM. (2020). *IBM Analytics solutions*. Consulta: 3 de agosto de 2020. Recuperado de <https://www.ibm.com/analytics/solutions>

El ciclo de vida analítico permite a las organizaciones adaptarse al crecimiento de datos gestionando de forma correcta la espiral de la información para minimizar el riesgo y aumentar la rentabilidad (figura 11). Por su parte, Desarrollo y Operaciones (DevOps) es un enfoque integral de desarrollo y

operaciones con el fin de asegurar el tiempo adecuado de salida al mercado mejorando el retorno de inversión de la producción (figura 12).

Figura 12. **IBM DevOps**



Fuente: IBM. (2020). *IBM DevOps*. Consulta: 3 de agosto de 2020. Recuperado de <https://www.ibm.com/cloud/devops>

Al hablar de modelos y normas de calidad, los principios de Shewhart y Deming se encuentran implícitos. Pero no son los únicos actores que han puesto en marcha los modelos de mejora continua.

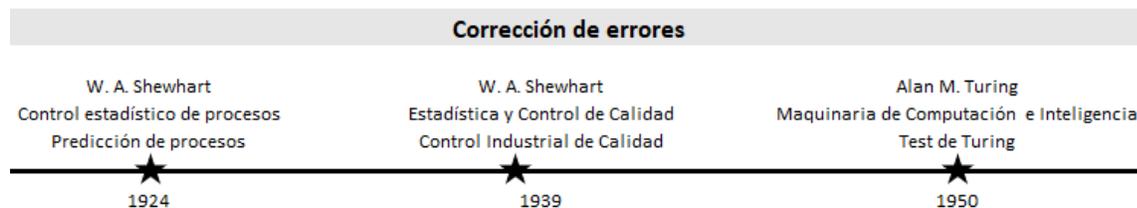
### 7.3. Modelo de pruebas de software

De acuerdo con la RAE (2020) modelo se define, entre otras acepciones, como un “Esquema teórico... de un sistema o de una realidad compleja... que se

elabora para facilitar su comprensión y el estudio de su comportamiento”. Por su parte, la academia define la palabra prueba así: “Ensayo o experimento que se hace de algo, para saber cómo resultará en su forma definitiva”. Por lo tanto, se entiende por modelo de pruebas de software el esquema de un sistema experimental que facilita la comprensión y el estudio del comportamiento de un programa informático para conocer su resultado final.

Antes de 1950 se consideraba que las aplicaciones debían ser probadas y verificadas por sus programadores, sin un esquema definido para este propósito. Un momento clave en la historia de las pruebas de software sucedió ese mismo año, cuando Alan M. Turing escribió el artículo “Maquinaria de Computación e Inteligencia”. En este ensayo propuso "La prueba de Turing", introduciendo la interrogante sobre la posibilidad de que las máquinas pudieran imitar el ingenio humano (Turing, 1950). Hasta 1956, las pruebas estaban enfocadas en corregir errores. Con base en el trabajo de Escobar (2014) se puede trazar una línea temporal sobre la evolución de las pruebas de software, como se aprecia en las siguientes imágenes.

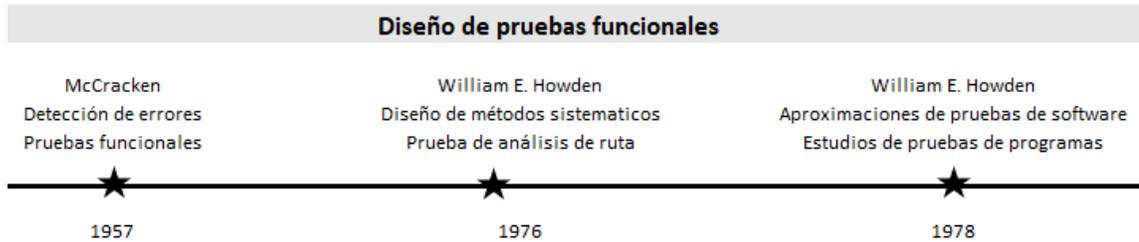
Figura 13. **Etapa de corrección de errores**



Fuente: elaboración propia.

A la etapa de corrección de errores, le siguió la de diseño de pruebas funcionales (figura 14).

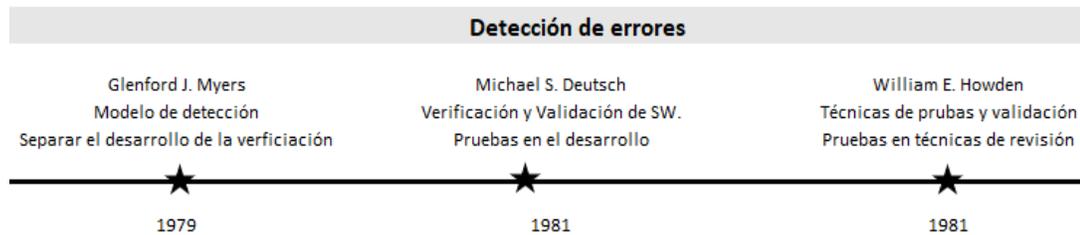
Figura 14. **Etapa de diseño de pruebas funcionales**



Fuente: elaboración propia.

A la etapa de pruebas funcionales, le siguió la de detección de errores. En la figura 15 se aprecia el primer período de la etapa.

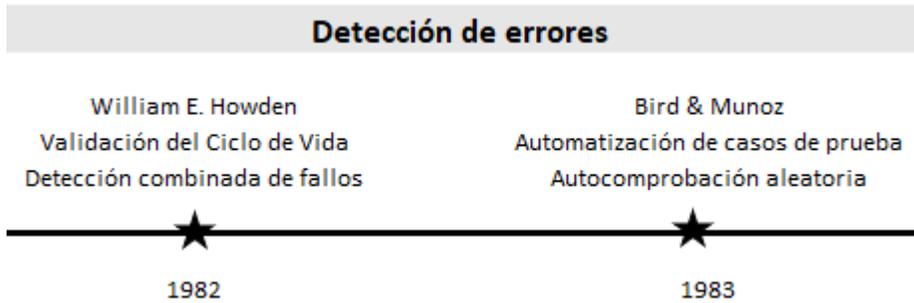
Figura 15. **Etapa de detección de errores (parte 1)**



Fuente: elaboración propia.

En la figura 16 se aprecia el segundo período de la etapa.

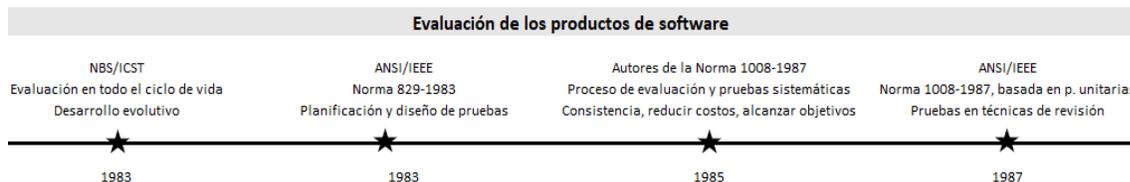
Figura 16. **Etapa de detección de errores (parte 2)**



Fuente: elaboración propia.

A la etapa de detección de errores, le siguió la de evaluación de los productos de software, según se aprecia en la figura 17.

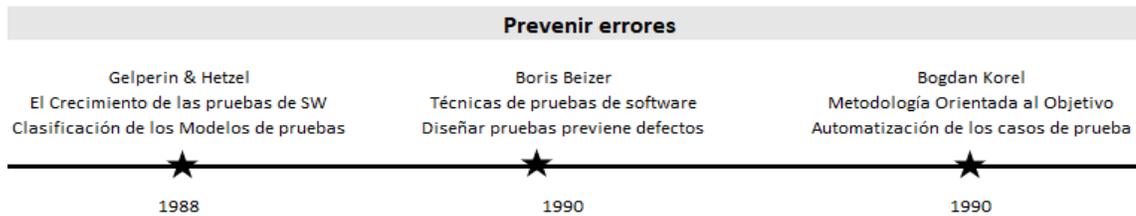
Figura 17. **Etapa de evaluación de los productos de software**



Fuente: elaboración propia.

A esta etapa le siguió la de prevención de errores. En la figura 18 se aprecia el primer período de la etapa.

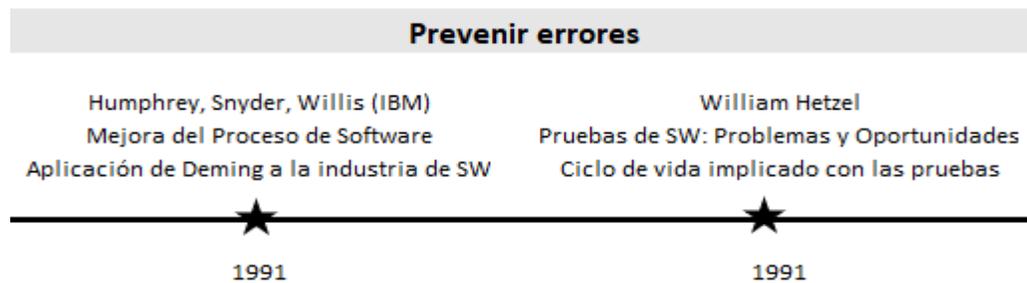
Figura 18. **Etapa de prevención de errores (parte 1)**



Fuente: elaboración propia.

En la figura 19 se aprecia el segundo período de la etapa.

Figura 19. **Etapa de prevención de errores (parte 2)**



Fuente: elaboración propia.

En 1984 se creó el Instituto de Ingeniería del Software (SEI, por sus siglas en inglés) a petición del Departamento de Defensa de los Estados Unidos, con sede en la Universidad Carnegie Mellon. Al SEI le fue comisionada la tarea de investigar y crear modelos para abordar los problemas de desarrollo de programas informáticos. También se le asignó la verificación de la capacidad y fiabilidad de las organizaciones que creaban aplicaciones para el Departamento de Defensa. En 1987 publicó la primera definición del modelo de madurez de

proceso en el desarrollo de software, que en 1991 evolucionó al Modelo de Madurez de las Capacidades para el Software (SW-CMM). En el año 2000 se publicó la primera versión de la Integración del Modelo de Madurez de Capacidad (CMMI, por sus siglas en inglés), basado en el Modelo de Madurez de Capacidad (CMM, por sus siglas en inglés) y se ha convertido en un mecanismo de mejora de procesos de las entidades que crean aplicaciones informáticas (Escobar, 2014).

Para el año 2020 se han popularizado algunos modelos de pruebas de software. A continuación, se estudiarán sus propiedades. Todos los modelos son de licenciamiento propietario.

Tabla I. **Modelos de pruebas de software más utilizados**

<b>Características</b>	<b>TMAP</b>	<b>TMM</b>	<b>TPI</b>	<b>Test PAI</b>	<b>TMMI</b>
<b>Publicación</b>	1995	1996	1998	2008	2008
<b>Enfoque</b>	En las actividades y tareas de pruebas. No se enfoca en procesos.	Procesos de pruebas.	Mejora de procesos de pruebas. No construye desde cero.	Mejora de procesos.	Procesos de pruebas.
<b>Complejo</b>	Flexible	Sí	Sí	Sí	Sí.
<b>Organización</b>	Todo tipo y tamaño	Mediana, grande	Mediana, grande	Mediana, grande	Mediana, grande
<b>Arquitectura</b>	Por etapas.	5 niveles de madurez.	Niveles de madurez A, B, C y D	Objetivos y prácticas.	5 niveles de madurez.
<b>Consideraciones</b>	Método pragmático.	Solo posee directrices.	Solo posee directrices.	Solo posee directrices	Solo posee directrices.

Fuente: elaboración propia.

Como se puede observar, a lo largo de más de medio siglo, las pruebas de software han tenido importantes avances, con aportes fundamentales desde diferentes enfoques. Diversos sistemas de procesos de prueba se han diseñado con el fin de abordar la necesidad de entregar software de calidad. La presente investigación se centrará en el Modelo de Madurez de Pruebas Integrado (TMMI) como un esquema para la mejora del proceso de pruebas.

### **7.3.1. Modelo TMMI**

TMMI es un modelo de madurez de pruebas de software cuya aplicación parte de la premisa de que todas las entidades inician en el nivel 1 de la arquitectura establecida por el modelo; cuanto más maduras son las prácticas de la organización, mayor es el nivel de madurez que cumple. Alcanzar nuevos niveles de madurez asegura que se han establecido las mejoras adecuadas como base para etapas posteriores. El enfoque de mejora se centra en las necesidades de la organización en el contexto empresarial y al pasar por los niveles de madurez aumenta la capacidad de gestión de la calidad del software y pruebas para alinearse con los objetivos del proyecto. El proceso apoya en la mejora de la calidad del producto de software con menos defectos (Modelo de Madurez de Pruebas Integrado [TMMI], 2020).

#### **7.3.1.1. Nivel Inicial**

En la primera etapa de la arquitectura, el proceso está fuera de control, y es dependiente del trabajo de unas cuantas personas excepcionales con el fin de obtener los resultados esperados. Es evidente cuando una organización gestiona en este nivel pues no entiende el costo de la calidad lo cual le genera desperdicio y poca efectividad.

#### **7.3.1.2. Nivel gestionado**

Cuando una organización se encuentra en el segundo nivel del modelo posee instrumentos como políticas, estrategias y planes. Así también controles, monitoreo, diseño y ejecución de pruebas.

#### **7.3.1.3. Nivel definido**

Adicional al nivel 2, en el tercer nivel se cuenta con instrumentos adicionales como una mejor organización, programa de entrenamiento, ciclo de vida e integración de pruebas. Aquí ya se encuentran pruebas no funcionales y revisiones por colegas como una forma adicional de validar que el desarrollo del software cumple con los requerimientos.

#### **7.3.1.4. Nivel medido**

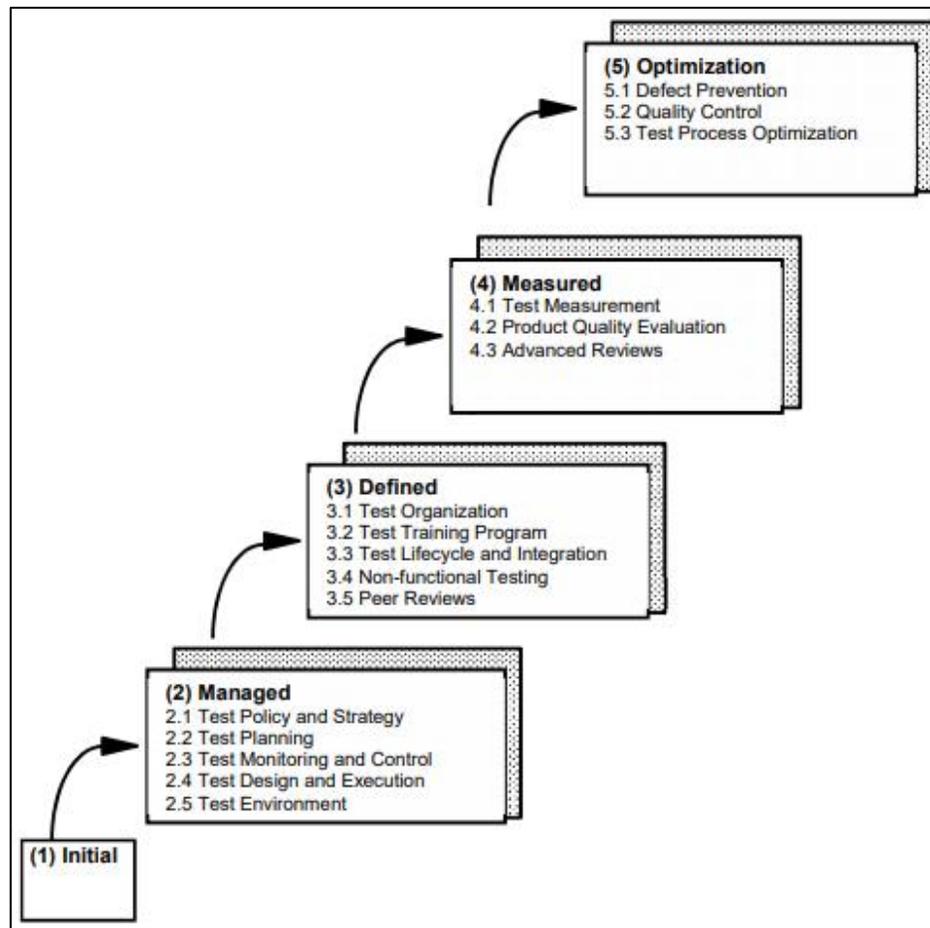
Conforme se avanza en las etapas del modelo, se garantizan mayores capacidades de gestión de la calidad del producto de software y pruebas. Para ello, en el cuarto nivel se cuenta con un sistema de medición de pruebas y evaluación de la calidad del software. Asimismo, las revisiones por colegas son mucho más avanzadas.

#### **7.3.1.5. Nivel optimizado**

En el nivel superior se optimiza al proceso de pruebas y control de calidad de los productos de software. Cumpliendo con las etapas previas, la organización se encuentra en la capacidad de prevenir defectos en los procesos de desarrollo y pruebas para entregar productos de alta calidad.

Como se puede apreciar en la arquitectura de TMMI (Figura 20), cada nivel de madurez indica las áreas de proceso que deben ser trabajadas por etapa. Según Velásquez, Vahos, López, Gómez, Londoño y Patiño (2018) la implementación del modelo tiene un impacto positivo en la calidad de las aplicaciones, el desempeño de las entidades de prueba y los plazos de entrega.

Figura 20. **Etapas de la arquitectura y áreas de proceso**



Fuente: Veenendaal. (2013). *TMMi maturity levels and process areas*. Consulta: 28 de agosto de 2020. Recuperado de <https://www.semanticscholar.org/paper/Results-of-the-first-TMMi-benchmark-%E2%80%93-where-are-we-Veenendaal-Cannegieter/435e5a09a023f700a2d2fd310eb0132d2f93b6ab/figure/0>

Las áreas de proceso guardan una relación cercana entre ellas. Para verificar y validar una entidad y obtener un grado de madurez, la guía TMMI utiliza formatos de cuestionarios y entrevistas. Estos permiten determinar el grupo de interrogantes según los objetivos de mejora, con el propósito de valorar el desarrollo de pruebas. Cabe resaltar que TMMI solo establece las indicaciones para su uso y es la organización interesada quien debe determinar la forma de llevarla a la práctica.

### 7.3.2. ISO/IEC/IEEE 29119

La ISO/IEC/IEEE 29119 es un conjunto de cinco normas internacionales para pruebas de software (AENOR, 2015). Su base se encuentra en los siguientes estándares internacionales.

Tabla II. **Normas internacionales que influenciaron la ISO/IEC/IEEE 29119**

Norma	Descripción	Influencia en 29119
<b>IEEE 829</b>	Documentación de pruebas	Documentación
<b>IEEE 1008</b>	Pruebas unitarias	Técnicas de pruebas
<b>BS 7925-1</b>	Glosario de términos	Conceptos y Vocabulario
<b>BS 7925-2</b>	Proceso y definiciones de técnicas de pruebas	Técnicas de pruebas
<b>IEEE 1028</b>	Revisión y auditoría	Revisión de productos de trabajo

Fuente: elaboración propia.

La arquitectura de la norma tiene la siguiente estructura:

Figura 21. **Estructura de la Norma ISO/IEC/IEEE 29119**



Fuente: elaboración propia.

- ISO/IEC/IEEE 29119-1:2013 (International Organization for Standardization [ISO], 2013)
  - Define los conceptos sobre pruebas de software y nociones clave que permiten entender la serie 29119.
- ISO/IEC/IEEE 29119-2:2013 (ISO, 2013)
  - Procesos del gobierno de pruebas.
  - Gestión e implantación de pruebas para cualquier entidad, actividad o proyecto.
  - Definiciones generales de procesos de pruebas, con ayuda gráfica.
  - Se puede utilizar por cualquier ciclo de desarrollo.
  - Enfocada al responsable del gobierno, gestión e implantación de pruebas.
  - Aplicable para directores, desarrolladores y probadores de proyectos.

- ISO/IEC/IEEE 29119-3:2013 (ISO, 2013)
  - Formatos para documentar pruebas
  - Definiciones sobre documentación de pruebas proveniente de los procesos de prueba.
  
- ISO/IEC/IEEE 29119-4:2015 (ISO, 2015)
  - Técnicas de diseño de casos y métodos de pruebas para emplear dentro del proceso definido por la parte 2 de la norma. Se pueden emplear para encaminar casos de prueba que permiten validar el cumplimiento del requerimiento de prueba. Asimismo, se presentan técnicas de pruebas con base en los riesgos, las cuales se definen en las partes 1 y 2 de la norma.
  
- ISO/IEC/IEEE 29119-5:2016 (ISO, 2016)
  - Define las pruebas con base en palabras clave.
  - Marco de referencia para implementación de pruebas y habilitar al evaluador a compartir sus artefactos de trabajo.
  - Características de las herramientas para pruebas de este tipo.
  - Características de interfaces y formatos de intercambio de datos estándar.
  - Jerarquía de palabras clave.

Luego de examinar los aspectos principales de la norma, es necesario reflexionar sobre las razones por las cuales es importante adoptarla para guiar los procesos de pruebas de nuestra organización. Su aplicación permitirá verificar, validar y evaluar a lo largo del ciclo de vida los elementos de prueba. Apoyará en la detección y prevención de defectos en etapas tempranas del desarrollo según las restricciones del calendario y los costos (Giraldo, 2015). Otros beneficios que aporta la norma son:

- Apoya en el desarrollo de un proceso de madurez de pruebas
- Provee técnicas aprobadas por la industria
- Aplicable a diferentes tipos de sistemas y organizaciones
- Ayuda a desarrollar métricas de pruebas
- Amplia selección de técnicas de pruebas
- Aplicable a desarrollos ágiles
- Guía para adaptar estándares al proceso de pruebas
- Guía para crear reportes de pruebas
- Desarrollo de habilidades y capacidades en los probadores
- Apoya la automatización
- Provee parámetros de seguridad
- Apoyo a la dirección de proyectos

#### **7.4. Efectividad en el desarrollo de software**

La efectividad se determina por la capacidad para cumplir los objetivos establecidos con el uso óptimo de los recursos asignados. En el desarrollo de software los insumos pueden ser la mano de obra, el tiempo, el presupuesto y los artefactos tecnológicos, entre otros. Hasta ahora se han estudiado las herramientas que permitirán conseguir los objetivos de gestión y calidad. En el presente apartado se estudiarán los procesos del desarrollo de software que se desea perfeccionar y la importancia en la consecución de este objetivo.

##### **7.4.1. Procesos**

El PMI se constituyó en 1969 con el fin de normalizar las mejores prácticas de administración de proyectos. En 1987 el American National Standards Institute (ANSI) publicó su trabajo hasta ese momento como el estándar Project Management Body of Knowledge (PMBOK). Luego, en 1996 se publica la primera

edición de la Guía del PMBOK. En octubre de 1998 ANSI acredita al PMI como una organización desarrolladora de estándares y ese mismo año la IEEE acepta el conjunto de conocimientos sobre gestión de proyectos de la Guía del PMBOK en la norma 1490-1998. Más tarde, en el año 2003, la IEEE adopta en su totalidad la Guía del PMBOK en la norma 1490-2003. La segunda y tercera edición del PMBOK se publicaron en los años 2000 y 2004 (Assaff, 2010; Instituto de Ingeniería Eléctrica y Electrónica [IEEE], 1998; IEEE, 2003). Tres ediciones más fueron publicadas en los años 2009, 2013 y 2017.

El propósito de trazar una línea de tiempo sobre el PMI y la Guía del PMBOK es el de valorar el trabajo que emite el instituto y los lineamientos que se enuncian en el manual. Esta guía establece un conjunto de procesos para la dirección de proyectos la cual es aceptada y utilizada para la gestión de proyectos de software. Asimismo, el PMI adaptó el ciclo PHVA a las interrelaciones dentro de los grupos de procesos y entre grupos (figura 22).

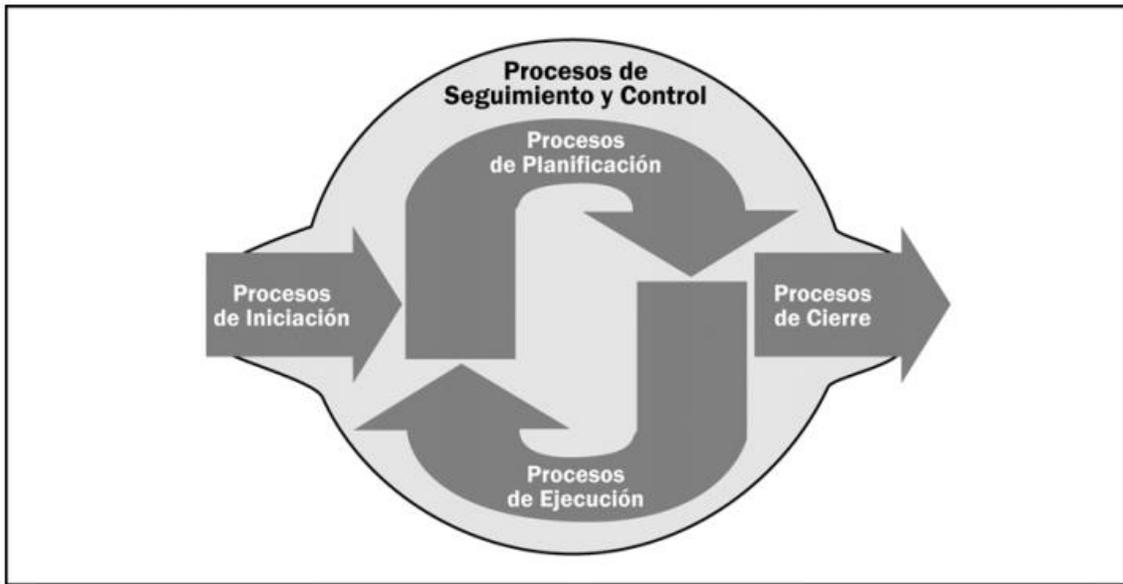
Tabla III. **Correspondencia de los grupos de procesos del PMBOK con el ciclo PHVA**

<b>Grupo de Procesos PMBOK</b>	<b>Etapas PHVA</b>
<b>Procesos de inicio</b>	Propios de PMBOK
<b>Procesos de planificación</b>	Planificar
<b>Procesos de ejecución</b>	Hacer
<b>Procesos de seguimiento y Control</b>	Revisar y actuar
<b>Procesos de cierre</b>	Propios de PMBOK

Fuente: elaboración propia.

A continuación, se aprecia en la imagen la adaptación del ciclo PHVA que realizó el PMI en la Guía del PMBOK.

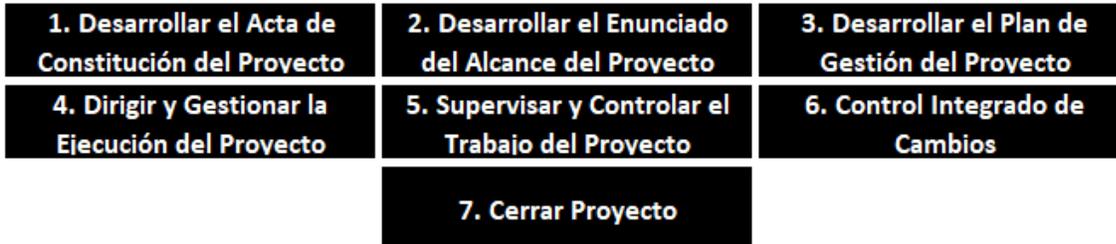
Figura 22. **Correspondencia de los grupos de procesos de dirección de proyectos al ciclo PHVA**



Fuente: PMI. (2004). *Guía de los Fundamentos de la Dirección de Proyectos*. Consulta: 28 de agosto de 2020. Recuperado de [https://www.academia.edu/8907738/PMBOK\\_TERCERA\\_EDICION\\_EN\\_ESPA%C3%91OL](https://www.academia.edu/8907738/PMBOK_TERCERA_EDICION_EN_ESPA%C3%91OL)

La Guía del PMBOK agrupa los procesos en diez áreas de conocimiento, que a su vez se corresponden con los cinco grupos de procesos. A continuación, se listan los procesos por el área de conocimiento a la que pertenecen. La primera área de conocimiento del PMBOK es la “Gestión de la Integración del Proyecto” (figura 23).

Figura 23. **Gestión de la integración del proyecto**



Fuente: elaboración propia.

La segunda área de conocimiento del PMBOK es la “Gestión del alcance del proyecto” (figura 24).

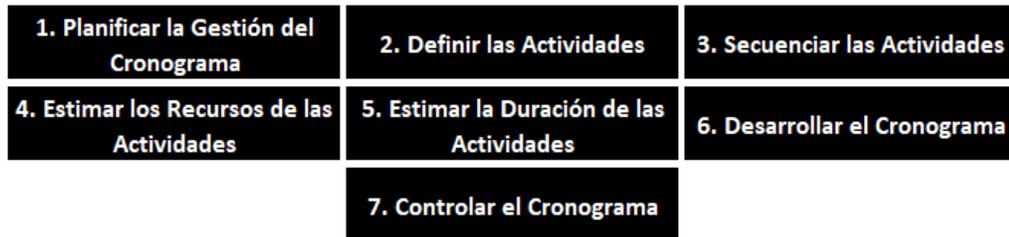
Figura 24. **Gestión del alcance del proyecto**



Fuente: elaboración propia.

La tercera área de conocimiento del PMBOK es la “Gestión del tiempo del proyecto” (figura 25).

Figura 25. **Gestión del tiempo del proyecto**



Fuente: elaboración propia.

La cuarta área de conocimiento del PMBOK es la “Gestión de los costos del proyecto” (figura 26).

Figura 26. **Gestión de los costos del proyecto**



Fuente: elaboración propia.

La quinta área de conocimiento del PMBOK es la “Gestión de la calidad del proyecto” (figura 27).

Figura 27. **Gestión de la calidad del proyecto**



Fuente: elaboración propia.

La sexta área de conocimiento del PMBOK es la “Gestión de los recursos humanos del proyecto” (figura 28).

Figura 28. **Gestión de los recursos humanos del proyecto**



Fuente: elaboración propia.

La séptima área de conocimiento del PMBOK es la “Gestión de las comunicaciones del proyecto” (figura 29).

Figura 29. **Gestión de las comunicaciones del proyecto**



Fuente: elaboración propia.

La octava área de conocimiento del PMBOK es la “Gestión de los riesgos del proyecto” (figura 30).

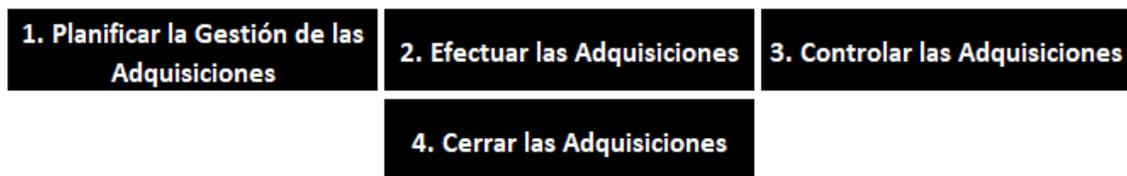
Figura 30. **Gestión de los riesgos del proyecto**



Fuente: elaboración propia.

La novena área de conocimiento del PMBOK es la “Gestión de las adquisiciones del proyecto” (figura 31).

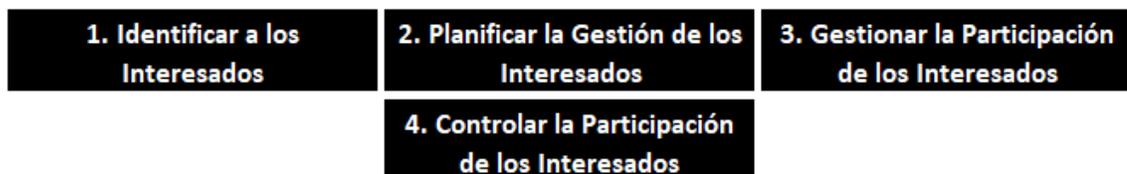
Figura 31. **Gestión de las adquisiciones del proyecto**



Fuente: elaboración propia.

La décima área de conocimiento del PMBOK es la “Gestión de los interesados del proyecto” (figura 32).

Figura 32. **Gestión de los interesados del proyecto**



Fuente: elaboración propia.

Cabe resaltar que los procesos del PMBOK son una guía para la gestión de proyectos y su uso no es de carácter obligatorio. De igual manera, cada equipo de desarrollo lo adaptará según sus necesidades.

#### **7.4.2. Eficiencia en los procesos**

Una vez identificados los procesos relevantes para nuestra dirección, es momento de ponerlos en marcha. Es esencial que cada proceso funcione de principio a fin. Si uno de estos no tiene el rendimiento que se espera y tampoco alcanza los objetivos para los cuales se concibió, está generando desperdicio. Y este es un fenómeno que ocurre todo el tiempo. Si se piensa en el principio de Pareto, y se considera que el 20 % del esfuerzo genera el 80 % de los resultados; o, por el contrario, el 20 % de los defectos provoca el 80 % de los problemas, en ambos casos se nota que hay una situación de desperdicio o mala utilización de recursos.

Los procesos que generan resultados deficientes o nulos ocupan recursos de forma inútil. Por lo cual, para tratar la eficiencia en procesos el enfoque debe estar en el desperdicio. Mientras un proceso se encuentra en funcionamiento agrega valor y acerca al cumplimiento de los objetivos del proyecto. Esto impacta la calidad de los productos por los cuales el cliente está dispuesto a pagar. Por el contrario, el desperdicio provoca costos adicionales, retrasa las actividades y genera productos de mala calidad por los cuales los clientes no pagarán.

El modelo de gestión de producción limpia enlista los ocho tipos diferentes de desperdicios en los procesos, que pueden ser extrapolados al desarrollo de proyectos (Cuatrecasas y González, 2017). Estos son:

- Superproducción
- Acarreo
- Tiempo muerto
- Exceso de actividades
- Inventario

- Desplazamiento
- Desperfectos en el producto
- Individuos subempleados

Entonces, una manera de medir la relación entre el trabajo de un proceso y el desperdicio que este genera es la eficiencia. Para medir la eficiencia en proyectos de desarrollo de software se puede echar mano de los algunos índices que sirven a este propósito (PMI, 2004).

- Índice de Desempeño de Costos o Cost Performance Index (CPI): mide la eficiencia en relación con los costos del proyecto.
  - EV: valor ganado
  - AC: costos reales

$$CPI = \frac{EV}{AC}$$

CPI  $\geq$  1, condición favorable

CPI < 1, condición desfavorable

Un CPI menor a 1.0 indica sobrecostos con respecto a lo planificado. Un CPI mayor a 1.0 indica costos menores con respecto a lo planificado.

- Índice de desempeño del cronograma o Schedule Performance Index (SPI): mide la eficiencia del cronograma del proyecto.
  - EV: valor ganado
  - PV: valor planificado

$$SPI = \frac{EV}{PV}$$

$SPI \geq 1$ , condición favorable

$SPI < 1$ , condición desfavorable

- Variación del costo o Cost Variance (CV): mide el desempeño según los costos del proyecto.
  - EV: valor ganado
  - AC: costos reales

$$CV = EV - AC$$

$CPI > 0$ , condición favorable

$CPI < 0$ , condición desfavorable

- Variación del cronograma o Schedule Variance (SV): mide el desempeño del cronograma del proyecto.
  - EV: valor ganado
  - PV: valor planificado

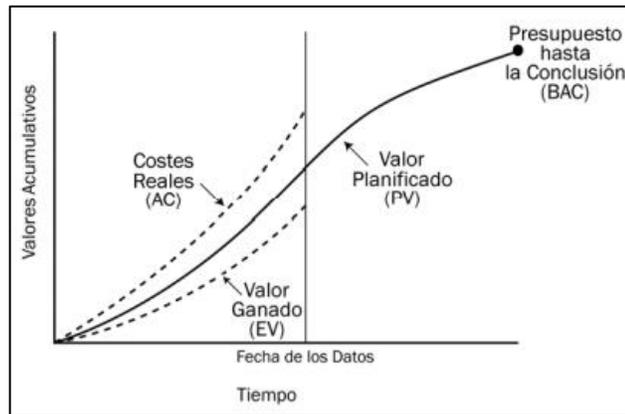
$$SV = EV - PV$$

$SV > 0$ , condición favorable

$SV < 0$ , condición desfavorable

Cuando el proyecto se culmina la variación del cronograma llegará a cero dado que todos los valores planeados se habrán concretado. El Presupuesto hasta la Conclusión (BAC) es el costo acumulado al completar el proyecto (Figura 33).

Figura 33. Rendimiento del proyecto



Fuente: PMI. (2004). *Guía de los Fundamentos de la Dirección de Proyectos*. Consulta: 23 de octubre de 2020. Recuperado de [https://www.academia.edu/8907738/PMBOK\\_TERCERA\\_EDICION\\_EN\\_ESPA%C3%91OL](https://www.academia.edu/8907738/PMBOK_TERCERA_EDICION_EN_ESPA%C3%91OL)

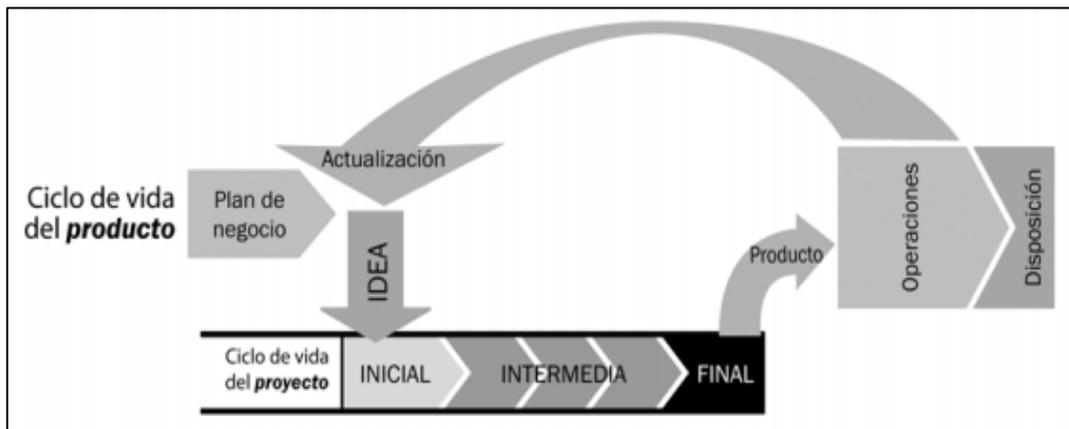
La mejora continua de procesos apoya en mitigar las actividades improductivas y en el aumento de la eficiencia en las operaciones. Asimismo, el enfoque de calidad propone acciones que incrementen los niveles de eficiencia de la dirección. Los estándares de calidad que se ponen en marcha dentro de la organización apoyan la consecución de los objetivos con eficiencia y aportan beneficios a los interesados del proyecto.

### 7.4.3. Calidad en procesos y productos de software

El conjunto de procesos de desarrollo del proyecto funcionando de forma eficiente producirá productos de calidad. Los procesos y productos guardan una estrecha relación y no se pueden esperar salidas adecuadas si no se emplean los procedimientos correctos y funcionales (figura 34). Para asegurar la calidad y mejora continua en procesos y productos, el PMBOK aporta procesos de gestión

de cambios, control de los riesgos y la calidad, según se enlistó en secciones previas.

Figura 34. **Ciclo de vida del producto y su integración con el ciclo de vida del proyecto**



Fuente: PMI. (2004). *Guía de los Fundamentos de la Dirección de Proyectos*. Consulta: 23 de octubre de 2020. Recuperado de [https://www.academia.edu/8907738/PMBOK\\_TERCERA\\_EDICION\\_EN\\_ESPA%C3%91OL](https://www.academia.edu/8907738/PMBOK_TERCERA_EDICION_EN_ESPA%C3%91OL)

Con el fin de alcanzar la efectividad en los objetivos de negocios como por ejemplo entregar software a tiempo, perfeccionar la calidad y reducir o erradicar el desperdicio la organización necesita implantar acciones para la mejora de procesos de software. Todo lo que sea sujeto de mejora, debe ser mejorado. Lo que se busca es construir productos de alta calidad, con mayor rapidez, más barato que la competencia y con inteligencia. Mejorar los procesos informáticos conlleva beneficios comerciales y aumenta las utilidades. Algunos de los beneficios que resultan de mejorar los procesos de software son:

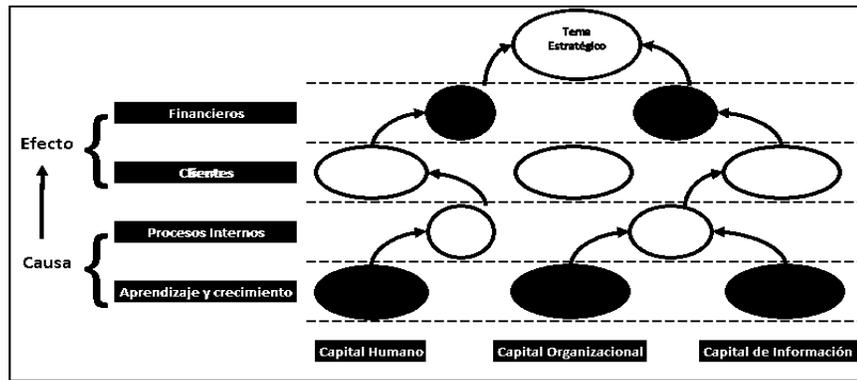
- Perfeccionamiento de la calidad
- Aumento de la moral de los empleados

- Incremento de la satisfacción del cliente
- Consistencia de entregas según el presupuesto y el calendario
- Reducciones de los costos del desarrollo de software
- Reducciones en los costos asociados con la baja calidad
- Aumento de la productividad
- Aumento de las entregas en tiempo

Introducir o adoptar un modelo de mejora es un hito en una iniciativa de cambio y aporta un lenguaje y una visión compartida de perfección (McDonald, Musson & Smith, 2007; O'Regan, 2017). Los modelos más reconocidos para la mejora de procesos de software son los siguientes:

- Six Sigma: es una doctrina de trabajo para la mejora y solución de problemas. Se enfoca en minimizar y erradicar los desperfectos incrementando la rentabilidad y el desempeño.
- Balanced ScoreCard: establece objetivos a partir de la estrategia midiéndolos por medio de indicadores. Estos a su vez se encuentran relacionados con planes de acción de corto plazo para emparejar la organización con la estrategia empresarial (figura 35).

Figura 35. Mapa estratégico, base del Balanced ScoreCard



Fuente: elaboración propia.

- Software Process Improvement Capability Determination (SPICE)/ISO 15504: apoya la mejora, verificación y validación de procesos de desarrollo, soporte de sistemas de datos y productos informáticos.
- ISO 9001: define los requerimientos de un Sistema de Gestión de Calidad para emplearlo a lo interno de las entidades.
- Modelo de Integración de Capacidad y Madurez (CMMI): apoya la mejora, verificación y validación de procesos de desarrollo, soporte y funcionamiento de sistemas informáticos (figura 36).

Figura 36. **Resultados iniciales de desempeño al adoptar CMMI V2.0, según el Instituto CMMI**



Fuente: CMMI Institute. (2020). *Take Organizational Performance to the Next Level with CMMI V2.0*. Consulta: 23 de octubre de 2020. Recuperado de <https://cmmiinstitute.com/getattachment/f182a1a8-1ebf-4737-9d40-4c2e4b9456a1/attachment.aspx>

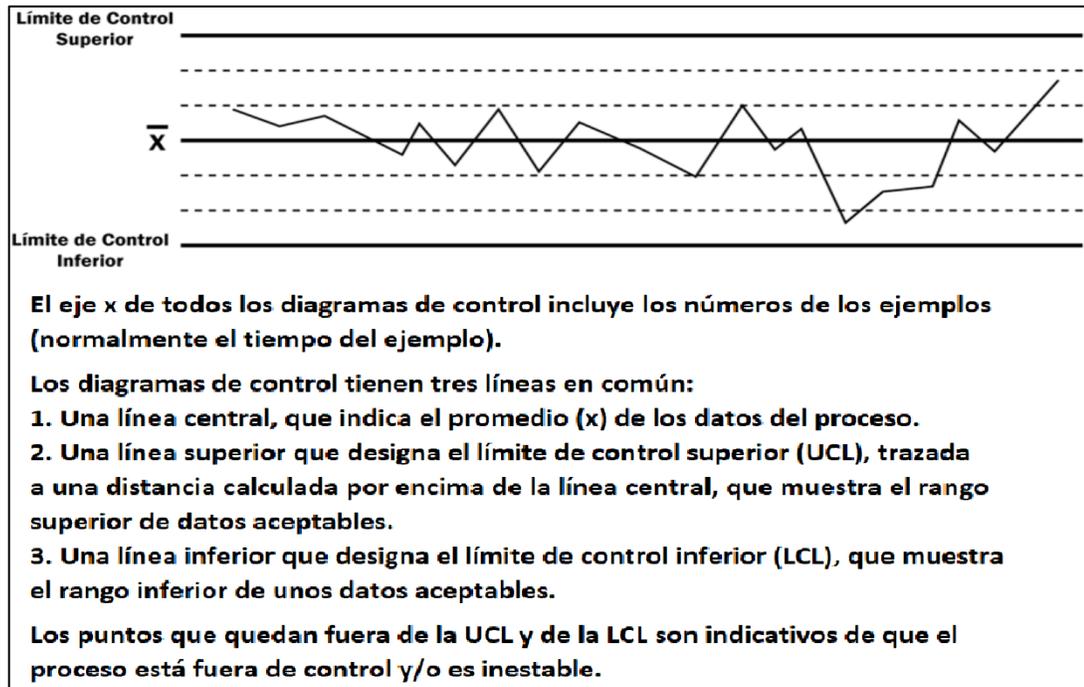
- **Análisis Causa-Raíz (RCA):** es una técnica para la solución de problemas a través de identificar las causas que le dieron vida. Busca prevenir así, la reincidencia en situaciones no deseadas y retroalimenta el sistema para la mejora.
- **Personal Software Process (PSP):** es una guía para las organizaciones que implementan CMMI y permite a los desarrolladores reducir el número de desperfectos en los productos. Apoya en la creación de acuerdos alcanzables, el incremento de las capacidades de planificación y pronóstico, así como en la gestión de la calidad de los procesos.
- **Team Software Process (TSP):** es un marco de trabajo para la mejora del desempeño y la calidad de un proyecto de desarrollo de software. Apoya a los programadores a alcanzar los objetivos de tiempo y costos.

#### **7.4.4. Gestión de errores del software**

El propósito del control de calidad de un producto es verificar y validar si el número de desperfectos hallados en la etapa de pruebas se encuentran en un rango aceptable por los estándares de calidad de la entidad. Este es un proceso de detección de errores. Como se ha estudiado con anterioridad, la evolución del proceso de pruebas lleva hasta un punto en el que se busca prevenir que los productos presenten defectos, pues esto impacta el tiempo y costo de entregas, entre otros aspectos. Dentro del proceso Realizar Aseguramiento de Calidad del PMBOK se proponen auditorías de calidad que autorizan la implantación de cambios, corrección de fallos, reparación de desperfectos y actividades de prevención.

Para estudiar el comportamiento de un proceso es necesario registrar información acerca de su funcionamiento. Con los datos registrados se puede entender el comportamiento real del proceso y contrastarlo con el esperado, es decir, si este se encuentra dentro de los parámetros que se han establecido como aceptables o si escapa de ellos. Las variaciones que salgan de los límites son de mayor relevancia para el estudio y pueden verse influenciadas por situaciones aleatorias o específicas. Son las causas puntuales las que se deben abordar y mitigar para aumentar la calidad de los procesos. Un estudio de este tipo fue el que W. Shewhart llevó a cabo en los laboratorios Bell Telephone del cual surgió el Diagrama de Control (figura 37).

Figura 37. Diagrama de control



Fuente: PMI. (2004). *Guía de los Fundamentos de la Dirección de Proyectos*. Consulta: 28 de octubre de 2020. Recuperado de [https://www.academia.edu/8907738/PMBOK\\_TERCERA\\_EDICION\\_EN\\_ESPA%C3%91OL](https://www.academia.edu/8907738/PMBOK_TERCERA_EDICION_EN_ESPA%C3%91OL)

Otro gráfico que permite detectar las principales causas de desperfectos es el Diagrama de Pareto (figura 5) del cual ya se mencionaron sus propiedades en secciones anteriores. Asimismo, el Diagrama de Ishikawa permite determinar las causas de un problema y las posibles soluciones a este (figura 4). Existen muchas más herramientas para detectar desperfectos, algunas de las cuales ya se han mencionado en el presente trabajo de investigación. Ahora es importante reflexionar en las razones por las cuales es necesario gestionar los errores. A continuación, se listan algunas de las razones por las cuales es importante gestionar y controlar las fallas:

- Los compromisos incumplidos impactan en la reputación de la organización.
- Impacto económico por falta de calidad.
- Se pierden oportunidades de negocios.
- Realizar cambios por calidad limita el inicio de nuevos proyectos.
- Carecer de calidad tiene repercute legalmente.
- Los clientes realizan previsiones con base en los compromisos que se realizan con estos.

#### **7.4.5. Efectividad en la gestión de errores**

Para perfeccionar la calidad es necesario incrementar la efectividad, verificando y validando los procedimientos en todo el ciclo de vida. Categorizar los riesgos, como se estudió con el gráfico de Pareto, ayuda a identificarlos de forma estándar, apoyando a la efectividad y precisión en la detección de fallos. Desvelar las densidades de los riesgos permite localizar de mejor manera las causas de la ineffectividad para introducir acciones correctivas (PMI, 2004).

De igual forma, dar seguimiento y control a los riesgos validando la ejecución de las soluciones permite verificar su efectividad. Durante la implantación de los planes para mitigar los riesgos se debe informar de forma periódica a la dirección sobre la efectividad que se está alcanzando con la implementación, con el fin de obtener una gestión adecuada. Por su parte, auditar los riesgos implica que la efectividad de las soluciones aplicadas será verificada y validada mejorando la gestión. Aplicar las tecnologías de la información en la automatización de la cadena de valor permitirá el intercambio común de información electrónica con a los interesados (PMI, 2004).

Seguir las prácticas de referencia pueden aumentar la efectividad del proyecto. En cada paso de la gestión existen procesos que requieren tomar en cuenta la efectividad pues esta se encuentra inmersa en cada pequeña parte de todo el sistema.

#### **7.4.6. Importancia en el cumplimiento de los tiempos de entrega**

Es importante cumplir con los tiempos de entrega porque:

- Se hacen compromisos y si no se cumplen, tiene un impacto en la reputación de la organización.
- Repercusiones monetarias por morosidad.
- Se pierden oportunidades de negocios futuros.
- Retrasos impiden arrancar con nuevos proyectos.
- No queda tiempo para la retroalimentación, cambios e innovación.
- Representa situaciones legales.
- El cliente hace previsiones en base a los compromisos que adquiere y espera que su producto se encuentre disponible según lo planificado.

#### **7.4.7. Reducción de costos**

Los costos de proyectos aumentan conforme:

- Se aceptan nuevos requerimientos no planificados
- No se gestiona el riesgo
- Se degradan las comunicaciones
- No se centraliza la información
- Se descuida la calidad

- No se establecen normas y límites para aceptar o rechazar defectos
- No se da importancia al desperdicio
- No se introducen acciones correctivas a tiempo
- No se da el adecuado seguimiento y control al cronograma
- No se registran ni atienden los fallos adecuadamente
- Se emplean métodos empíricos para abordar los problemas
- No se tiene una visión de procesos
- No se evalúan las salidas del sistema

Estas son solamente algunas de las razones por las cuales los costos de un proyecto se elevan.

#### **7.4.7.1. Técnicas y procedimientos para reducir costos**

Es necesario estudiar los datos históricos y obtener respuesta de las áreas de gestión y sobre todo del departamento financiero, para entender en donde se está perdiendo dinero (PMI, 2004). Y para ello es necesario:

- Poner en marcha los procesos de auditoria, mejora y optimización.
- Pagar por objetivos. Cumplimiento de objetivos.
- Controlar el horario de trabajo y la producción.
- Mejorar las comunicaciones, gestión.
- Introducir mejoras tecnológicas.
- Gestión de pagos y cobros riguroso y cuidadoso.
- Mejorar los contratos de proyectos.
- Automatizar procesos.
- Obtener retroalimentación de las áreas.

- Mejorar la gestión.
- Eliminar cuellos de botella.

## **7.5. Competitividad de las empresas de desarrollo de software**

La competitividad de una organización viene dada por:

- La calidad en el producto y el servicio que presta.
- Por cumplir con tiempos de entrega.
- Por cumplir en tiempo y calidad.
- Por trabajar de forma ordenada, organizada.
- Por tener buena comunicación con el cliente.

Para aumentar la competitividad es necesario poner en marcha una estrategia de transformación de la cultura de la entidad. Esto implica a la gerencia y dirección de la organización. Cuando los altos mandos están al tanto de los problemas será más fácil darles la importancia suficiente para abordarlos con proactividad y obtener cambios. Las estrategias pueden ser de cualquier índole, buscando impactar todas las partes de la cadena de valor, de modo que se obtengan mejoras relevantes y duraderas (PMI, 2004).

### **7.5.1. Estrategias**

Algunas de las estrategias que se pueden poner en marcha para incrementar la competitividad son las siguientes.

- Emplear herramientas tecnológicas.
- Adoptar el constante cambio para la evolución.
- Verificación y validación para implementar cambios.

- Innovación.
- Auditoría.
- Enfoque productivo.
- Retroalimentación del cliente interno y externo.
- Alinear a los equipos de trabajo con la estrategia de la organización.
- Desarrollar un enfoque de recursos humanos.

Estas no es una lista exhaustiva, ya que, en materia de estrategia, todo aquello que evoque mejoras y proporcione una ventaja competitiva puede ser adoptado y adaptado a las necesidades de la organización (PMI, 2004).

### **7.5.2. Herramientas para la competitividad**

La recopilación de las herramientas que se enlistan a continuación no es exhaustiva, pero estas son las más utilizadas.

- Prince 2
- Information Technology Infrastructure Library (ITIL)
- Objetivos de Control para las Tecnologías de la Información y Relacionadas (COBIT)
- PMBOK
- Ingeniería de software
- Modelos de Mejora continua
- Normas de calidad

Otras herramientas que permiten estudiar la situación de la organización para crear estrategias son las siguientes:

- Fortalezas, Oportunidades, Debilidades y Amenazas (FODA): es una herramienta para estudiar la situación de cualquier entidad observando sus propiedades internas y el comportamiento externo (García & Cano, 2013; Rojas, 2016). En la figura 38 se observan las propiedades de la matriz FODA.

Figura 38. **Propiedades de la matriz FODA**

<b>Propiedad</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Fortaleza</b>	Interna	Lo que diferencia de otros, aquello en lo sé es bueno
<b>Debilidad</b>	Interna	Desventajas y oportunidades de mejora
<b>Oportunidad</b>	Externa	Situaciones ventajosas o beneficiosas
<b>Amenaza</b>	Externa	Lo que hacen los competidores y las situaciones del entorno que afectan.

Fuente: elaboración propia.

Apoya a los gerentes a definir estrategias según el análisis situacional (figura 39).

Figura 39. Estrategias que derivan de la matriz FODA

	<b>FUERZAS-F</b>	<b>DEBILIDADES-D</b>
	Anotar las fuerzas	Anotar las debilidades
<b>OPORTUNIDADES-O</b> Anotar las oportunidades	<b>ESTRATEGIAS-FO</b> Anotar las fuerzas para aprovechar las oportunidades	<b>ESTRATEGIAS-DO</b> Superar las debilidades aprovechando las oportunidades
<b>AMENAZAS-A</b> Anotar las amenazas	<b>ESTRATEGIAS-FA</b> Usar las fuerzas para evitar las amenazas	<b>ESTRATEGIAS-DA</b> Reducir las debilidades y evitar las amenazas.

Fuente: Rojas. (2016). *Estrategias FA, FO, DA, DO*. Consulta: 28 de octubre de 2020.  
 Recuperado de <https://slideplayer.es/slide/10887889/>

- Político, Económico, Sociocultural, Tecnológico, Ecológico y Legal (PESTEL)  
 Es una herramienta para estudiar el comportamiento externo y de qué manera este afecta a la organización, así como determinar el grado de competitividad. Se enfoca en analizar seis factores que impactan a la entidad en cuestión, como se observa en la figura 40 (Martínez & Milla, 2012).

Figura 40. **Gráfico de PESTEL**



Fuente: elaboración propia.

A continuación, se presentan las propiedades más relevantes que estudia cada factor del análisis situacional PESTEL.

Tabla IV. **Factores del análisis estratégico**

<b>Factor</b>	<b>Descripción</b>
<b>Político</b>	Analiza la política fiscal, tratados comerciales, partidos políticos de turno, administración pública, políticas públicas.
<b>Económico</b>	Estudia la política económica, inflación, tipo de cambio y otros indicadores macroeconómicos. Asimismo, estudia la segmentación del poder adquisitivo, la tasa de desempleo, entre otros.

Continuación de la tabla IV.

<b>Social-Cultural</b>	<b>Estudia las tendencias sociales como modas y gustos que afectan el consumo. Asimismo, estudia las relaciones sociales, cambio poblacional, religión, otros.</b>
<b>Tecnológico</b>	Estudia la inversión en investigación y desarrollo, oportunidades de innovación, tecnologías de diferenciación, uso eficiente de factores que afectan la tecnología (energía eléctrica, internet), formas de producción, distribución, logística.
<b>Ecológico</b>	Analiza las políticas de protección del medioambiente, regulaciones en el consumo de recursos naturales y tratamiento de residuos; conciencia ecológica, contaminación y cambio climático.
<b>Legal</b>	Analiza los permisos y leyes sobre contrataciones, compras, empleo, salud, derecho de propiedad, protección y otras regulaciones.

Fuente: elaboración propia.

## 8. PROPUESTA DE ÍNDICE DE CONTENIDOS

ÍNDICE DE ILUSTRACIONES

LISTA DE SÍMBOLOS

GLOSARIO

RESUMEN

PLANTEAMIENTO DEL PROBLEMA Y FORMULACIÓN DE PREGUNTAS

ORIENTADORAS

OBJETIVOS

RESUMEN DEL MARCO METODOLÓGICO

INTRODUCCIÓN

### 1. MARCO TEÓRICO

#### 1.1. Industria de desarrollo de Software

1.1.1. Historia

1.1.2. Ingeniería de Software

1.1.3. El valor de la ingeniería de software

1.1.4. Procesos y ciclos de vida de software

1.1.5. Actividades en el ciclo de vida en cascada

1.1.6. Gestión de proyectos de software

1.1.7. Mejora de procesos de software

1.1.8. Mercado de software en Guatemala

1.1.8.1. Historia

1.1.8.2. Geografía

1.1.8.3. Economía

1.1.8.4. Política

1.1.8.5. Marco Jurídico

- 1.1.9. Empresa de desarrollo de software analizada
- 1.2. Calidad y mejora continua
  - 1.2.1. Calidad
  - 1.2.2. Herramientas
  - 1.2.3. Medición de la calidad
  - 1.2.4. Modelo de mejora continua de Deming
    - 1.2.4.1. Planificar
    - 1.2.4.2. Hacer
    - 1.2.4.3. Verificar
    - 1.2.4.4. Actuar
    - 1.2.4.5. Implementación del modelo de mejora continua
- 1.3. Modelo de pruebas de software
  - 1.3.1. Modelo TMMI
    - 1.3.1.1. Nivel Inicial
    - 1.3.1.2. Nivel Gestionado
    - 1.3.1.3. Nivel Definido
    - 1.3.1.4. Nivel Medido
    - 1.3.1.5. Nivel Optimizado
  - 1.3.2. ISO/IEC/IEEE 29119
- 1.4. Efectividad en el desarrollo de software
  - 1.4.1. Procesos
  - 1.4.2. Eficiencia en los procesos
  - 1.4.3. Calidad en procesos y productos de software
  - 1.4.4. Gestión de errores del software
  - 1.4.5. Efectividad en la gestión de errores
  - 1.4.6. Importancia en el cumplimiento de los tiempos de entrega
  - 1.4.7. Reducción de costos

1.4.7.1. Técnicas y procedimientos para reducir costos

1.5. Competitividad de las empresas de desarrollo de software

1.5.1. Estrategias

1.5.2. Herramientas para la competitividad

2. DESARROLLO DE LA INVESTIGACIÓN

3. PROPUESTA DE SOLUCIÓN

4. PRESENTACIÓN Y DISCUSIÓN DE RESULTADOS

CONCLUSIONES

RECOMENDACIONES

REFERENCIAS

ANEXOS



## **9. METODOLOGÍA DE INVESTIGACIÓN**

La investigación por realizar tendrá un enfoque mixto cualitativo-cuantitativo, de diseño transversal no experimental, de tipo descriptivo.

### **9.1. Enfoque de la investigación**

El enfoque de la investigación es mixto, debido a las siguientes razones:

Es de enfoque cualitativo porque parte de datos subjetivos, sin medición numérica, que se obtendrán directamente de cuestionarios a los colaboradores del departamento de TI, así como a usuarios finales. Con estos datos se busca determinar los niveles de satisfacción, calidad y exactitud de la resolución de errores, cuellos de botella y cualquier otra variable que permita entender el funcionamiento del área y sus procesos. Asimismo, indirectamente se analizarán datos recopilados en sistemas de software como opiniones, comentarios y observaciones sobre la gestión de errores. Los datos recopilados se reducirán a categorías. Después se utilizarán tablas comparativas dando lugar a gráficas, diagramas y mapas para analizar los resultados. Se emplearán técnicas para análisis situacional como FODA y PESTEL.

Por otro lado, es cuantitativo porque parte del estudio de datos numéricos que se registran en sistemas de gestión y rastreo de errores del software con cada proyecto en fase de pruebas. Por medio de cálculos matemáticos se establecerán patrones de comportamiento sobre la gestión y los costos en que se incurre cuando no se cumple con los tiempos y recursos planificados. Se observarán y analizarán indirectamente los datos registrados de errores del

software ya sean de programación, funcionales, otros. También se revisará la cantidad de quejas y reclamos, los recursos utilizados para la resolución de errores y otras variables que permitan determinar la productividad de los equipos y realizar otros análisis de relevancia. Todo esto se llevará a cabo por medio de técnicas de estadística descriptiva, como, por ejemplo, análisis de Pareto, medidas de tendencia central, variabilidad y de dispersión.

## **9.2. Diseño de la investigación**

El estudio se basa en un diseño no experimental transversal por las siguientes razones:

Es no experimental porque no existirá un control en los fenómenos observados por medio de entrevistas y encuestas, así como en los datos que se recogen en sistemas informáticos e impresos. No se realizará intervención en el registro y rastreo de errores del software para los proyectos en fase de pruebas, así como en los procesos del departamento de TI.

Asimismo, tendrá un diseño transversal ya que los datos para la investigación serán recolectados al inicio del estudio, lo que significa que se obtendrán una única vez para su posterior depuración, análisis y evaluación, con la finalidad de entender el estado inicial de los procesos de gestión del área de pruebas y a lo largo de la investigación hacer el análisis para llegar a la propuesta de solución.

## **9.3. Tipo de estudio**

El alcance de la investigación será descriptivo porque su desarrollo busca analizar la gestión que realiza el departamento de pruebas y caracterizarlo

fehacientemente. Para este propósito se levantarán todos los datos posibles de los procesos del área y de los errores de software registrados para entender y explicar la situación inicial. En función de las deficiencias encontradas en el análisis se realizará una propuesta de mejora de la efectividad de la gestión que permita erradicar los errores de software en tiempo, costo y calidad. Se realizará una comparativa entre la situación inicial y la solución propuesta, detallando todas las variables encontradas y las posibles mejoras del modelo.

#### **9.4. Variables e indicadores**

Las variables que serán estudiadas se describen a continuación:

- **Efectividad:** se medirá el nivel de cumplimiento de los objetivos del área de pruebas al atender los errores del software con un nivel óptimo de recursos.
- **Costos de desarrollo de software:** son las erogaciones monetarias en las que se incurre para soportar todo el ciclo de vida de los proyectos de software. Se estudiarán y medirán en función de la resolución de errores del software. Son parte medular de esta investigación y se medirá su nivel de mitigación de acuerdo con el modelo propuesto.
- **Actividades del área de pruebas:** se cuantificarán todas las actividades que realiza el área de pruebas durante el ciclo de vida de un proyecto de software.
- **Inefectividad en la gestión:** se medirá el nivel de cumplimiento de los objetivos del área de pruebas con respecto a los proyectos de software.

- Elementos necesarios en el modelo: se caracterizarán y cuantificarán todos los elementos que integran el modelo de pruebas de un proyecto de software bajo el ciclo de mejora continua.
- Proceso de pruebas de software: se estudiará el conjunto de actividades del proceso de pruebas con la finalidad de determinar la forma de ajustar el modelo propuesto.
- Objetivos de la organización: se estudiarán con el fin de garantizar que la propuesta es realista y está basada en prácticas que van en función de la consecución de estos.
- Errores: son los errores encontrados en el software y que el área de pruebas detecta con cada iteración de prueba al software; estos se analizan y resuelven con la finalidad de garantizar el funcionamiento adecuado de las aplicaciones de software. Son parte medular de esta investigación y se medirá su nivel de mitigación basándose en el modelo propuesto.

Para el desarrollo organizado del trabajo de investigación, se describen en la siguiente tabla, las principales variables que se analizarán en el mismo.

Tabla V. Cuadro de variables e indicadores

	Objetivos	Variables	Tipo de variable	Indicadores	Instrumentos
General	Diseñar un modelo de mejora basado en ISO/IEC/IEEE 29119 y pruebas TMMI para aumentar la efectividad en la gestión de errores y reducir los costos en una empresa de desarrollo de software.	Efectividad Costos	Cuantitativa continua Cuantitativa continua	% de efectividad en la resolución de errores % de gasto en la resolución de errores respecto del costo total del proyecto	Estadística descriptiva
	Identificar las actividades del área de pruebas que reducen la efectividad en la gestión e incrementan los costos de desarrollo de software.	<ul style="list-style-type: none"> <li>• Actividades del área de pruebas</li> <li>• Inefectividad en la gestión</li> <li>• Costos de desarrollo de software</li> </ul>	<ul style="list-style-type: none"> <li>• Cuantitativa discreta</li> <li>• Cuantitativa continua</li> <li>• Cuantitativa continua</li> </ul>	<ul style="list-style-type: none"> <li>• Cantidad de actividades</li> <li>• Escala de efectividad en la gestión</li> <li>• % de gasto en la resolución de errores</li> </ul>	<ul style="list-style-type: none"> <li>• Análisis documental</li> <li>• Cuestionario al personal de TI y usuario final</li> </ul>
Específicos	Establecer los elementos necesarios en el modelo para garantizar la efectividad en la gestión y reducir los costos de desarrollo de software.	Elementos necesarios en el modelo	Cuantitativa discreta	Cantidad de elementos del modelo de pruebas	<ul style="list-style-type: none"> <li>• FODA</li> <li>• PESTEL</li> <li>• Metodologías de desarrollo</li> <li>• Modelo TMMI</li> <li>• Mejora de Deming</li> <li>• ISO/IEC/IEEE 29119</li> </ul>
	Establecer la metodología para evaluar y ajustar el proceso de pruebas de software para garantizar el cumplimiento de los objetivos de la organización.	Proceso de pruebas de software Objetivos de la organización	Cuantitativa continua Cuantitativa continua	% de ajuste con el proceso establecido % de cumplimiento con los objetivos de la organización	<ul style="list-style-type: none"> <li>• Cuadro comparativo</li> <li>• Lista de inspección</li> </ul>
	Determinar cómo se reducen los errores y costos de desarrollo de software con el modelo diseñado.	Errores Costos	Cuantitativa continua Cuantitativa continua	% de mejora en la resolución de errores % de mejora en los costos del proyecto según los errores asociados	Análisis de Pareto

Fuente: elaboración propia.

## **9.5. Fases de la investigación**

La propuesta de la metodología para dar solución a la situación planteada estará conformada por cinco fases principales, el desarrollo adecuado permitirá alcanzar los objetivos de la investigación. La estructura en que se desarrollarán las fases se describe a continuación.

### **9.5.1. Fase 1: revisión documental de la teoría y bibliografía existente**

Para estructurar el marco teórico que fundamente la investigación y permita proporcionar la solución más certera a las necesidades que se identificaron y plantearon, esta fase se apoyará de la observación indirecta de información disponible en fuentes bibliográficas digitales, entrevistas, artículos científicos, tesis a nivel de maestría, doctorado y otros; así como en la consulta de registros del área de estudio para obtener una contextualización del entorno en el cual se realizará la investigación. Este proceso se apoyará en resúmenes, cuadros comparativos, lluvia de ideas y mapas conceptuales para ordenar y analizar la información.

### **9.5.2. Fase 2: análisis de las actividades del área de pruebas que reducen la efectividad en la gestión**

Se realizarán tres diferentes estudios para recopilar datos del ciclo de vida de desarrollo del software con la finalidad de contar con una base para extracción y análisis de información exhaustiva. Para ello se llevará a cabo un estudio de los proyectos de desarrollo de software que la empresa ha realizado durante el último año. Este primer estudio permitirá conocer los tiempos y costos marginales

de proyectos, errores por proyecto, naturaleza de las pruebas y de la gestión del proyecto en su estado inicial.

Para el cálculo de la muestra se utilizará la siguiente fórmula tomando un error uniforme del 5 % y una confiabilidad de 95 %:

$$n = \frac{N\sigma^2 Z^2}{(N - 1)e^2 + \sigma^2 Z^2}$$

Donde:

N = tamaño de muestra

N = tamaño de la población (Proyectos de software)

$\sigma$  = desviación normal de la población, que, al no tener su valor, convencionalmente suele utilizarse el valor de 0.50.

z = tipificación del nivel de confianza en la distribución normal, y cuyo valor a un nivel de confianza del 95 % y dos colas es 1.96.

e = error en la muestra, que varía entre 0.01 y 0.09. Que para esta investigación será de 0.05.

Al sustituir los datos en la formula se obtiene el tamaño de la muestra de proyectos a tomar en cuenta en el desarrollo de la investigación:

Tabla VI. **Cuadro de la muestra de proyectos a evaluar**

<b>N</b>	<b><math>\sigma</math></b>	<b>Z</b>	<b>e</b>
<b>130</b>	0.5	1.96	0.05
<b>n</b>	97 proyectos		

Fuente: elaboración propia.

Por tanto, se utilizará una muestra de 97 proyectos para recopilar datos y estudiar la naturaleza del desarrollo de software de la organización.

En el segundo estudio, se elaborará una serie de cuestionarios al personal del departamento de TI, segmentando por áreas (sistemas, desarrollo, pruebas) para profundizar en los procesos según corresponde a cada una (ver Anexo 1). Este estudio permitirá analizar la naturaleza de la gestión, el nivel de cumplimiento de los tiempos y requerimientos de proyectos; evaluar el nivel de conocimiento de técnicas, herramientas y metodologías de desarrollo de software y pruebas de los equipos de trabajo, así como el uso de estas para llevar a cabo las atribuciones asignadas.

Para el cálculo de la muestra se utilizará la siguiente fórmula tomando un error uniforme del 5 % y una confiabilidad de 95 %:

$$n = \frac{N\sigma^2Z^2}{(N-1)e^2 + \sigma^2Z^2}$$

Donde:

N = tamaño de muestra

N = tamaño de la población

$\sigma$  = desviación normal de la población, que, al no tener su valor, convencionalmente suele utilizarse el valor de 0.50.

z = tipificación del nivel de confianza en la distribución normal, y cuyo valor a un nivel de confianza del 95 % y dos colas es 1.96.

e = error en la muestra, que varía entre 0.01 y 0.09. Que para esta investigación será de 0.05.

Al sustituir los datos en la formula se obtiene el tamaño de la muestra de personal del departamento de TI a tomar en cuenta en el desarrollo de la investigación:

Tabla VII. **Cálculo de la muestra de personal del departamento de TI**

<b>N</b>	<b><math>\sigma</math></b>	<b>Z</b>	<b>e</b>
<b>153</b>	0.5	1.96	0.05
<b>n</b>	110 personas		

Fuente: elaboración propia.

Por tanto, se utilizará una muestra de 110 personas para recopilar datos y estudiar la naturaleza de la gestión del personal de pruebas de la organización.

Un tercer estudio se realizará por medio de cuestionarios al usuario final que se encuentra relacionado con los proyectos analizados en el primer estudio, de manera que se conozca el nivel de satisfacción sobre los proyectos que dichos usuarios solicitaron y que les fueron entregados (ver Anexo 2). Se estudiarán los comentarios de los usuarios sobre el nivel de cumplimiento de requerimientos, tiempos, costos, funcionalidad, usabilidad y productividad de las aplicaciones.

La muestra por analizar es la misma que la calculada en el estudio de proyectos, ya que ambos guardan una relación unívoca. Por lo tanto, se utilizará una muestra de 97 usuarios para recopilar datos y estudiar el nivel de satisfacción acerca de las aplicaciones de software que les fueron entregadas.

Al finalizar el conjunto de estudios se tendrá una visión holística de la situación que fundamentará las razones de esta investigación, de manera que

permita realizar una propuesta basada en las mejores prácticas y recomendaciones de la industria, empleando para ello modelos y normas de mejora continua y de madurez de pruebas de software.

### **9.5.3. Fase 3: elaboración del modelo de gestión del área de pruebas para garantizar la efectividad y reducir los costos de desarrollo**

Se desarrollará un diagnóstico situacional para determinar de forma analítica las debilidades, fortalezas, amenazas y oportunidades de mejora por medio de la técnica FODA, complementándose con un análisis PESTEL para obtener una visión complementaria y periférica de la gestión, así como un análisis de Pareto para identificar los cuellos de botella y procesos cuyo abordaje es vital para mitigar deficiencias dentro del departamento.

A partir del análisis anterior, se elaborará cada uno de los componentes del proceso de la gestión de pruebas enfocados en los errores del software de acuerdo con las mejores prácticas y técnicas aprobadas por la industria, según la identificación de las necesidades del departamento de desarrollo de software. Se modelará la nueva organización del área, manuales de gestión, desarrollo y pruebas, detallando los pasos recomendados dependiendo de la tipología de los proyectos y requerimientos. Asimismo, se determinarán los sistemas y áreas que se recomienda involucrar en cada tipo de prueba en particular y otra serie de procesos que sean relevantes. Se complementará el modelo con un diccionario de lecciones aprendidas y casos de éxito de la gestión, a manera de sentar precedentes para posteriores iteraciones de pruebas.

#### **9.5.4. Fase 4: propuesta de evaluación y ajuste del proceso de pruebas para cumplir los objetivos**

Con el fin de completar el ciclo de mejora continua que se propone, es necesario determinar la forma de evaluar el desempeño del área de pruebas al emplear el modelo final, con lo cual, basándose en la información obtenida de este proceso, se puedan tomar acciones correctivas para ajustar el modelo, de manera que en ciclos posteriores de pruebas se reduzcan las probabilidades de error y se aumente así la efectividad de la gestión de forma continua. La importancia de este ajuste es vital para garantizar la evolución constante y fiable de los procesos que produzcan proyectos de software exitosos.

Los manuales creados en la fase anterior contarán con herramientas de evaluación (cuestionarios, listas de inspección, retroalimentación del usuario final) para determinar el nivel de cumplimiento del equipo de pruebas con las guías propuestas, con el fin de evaluar la calidad de los productos que son entregados, así como el nivel de cumplimiento con procedimientos y políticas propuestas para una gestión exitosa. El procedimiento permitirá entender la causa-raíz de los inconvenientes suscitados por entregas fuera del plan (recursos e insumos). Con esto se evitará reincidir en malas prácticas, al emplear acciones correctivas para fortalecer y hacer más robusta la gestión de pruebas, manteniendo una mejora constante que se traduzca en proyectos exitosos y negocios futuros.

#### **9.5.5. Fase 5: análisis comparativo de las metodologías de gestión para proyectar los beneficios del modelo propuesto.**

El informe final del trabajo de graduación presentará un análisis comparativo de los modelos de gestión inicial y final del área de pruebas, por medio del análisis de Pareto, con el fin de resaltar los beneficios que se pueden obtener a través de la propuesta realizada contrastándolos con los beneficios de la gestión tradicional. Haciendo evidentes las carencias se proyectarán los beneficios del modelo propuesto.

Comparando las fortalezas y debilidades de ambos modelos, el estudio permitirá visualizar las razones por las cuales es necesario realizar una reingeniería a los procesos de pruebas y mejorar constantemente dichos procesos. Los aspectos más relevantes para esta comparativa son los costos y tiempos invertidos en mejoras y correcciones de errores. Otro aspecto relevante es la calidad de los productos entregados y el nivel de satisfacción de los usuarios finales. El análisis mostrará aquellos aspectos en los cuales el modelo inicial carece de buenas prácticas con un abordaje empírico y la manera en la que el modelo propuesto es capaz de controlar estos vacíos de manera sistemática. Finalmente, se resaltarán las bondades del modelo y el valor agregado para el área de pruebas, el departamento de TI, la organización y el usuario final.

## 10. TÉCNICAS DE ANÁLISIS DE INFORMACIÓN

Para llevar a cabo los objetivos correspondientes a cada fase de la investigación, se detallan a continuación las técnicas y herramientas que permitirán analizar e inferir en la información recolectada para generar conclusiones de lo que se logró con la elaboración del estudio.

Para la primera fase se realizará observación indirecta, recabando datos sobre desarrollo de proyectos, ingeniería de software, metodologías de desarrollo de software, modelos de pruebas de software, de calidad y costos. Los datos se recolectarán sintetizando en categorías, mapas conceptuales y resúmenes. A continuación, se analizará cualitativamente el contenido por medio de cuadros comparativos para plasmar esquemas y diagramas con el fin de extraer la información relevante para la investigación. Las fuentes bibliográficas que se consultarán para extraer datos serán libros, artículos científicos, publicaciones digitales, tesis de maestría y posgrado.

En la segunda fase, se realizará observación indirecta de datos en fuentes digitales de la organización relacionadas con errores de software, así como datos derivados de cuestionarios al personal de TI y usuarios finales sobre la atención de errores y satisfacción del cliente. Para organizar lo recolectado, se tabularán los datos a través de herramientas de software con la finalidad de realizar un análisis cuantitativo y extraer información pertinente a la gestión de errores y del proceso de desarrollo. Por medio de estadística descriptiva se planteará la descripción de variables con promedios, porcentajes, gráficos de barras, gráficos de líneas, desviación normal y diagramas de pie haciendo posible encontrar los cuellos de botella y puntos clave de mejora de la gestión.

Tomando como base la información recabada en fases previas sobre el desarrollo y pruebas de software, modelos de pruebas y de mejora continua, en la tercera fase, se realizará un análisis cualitativo por medio de las herramientas FODA y PESTEL, así como diagramas de flujo, de procesos y de la estructura organizacional. Asimismo, se construirá un cuadro comparativo acerca del uso de los modelos de madurez de pruebas TMMI y la norma de pruebas de software ISO/IEC/IEEE 29119 con el fin de analizar cualitativamente cuáles son las mejores técnicas y prácticas que la organización podría utilizar para aumentar la efectividad de la gestión. De igual manera se realizará un análisis cuantitativo costo-beneficio sobre la resolución de errores. A partir de este estudio se creará el modelo de mejora continua del área de pruebas.

Para garantizar un modelo continuo de calidad en los procesos de pruebas será necesario retroalimentar el ciclo con datos de la última gestión con el fin de determinar los puntos de mejora. Para ello, en la cuarta fase, se introducirán al modelo de pruebas, a través de un análisis cualitativo con cuadros comparativos y listas de inspección, herramientas de recopilación y análisis de datos que permitan medir y evaluar el desempeño y cumplimiento de los protocolos establecidos por el nuevo modelo. Se propondrá emplear paquetes de software para analizar cuantitativamente los datos de este proceso con estadística descriptiva utilizando gráficos de barras, diagramas de pie y diagrama de Pareto para fundamentar el proceso de verificación. A partir de este estudio la organización podrá determinar los cambios a introducir en el modelo que sean capaces de producir mejores resultados en iteraciones posteriores.

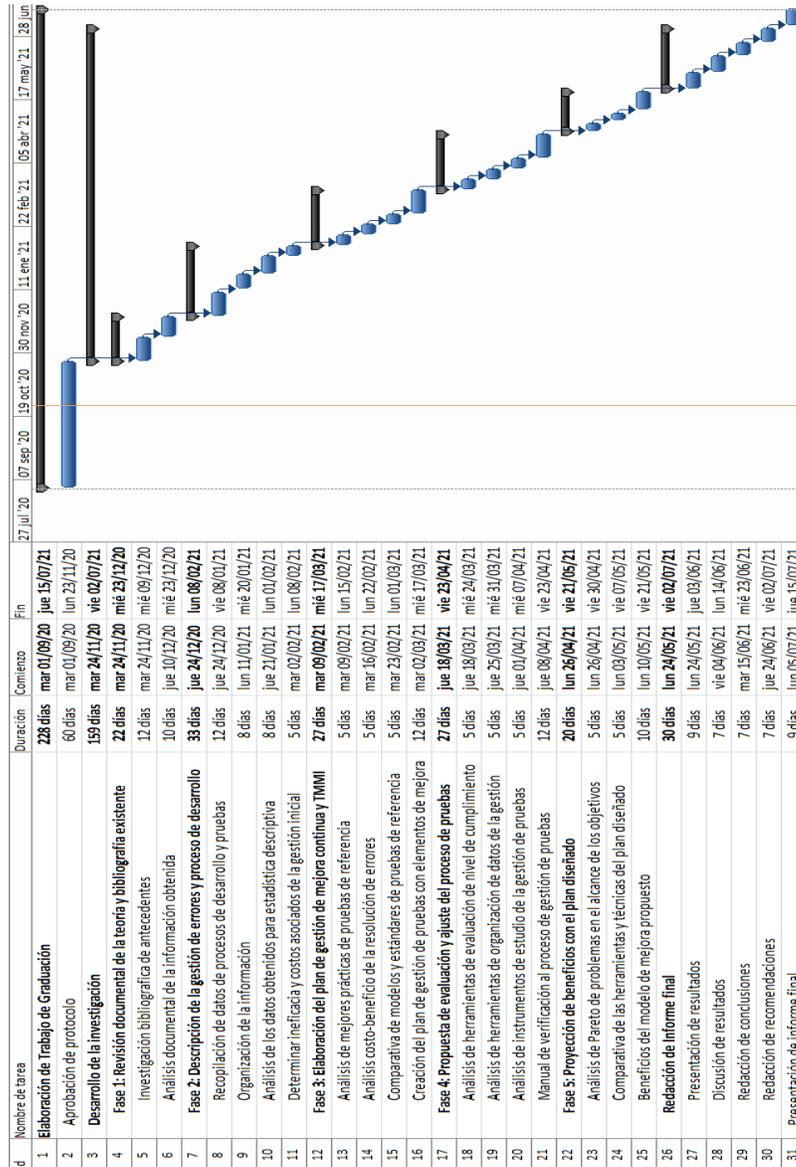
En la quinta fase, se estudiará la reducción de errores a partir del modelo propuesto, para lo cual se realizará un análisis cualitativo por medio de un cuadro comparativo para listar las herramientas y técnicas que el modelo aporta a la gestión analizando sus beneficios y la mejora que se obtiene de su utilización.

Para determinar cuáles son las deficiencias que se deben resolver se realizará un análisis cuantitativo a través de un diagrama de Pareto, con el fin de hacer visibles dichas deficiencias que afectan el alcance de los objetivos de la empresa y cómo se reducirían los costos no planificados. Con este análisis se determinará la forma de aportar mejoras a los procesos de desarrollo y pruebas de software a través de la propuesta de investigación, mejorando los tiempos de entrega y calidad de los productos y servicios.



# 11. CRONOGRAMA DE ACTIVIDADES

Figura 41. Cronograma



Fuente: elaboración propia.



## 12. FACTIBILIDAD DEL ESTUDIO

Para garantizar el éxito del presente trabajo de investigación es necesario contar con recursos humanos, materiales, tecnológicos, de información y financieros, los cuales se detallan a continuación:

- **Humanos:** deben estar conformados por el investigador, profesional asesor que avalará el cumplimiento de los requisitos técnicos, los colaboradores del área de TI de la organización y los usuarios finales de proyectos de software realizados por la organización.
- **Materiales y tecnológicos:** es necesario disponer de recursos materiales como papelería y útiles, combustible para movilización, equipo de computación para el análisis documental, equipos de impresión y una línea telefónica para coordinar las actividades de investigación. De igual manera, se debe contar con acceso a internet y a aplicaciones en línea como Google Forms para recabar información, Microsoft Visio para crear recursos visuales como cuadros comparativos, diagramas, esquemas, flujogramas, organigramas, entre otros. Se necesitan herramientas como Microsoft Project para dar seguimiento al calendario de trabajo, programas de ofimática como Microsoft Office para organizar y analizar datos, así como Microsoft Power BI para visualizar la información.
- **Información:** para analizar las variables de la investigación, es necesario contar con acceso a documentación de proyectos de software, protocolos y procedimientos del área de pruebas, así como a registros digitales sobre la gestión de proyectos y errores del software.

- **Financieros:** los recursos económicos que se describen a continuación son importantes para realizar la investigación y estos serán financiados por el investigador.

Tabla VIII. **Recursos financieros**

<b>Descripción</b>	<b>Tipo de Recurso</b>	<b>Monto</b>
Honorarios del asesor de la investigación	Humano	Q2,500.00
Papelería, útiles e insumos de oficina	Material y tecnológico	Q1,000.00
Gastos de movilización	Material	Q1,000.00
<b>Total de la inversión</b>		<b>Q4,500.00</b>

Fuente: elaboración propia.

El investigador deberá contar con un presupuesto personal aproximado de Q. 4,500.00, suma que representa el presupuesto total para realizar correctamente esta investigación y cumplir con los objetivos planteados.

### 13. REFERENCIAS

1. AENOR. (20 de mayo de 2015). *ISO/IEC/IEEE 29119 El nuevo estándar internacional para pruebas de software*. Zaragoza: AENOR. Recuperado de <https://in2test.lsi.uniovi.es/gt26/presentations/ISO29119-Presentacion-GT26-20150520.Zaragoza.pdf>
2. AGEXPORT (18 de mayo de 2011). *Guatemala como líder en proveedores de Internet y desarrollo de Software parte 2*. [Archivo de Vídeo]. Recuperado de <https://www.youtube.com/watch?v=tsGlxTK-JmY&t=151s>
3. Aspen. (agosto 2015). *Ecosistema Emprendedor en Guatemala*. México: ANDE. Recuperado de [https://cdn.ymaws.com/ande.site-ym.com/resource/resmgr/Docs/Ecosistema\\_Emprendedor\\_-\\_Gua.pdf](https://cdn.ymaws.com/ande.site-ym.com/resource/resmgr/Docs/Ecosistema_Emprendedor_-_Gua.pdf)
4. Assaff, R. (2010). *PMBOK - El Cuerpo de Conocimientos de la Gestión de Proyectos*. [Artículo]. Universidad de Palermo, Italia. Recuperado de <https://www.palermo.edu/ingenieria/downloads/CyT6/6CyT%2010.pdf>

5. Bolaños, R. M. (31 de enero de 2019). *El plan que busca subir del 12 al 26 % del PIB la inversión en infraestructura y crear ciudades intermedias*. [Mensaje de blog]. Recuperado de <https://www.prensalibre.com/economia/el-plan-que-busca-subir-del-12-al-26-del-pib-la-inversion-en-infraestructura-y-crear-ciudades-intermedias/>
6. Carro, R. y González, D. (2012). *Administración de la Calidad Total*. Argentina: Universidad Nacional del Mar de Plata. Recuperado de [http://nulan.mdp.edu.ar/1614/1/09\\_administracion\\_calidad.pdf](http://nulan.mdp.edu.ar/1614/1/09_administracion_calidad.pdf)
7. Central America Data. (13 de septiembre de 2018). *Cifras de la industria digital en Guatemala*. [Mensaje de blog]. Recuperado de [https://www.centralamericadata.com/es/article/home/Cifras\\_de\\_la\\_industria\\_digital\\_en\\_Guatemala](https://www.centralamericadata.com/es/article/home/Cifras_de_la_industria_digital_en_Guatemala)
8. Central America Data. (2 de enero de 2019). *Crece los ingresos de la industria digital*. [Mensaje de blog]. Recuperado de [https://www.centralamericadata.com/es/article/home/Crece\\_los\\_ingresos\\_de\\_la\\_industria\\_digital](https://www.centralamericadata.com/es/article/home/Crece_los_ingresos_de_la_industria_digital)
9. CEPAL. (mayo de 2011). *Políticas públicas: formulación y evaluación*. Recuperado de [https://www.cepal.org/ilpes/noticias/paginas/3/43323/LW\\_Polpub\\_antigua.pdf](https://www.cepal.org/ilpes/noticias/paginas/3/43323/LW_Polpub_antigua.pdf)

10. Chacón, L. (2007). *Desarrollo de la Industria del Software en Guatemala*. (Tesis de grado). Universidad de San Carlos de Guatemala, Guatemala. Recuperado de [http://biblioteca.usac.edu.gt/tesis/08/08\\_0358\\_CS.pdf](http://biblioteca.usac.edu.gt/tesis/08/08_0358_CS.pdf)
11. CMMI Institute. (19 de mayo de 2020). *Take Organizational Performance to the Next Level with CMMI V2.0*. [Artículo electrónico]. Recuperado de <https://cmmiinstitute.com/getattachment/f182a1a8-1ebf-4737-9d40-4c2e4b9456a1/attachment.aspx>
12. ConnectAmericas (14 de enero de 2015). *Entrevista a Gustavo Tasner, director del BPO Center de Capgemini*. [Archivo de Vídeo]. Recuperado de [https://www.youtube.com/watch?time\\_continue=74&v=qEQ9ihbRwG0&feature=emb\\_logo](https://www.youtube.com/watch?time_continue=74&v=qEQ9ihbRwG0&feature=emb_logo)
13. Cuatrecasas, L. y González, J. (2017). *Gestión Integral de la Calidad*. España: Profit. Recuperado de <http://reader.digitalbooks.pro/book/preview/101085/cover?1603912353967>
14. Cubillos, M., y Rozo, D. (enero de 2009). *El concepto de calidad: Historia, evolución e importancia para la competitividad*. Ciencia Unisalle: Revista de la Universidad de La Salle, 48, 80-99. Recuperado de <https://ciencia.lasalle.edu.co/cgi/viewcontent.cgi?article=1170&context=ruls>

15. De León, H. (28 de agosto de 2020). *Historia*. [Mensaje de blog]. Recuperado de <http://mate.ingenieria.usac.edu.gt/historia.php>
16. Diario La Hora. (1 de febrero de 2019). *Certeza jurídica*. [Artículo de periódico]. Recuperado de <https://lahora.gt/certeza-juridica/>
17. EAE. (3 de agosto de 2020). *Retos en Supply Chain*. [Mensaje de blog]. Recuperado de <https://retos-operaciones-logistica.eae.es/indicadores-de-calidad-optimizar-gestion/>
18. Escobar, M. (2014). *Diseño de un modelo para el desarrollo de prueba de software a fin de alcanzar el nivel 2 de Test Maturity Model en la empresa Siredcom en la ciudad de Quito*. (Tesis de maestría). Universidad de las Fuerzas Armadas, Ecuador. Recuperado de <https://repositorio.espe.edu.ec/bitstream/21000/8291/1/T-ESPEL-MAS-0009.pdf>
19. Fernández, M. (2015). *Aplicación de técnicas de pruebas automáticas basadas en propiedades a los diferentes niveles de prueba del software*. (Tesis doctoral). Universidad de La Coruña, España. Recuperado de <https://core.ac.uk/download/pdf/61915569.pdf>
20. FIUSAC. (28 de agosto de 2020). *Historia*. [Mensaje de blog]. Recuperado de <https://portal.ingenieria.usac.edu.gt/index.php/aspirante/antecedentes>

21. Furlán, L. (9 de enero de 2007). Guatemala: *Una pequeña historia de Internet*. [Mensaje de blog]. Recuperado de <https://interred.wordpress.com/2007/01/09/una-pequena-historia-de-internet-en-guatemala/>
22. Galileo. (28 de agosto de 2020). *Reseña Histórica*. [Mensaje de blog]. Recuperado de <https://www.galileo.edu/acerca-de-galileo/resena-historica/>
23. García, T y Cano, M. (2013). *El FODA: una técnica para el análisis de problemas en el contexto de la planeación en las organizaciones*. Instituto de Investigaciones y Estudios Superiores de las Ciencias Administrativas, México. Recuperado de <https://www.uv.mx/iiesca/files/2013/01/foda1999-2000.pdf>
24. GBM. (28 de agosto de 2020). *Historia de GBM*. [Mensaje de blog]. Recuperado de <http://www.gbm.net/archivo/es/historia-de-gbm>
25. Giraldo, F. (20 de mayo de 2015). *ISO 29119 and Software Testing – now what?* Universidad del Quindío, Colombia. Recuperado de <https://www.slideshare.net/fdgiraldo/iso-29119-and-sw-testing>
26. Herrera, M. (2016). *Impacto que tiene el proceso de control de pruebas en la calidad del software*. (Tesis de maestría). Universidad de Guayaquil, Ecuador. Recuperado de <http://repositorio.ug.edu.ec/bitstream/redug/15767/1/TRABAJO%20ODE%20TITULACION%20ESPECIAL%20-%20MARIUXI%20HERRERA%20ZUC3%91IGA.pdf>

27. IBM. (2020). *DevOps tools and software*. [Mensaje de blog]. Recuperado de <https://www.ibm.com/cloud/devops>
28. IBM. (2020). *IBM DevOps*. [Mensaje de blog]. Recuperado de <https://www.ibm.com/analytics/solutions>
29. Idrovo, F. (2016). *Diseño de manual de procedimientos de fase de pruebas para mejorar la calidad del software*. (Tesis de maestría). Universidad de Guayaquil, Ecuador. Recuperado de <http://repositorio.ug.edu.ec/bitstream/redug/44005/1/TRABAJO%20DE%20TITULACION%20ESPECIAL-%20Fanny%20Idrovo%20Chiriboga.pdf>
30. IEEE. (1995). *610.10-1994 - IEEE Standard Glossary of Computer Hardware Terminology*. [Artículo científico]. Recuperado de <https://ieeexplore.ieee.org/document/477873>
31. IEEE. (22 de marzo de 1999). *IEEE 1490-1998 - IEEE Guide - Adoption of PMI Standard, A Guide to the Project Management Body of Knowledge*. [Mensaje de blog]. Recuperado de <https://standards.ieee.org/standard/1490-1998.html>
32. IEEE. (24 de mayo de 2004). *IEEE 1490-2003 - IEEE Guide Adoption of PMI Standard - A Guide to the Project Management Body of Knowledge*. [Mensaje de blog]. Recuperado de <https://standards.ieee.org/standard/1490-2003.html>

33. INCAE. (2016). *Emprendimiento Juvenil en Guatemala*. Escuela Universitaria en Costa Rica, Costa Rica. Recuperado de [https://www.incae.edu/sites/default/files/reporte\\_nacional\\_-\\_guatemala\\_final\\_corregido.pdf](https://www.incae.edu/sites/default/files/reporte_nacional_-_guatemala_final_corregido.pdf)
34. Ingesis. (28 de agosto de 2020). *Historia*. [Mensaje de blog]. Recuperado de <http://www.ingesis.com/historia>
35. ISO. (1 de diciembre de 2015). *ISO/IEC/IEEE 29119-4:2015*. [Mensaje de blog]. Recuperado de <https://www.iso.org/standard/62821.html>
36. ISO. (1 de noviembre de 2016). *ISO/IEC/IEEE 29119-5:2016*. Recuperado de <https://www.iso.org/standard/62821.html>
37. ISO. (1 de septiembre de 2013). *ISO/IEC/IEEE 29119-1:2013*. [Mensaje de blog]. Recuperado de <https://www.iso.org/standard/45142.html>
38. ISO. (5 de septiembre de 2013). *ISO/IEC/IEEE 29119-2:2013*. [Mensaje de blog]. Recuperado de <https://www.iso.org/standard/62821.html>
39. ISO. (8 de septiembre de 2013). *ISO/IEC/IEEE 29119-3:2013*. [Mensaje de blog]. Recuperado de <https://www.iso.org/standard/62821.html>
40. Jiguan, B. (6 de diciembre de 2018). La calidad del internet está aumentando en Guatemala. [Mensaje de blog]. Recuperado de <https://dca.gob.gt/noticias-guatemala-diario-centro-america/la-calidad-del-internet-esta-aumentando-en-guatemala/>

41. Lewis, J. (2001). *El Internet como herramienta para la promoción y desarrollo de servicios financieros en Guatemala, caso: Bancafé, grupo financiero del país*. (Tesis de grado). Universidad Rafael Landívar, Guatemala. Recuperado de <http://biblio3.url.edu.gt/Publi/Tesis/2001/01/04/Lewis-Jorge.pdf>
42. Mardomingo, C. (2012). *Desarrollo de una oficina de control de calidad de los servicios de tecnologías de Información*. (Tesis de maestría). Universidad Católica Andrés Bello, Venezuela. Recuperado de <http://biblioteca2.ucab.edu.ve/anexos/biblioteca/marc/texto/AAS3598.pdf>
43. Martínez, D. y Milla, A. (2012). *Análisis del Entorno*. Madrid: Díaz de Santos. Recuperado de [https://books.google.com.mx/books?id=LDStM0GQPkgC&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.mx/books?id=LDStM0GQPkgC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
44. McDonald, M., Musson, R. y Smith, R. (2007). *The Practical Guide to Defect Prevention*. Estados Unidos: Microsoft Press. Recuperado de <https://epdf.pub/the-practical-guide-to-defect-prevention.html>
45. Medrano, G. (2001). *Datos históricos de la Universidad Rafael Landívar*. Universidad Rafael Landívar, Guatemala. Recuperado de [http://biblio3.url.edu.gt/Libros/2011/pas\\_pre/5.pdf](http://biblio3.url.edu.gt/Libros/2011/pas_pre/5.pdf)

46. Menéndez-Barzanallana, R. (05 de abril de 2014). *Ingeniería de Software*. Universidad de Murcia, España. Recuperado de <https://www.um.es/docencia/barzana/IAGP/IAGP2-Ingenieria-software-introduccion.html>
47. Menéndez-Barzanallana, R. (13 de agosto de 2013). *Apuntes de las asignaturas que imparto sobre informática, y algo más*. Universidad de Murcia, España. [Mensaje de blog]. Recuperado de <https://www.um.es/docencia/barzana/II/li04.html>
48. Mera-Paz, J. A. (19 de julio de 2016). *Análisis del proceso de pruebas de calidad de software*. *Ingeniería Solidaria*, 12 (20), 163-176. [Artículo científico]. Recuperado de <https://revistas.ucc.edu.co/index.php/in/article/download/1482/1724/>
49. MINECO. (2018). *Evaluación De Las Relaciones Comerciales Entre Guatemala Taiwán*. Viceministerio de Integración y Comercio Exterior Dirección de Análisis Económico, Guatemala. Recuperado de [http://www.sice.oas.org/TPD/GTM\\_TWN/Studies/Review\\_Aug\\_2018\\_GTM\\_TWN\\_s.pdf](http://www.sice.oas.org/TPD/GTM_TWN/Studies/Review_Aug_2018_GTM_TWN_s.pdf)
50. O'Regan, G. (2017). *Concise Guide to Software Engineering*. Estados Unidos: Springer. Recuperado de <https://link.springer.com/book/10.1007/978-3-319-57750-0>

51. PMI. (2004). *Guía de los Fundamentos de la Dirección de Proyectos*. España: ANSI. Recuperado de [https://www.academia.edu/8907738/PMBOK\\_TERCERA\\_EDICION\\_EN\\_ESPA%C3%91OL](https://www.academia.edu/8907738/PMBOK_TERCERA_EDICION_EN_ESPA%C3%91OL)
52. PMI. (2016). *El alto costo de un bajo desempeño*. España: CEO. Recuperado de [https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2016.pdf?sc\\_lang\\_temp=es-ES](https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2016.pdf?sc_lang_temp=es-ES)
53. Ponce, J., Domínguez, F., Gutiérrez, J., y Escalona, M. (2014). *Pruebas de aceptación orientadas al usuario: contexto ágil para un proyecto de gestión documental*. Ibersid: Revista De Sistemas De Información Y Documentación, 8, 73-80. Recuperado de <https://www.iversid.eu/ojs/index.php/iversid/article/view/4146/3785>
54. Pulse. (2020). *Ahead of the Curve*. [Mensaje de blog]. Recuperado de <https://www.pmi.org/learning/library/forging-future-focused-culture-11908>
55. RAE. (23 de octubre de 2020). *Modelo*. [Mensaje de blog]. Recuperado de <https://dle.rae.es/modelo>
56. RAE. (23 de octubre de 2020). *Prueba*. [Mensaje de blog]. Recuperado de <https://dle.rae.es/prueba>
57. RAE. (28 de octubre de 2020). *Calidad*. [Mensaje de blog]. Recuperado de <https://dle.rae.es/calidad>

58. Registro de Dominios.gt. (28 de agosto de 2020). *Nuestra Historia*. [Mensaje de blog]. Recuperado de <https://www.gt/sitio/ourhistory.php>
59. Rojas, H. (12 de febrero de 2016). Estrategias FA, FO, DA, DO. [Archivo de video]. Recuperado de <https://slideplayer.es/slide/10887889/>
60. Santos, V. (28 de agosto de 2020). *La Industria del Software. Estudio a nivel global y América Latina*. [Artículo científico]. Recuperado de <https://www.eumed.net/cursecon/ecolat/la/09/vsh.htm>
61. SIB. (03 de septiembre de 2020). *SIB Innovation Hub*. Guatemala: Super Intendencia de Bancos. Recuperado de [https://www.sib.gob.gt/SIBInnovationHUB/c/document\\_library/get\\_file?folderId=10623&name=DLFE-201.pdf](https://www.sib.gob.gt/SIBInnovationHUB/c/document_library/get_file?folderId=10623&name=DLFE-201.pdf)
62. TMMI. (1 de octubre de 2020). *TMMi Model*. [Mensaje de blog]. Recuperado de <https://www.tmmi.org/tmmi-model/>
63. Turing, A. (Octubre de 1950). *Computing Machinery and Intelligence*. Mind: Revista de la Asociación Mind, 59, 433-460. Recuperado de <http://www.jstor.org/stable/2251299>
64. UPV. (4 de enero de 2011). *La Crisis del Software*. [Mensaje de blog]. Recuperado de <https://histinf.blogs.upv.es/2011/01/04/la-crisis-del-software/>

65. UVG. (28 de agosto de 2020). *UVG en el tiempo*. Universidad Rafael Landívar, Guatemala. Recuperado de <https://www.uvg.edu.gt/nosotros/historia/>
66. Van, E. (2018). *Test Maturity Model Integration*. Irlanda: TMMi. Recuperado de <https://tmmi.org/tm6/wp-content/uploads/2018/11/TMMi-Framework-R1-2.pdf>
67. Van, E. y Cannegieter, J. (2013). *Test Maturity Model Integration (TMMi) Results of the First TMMI Benchmark – Where are we today?* [Mensaje de blog] Recuperado de <https://pdfs.semanticscholar.org/435e/5a09a023f700a2d2fd310eb0132d2f93b6ab.pdf>
68. Vásquez, E. (20 de junio de 2019). *Industria digital rebasa meta de ventas al exterior*. [Artículo de periódico]. Recuperado de <https://elperiodico.com.gt/inversion/2019/06/20/industria-digital-rebasa-meta-de-ventas-al-exterior/>
69. Velásquez, J. (2010). *Estudio tecnológico y de mercado acerca de los sistemas de banca electrónica en Guatemala*. (Tesis de grado). Universidad de San Carlos de Guatemala, Guatemala. Recuperado de [http://biblioteca.usac.edu.gt/tesis/08/08\\_0459\\_CS.pdf](http://biblioteca.usac.edu.gt/tesis/08/08_0459_CS.pdf)
70. Velásquez, S., Vahos, J., López, C., Gómez, M., Londoño, J. y Patiño, S. (15 de diciembre de 2018). *El rol de las normas en la acreditación y certificación de pruebas de software*. *Revista Espacios*, 39, 17-29. Recuperado de <http://www.jstor.org/stable/2251299>

71. Vides, A. (06 de enero de 2015). *Guatemala cierra el 2014 como mercado emergente de Outsourcing*. [Mensaje de blog] Recuperado de <https://agexporthoy.export.com.gt/sectores-de-exportacion/sector-servicios/guatemala-cierra-el-2014-como-mercado-emergente-de-outsourcing/>
  
72. Vides, A. (19 de junio de 2019). *“ON” primer encuentro que impulsará la innovación, emprendimiento digital y lo que la tecnología puede hacer por las empresas*. [Mensaje de blog] Recuperado de <https://agexporthoy.export.com.gt/sectores-de-exportacion/sector-servicios/on-primer-encuentro-que-impulsara-la-innovacion-emprendimiento-digital-y-lo-que-la-tecnologia-puede-hacer-por-las-empresas/>



## 14. APÉNDICES

### Apéndice 1. Departamento de TI

#### **Modelo de cuestionario para recolectar datos sobre la gestión del departamento de TI.**

Sus respuestas al siguiente cuestionario permitirán aumentar la efectividad del trabajo conjunto de TI. Conteste con libertad.

1. ¿A qué área de TI pertenece?
  - Pruebas
  - Desarrollo
  - Sistemas
  - Soporte Técnico
  - Otra
  
2. ¿Conoce cómo es la organización de su área?

---

  - Sí
  - No
  
3. ¿Conoce cuáles son las fases del ciclo de vida de los proyectos de software que utiliza su área?
  - Sí
  - No

Continuación del apéndice 1.

4. ¿Cuál es la metodología de desarrollo de software que más utiliza en proyectos de software?
  - RUP
  - SCRUM
  - XP
  - Otra
5. ¿Cuándo trabaja en un proyecto de software, conoce a los encargados y responsables de dichos proyectos?
  - Sí
  - No
6. ¿Conoce el calendario de trabajo del proyecto y da seguimiento a este?
  - Sí
  - No
7. ¿Conoce los objetivos de los proyectos de software en los que está involucrado?
  - Sí
  - No
8. ¿Qué tan eficientes son las comunicaciones durante la gestión del proyecto?
  - Malas
  - Regulares
  - Buenas
  - Muy buenas
  - Excelentes

Continuación del apéndice 1.

9. ¿La gestión de las actividades de los proyectos es sistemática y documentada o le resulta caótica?
  - Sistemática y documentada
  - Caótica
10. ¿La gestión de pruebas y errores del software es sistemática y documentada o le resulta caótica?
  - Sistemática y documentada
  - Caótica
11. ¿Los sistemas de software de gestión y rastreo de errores del software le resultan efectivos o inefectivos?
  - Efectivos
  - Inefectivos
  - Regular
12. ¿Los criterios de certificación de un proyecto le resultan aceptables o deberían de ser más estrictos?
  - Aceptables
  - Deberían de ser más estrictos
  - Deberían de ser más flexibles
13. ¿Cuáles tipos de pruebas de software conoce?
  - Unitarias
  - De componentes
  - De integración
  - De sistema
  - De humo
  - Alpha
  - Beta

Continuación del apéndice 1.

- De aceptación
  - De regresión
  - De caja blanca
  - De caja negra
  - Aleatorias
  - Compatibilidad
  - Seguridad
  - Estrés
  - Usabilidad
  - Rendimiento
  - Internacionalización y localización
  - Escalabilidad
  - Mantenibilidad
  - Inestabilidad
  - Portabilidad
  - Manuales
  - Automáticas
  - Otras
14. ¿Al realizar pruebas de software utiliza procesos y formatos estructurados, empíricos o aleatorios?
- Procesos y/o formatos estructurados
  - Procesos y/o formatos empíricos
  - Procesos y/o formatos aleatorios

Continuación del apéndice 1.

15. ¿En términos generales, considera que la cantidad de pruebas por requerimiento son exhaustivas?
- Sí
  - No
  - Regular
16. ¿Alguna vez trasladó un proyecto a producción con pequeños errores que serían resueltos posteriormente, luego de llegar a un convenio con las partes involucradas?
- Sí
  - No
17. ¿Ha trabajado en proyectos de software que se retrasan por la cantidad de errores que presentan?
- Sí
  - No
18. ¿Qué tan frecuente es recibir reportes de error posterior a la puesta en producción de un proyecto de software?
- Frecuencia nula
  - Poca frecuencia
  - Frecuencia moderada
  - Frecuencia considerable
  - Alta frecuencia
  - Extrema frecuencia
- ¿Con qué frecuencia trabaja horas extra para finalizar un proyecto?
- Frecuencia nula
  - Poca frecuencia
  - Frecuencia moderada

Continuación del apéndice 1.

- Frecuencia considerable
  - Alta frecuencia
  - Extrema frecuencia
19. ¿El resto de sus compañeros experimenta similar carga de trabajo?
- Sí
  - No
20. ¿Qué herramientas de software utiliza para realizar pruebas y certificar proyectos?
- JIRA
  - Jenkins
  - Junit
  - Selenium
  - JMeter
  - Mantis
  - Postman
  - GitHub
  - Bugzilla
  - Google Analytics
  - Redmine
  - Rational
  - Azure DevOps
  - Robot Framework
  - TestLog
  - Test Complete
  - ANTS
  - Load Impact

Continuación del apéndice 1.

- Test Studio
  - Testdroid
  - qaManager
  - TestLink
  - Otra
21. ¿Cuáles de los siguientes modelos y normas utilizan en su área durante la ejecución de proyectos de software?
- CMMI
  - TMMI
  - ISO/IEC/IEEE 29119
  - ISO 27001
  - ISO 27002

Fuente: elaboración propia.

## Apéndice 2. Usuario final

### **Modelo de cuestionario para conocer el nivel de satisfacción del cliente sobre los proyectos de software que ha solicitado al departamento de TI.**

Sus respuestas al siguiente cuestionario permitirán aumentar la efectividad del trabajo conjunto de TI. Conteste con libertad.

1. ¿Cuál es el nombre del proyecto?

---

2. ¿En qué fecha solicitó formalmente el proyecto?

---

3. ¿Cuál es la fecha de inicio del proyecto?

---

4. ¿Cuál es la fecha planificada de entrega del proyecto?

---

5. ¿Cuál fue la fecha real de entrega del proyecto?

---

6. ¿Cuál fue su nivel de involucramiento en el desarrollo de los requerimientos?

- Nulo
- Bajo
- Moderado
- Alto
- Muy alto

Continuación del apéndice 2.

7. ¿Cuál fue su nivel de involucramiento durante el desarrollo del proyecto?
  - Nulo
  - Bajo
  - Moderado
  - Alto
  - Muy alto
8. ¿Cuál fue su nivel de involucramiento durante las pruebas y certificación del proyecto?
  - Nulo
  - Bajo
  - Moderado
  - Alto
  - Muy alto
9. ¿Cuál fue su nivel de involucramiento durante la puesta en producción del proyecto?
  - Nulo
  - Bajo
  - Moderado
  - Alto
  - Muy alto
10. ¿Cuál fue su nivel de involucramiento durante el mantenimiento del proyecto? Posterior a su puesta en producción.
  - Nulo
  - Bajo
  - Moderado
  - Alto
  - Muy alto

Continuación del apéndice 2.

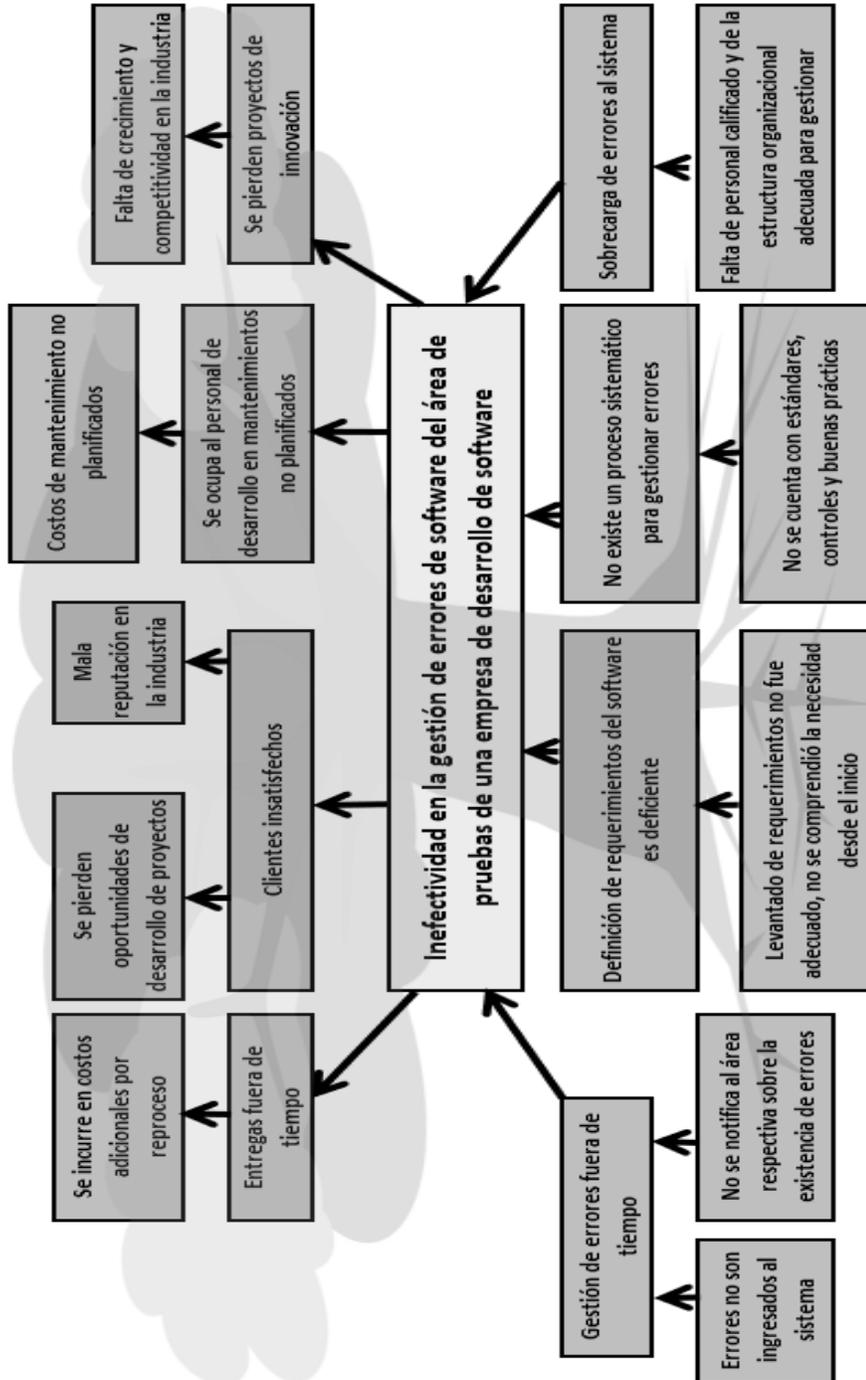
11. ¿Encontró errores en el software durante el proceso de pruebas y certificación?
  - Sí
  - No
12. ¿Encontró errores en el software durante la puesta en producción?
  - Sí
  - No
13. ¿Encontró errores en el software posterior a la puesta en producción?
  - Sí
  - No
14. ¿Cuál es su nivel de satisfacción acerca de la solución de software que recibió?
  - Nula
  - Baja
  - Moderada
  - Alta
  - Muy alta
15. ¿El software es usable?
  - Sí
  - No
16. ¿El software es funcional?
  - Sí
  - No
17. ¿El software es explotable? Le apoya en sus actividades del día a día.
  - Sí
  - No

Continuación del apéndice 2.

18. ¿El software satisface los requerimientos?
- Sí
  - No
19. ¿El software presenta oportunidades de mejora?
- Sí
  - No
20. ¿Cuál es su nivel de satisfacción acerca de la gestión del proyecto por parte de las áreas de TI?
- Nula
  - Baja
  - Moderada
  - Alta
  - Muy alta
21. ¿Recibió algún tipo de capacitación para abordar el proyecto en cualquiera de sus fases (Requerimientos, análisis y diseño, desarrollo, pruebas, certificación, puesta en producción)?
- Sí
  - No
22. Fue responsable por la gestión del proyecto
- Sí
  - No

Fuente: elaboración propia.

### Apéndice 3. Árbol del problema



Fuente: elaboración propia.

## Apéndice 4. Matriz de coherencia

<b>Diseño de investigación del modelo de mejora basado en ISO/IEC/IEEE 29119 y pruebas TMMI para aumentar la efectividad en la gestión de errores y reducir los costos en una empresa de desarrollo de software.</b>					
	Preguntas	Objetivos	Variables	Método de solución	Resultados esperados
<b>CENTRAL</b>	¿Qué modelo aumenta la efectividad en la gestión de errores de la organización y reduce los costos de desarrollo de software?	Diseñar un modelo de mejora basado en ISO/IEC/IEEE 29119 y pruebas TMMI para aumentar la efectividad en la gestión de errores y reducir los costos en una empresa de desarrollo de software.	Efectividad  Costos	<ul style="list-style-type: none"> <li>• Ciclo de Deming.</li> <li>• ISO/IEC/IEEE 29119</li> <li>• Modelo TMMI</li> </ul>	Modelo para la mejora continua del área de pruebas de la organización para aumentar la efectividad en la gestión de errores y reducir los costos de desarrollo de software.
<b>AUX. 1</b>	¿Cuáles son las actividades del área de pruebas que reducen la efectividad en la gestión e incrementan los costos de desarrollo de software?	Identificar las actividades del área de pruebas que reducen la efectividad en la gestión e incrementan los costos de desarrollo de software.	<ul style="list-style-type: none"> <li>• Actividades del área de pruebas</li> <li>• Inefectividad en la gestión</li> <li>• Costos de desarrollo de software</li> </ul>	<ul style="list-style-type: none"> <li>• Recopilación de datos: observación indirecta, cuestionarios</li> <li>• Análisis de datos</li> <li>• Informe situacional</li> </ul>	Reporte situacional para entender el contexto del área de pruebas, identificar debilidades, puntos de mejora y áreas de oportunidad.
<b>AUX. 2</b>	¿Qué elementos son necesarios en el modelo para garantizar la efectividad en la gestión y reducir los costos de desarrollo de software?	Establecer los elementos necesarios en el modelo para garantizar la efectividad en la gestión y reducir los costos de desarrollo de software.	Elementos necesarios en el modelo	<ul style="list-style-type: none"> <li>• Ciclo de Deming</li> <li>• ISO/IEC/IEEE 29119</li> <li>• Modelo TMMI</li> </ul>	<ul style="list-style-type: none"> <li>• Definición del proceso, recursos, estructura organizacional</li> <li>• Plan operativo de desarrollo y pruebas</li> </ul>
<b>AUX. 3</b>	¿Cómo se debe evaluar y ajustar el proceso de pruebas de software para garantizar el cumplimiento de los objetivos de la organización?	Establecer la metodología para evaluar y ajustar el proceso de pruebas de software para garantizar el cumplimiento de los objetivos de la organización.	Proceso de pruebas de software  Objetivos de la organización	<ul style="list-style-type: none"> <li>• Ciclo de Deming</li> <li>• ISO/IEC/IEEE 29119</li> <li>• Modelo TMMI</li> </ul>	Instrumento de evaluación y ajuste de la gestión que garantice el cumplimiento de los objetivos de la organización.
<b>AUX. 4</b>	¿Cómo se reducen los errores y costos de desarrollo de software con el modelo diseñado?	Determinar cómo se reducen los errores y costos de desarrollo de software con el modelo diseñado.	Errores  Costos	Análisis comparativo de la metodología de gestión tradicional contra la propuesta.	Definición cualitativa y cuantitativa de la reducción de errores y costos de desarrollo de software

Fuente: elaboración propia.

