

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
UNIDAD DE POSGRADO**



Modelo de aseguramiento de la calidad en el proceso de desarrollo de software basado en los modelos de madurez de capacidades (CMMi), proceso de software para equipos (TSP) y personas (PSP)

TESIS PARA OBTENER EL GRADO ACADÉMICO DE

Magíster en Gobierno de Tecnologías de Información

AUTOR

Ing. Alex José Espejo Chavarría

ASESOR

Dr. Carlos Pastor Carrasco

LIMA – PERÚ

2016

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

**FACULTAD DE INGENIERÍA DE SISTEMAS E
INFORMÁTICA**

UNIDAD DE POSGRADO

**Modelo de aseguramiento de la calidad en el proceso
de desarrollo de software basado en los modelos de
madurez de capacidades (CMMi), proceso de software
para equipos (TSP) y personas (PSP)**

TESIS

Para optar el Grado Académico de Magíster en Gobierno de
Tecnologías de Información

AUTOR

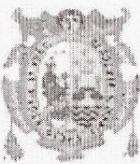
Alex José Espejo Chavarría

ASESOR

Pastor Carrasco Carlos

Lima – Perú

2016



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú, DECANA DE AMERICA
Facultad de Ingeniería de Sistemas e Informática
UNIDAD DE POSGRADO



SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO DE MAGÍSTER EN GOBIERNO DE TECNOLOGÍAS DE INFORMACIÓN

En la Ciudad Universitaria, a los cuatro (04) días del mes de noviembre del 2016, siendo las 20:00 horas, se reunieron en el Aula Magna de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos, el Jurado Examinador de tesis conformado por los siguientes profesores:

Dra. Nora Bertha La Serna Palomino (Presidente).
Dr. Carlos Alberto Pastor Carrasco (Miembro Asesor).
Dr. Glen Darío Rodríguez Rafael (Miembro).
Dra. Luz Sussy Bayona Oré (Miembro).
Mg. Marco Sobrevida Cabezudo (Miembro).

Se inició la Sustentación de la tesis invitando al graduando **Alex José Espejo Chavarría**, para que realizara la exposición oral y pública de la tesis para optar el Grado Académico de Magíster en Gobierno de Tecnologías de Información, siendo la Tesis intitulada:

“Modelo de Aseguramiento de la Calidad en el Proceso de Desarrollo de Software Basado en los Modelos de Madurez de capacidades (CMMi), Proceso de Software para Equipos (TSP) y Personas (PSP)”

Concluida la exposición, los miembros del Jurado Examinador procedieron a formular sus preguntas que fueron absueltas por el graduando; acto seguido se procedió a la evaluación correspondiente, habiendo obtenido la siguiente calificación:

B JF HUY BUENO

Por tanto el Presidente del Jurado, de acuerdo al Reglamento de Grados y Títulos, le otorga al bachiller **Alex José Espejo Chavarría** el Grado Académico de Magíster en Gobierno de Tecnologías de Información.

Siendo las **21:00** horas, el Presidente del Jurado Examinador da por concluido el acto académico de Sustentación de Tesis.

DRA. NORA BERTHA LA SERNA PALOMINO
Presidente

DR. CARLOS ALBERTO PASTOR CARRASCO
Miembro Asesor

DR. GLEN DARÍO RODRÍGUEZ RAFAEL
Miembro

DRA. LUZ SUSSY BAYONA ORÉ
Miembro

MG. MARCO SOBREVILLA CABEZUDO
Miembro

© Alex José Espejo Chavarría, 2016

Todos los derechos reservados.

FICHA CATALOGRÁFICA

**MODELO DE ASEGURAMIENTO DE LA CALIDAD EN EL
PROCESO DE DESARROLLO DE SOFTWARE BASADO
EN LOS MODELOS DE MADUREZ DE CAPACIDADES
(CMMI), PROCESO DE SOFTWARE PARA EQUIPOS
(TSP) Y PERSONAS (PSP)**

Alex José Espejo Chavarría
Lima - Perú, 2016
Orientador: Mg. Carlos Pastor Carrasco
Disertación: Magister en Gobierno de Tecnologías de Información

Universidad Nacional Mayor de San Marcos
Escuela de Posgrado
Facultad de Ingeniería de Sistemas e Informática, 2016
Unidad de Posgrado
Páginas: 184

DEDICATORIA

A mis padres, por su amor, aliento, enseñanzas y por desear lo mejor para mi vida, gracias por ser mi fuente inagotable de aliento para seguir adelante.

A mis hermanas Janet, Cecilia y Luz por su apoyo, por compartir el ideal que con sacrificio, esfuerzo y dedicación nuestros sueños se vuelven alcanzables.

A Milagros por sus palabras llenas de vida y motivación hacían que cada día me inspire a ser mejor persona y profesional.

A todas aquellas personas que con su incansable aliento y profesionalismo me ayudaron desde el principio a concretar este sueño.

Y, por encima de todo, doy gracias a Dios por conducir nuestras vidas por un camino lleno de fe y esperanza.

AGRADECIMIENTOS

Al Dr. Carlos Pastor Carrasco, por su paciencia, orientación y dedicación en su función de asesor y colaborador, para que este trabajo cumpla con los objetivos trazados.

A la Dra. Sussy Bayona y Dr. Glen Rodríguez por sus sugerencias, orientación y revisiones del presente trabajo, en su función de jurados informantes.

A mis padres Matías y Adriana a quienes les debo todo lo que soy, por su amor, por sus consejos y por sus enseñanzas.

A mis hermanas Janet, Cecilia y Luz por su incesante motivación y apoyo incondicional.

A todas aquellas personas que de diversas maneras me ayudaron para culminar el presente trabajo de investigación y se preocuparon por el cumplimiento de mis objetivos profesionales y personales.

A Dios, por su inmensa gracia y amor.

TABLA DE CONTENIDO

LISTA DE CUADROS.....	XII
LISTA DE FIGURAS.....	XIII
LISTA DE GRÁFICOS	XIV
RESUMEN.....	XVI
SUMMARY	XVII
CAPITULO I: INTRODUCCIÓN.....	1
1.1. Situación Problemática.....	1
1.2. Formulación del Problema.....	6
1.2.1. Problema General	6
1.2.2. Problemas Específicos	6
1.3. Justificación teórica	7
1.4. Justificación práctica	9
1.5. Objetivos	10
1.5.1. Objetivo General.....	10
1.5.2. Objetivos Específicos	10
CAPITULO II: MARCO TEÓRICO.....	11
2.1. Marco Filosófico o epistemológico de la investigación.....	11
2.2. Antecedentes de Investigación.....	12
2.2.1. Tesis Doctoral: Un marco metodológico para la mejora en la gestión de los equipos de desarrollo de software global	12
2.2.2. Tesis Maestría: Estudio comparativo de los modelos y estándares de calidad del software	14
2.2.3. Nota Técnica: La integración de CMMI y TSP/PSP: El uso de TSP de datos para crear modelos de rendimiento de proceso	15
2.2.4. Nota Técnica: Mejora de Procesos de Software para Pequeñas Organizaciones Basado en CMMI / TSP / PSP.....	17
2.2.5. Tesis Maestría: Integración entre PSP y PMBOK aplicada al desarrollo de un sistema experto para el diagnóstico e identificación automática de enfermedades profesionales.....	18
2.2.6. Tesis Maestría: Agilizando lo Ágil: Un Framework para el desarrollo de software bajo el modelo CMMI en Compañías que usan metodologías ágiles de desarrollo de software usando el Modelo Acelerado de Implementación (AIM)	19
2.2.7. Tesis Maestría: Bug Manager: Introduciendo Calidad a las Organizaciones de Software	21
2.3. Bases Teóricas.....	22
2.3.1. Modelos de Aseguramiento de la Calidad.....	22
2.3.1.1. Modelo de Referencia.....	22

2.3.1.2.	ISO/IEC 15504	24
2.3.1.3.	ISO 9001-2015	26
2.3.1.4.	PMBOK	31
2.3.1.5.	Modelo CMMI	34
2.3.1.6.	Proceso Personal de Software PSP	37
2.3.1.7.	Proceso de Software de Equipo TSP.....	46
2.3.1.8.	Integración de los Modelos CMMI, TSP y PSP	52
2.3.2.	Conceptos de Calidad	54
2.3.3.	Calidad en el Proceso de Desarrollo de Software	55
2.3.3.1.	Definición de Proceso de Software	55
2.3.3.2.	Importancia del Proceso de Desarrollo de Software	55
2.3.3.3.	Calidad del producto.....	56
2.3.3.4.	Calidad del proceso.....	59
2.3.3.5.	Aseguramiento de la calidad.....	60
2.3.3.6.	Proceso disciplinado.....	62
2.3.3.7.	Productividad en el Proceso de Desarrollo de Software	62
2.3.4.	Cuadro Comparativo de Modelos de Aseguramiento de la Calidad	63
2.3.4.1.	Cuadro Comparativo de Modelos a Nivel de Proceso	63
2.3.4.2.	Criterios de Selección del Modelo	69
2.4.	Hipótesis y Variables.....	71
2.4.1.	Hipótesis general.....	71
2.4.2.	Hipótesis específicas	71
2.4.3.	Identificación de variables.....	71
2.4.4.	Operacionalización de variables	72
CAPITULO III: METODOLOGIA.....		74
3.1.	Tipo y Diseño de Investigación.....	74
3.2.	Unidad de análisis	76
3.3.	Población de estudio	76
3.4.	Tamaño de muestra	77
3.5.	Selección de muestra.....	77
3.6.	Técnicas de recolección de Datos	78
3.6.1.	Técnicas	79
3.6.2.	Instrumentos.....	79
3.7.	Métodos y técnicas de procesamiento y análisis de datos.....	81
CAPITULO IV: RESULTADOS Y DISCUSIÓN.....		82
4.1.	Diagnóstico inicial.....	82

4.1.1. Tasa de respuesta.....	82
4.1.2. Información de la encuesta.....	83
4.2. Análisis, interpretación y discusión de resultados	92
4.3. Pruebas de hipótesis.....	96
4.3.1. Indicador Porcentaje de Costo de Calidad.....	96
4.3.2. Indicador Cantidad de Defectos.....	100
4.3.3. Indicador Porcentaje de Desviación de Costo.....	104
4.3.4. Indicador Porcentaje de Clientes Satisfechos	108
4.4. Presentación de resultados	112
4.4.1. Indicador Porcentaje de Costo de Calidad.....	112
4.4.2. Indicador Cantidad de Defectos.....	113
4.4.3. Indicador Porcentaje de Desviación de Costo.....	114
4.4.4. Indicador Porcentaje de Clientes Satisfechos	115
CAPITULO V: IMPACTOS	117
5.1. Modelo Propuesto	117
5.1.1. Fundamentos del Aporte	117
5.1.2. Descripción del Aporte.....	119
5.1.3. Descripción Detallada del Aporte.....	121
5.1.3.1. Introducción.....	121
5.1.3.2. Encuadramiento del modelo propuesto	122
5.1.3.3. Modelo propuesto.....	122
5.1.3.4. Componente 1. Procesos, Prácticas y Herramientas PSP	122
5.1.3.5. Componente 2. Procesos y Prácticas TSP	125
5.1.3.6. Componente 3. Procesos CMMi	127
5.1.3.7. Consideraciones sobre la implementación de modelo	130
5.1.4. Supuestos	130
5.1.5. Limitaciones	131
5.1.6. Alcance.....	131
5.2. Metodología de Implementación.....	131
5.2.1. Alcance.....	131
5.2.2. Descripción General	131
5.2.3. Descripción Específica.....	132
5.2.3.1. Capacitación.....	132
5.2.3.2. Lanzamiento y Seguimiento del Modelo	134
CONCLUSIONES.....	138
RECOMENDACIONES.....	140

FUTURAS INVESTIGACIONES	141
REFERENCIAS BIBLIOGRÁFICAS.....	142
ANEXOS	146
ANEXO A: Diagrama de Ishikawa	146
ANEXO B: Árbol de problemas	148
ANEXO C: Población de estudio	149
ANEXO D: Ficha de validación de instrumento de investigación	150
ANEXO E: Carta de presentación	151
ANEXO F: Cuestionario de diagnóstico.....	152
ANEXO G: Ficha de registro de datos	156
ANEXO H: Cuestionario	160
ANEXO I: Matriz de consistencia.....	161
ANEXO J: Modelo de aseguramiento de la calidad	163
ANEXO K: Formatos PSP	167
ANEXO L: Formatos TSP.....	170
ANEXO M: Checklist CMMi	177
ANEXO N: Glosario de Términos	181

LISTA DE CUADROS

Cuadro 2.1 Comparativo entre CMMI, PSP, TSP, ISO 9001-2000, ISO/IEC 15504, PMBOK.....	64
Cuadro 2.2 Formulario de criterios de selección del modelo.....	70
Cuadro 2.3 Operacionalización de variables.....	72
Cuadro 2.4 Indicadores de la variable dependiente.....	73
Cuadro 3.1 Simbología del diseño Pre Experimental.....	75
Cuadro 3.2 Criterios de selección de la muestra.....	78
Cuadro 3.3 Instrumentos de medición asociados al indicador.....	80
Cuadro 4.1 Prueba de normalidad indicador porcentaje de costo de calidad.....	97
Cuadro 4.2 Resultados de la prueba de T del indicador porcentaje de costo de Calidad.....	100
Cuadro 4.3 Prueba de normalidad indicador cantidad de defectos.....	100
Cuadro 4.4 Resultados de la prueba de T del indicador cantidad de defectos....	103
Cuadro 4.5 Prueba de normalidad indicador porcentaje de desviación de costo.	104
Cuadro 4.6 Resultados de la prueba de T del indicador porcentaje de desviación de costo.....	108
Cuadro 4.7 Prueba de normalidad indicador porcentaje de clientes satisfechos.	108
Cuadro 4.8 Resultados de la prueba de T del indicador porcentaje de desviación de costo.....	112

LISTA DE FIGURAS

Figura 1.1 Informe del Caos 2015.....	3
Figura 1.2 Resolución Caos para los años 2011-2015.....	4
Figura 1.3 Informe de Caos por Ciclo de Vida de Proyecto.....	5
Figura 2.1 Fases de la metodología de mejora de gestión de equipos.....	13
Figura 2.2 Modelos de rendimiento de proceso y TSP.....	16
Figura 2.3 Marco de procesos de software en una pequeña organización.....	17
Figura 2.4 Integración de los componentes de un proceso de desarrollo.....	22
Figura 2.5 Modelo de Sistema de Gestión de la Calidad basado en Procesos ISO 9001:2015.....	27
Figura 2.6 Proceso de gestión de la calidad PMBOK.....	33
Figura 2.7 Niveles de Madurez del Modelo CMMi.....	37
Figura 2.8 Flujo de proceso PSP.....	40
Figura 2.9 Etapas de PSP.....	42
Figura 2.10 Mapa Conceptual PSP.....	45
Figura 2.11 Mapa Conceptual TSP.....	51
Figura 2.12 Integración Modelos CMMi, TSP y PSP.....	52
Figura 2.13 Jerarquía Conceptual Modelos CMMi, TSP y PSP.....	53
Figura 2.14 Vista Conceptual Integración Modelos CMMi, TSP y PSP.....	53
Figura 2.15 Importancia de la mejora de procesos de software.....	56
Figura 3.1 Simbología del diseño pre-experimental	75
Figura 5.1 Modelo de aseguramiento de la calidad alto nivel.....	120
Figura 5.2 Modelo de aseguramiento de la calidad del proceso de desarrollo de software.....	121
Figura 5.3 Procesos PSP que integran el modelo de Aseguramiento de la Calidad en el Proceso de Desarrollo de Software.....	124
Figura 5.4 Procesos TSP que integran el modelo de Aseguramiento de la Calidad en el Proceso de Desarrollo de Software.....	126
Figura 5.5 Procesos CMMi que integran el modelo de Aseguramiento de la Calidad en el Proceso de Desarrollo de Software.....	128
Figura 5.6 Lanzamiento y seguimiento del modelo.....	134

LISTA DE GRÁFICOS

Gráfico 4.1. Tasa de respuesta.....	82
Gráfico 4.2. Qué metodología de desarrollo de software aplican habitualmente.....	83
Gráfico 4.3. Cuál es el esfuerzo promedio dedicado a cada fase del proyecto.....	84
Gráfico 4.4. Porcentaje de esfuerzo post-construcción.....	85
Gráfico 4.5. Qué tipos de herramientas dan soporte a su proceso de desarrollo..	86
Gráfico 4.6. Qué factores principales afectan la calidad.....	86
Gráfico 4.7. Políticas de Calidad a nivel de Empresa/Consultora.....	87
Gráfico 4.8. El modelo de proceso utilizado asegura la calidad.....	88
Gráfico 4.9. La metodología utilizada asegura la calidad del producto.....	88
Gráfico 4.10. Qué factores cree usted que aseguran la calidad.....	89
Gráfico 4.11. Su empresa y/o consultora posee gestión de defectos.....	90
Gráfico 4.12. Su empresa tiene definidos procesos de testing.....	90
Gráfico 4.13. Qué porcentaje de esfuerzo le llevan las siguientes actividades.....	91
Gráfico 4.14. Quién realiza las pruebas del software.....	92
Gráfico 4.15. Comparativo Indicador Porcentaje de Costo de Calidad.....	93
Gráfico 4.16. Comparativo Indicador Cantidad de Defectos por Proyecto.....	94
Gráfico 4.17. Comparativo Indicador Porcentaje de Desviación de Costo.....	95
Gráfico 4.18. Comparativo Indicador Grado de Satisfacción del Cliente.....	96
Gráfico 4.19. Q-Q de normalidad para datos del indicador porcentaje de costo de calidad – Prueba Previa.....	97
Gráfico 4.20. Q-Q de normalidad para datos del indicador porcentaje de costo de calidad – Post Prueba.....	98
Gráfico 4.21. Q-Q de normalidad para datos del indicador cantidad de defectos – Prueba Previa.....	101
Gráfico 4.22. Q-Q de normalidad para datos del indicador cantidad de defectos – Post Prueba.....	102
Gráfico 4.23. Q-Q de normalidad para datos del indicador porcentaje de desviación de costo – Prueba Previa.....	105
Gráfico 4.24. Q-Q de normalidad para datos del indicador porcentaje de desviación de costo – Post Prueba.....	106
Gráfico 4.25. Q-Q de normalidad para datos del indicador porcentaje de clientes satisfechos – Prueba Previa.....	109
Gráfico 4.26. Q-Q de normalidad para datos del indicador porcentaje de clientes satisfechos – Post Prueba.....	110
Gráfico 4.27. Porcentaje promedio de costo de calidad.....	113

Gráfico 4.28. Cantidad promedio de defectos.....	114
Gráfico 4.29. Cantidad promedio de defectos.....	115
Gráfico 4.30. Porcentaje promedio de clientes satisfechos.....	116

RESUMEN

En la actualidad las empresas que fabrican software se han convertido en socios estratégicos de sus clientes a través de la generación de valor, mediante la entrega de productos que soportan sus diversos procesos de negocio, mejorando su competitividad y facilitando su adaptación a los cambios del entorno.

Pero existen diversos problemas en los proyectos de desarrollo de software, el principal problema es el incumplimiento de los objetivos, es decir terminar el proyecto dentro del plazo y presupuesto planificado, alcance y la calidad prevista.

El objetivo de la investigación es generar e implementar un modelo de aseguramiento de la calidad en el proceso de desarrollo de software con un enfoque orientado al factor humano; factor que es responsable de introducir y gestionar la calidad en el proceso de desarrollo de software. El modelo propuesto se basa en tres modelos “Modelos de Madurez de Capacidades Integrado, Proceso de Software para Equipos y Proceso de Software para Personas”.

Palabras Clave: *Modelo de Madurez de Capacidades Integrado (CMMi), Proceso de Software Personal (PSP), Proceso de Software para Equipos (TSP), Calidad, Proceso, Modelo*

SUMMARY

At present the companies which manufacture software have become strategic partners of their customers through value creation by delivering products that support its various business processes, improving their competitiveness and facilitating their adaptation to the changing environment.

But there are several problems in software development projects; the main problem is the failure of the objectives, which means to complete the project within the planned time and budget, scope and expected quality.

The objective of the research is to generate and implement a model of quality assurance in the software development process with a focus oriented human factor; factor that is responsible for introducing and managing quality in the software development process. The proposed model is based on three models "Capability Maturity Model Integration, Team Software Process and Personal Software Process".

Keywords: *Maturity Model Integrated Capabilities (CMMi), Personal Software Process (TSP), Team Software Process (PSP), Quality, Process, Model.*

CAPITULO I: INTRODUCCIÓN

1.1. Situación Problemática

El desarrollo de software es una actividad reciente, comparada con las demás ingenierías. En sus inicios, ésta actividad se desarrolló con base en habilidades personales y con la firme creencia de que su naturaleza era artesanal. La falta de procesos, modelos, metodologías, la indisciplina personal y la falta de visión para conceptualizar al desarrollo de software como una ingeniería se materializó en la crisis del software en los años 70 y desde entonces se han tomado acciones para cambiar las malas prácticas y considerar al desarrollo de software como una ingeniería.

El término de “crisis del software” comenzó a ser usado a finales de los años 60 para definir los diversos problemas que se estaban encontrando dentro de los proyectos de software, donde solía ocurrir que los diversos proyectos de desarrollo de software terminaban sobrepasando dramáticamente el tiempo y costo previstos al inicio del proyecto o que sencillamente la calidad del producto no fuera aceptable. A partir de ese momento se comenzó a concebir la idea de la ingeniería del software, una disciplina que buscaba ver la creación de software como un proceso de producción sistematizado que facilitara la entrega de productos de calidad que satisfaga a clientes y usuarios.

Una de las principales causas por la cual sucedió la “crisis del software” fue la inmadurez de la industria, pero con la proliferación de la computación personal y el gran uso que hoy en día se le está dando a los equipos inteligentes (teléfonos inteligentes o

tabletas) la industria del software representa un mercado de más de 300 mil millones de dólares y con productos de adopción mundial como (Windows, SAP, Facebook, Android).

En el presente trabajo de investigación se hace una distinción entre dos grandes tipos de productos de software, los productos genéricos, también llamados software pre-empacado, y productos de desarrollo a la medida. La principal diferencia entre estos productos radica en el origen de los requerimientos, siendo en el primer caso impuestos por la misma empresa que desarrolla y en el segundo caso los requerimientos los definen los usuarios.

La adopción de principios de ingeniería del software, el uso de cadenas de producción para el desarrollo y la estrategia de llevar al máximo la reutilización de código, ha llevado al éxito a las compañías especializadas en software pre-empacado, hay estudios que revelan como en este tipo de productos los costos y tiempo de desarrollo se han venido reduciendo a lo largo de los años, igualmente muestran como en el caso de los productos personalizados los costos y el tiempo han tendido a aumentar (Yu, 2009). El presente trabajo de investigación se orienta a la industria del software que desarrolla productos a medida (cuyos requerimientos son definidos por los usuarios.)

Las principales causas de que aun hoy haya fallas en los desarrollos de software radica en, primero, la informalidad con la que se ha tratado la programación desde sus inicios, recordemos que el desarrollo de software en sus primeros años tenía un enfoque científico y hasta a veces artístico, dada a la facilidad que representa solo enfocarse a la programación, y en segundo lugar los cambios constantes que pueden sufrir los requerimientos del mercado, exigiendo así cada vez sistemas más grandes y complejos (Sommerville, 2010).

Los principales retos actuales para la ingeniería del software es garantizar la calidad en el proceso de desarrollo, siendo el caso de los proyectos cuyos requerimientos son gobernados por los clientes, esto debido a los cambios del entorno en general que se pueden llegar a materializar durante la ejecución del proyecto, incrementando la

complejidad del desarrollo al incluir nuevas tareas no contempladas al inicio del desarrollo. Es por ello que es labor del equipo y el ingeniero de software hacer concesiones para adaptar los lineamientos y alcances del proyecto, tratar de reducir la complejidad asociada a los problemas que surjan durante el proyecto y poder enfocar los esfuerzos a los problemas esenciales o asociados a los requerimientos (planificación, disciplina y cultura) soportándose sobre un modelo de aseguramiento de calidad.

Ciertamente la situación actual de la industria del software indica que aquel problema llamado “crisis del software” que venía afectando desde mediados del siglo pasado a los grandes proyectos de software ha quedado atrás. Aun así recordemos que el mundo de la tecnología y el software es altamente cambiante y en un mundo altamente conectado con gran demanda por productos y servicios tecnológicos y una capacidad de generar grandes volúmenes de información y datos, hacen que muchos preñan las alarmas de alerta sobre una segunda crisis del software, pero así como cambia y se mejoran los procesos de ingeniería clásicos para mejorar la tecnología y los procesos de desarrollo, aún hay mucho espacio y oportunidad para mejorar en la ingeniería del software, permitiendo que el uso de buenas prácticas, modelos y estándares de desarrollo que favorezcan el poder llevar adelante grandes proyectos de la ingeniería de software garantizando el cumplimiento de sus objetivos.

A continuación se presentan los resultados del estudio realizado por “Standish Group Survey”, en su publicación “CHAOS MANIFESTO 2015”.



Figura 1.1 Informe del Caos 2015

Fuente. Datos tomados de The Standish Group International – Chaos Report 2015

El informe del caos, estudio realizado en el año 2015 muestran un avance en el porcentaje de éxito de los proyectos respecto al año 2014, con un 29% de todos los proyectos (entregado a tiempo, dentro del presupuesto, con las características y funciones necesarias), el 52% fueron comprometidos, cuestionados o sufrieron cambios (fuera de tiempo, por encima del presupuesto, y/o con menos de las características requeridas y las funciones) y el 19% fracasaron (cancelado antes de su finalización o entregado y nunca se usa). Estas cifras representan un avance de 1% en proyectos exitosos respecto al estudio anterior, en proyectos que tuvieron algún tipo de problema se redujo en 3% respecto al estudio anterior y hubo un incremento de 2% en proyectos fracasados teniendo como referencia el estudio anterior. Basandonos en las estadísticas contenidas en el informe se puede indicar que no existe mayor variación respecto a los informes anteriores (2011 al 2015). Pero es oportuno mencionar que hubo un incremento en el porcentaje de proyectos fallidos.

MODERN RESOLUTION FOR ALL PROJECTS					
	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Figura 1.2 Resolución Caos para los años 2011-2015

Fuente. Datos tomados de The Standish Group International – Chaos Report 2015

En la siguiente figura se presenta un comparativo entre dos marcos de trabajo (ciclos de vida) más utilizados en el desarrollo de software, Ágil versus Cascada. Se coloca como referencia esta información en nuestra problemática porque en los ciclos de vida ágiles existe una mayor interacción entre integrantes del equipo, colaboración con el cliente y respuesta frente al cambio.

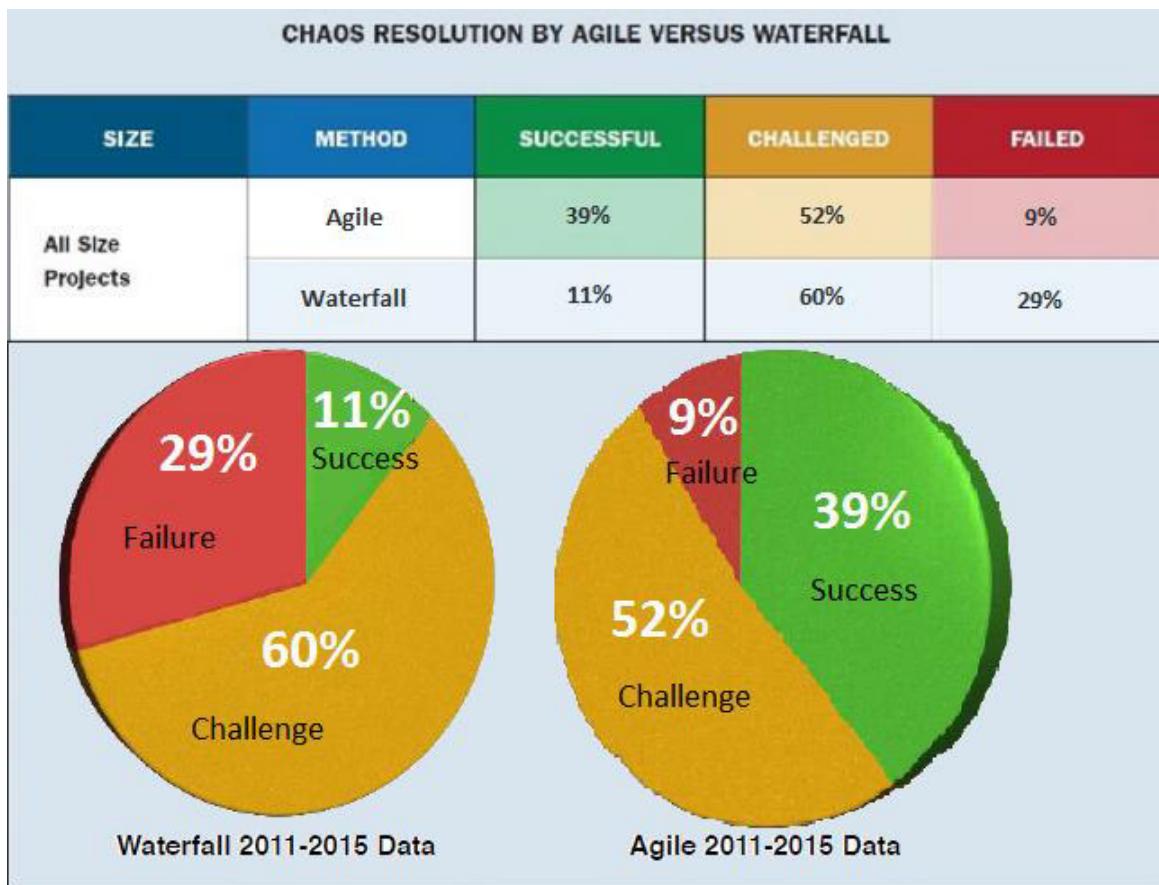


Figura 1.3 Informe de Caos por Ciclo de Vida de Proyecto

Fuente. Datos tomados de The Standish Group International – Chaos Report (2015)

Aplicar Agile (marco de trabajo ágil) aumenta el porcentaje de éxito del 11% al 39% y disminuye el porcentaje de fracaso del 29% al 9%. Este aumento de proyectos exitosos utilizando el marco Agile se fundamenta básicamente en la interacción entre los ingenieros de software (equipos auto-gestionados), colaboración eficaz con el cliente y respuesta a los cambios producidos en el entorno contemplados en sus iteraciones cortas.

Con esta problemática que adolece el proceso de desarrollo de software, se ha considerado oportuno y adecuado tomar algunas medidas correctivas que orienten los esfuerzos que despliegan las empresas en obtener los objetivos esperados, acción que a través de nuestra especialidad profesional, nos centraremos en generar un modelo que nos permitan el aseguramiento de la calidad del proceso de desarrollo de software, donde el talento humano sea uno de los principales factores de éxito.

Considerando los esfuerzos desplegados a la fecha por diversas organizaciones a nivel mundial, los resultados continúan siendo desalentadores, por ello la presente investigación se enfoca en el factor humano; factor que es determinante para lograr el éxito de los proyectos, puesto que son los responsables de introducir y gestionar la calidad en el proceso de desarrollo de software.

Para ello se necesita que el factor humano de forma individual o grupal consiga interiorizar y fomentar una cultura de calidad, la cual se ponga de manifiesto en el desarrollo de trabajo diario.

1.2. Formulación del Problema

1.2.1. Problema General

En base a la situación problemática surge la siguiente pregunta *¿De qué manera la integración de un modelo basado en los modelos de Madurez de Capacidades Integrado, Procesos de Software de Equipos y Proceso de Software de Personas permite asegurar la calidad en el proceso de desarrollo de software?*

1.2.2. Problemas Específicos

Para tal efecto se determina los siguientes problemas específicos:

¿De qué manera la integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software reduce el costo de calidad en los proyectos?

¿De qué manera la integración del modelo de aseguramiento de calidad reduce los defectos en los proyectos incurridos en el proceso de desarrollo de software?

¿De qué manera la integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software incrementa la rentabilidad de los proyectos?

¿De qué manera la integración del modelo de aseguramiento mejora la satisfacción de los clientes?

Partiendo de la premisa de que existen n factores que afectan la calidad en el proceso de desarrollo de software, basados en nuestra problemática nos orientaremos a los factores como los equipos de trabajo y a las personas (ingenieros) y los modelos de referencia asociados a estos últimos, los cuales consideramos deben ser el punto de partida para asegurar la calidad en el proceso de desarrollo.

Por tanto la presente investigación se orienta a asegurar la calidad en proceso de desarrollo en aquellas consultoras y empresas que desarrollan software para consumo interno (In-House) que posean procesos de desarrollo definidos e institucionalizados (Certificación de Nivel de Madurez CMMI nivel 2 y 3).

Se está tomando como universo a todas las consultoras y empresas de desarrollo (In-House) certificadas en CMMI nivel de madurez 2 y 3 del departamento de Lima – Perú obtenidas desde el portal de consultas del CMMI Institute. (CMMI Institute Published Appraisal Results, 2013).

1.3. Justificación teórica

La presente investigación es importante porque se enfoca en el proceso de desarrollo de software, una actividad que tiene muchos problemas, según el último reporte “**Resumen del Caos 2015**” del The Grupo Standish International se indica que tan solo el 29% de los proyectos evaluados fueron exitosos, es decir cumplieron los objetivos del proyecto (Tiempo, Alcance, Costo y Calidad).

En los últimos años se realizaron diversos esfuerzos por mejorar el proceso de desarrollo de software, se integraron prácticas universales de gestión de proyectos de TI, se refinaron los ciclos de vida de software, la asociación de estandarización "IEEE" liberó el libro de conocimiento de la ingeniería de software (SWBOK), adicionalmente se crearon diversos marcos de trabajo CMMI, ISO 12207, etc., metodologías de desarrollo de software pero desafortunadamente todos los esfuerzos antes mencionados no han sido suficientes.

Como se mencionó anteriormente, existen diversos modelos de aseguramiento de la calidad orientados al proceso. El objetivo de la presente investigación es generar un modelo de aseguramiento de la calidad del proceso de desarrollo que se orienten al equipo de desarrollo y al ingeniero, partiendo de la premisa que el factor humano es el activo más importante de las organizaciones.

Con la integración de los modelos orientados al proceso, equipo e ingeniero que aseguran la calidad en el proceso de desarrollo de software se espera tener los siguientes beneficios:

Beneficios para el Ingeniero:

- Producir software usando un enfoque estructurado y disciplinado.
- Administrar la calidad de los productos y aplican una retroalimentación (feedback) cuantitativa para mejorar sus procesos personales de trabajo, obteniendo así:
 - Mejores estimaciones.
 - Mejor planificación y seguimiento.
 - Protección contra compromisos que nunca se cumplen.
 - Un compromiso personal hacia la calidad.
 - Involucrarse en un proceso de mejoramiento continuo.
- Entregar valor a los clientes.

Beneficios para el Equipo

- Equipos de desarrollo de alto rendimiento.
- Mejorar la productividad del equipo.
- Entregar valor a los clientes.
- Proporcionar equipos de proyectos con guías explícitas sobre cómo alcanzar sus objetivos.
- Construir equipos autosuficientes que planifiquen y documenten su trabajo, estableciendo metas además de sus progresos y planificaciones.

1.4. Justificación práctica

La presente investigación tiene como objetivo principal elaborar un modelo para el aseguramiento de la calidad en el proceso de desarrollo de software. La adopción del presente modelo en las consultoras y empresas de desarrollo de software permite asegurar la calidad en el proceso de desarrollo de software, centrándose principalmente en el factor humano (Equipo e Ingeniero).

Adicionalmente los diversos marcos de referencia sobre calidad, sistemas de calidad, aseguramiento de la calidad no consideran a los ingenieros y equipos de software como factores elementales en el aseguramiento de la calidad en el proceso de desarrollo de software.

El producto de la presente investigación puede ser utilizado e integrado al proceso de desarrollo de las empresas y consultoras que cuenten con certificación CMMi nivel 2 o 3.

Asimismo es preciso indicar que los procesos de los modelos de madurez de capacidad integrado, proceso de software para equipos y proceso de software personal se integran y complementan sin que exista un traslapé o superposición entre sus procesos.

1.5. Objetivos

1.5.1. Objetivo General

- Generar un modelo que permita asegurar la calidad en el proceso de desarrollo de software basado en los modelos de madurez de capacidades, Proceso de Software de Equipos y Proceso de Software de Personas.

1.5.2. Objetivos Específicos

- Reducir el costo de calidad en los proyectos a través del aseguramiento de la calidad en el proceso de desarrollo de software.
- Reducir los defectos en los proyectos a través del aseguramiento de la calidad en el proceso de desarrollo de software.
- Incrementar la rentabilidad de los proyectos a través del aseguramiento de la calidad en el proceso de desarrollo de software.
- Aumentar el grado de satisfacción de los clientes a través del aseguramiento de la calidad en el proceso de desarrollo de software.

CAPITULO II: MARCO TEÓRICO

2.1. Marco Filosófico o epistemológico de la investigación

La **epistemología** es una disciplina que estudia **cómo se genera y se valida** el **conocimiento** de las ciencias. Su función es analizar los preceptos que se emplean para justificar los datos científicos obtenidos, considerando diversos factores como los sociales, psicológicos y hasta los factores históricos que intervengan.

La presente investigación tiene un diseño transversal que nos permite analizar la relación entre un conjunto de variables en un determinado punto del tiempo. Para ello se hizo uso de un procedimiento de comparaciones para determinar el impacto de las variables independientes sobre la variable dependiente, y para la validación de dicho conocimiento se hizo uso de métodos cuantitativos y cualitativos sobre los datos obtenidos del trabajo de campo y documentación de la oficina de gestión de proyectos.

Así mismo en la presente investigación se incorpora al ser humano como factor principal y elemental para lograr la calidad en el proceso de desarrollo de software.

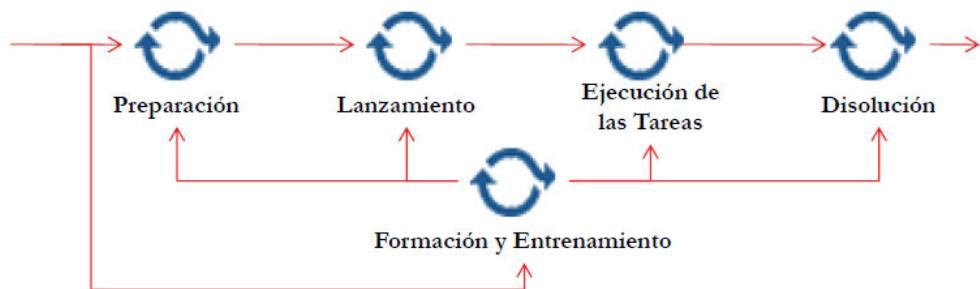
2.2. Antecedentes de Investigación

2.2.1. Tesis Doctoral: Un marco metodológico para la mejora en la gestión de los equipos de desarrollo de software global

- **Autor:** Javier Saldaña Ramos – Departamento de Informática, Universidad Carlos III de Madrid (2010).
- **Objetivo:** Definir una metodología específica para la gestión de equipos de desarrollo software global, que describa de forma detallada las actividades y tareas que tienen que ser realizadas por los miembros del equipo y el orden en el que éstas tienen que realizarse, así como los diferentes roles que participan en el desarrollo de las mismas. La metodología definida trata de reducir tanto los problemas de colaboración y coordinación entre los miembros del equipo de desarrollo software global como de gestión del conocimiento, facilitando la transferencia de conocimiento y evitando que éste se encuentre repetido en diferentes lugares.
- **Solución Propuesta:** La solución que plantea esta tesis doctoral es una solución práctica a la gestión de equipos de desarrollo software global. Esta solución permitirá conocer a cada uno de los integrantes del proyecto la tarea que deben desempeñar en cada fase del mismo, así como las habilidades y capacidades que debe tener los participantes en la misma y, las tecnologías que pueden facilitar la ejecución de la tarea en el entorno distribuido. La solución propuesta se basa en la definición de un marco metodológico que:
 - ✓ Incluya la definición de un conjunto de buenas prácticas que mejoran la gestión de los equipos de desarrollo software global.

- ✓ Identifica el conjunto de las competencias requeridas para incrementar el rendimiento del equipo en el desarrollo de las tareas, y mejora la eficiencia de la gestión del mismo.
- ✓ Define mecanismos de colaboración y las capacidades tecnológicas que facilitan la coordinación, comunicación y colaboración entre los integrantes del equipo de desarrollo software global.
- ✓ Constituye un marco de referencia para llevar a cabo procesos de evaluación y mejora en la organización, en relación a la gestión de los equipos de desarrollo software global.

Figura 2.1. Fases de la metodología de mejora de gestión de equipos



Fuente. Datos tomados de

http://e-archivo.uc3m.es/bitstream/handle/10016/10286/Tesis_Javier_Saldana.pdf

- **Conclusiones:** La implementación de las actividades descritas en el marco metodológico propuesto en la tesis doctoral permitió mejorar la eficiencia en los equipos de desarrollo de software global e incrementar el rendimiento de los mismos, contribuyendo, por tanto, a incrementar la posibilidad de éxito del proyecto de desarrollo de software global.
- **Relación con la Tesis de Investigación:** Ambos trabajos de investigación coinciden que mejorando la gestión de equipos se logra ser más eficientes y se logra incrementar el rendimiento de los equipos de desarrollo de software. Adicionalmente nos permite

disponer de personal mejor calificado facilitando la consolidación y desarrollo de buenas prácticas.

2.2.2. Tesis Maestría: Estudio comparativo de los modelos y estándares de calidad del software

- **Autor:** Lic. Fernanda Scalone – Universidad Tecnológica Nacional Facultad Regional Buenos Aires (2006).
- **Conclusiones:** Respecto de la Calidad del Software, se puede decir que el software juega un papel muy importante para el desarrollo de las organizaciones, ya que sirve de soporte a los procesos de negocio, productivos y administrativos; y como parte integral de las estrategias corporativas para la generación de ventajas competitivas. Esto significa que resulta fundamental evaluar la Calidad del Software. Para el logro de esta Calidad será necesario efectuar una Gestión de la Calidad del Software, la cual consiste en un conjunto de actividades que permiten dirigir y controlar la organización en lo relativo a la Calidad del Software. Esta Gestión de la Calidad del Software está formada por la Planificación de la Calidad del Software, el Control de Calidad del Software, el Aseguramiento de la Calidad del Software y el Mejoramiento de la Calidad del Software.
- **Recomendaciones:**
 - ✓ Capacitar a todo el personal de la empresa respecto de la Filosofía de Calidad del Software.
 - ✓ Dar a conocer los objetivos de la empresa a todo el personal.
 - ✓ Armar grupos de trabajo eficaz y eficiente.
 - ✓ Evaluar y controlar periódicamente los procesos de la empresa y sus productos o resultados asociados.

- ✓ Establecer el mejoramiento continuo de los procesos de negocio de la empresa y de sus productos o servicios.
 - ✓ Mejorar la administración de los recursos humanos, materiales, tiempos y costos.
 - ✓ Mejorar la competitividad de la empresa.
-
- **Relación con la Tesis de Investigación:** Ambas investigaciones coinciden en que asegurando la calidad en el proceso de desarrollo de software se está generando valor, porque permite a las organizaciones mejorar sus procesos y fundamentalmente alinear TI con los objetivos del negocio.

2.2.3. Nota Técnica: La integración de CMMI y TSP/PSP: El uso de TSP de datos para crear modelos de rendimiento de proceso

- **Autor:** Shurei Tamura - Software Engineering Measurement and Analysis (2009).

- **Resumen:** Equipos de TSP recogen y utilizan medidas detalladas de tamaño, defectos, el esfuerzo, la programación y re-trabajo. Estas medidas proporcionan información valiosa sobre el rendimiento y el estado del proyecto y sirven de base para la predicción de varios aspectos de la finalización del proyecto.

Como se muestra en la Figura 2.2, un depósito de datos que incluye datos de TSP proporciona la información necesaria para crear los PMP. Equipos de TSP definen los procesos del proyecto que especifican cómo y cuándo los datos se recogen, y en equipo TSP miembros entrenados en PSP-Comprender y utilizar datos para la planificación y la realización de su trabajo. Por lo tanto, los equipos de TSP pueden proporcionar alta calidad, datos de grano fino, que

puede ser una excelente base para la creación y uso de los Modelos de Rendimiento de Proceso.

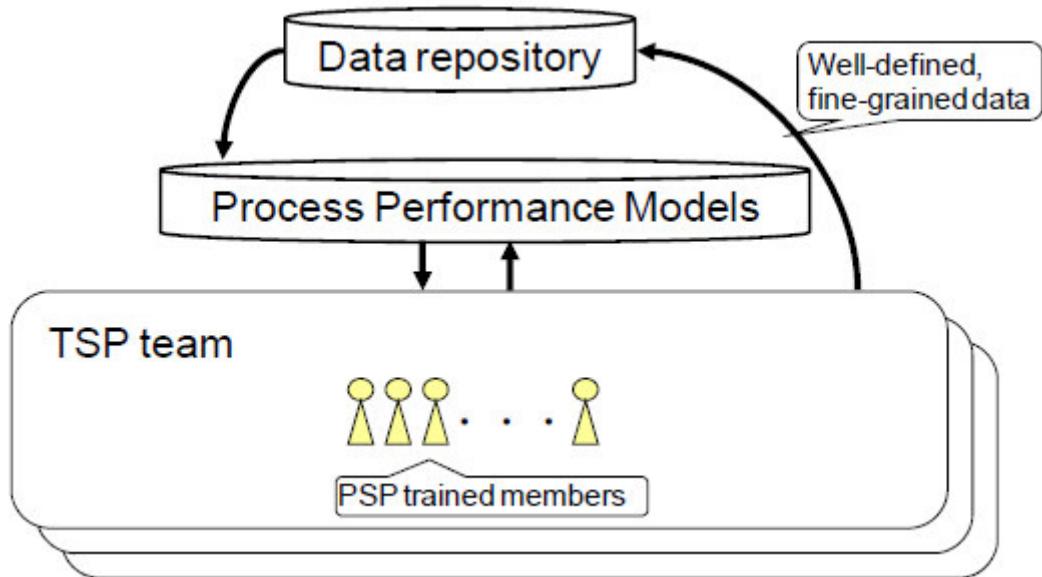


Figura 2.2 Modelos de rendimiento de proceso y TSP

Fuente. Datos tomados de

http://resources.sei.cmu.edu/asset_files/TechnicalNote/2009_004_001_15089.pdf

- **Relación con la Tesis de Investigación:** La nota técnica muestra la importancia de contar con información histórica sobre el trabajo diario de los ingenieros de software y su uso para la estimación, calidad y planificación de proyectos futuros. Así mismo se hace hincapié en la importancia de contar con ingenieros y equipos formados para realizar trabajo de desarrollo de software. La presente investigación se basa en la aplicación de modelos orientados a personas y equipos que desarrollan software dentro de un marco de proceso definido (CMMi) tal como lo presenta el modelo de la nota técnica.

2.2.4. Nota Técnica: Mejora de Procesos de Software para Pequeñas Organizaciones Basado en CMMI / TSP / PSP

- **Autor:** Zhang Lina - School of Computer Science, Henan Polytechnic University, P.R.China.
- **Conclusiones:** El CMMI, PSP, TSP y proporcionan un marco tridimensional integrado para la mejora de procesos en una organización pequeña. Estos métodos no sólo ayudan a los ingenieros a ser más eficaz, pero también proporcionan la comprensión profunda necesaria para acelerar la mejora de procesos de la organización.

El proceso de mejora de software es un proceso gradual, no un evento. En la ejecución del proceso de mejora de proceso de software bajo CMMI / PSP / TSP, primero es el estudio y la comprensión de la teoría a fondo, de acuerdo con sus propias condiciones reales para el desarrollo de programas viables, y en la práctica resumir constantemente experiencia y encontrar su camino.

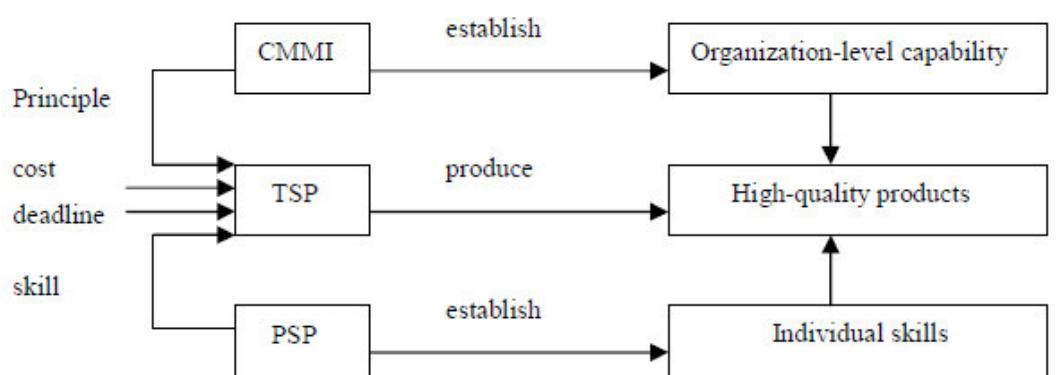


Figura 2.3 Marco de procesos de software en una pequeña organización

Fuente. Datos tomados de

<http://www.seiofbluemountain.com/upload/product/201001/1264727480s3xvkd4g.pdf>

- **Relación con la Tesis de Investigación:** La nota técnica muestra un marco tridimensional compuesto por los modelos CMMI, TSP y PSP los cuales de manera articulada mejoran los procesos de las organizaciones de desarrollo de software. La presente investigación hará uso de los tres modelos para asegurar la calidad del proceso en el desarrollo de software, visto desde otro punto, basado en el marco tridimensional de PROCESO, EQUIPO e INGENIERO.

2.2.5. Tesis Maestría: Integración entre PSP y PMBOK aplicada al desarrollo de un sistema experto para el diagnóstico e identificación automática de enfermedades profesionales

- **Autor:** Lic. Claudia Marcela Ramírez Zuluaga, Universidad Autónoma de Manizales, Maestría en gestión y desarrollo de proyectos de software Manizales, 2011.

- **Conclusiones:**

Las organizaciones donde se desarrolla software y que aplican las prácticas del PMBOK® pueden verse beneficiadas con esta trazabilidad ya que en ella se define explícitamente el COMO hacer las prácticas que en el PMBOK® se establece de manera genérica el QUE debe hacerse.

La aplicación de prácticas de PSP y PMBOK® ayuda a desarrollar productos de software de calidad ya que enseñan y guían a los ingenieros a administrar el tiempo y los recursos, realizar mejores estimaciones y planes de trabajo, medir y prevenir los defectos, y comprometerse con la calidad del producto.

La calidad es el principal objetivo cuando se desarrolla una aplicación y como lo comenta Humphrey: "la calidad de un producto de desarrollo de software depende de la calidad del proceso que

se utiliza para hacerlo”, si bien el marco propuesto puede parecer muy extenso este se ve recompensado en los atributos de calidad del producto obtenido lo cual es beneficioso no solo para el desarrollador sino también para la organización.

- **Relación con la Tesis de Investigación:** La importancia de los modelos PSP y TSP en el proceso de desarrollo, ya que dichos modelos guían a los ingenieros de software a comprometerse con la calidad en cada etapa o fase del proceso de desarrollo.

- **Recomendaciones:**

El proceso de trazabilidad que fue desarrollado en este proyecto podría ser más sencillo aplicándolo con TSP ya que se puede adaptar más fácil el proceso a la guía del PMBOK® por tener, TSP, un marco más general aplicado a equipos de trabajo.

El conjunto de prácticas propuesto en este proyecto puede aplicarse al desarrollo de proyectos académicos para obtener una base de datos estadística que pueda ser analizada desde la perspectiva de la ingeniería de software experimental y evaluar su efectividad práctica en la creación de aplicaciones computacionales.

2.2.6. Tesis Maestría: Agilizando lo Ágil: Un Framework para el desarrollo de software bajo el modelo CMMI en Compañías que usan metodologías ágiles de desarrollo de software usando el Modelo Acelerado de Implementación (AIM)

- **Autor:** Lic. María Eugenia Rojas Izaquita, Universidad Nacional de Colombia, 2011.

- **Conclusiones:**

El modelo CMMI permite a las organizaciones tener un panorama de ¿Qué se debe hacer? pero como lo mencionamos al inicio de este documento no define el ¿Cómo hacer las cosas? La interpretación del modelo y la forma de implementarlo queda abierta a cada organización en particular, por esto es de vital importancia contar un equipo de procesos que oriente las interpretaciones y las definiciones.

El modelo de implementación AIM parte de la premisa de que en cada organización se pueden conformar grupos maduros que inicien el proceso de implementación de CMMI y sean ellos con su experiencia, ejemplo y madurez ayuden a agilizar el proceso de implementación.

- **Relación con la Tesis de Investigación:**

Las características de TSP y PSP que lo hacen compatible con las prácticas de CMMI, en donde el efecto total esperado es entregar la funcionalidad requerida sin defectos, a tiempo y dentro del presupuesto establecido son:

1. Entrenamiento en Proceso Personal de Software (PSP).
2. Mediciones dentro del ámbito de TSP.
3. Entrenamiento y equipos auto-dirigidos con técnicas de TSP.
4. Integración del ciclo de vida de desarrollo con las prácticas de calidad.
5. Un equipo de proyecto centrado en la estrategia de mejora.

- **Recomendaciones:**

Si bien el proceso de implementación del modelo CMMI puede ser costoso, es necesario revisar las estrategias para que se puedan

llevar a cabo estos procesos de forma tal que cada unidad monetaria invertida pueda tener una retribución en el mediano plazo.

2.2.7. Tesis Maestría: Bug Manager: Introduciendo Calidad a las Organizaciones de Software

- **Autor:** Lic. Marco Antonio Rangel Bocardo, Instituto Tecnológico y de Estudios Superiores de Monterrey, Campos Guadalajara, 2011.

- **Conclusiones:** El producto Bug Manager permitirá que las empresas tengan control sobre el avance y calidad de sus proyectos, por medio de las siguientes estrategias:
 1. Guía en la elaboración del plan de calidad.
 2. Definición del ciclo de vida y actividades de desarrollo.
 3. Registro y seguimiento de actividades de aseguramiento de calidad.
 4. Registro y seguimiento de defectos.
 5. Generación de estadísticas personales, por proyecto, por equipo y por empresa.

- **Relación con la Tesis de Investigación:** Ambas trabajos de investigación consideran como elementos claves las siguientes actividad para el aseguramiento de la calidad en los procesos de desarrollo de software:
 1. Registro de actividades con su esfuerzo y tamaño dedicados.
 2. Registro y seguimiento de defectos encontrados.
 3. Realización de revisiones de código.
 4. Medición de la productividad personal y global de la empresa.

2.3. Bases Teóricas

El presente trabajo de investigación tiene como objetivo elaborar un modelo de aseguramiento de la calidad del proceso de desarrollo de software, por tanto en el presente acápite se detalla los componentes de un proceso de desarrollo de software y los diversos modelos - marcos de referencia que aseguran la calidad del proceso.

La figura N° 2.4 nos muestra los tres componentes “Proceso, Tecnología y Personas” del proceso de desarrollo y como una adecuada integración soportada en modelos nos aportará calidad al proceso generando valor y permitiendo alcanzar los objetivos.

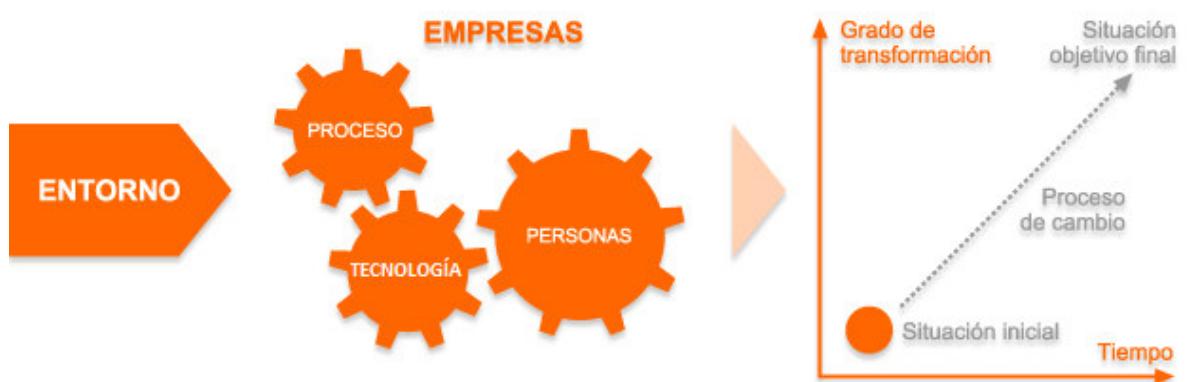


Figura 2.4 Integración de los componentes de un proceso de desarrollo

Fuente. Elaboración propia

2.3.1. Modelos de Aseguramiento de la Calidad

2.3.1.1. Modelo de Referencia

Un modelo de referencia es un conjunto de prácticas de un tema de interés específico (software, servicios, proyectos, tercerización, relacionamiento con cliente, gestión humana, gestión financiera, IT, etc.), las cuales han sido producto de investigación de muchos años y han sido probadas en diferentes empresas, lo cual las convierte en un

referente de mejores prácticas para que otras organizaciones las utilicen en el mejoramiento de sus procesos.

Los modelos de referencia, a diferencia de las metodologías, no especifican cómo ejecutar las prácticas sino que dan un marco general de qué se debe tener en cuenta para implementarlas. Es responsabilidad de la organización definir, según su naturaleza, cultura, funciones, servicios y negocios, cómo aplicar dichas prácticas.

Teniendo en cuenta la importancia que tiene para las organizaciones una orientación por procesos y considerando la existencia de estos modelos de referencia, las organizaciones crean iniciativas o proyectos de mejoramiento de procesos con el objetivo de estandarizar sus procesos tomando como referencia éstas prácticas reconocidas (SIE CENTER, 2011).

Modelo de referencia es una colección estructurada de elementos que describen las características de un proceso efectivo. Los procesos incluidos en un modelo son los que en la experiencia han demostrado ser efectivos (SIE CENTER, 2011).

Los beneficios de usar modelos de referencia son los siguientes:

- Mejorar tiempos de entrega
- Mejorar la productividad
- Mejorar la calidad
- Mejorar la satisfacción del cliente
- Mejorar el nivel de vida de los empleados
- Reducir el costo de la calidad
- Incrementar el retorno de la inversión

2.3.1.2. ISO/IEC 15504

Descripción del Estándar

La norma internacional ISO/IEC 15504:2004 fue desarrollada inicialmente por el proyecto denominado SPICE que se orientó a elaborar normas para la evaluación de los procesos software. Sin embargo después de la primera publicación, la norma ISO/IEC 15504:2004 se convirtió en una norma para evaluar procesos en general. La norma está orientada a los procesos y se integra por cinco partes (Conceptos y vocabularios, Realizando una evaluación, guía para la realización de evaluaciones, guía para el uso de la mejora y determinación de la capacidad de procesos y un ejemplar de modelo de evaluación de proceso.

ISO/IEC 15504 es un emergente estándar internacional de evaluación y determinación de la capacidad y mejora continua de procesos de ingeniería del software, con la filosofía de desarrollar un conjunto de medidas de capacidad estructuradas para todos los procesos del ciclo de vida y para todos los participantes. Es el resultado de un esfuerzo internacional de trabajo y colaboración y tiene la innovación, en comparación con otros modelos, del proceso paralelo de evaluación empírica del resultado. Norma que trata los procesos de ingeniería, gestión, relación cliente-proveedor, de la organización y del soporte. Se creó por la alta competencia del mercado de desarrollo de software, a la difícil tarea de identificar los riesgos, cumplir con el calendario, controlar los costos y mejorar la eficiencia y calidad. Este engloba un modelo de referencia para los procesos y sus potencialidades sobre la base de la experiencia de compañías grandes, medianas y pequeñas.

Estructura del Estándar

ISO/IEC desarrolla un modelo 2D de evaluación de la capacidad del proceso, donde se valora la organización de desarrollo software en

la dimensión del proceso contra los atributos del proceso en la dimensión de capacidad. La primera versión estructuraba el modelo en nueve partes, pero en el curso de los debates y votaciones, en aras de reducir el tamaño del estándar, se decide que se divide en cinco partes:

- Parte 1: Conceptos y vocabulario.
- Parte 2: Realización de una evaluación.
- Parte 3: Guía para la realización de una evaluación.
- Parte 4: Guía en uso para la mejora y determinación de la capacidad del proceso.
- Parte 5: Un ejemplo de modelo de evolución del proceso

Ventajas del ISO/IEC 15504

- Primer modelo de procesos de dos dimensiones, es decir, dimensiones independientes para los procesos y la capacidad.
- El resultado de una evaluación de proceso puede ser representado por un perfil de proceso.
- Define un conjunto de criterios de conformidad para permitir la comparación de modelos externos de procesos y encontrar requisitos comunes.

Desventajas del ISO/IEC 15504

- Permite que el dominio de proceso sea tan amplio para abarcar todos los posibles ciclos de vida, de tal manera que hace difícil que todos los atributos del proceso sean universales, provocando dificultad y confusión durante la evaluación.
- La dimensión de capacidad ha alcanzado un alto grado de dificultad y existen solapamientos con la dimensión de procesos.
- La complejidad de las evaluaciones y por consiguiente el costo es significativamente alto. (ISO IEC 15504-1, 2004)

2.3.1.3. ISO 9001-2015

Resultados consistentes y predecibles se alcanzan de manera más eficaz y eficientemente cuando las actividades se comprenden y gestionan como procesos interrelacionados que funcionan como un sistema coherente. La Norma Internacional ISO 9001:2015 promueve la adopción de un enfoque basado en procesos cuando se desarrolla, implementa y mejora la eficacia de un sistema de gestión de la calidad, para mejorar la satisfacción del cliente mediante el cumplimiento de los requisitos del cliente.

En el enfoque de proceso se aplica la definición sistemática y la gestión de los procesos, así como sus interacciones, con el fin de lograr los resultados previstos de acuerdo con la política de calidad y la dirección estratégica de la organización. La gestión de los procesos y el sistema en su conjunto pueden ser logrados mediante la metodología (PDCA) "Planificar-Hacer-Verificar-Actuar" con un enfoque global sobre el "pensamiento basado en el riesgo ", para prevenir "resultados no deseables". Cuando se utiliza dentro de un sistema de gestión de calidad, el enfoque de procesos garantiza:

- La consistente comprensión y cumplimiento de los requisitos
- La consideración de los procesos en términos de valor agregado.
- El logro de un desempeño eficaz del proceso
- Mejora de los procesos, mediante en la evaluación de datos e información.

El modelo esquemático mostrado en la figura 2.5 cubre todos los requisitos de la norma internacional ISO 9001:2015.

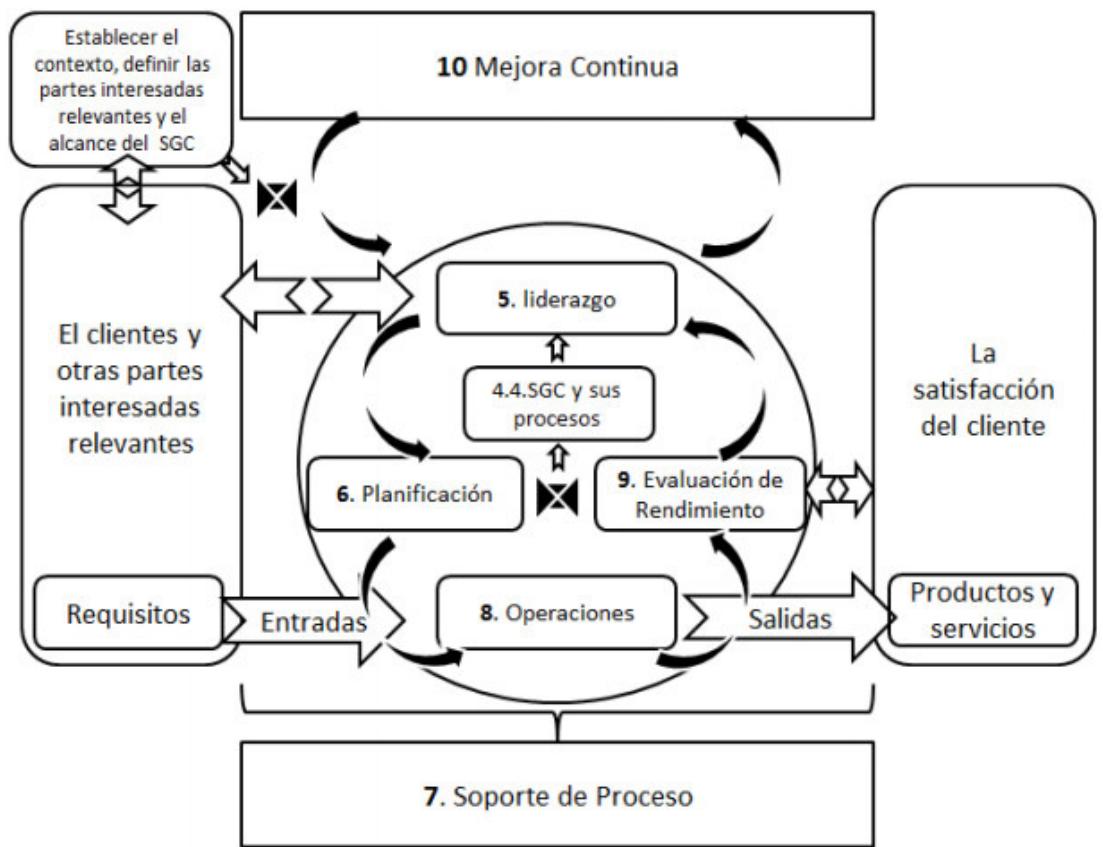


Figura 2.5 Modelo de Sistema de Gestión de la Calidad basado en Procesos ISO 9001:2015

Fuente. Datos tomados de

<http://www.accesoglobal.com.mx/archivosmios/NuevaISO90012015Interpretacion.pdf>

Contexto de la Organización

La organización debe determinar los aspectos externos e internos que son relevantes para su propósito y dirección estratégica y que afectan a su capacidad para lograr el resultado deseado de su sistema de gestión de calidad (sus objetivos, p.ej.). La organización debe controlar y revisar la información sobre estas cuestiones externas e internas.

Debido al impacto o potencial impacto en la capacidad de la organización para suministrar de forma coherente productos y/o

servicios que satisfagan al cliente y los requisitos legales y reglamentarios aplicables, la organización debe determinar:

- Las partes interesadas que son relevantes para el sistema de gestión de calidad.
- Los requisitos de estas partes interesadas que son relevantes para el sistema de gestión de calidad.

La organización debe determinar los límites y la aplicabilidad del sistema de gestión de calidad para establecer su ámbito de acción. En la determinación de este ámbito, la organización debe considerar:

- Los aspectos externos e internos.
- Los requisitos de las partes interesadas pertinentes.
- Los productos y servicios de la organización.

Sistema de gestión de la calidad y sus procesos

La organización debe establecer, implementar, mantener y mejorar continuamente su sistema de gestión de calidad, incluyendo los procesos necesarios y sus interacciones, de conformidad con los requisitos de la Norma Internacional ISO 9001:2015. La organización debe determinar los procesos necesarios para su sistema de gestión de calidad y su aplicación en toda la organización; y debe determinar:

- Las entradas necesarias y los resultados esperados de estos procesos.
- La secuencia e interacción de estos procesos.
- Los criterios, métodos, incluyendo mediciones e indicadores de desempeño relacionados necesarios para garantizar el funcionamiento eficaz y el control de tales procesos.
- Los recursos necesarios y garantizar su disponibilidad.

- La asignación de responsabilidades y autoridades para estos procesos.
- Los riesgos y oportunidades en conformidad con los requisitos; y planificar y ejecutar las acciones apropiadas para hacerles frente.
- Los métodos de seguimiento, medición, según el caso, y la evaluación de los procesos, y si es necesario, los cambios en los procesos para asegurar que se alcanzan los resultados previstos.
- Las oportunidades de mejora de los procesos y el sistema de gestión de calidad.

La organización debe mantener la información documentada en la medida necesaria para apoyar la operación de procesos y retener la información documentada en la medida necesaria para tener la confianza de que los procesos se llevan a cabo según lo planificado.

Liderazgo

La alta dirección debe demostrar su liderazgo y compromiso con respecto al sistema de gestión de calidad mediante:

- Tomar la responsabilidad de la eficacia del sistema de gestión de la calidad.
- Asegurarse de que los objetivos y política de calidad se establecen para el sistema de gestión de la calidad y son compatibles con la dirección estratégica y el contexto de la organización.
- Garantizar que la política de calidad es comunicada, entendida y aplicada dentro de la organización.
- Garantizar la integración de los requisitos del sistema de gestión de calidad en los procesos de negocio de la organización.
- Promover la toma de conciencia del enfoque basado en procesos.

- Asegurar la disponibilidad de recursos necesarios para el sistema de gestión de calidad.
- Comunicar la importancia de la eficacia en la gestión de la calidad y de la conformidad de los requisitos del sistema de gestión de calidad.
- Garantizar que el sistema de gestión de calidad logra sus resultados previstos.
- Participar, dirigir y apoyar a las personas para contribuir a la eficacia del sistema de gestión de la calidad.
- Promover la mejora continua.
- Apoyar a otros roles de gestión relevantes para demostrar su liderazgo, tal como aplica en sus áreas de responsabilidad.

Evaluación de desempeño

La organización debe determinar:

- Qué necesita elementos necesitan seguimiento y medición.
- Los métodos de seguimiento, medición, análisis y evaluación, según corresponda, para asegurar resultados válidos.
- En qué momento deben ser ejecutados el seguimiento y la medición.
- En qué momento deben ser analizados y evaluados los resultados del seguimiento y la medición.

La organización debe asegurar que se implementen las actividades de seguimiento y medición, de acuerdo con los requisitos determinados y debe conservar la información documentada apropiada como evidencia de los resultados.

La organización debe mejorar continuamente la conveniencia, adecuación y eficacia de su sistema de gestión de calidad. (Jesús Araque, 2015).

2.3.1.4. PMBOK

La guía del PMBOK® ofrece un enfoque profesional de la dirección de proyectos, aplicable a la mayoría de ellos, en casi todas las ocasiones. Este método está basado en prácticas contrastadas y de probado valor, gracias a la contribución de profesional de todo el mundo.

La guía del PMBOK® es conocida como un estándar y también como un marco de trabajo que puede definirse como una estructura conceptual básica que permite manejar, de forma homogénea y agrupada, diferentes procesos de negocio, lo que aumenta la disciplina en la gestión. La utilización de un marco de trabajo es una táctica bien definida que permite dominar el entorno complejo de las organizaciones de manera sencilla.

Áreas de Conocimiento

En base a la Guía del PMBOK® existen diez áreas del conocimiento. Estas áreas no son islas independientes entre sí, sino que generalmente están interrelacionadas.

- **Gestión de la Integración:** Identificar, definir, combinar, unificar y coordinar los diversos procesos y actividades de dirección de proyectos.
- **Gestión del Alcance:** Garantizar que el proyecto incluya todo el trabajo requerido (y únicamente el trabajo requerido) para completarlo con éxito.

- **Gestión del Tiempo:** Gestionar que el proyecto termine dentro del plazo previsto.
- **Gestión del Costo:** Planificar, estimar, presupuestar, financiar, obtener financiamiento, gestionar y controlar los costes de modo que se complete el proyecto dentro del presupuesto aprobado.
- **Gestión de la Calidad:** Determinar responsabilidades, objetivos y políticas de calidad a fin de que el proyecto satisfaga las necesidades por las que se lleva acabo.
- **Gestión de los Recursos Humanos:** Organizar, gestionar y liderar al equipo de proyecto.
- **Gestión de las Comunicaciones:** garantizar la oportuna y adecuada recopilación, creación, distribución, almacenamiento, recuperación, gestión, control, monitorización y disposición final de la información final del proyecto.
- **Gestión de los Riesgos:** Identificar, analizar, planificar respuestas y controlar las incertidumbres del proyecto.
- **Gestión de las Adquisiciones:** Comprar o adquirir los productos, servicios o resultados requeridos por terceros ajenos a la organización.
- **Gestión de los Interesados:** Identificar a todas las organizaciones o personas impactadas por el proyecto, analiza sus expectativas e impacto en el proyecto, y desarrollar estrategias de gestión adecuadas para lograr la participación eficaz de los interesados en las decisiones y ejecución del proyecto.

Grupo de Procesos

En la Guía del PMBOK® se mencionan cinco grupos de procesos de la dirección de proyectos:

- **Procesos de inicio:** Se definen los objetivos del proyecto, se identifican a los principales interesados, se nombra al DP y se autoriza formalmente el inicio del proyecto.
- **Procesos de planificación:** se define el alcance del proyecto, se refinan los objetivos y se desarrolla el plan para la dirección del proyecto, que será el curso de acción para un proyecto exitoso.
- **Procesos de ejecución:** se integran todos los recursos a los fines de implementar el plan para la dirección del proyecto.
- **Procesos de monitoreo y control:** se supervisa el avance del proyecto y se aplican acciones correctivas.
- **Procesos de cierre:** se formaliza con el cliente la aceptación de los entregables del proyecto. (Project Management Institute, 2014).

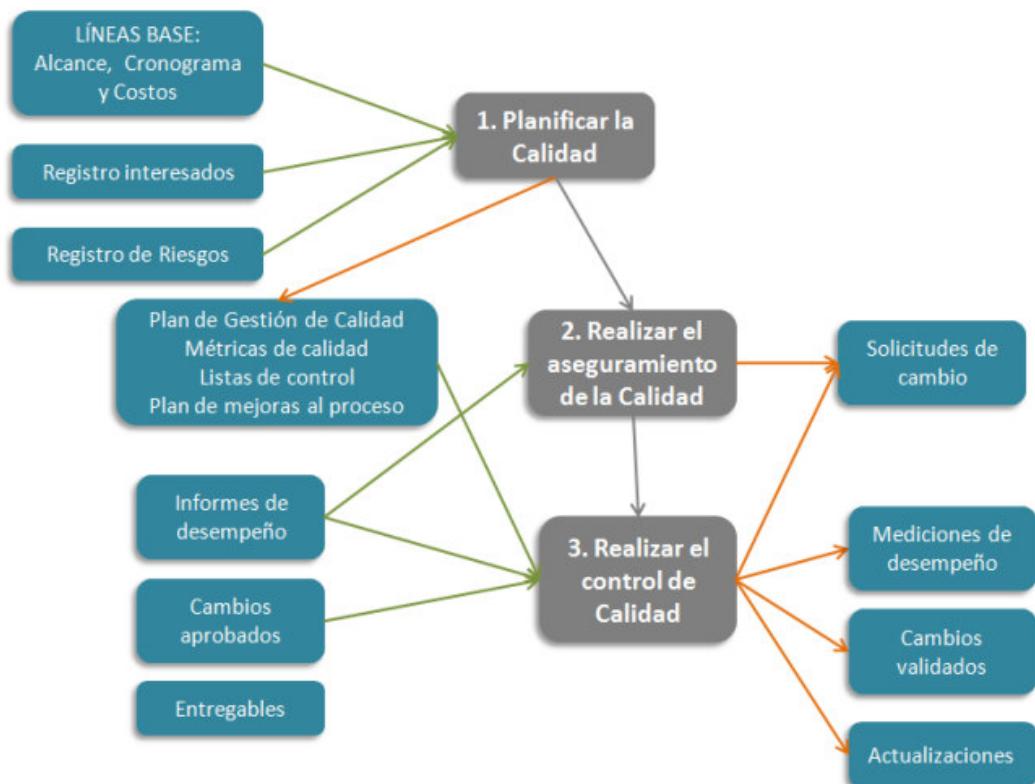


Figura 2.6 Proceso de gestión de la calidad PMBOK

Fuente, Datos tomados de

<https://whatisprojectmanagement.wordpress.com/2013/01/10/gestion-de-la-calidad/>

2.3.1.5. Modelo CMMi

El Modelo de Madurez de Capacidad Integrado (CMMI por sus siglas en inglés) es un modelo de procesos que contiene las mejores prácticas de la industria para el desarrollo, mantenimiento, adquisición y operación de productos y servicios.

CMMi es un enfoque de mejora de procesos que provee a las organizaciones de los elementos esenciales para un proceso efectivo. CMMi es el Modelo de Madurez de Capacidades Integrado. Fue desarrollado por el SEI (Software Engineering Institute).

Mide la madurez del desarrollo del software en una escala del 1 al 5. Integra disciplinas como sistemas y software en un solo marco de trabajo.

Describe formas efectivas y probadas de hacer las cosas, no es un enfoque radical.

Objetivos de CMMi:

- Producir servicios y Productos de alta calidad.
- Crear valor para los accionistas.
- Mejorar la satisfacción del cliente.
- Incrementar la participación en el mercado.
- Ganar reconocimiento en la industria.

Modelos (Constelaciones)

- CMMi for development (CMMi para desarrollo)
- CMMi for Acquisition (CMMi para adquisición)

- CMMI for Services (CMMI para servicios)

Formas de Usar

- Niveles de madurez (mejora organizacional)
- Niveles de capacidad (mejora enfocada en algunos procesos)

Disciplinas

- Ingeniería de Sistema - Cubre la construcción de un sistema con o sin software.
- Ingeniería de Software - Cubre la construcción de soluciones software.
- Integración de productos y procesos de desarrollo - Cubre la relación a largo plazo con el cliente.
- Relación con proveedores - Cubre los procesos relacionados con la subcontratación de partes del sistema.

CMMI es un modelo de calidad del software que clasifica las empresas en niveles de madurez. Estos niveles sirven para conocer la madurez de los procesos que se realizan para producir software. (SEI, 2012)

CMMI propone 5 distintos modelos de madurez de las organizaciones:

1. Inicial - Estado inicial donde el desarrollo se basa en la heroicidad y responsabilidad de los individuos.

- Los procedimientos son inexistentes o localizados a áreas concretas.
- No existen plantillas definidas a nivel corporativo.

2. Gestionado - Se normalizan las buenas prácticas en el desarrollo de proyectos (en base a la experiencia y al método).

- En este nivel consolidado, las buenas prácticas se mantienen en los momentos de estrés.
- Están definidos los productos a realizar.
- Se definen hitos para la revisión de los productos.

3. Definido - La organización entera participa en el proceso eficiente de proyecto software.

- Se conoce de antemano los procesos de construcción de software.
- Existen métodos y plantillas bien definidas y documentados.
- Los procesos no solo afectan a los equipos de desarrollo sino a toda la organización relacionada.
- Los proyectos se pueden definir cualitativamente.

4. Cuantitativamente Gestionado

- Se puede seguir con indicadores numéricos (estadísticos) la evolución de los proyectos.
- Las estadísticas son almacenadas para aprovechar su aportación en siguientes proyectos.
- Los proyectos se pueden pedir cuantitativamente.

5. Optimizado

- En base a criterios cuantitativos se pueden determinar las desviaciones más comunes y optimizar procesos.
- En los siguientes proyectos se produce una reducción de costes gracias a la anticipación de problemas y la continua revisión de procesos conflictivos.



Figura 2.7 Niveles de Madurez del Modelo CMMI

Fuente. Datos tomados de

<http://einsupiicsa.wordpress.com/2011/12/06/cmmi-sus-niveles-y-su-importancia>

2.3.1.6. Proceso Personal de Software PSP

PSP es un proceso de auto-mejoramiento que ayuda a controlar, gestionar y mejorar la forma de trabajar. Se trata de un marco estructurado de formas, pautas y procedimientos para el desarrollo de software. Usar PSP correctamente, proporciona los datos que se necesita para realizar y cumplir con los compromisos, y hacer que los elementos de rutina del trabajo sean más predecibles y eficientes. El único propósito de PSP es ayudar a mejorar las habilidades de ingeniería de software.

El PSP (Personal Software Process) es un modelo que tiene como justificación la premisa de que la calidad de software depende del trabajo de cada uno de los ingenieros de software y de aquí que el proceso diseñado debe ayudar a controlar, manejar y mejorar el trabajo de los ingenieros (Software Engineering Institute, 2010a).

El modelo PSP nos permite llevar un mejor control a nivel personal durante el proceso de desarrollo de software, ya que permite a la persona (Ingeniero) estimar y planear su trabajo, lograr sus compromisos y responder de la mejor manera ante un trabajo bajo presión.

PSP también nos proporciona las bases para desarrollar y practicar una disciplina personal con fuerza industrial, una técnica que le muestra cómo mejorar su proceso personal y los datos necesarios para continuamente mejorar la productividad, calidad de su trabajo. (HUMPHREY, 2007)

PSP se centra en la administración del tiempo y en la administración de la calidad a través de la eliminación temprana de defectos.

PSP busca proporcionar un marco de trabajo para el personal involucrado en el proceso de desarrollo de software.

PSP demuestra cómo manejar la calidad desde el principio del trabajo.

PSP utiliza procesos y herramientas para obtener medios para crear escalabilidad de los participantes, corregir desviaciones de cronograma y subsanar problemas de calidad, generados por los procesos utilizados en el grupo.

Objetivos de PSP:

- Lograr una disciplina de mejora continua en el proceso de desarrollo.
- Medir, estimar, planificar, seguir y controlar el proceso de desarrollo.
- Mejorar la calidad del proceso de desarrollo.
- En general, PSP provee calidad y productividad.

El diseño de PSP se basa en los siguientes principios de planeación y de calidad:

- Cada ingeniero es esencialmente diferente; para ser más precisos, los ingenieros deben planear su trabajo y basar sus planes en sus propios datos personales.
- Para mejorar constantemente su funcionamiento, los ingenieros deben utilizar personalmente procesos bien definidos y medidos.
- Para desarrollar productos de calidad, los ingenieros deben sentirse personalmente comprometidos con la calidad de sus productos.
- Cuesta menos encontrar y arreglar errores en la etapa inicial del proyecto que encontrarlos en las etapas subsecuentes.
- Es más eficiente prevenir defectos que encontrarlos y arreglarlos.
- La manera correcta de hacer las cosas es siempre la manera más rápida y más barata de hacer un trabajo.

Para hacer un trabajo de ingeniería de software de la manera correcta, los ingenieros deben planear de la mejor manera su trabajo antes de comenzarlo y deben utilizar un proceso bien definido para realizar de la mejor manera la planeación del trabajo.

Para que los desarrolladores lleguen a entender su funcionamiento de manera personal, deben medir el tiempo que pasan en cada proceso, los defectos que inyectan y remueven de cada proyecto y finalmente medir los diferentes tamaños de los productos que llegan a producir.

Para producir constantemente productos de calidad, los ingenieros deben planear, medir y rastrear constantemente la calidad del producto y deben centrarse en la calidad desde el principio de un trabajo.

Finalmente, deben analizar los resultados de cada trabajo y utilizar estos resultados para mejorar sus procesos personales.

La estructura del proceso de PSP se muestra conceptualmente en la figura N° 2.8. A partir de la declaración de un requisito, el primer paso en el proceso de PSP es la planificación. Lo cual comprende una secuencia de comandos de planificación que guían el trabajo y presentan un plan para la grabación de los datos de la planificación. Los ingenieros siguen los lineamientos para hacer el trabajo, donde registran el tiempo y los datos de defecto en el tiempo y defecto de los registros. Al final del trabajo, durante la fase de post-mortem (PM), se presenta un resumen de los registros, los cuales permiten medir el tamaño del programa, y un plan de forma resumida.

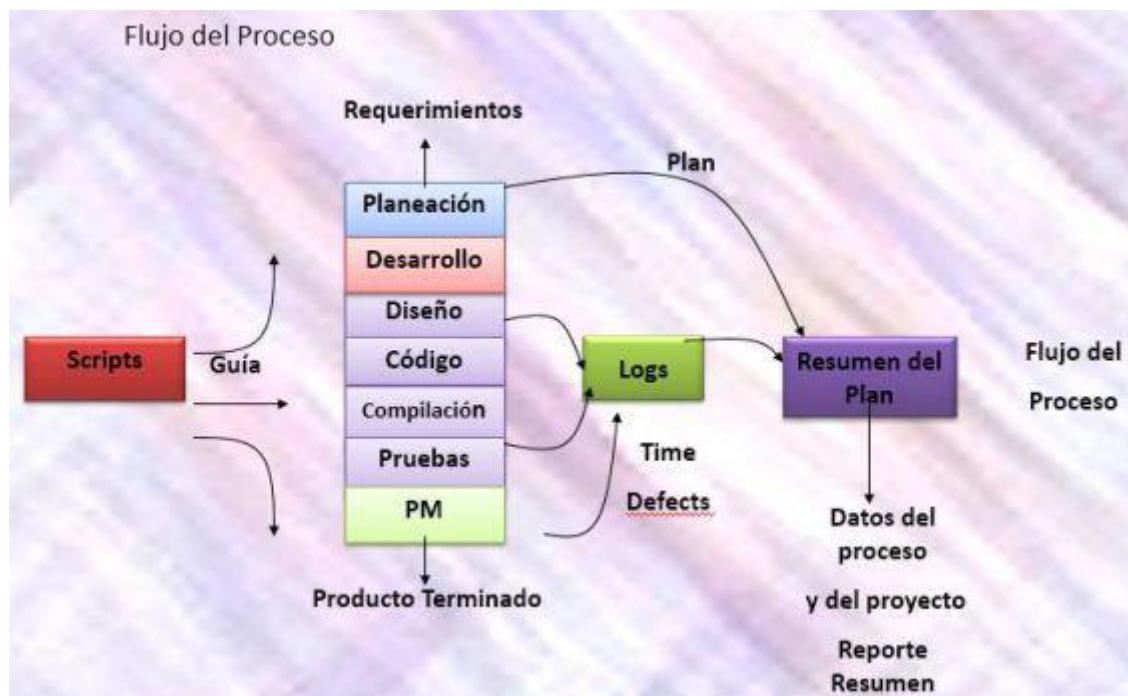


Figura 2.8 Flujo de proceso PSP

Fuente. Datos tomados de <http://karron10.wordpress.com/2013/04/22/el-proceso-personal-del-software-psp/>

Etapas de PSP:

PSP cuenta con 4 etapas base: Medición de Personal, Planificación Personal, Calidad Personal y Proceso personal cíclico.

- **PSP 0:**

- ✓ Define el proceso de trabajo personal identificando y ordenando las principales actividades.
- ✓ Introduce la recolección de datos para medir la productividad y calidad a través del registro de tiempos y defectos.
- ✓ Establece las bases para las mejoras en planificación de trabajo por tiempos y evaluación de resultados.
- ✓ Documentan el proceso usando formas específicas.

- **PSP 0.1:**

- ✓ Establecer estándares de código (Definir “Líneas de código”).
- ✓ Realizar mediciones.
- ✓ Registra el tamaño del producto utilizando puntos funcionales y la estandarización de la codificación.
- ✓ Registra los problemas y propuestas de mejora.

- **PSP 1:**

- ✓ Mejora la planeación introduciendo la estimación del tamaño del producto.
- ✓ Introduce los reportes de pruebas.

- **PSP 1.1**

- ✓ Planeación de tareas (Estimación de recursos).
- ✓ Planeación de tiempos (Calendarización).

- **PSP 2**

- ✓ Introduce las actividades de detección temprana de defectos mediante revisiones de diseño, código y uso de listas de verificación.

- **PSP 2.1**

- ✓ Plantillas de diseño (Marco de trabajo y listas).
- ✓ Verificación de tareas de diseño.
- ✓ Introduce las estimaciones para el diseño detallado, facilitando la revisión del diseño.

- **PSP 3**

- ✓ Introduce el proceso cíclico para crear programas de mayor tamaño
- ✓ Introduce el registro de seguimiento de asuntos.
- ✓ Lleva el resumen de planeación y registro de tiempo, tamaño y defectos por ciclo.



Figura 2.9 Etapas de PSP

Fuente. Datos tomados de <http://karron10.wordpress.com/2013/04/22/el-proceso-personal-del-software-psp/>

Elementos de PSP: Un conjunto de elementos de proceso que incluye lo siguiente:

- **Scripts:** Documentan los criterios de entrada, las fases/pasos y los criterios de salida de cada proceso. Su propósito es guiar en el uso del proceso.
- **Métricas:** Miden el proceso y el producto. Proveen visión sobre cómo funciona el proceso y sobre el estado actual del trabajo.
- **Formatos:** Proveen una forma conveniente y consistente de capturar y retener los datos medidos. Sirven para capturar información específica generada por la ejecución de una o más scripts requeridos por alguna parte del proceso.
- **Especificaciones de Rol** para guiar a los individuos en un proyecto en el desempeño de actividades críticas.
- **Otros activos**, como la estrategia de introducción de TSP, listas de control, lineamientos y especificaciones que no están relacionadas a los roles.
- **Cursos de formación y actividades de autorización** de las tecnologías de TSP y PSP.
- **Estándares:** Proveen definiciones consistentes que guían el trabajo y la recolección de datos.

Métricas de PSP:

- **Esfuerzo:** Total Horas Invertidas.

- **Progreso:** variación entre tiempo estimado vs esfuerzo real.
- **Productividad:** KLOC/hora
- **Calidad:** Defectos / KLOC (miles de líneas de código)
- **Estabilidad:** Cantidad de modificaciones al producto.

Ventajas de usar PSP:

- Los ingenieros de software producen software usando un enfoque estructurado y disciplinado.
- Los ingenieros de software administran la calidad de los productos y aplican una retroalimentación (feedback) cuantitativa para mejorar sus procesos personales de trabajo, obteniendo así:
 - ✓ Mejores Estimaciones.
 - ✓ Mejor Planificación y Seguimiento.
 - ✓ Protección contra compromisos que nunca se cumplen.
 - ✓ Un compromiso personal hacia la calidad.
 - ✓ Involucrarse en un proceso de mejoramiento continuo.

A continuación se presenta un mapa conceptual del modelo de Proceso de Software para Personas PSP donde se esquematiza e integra las diversas fases, niveles, ventajas y desventajas del modelo.

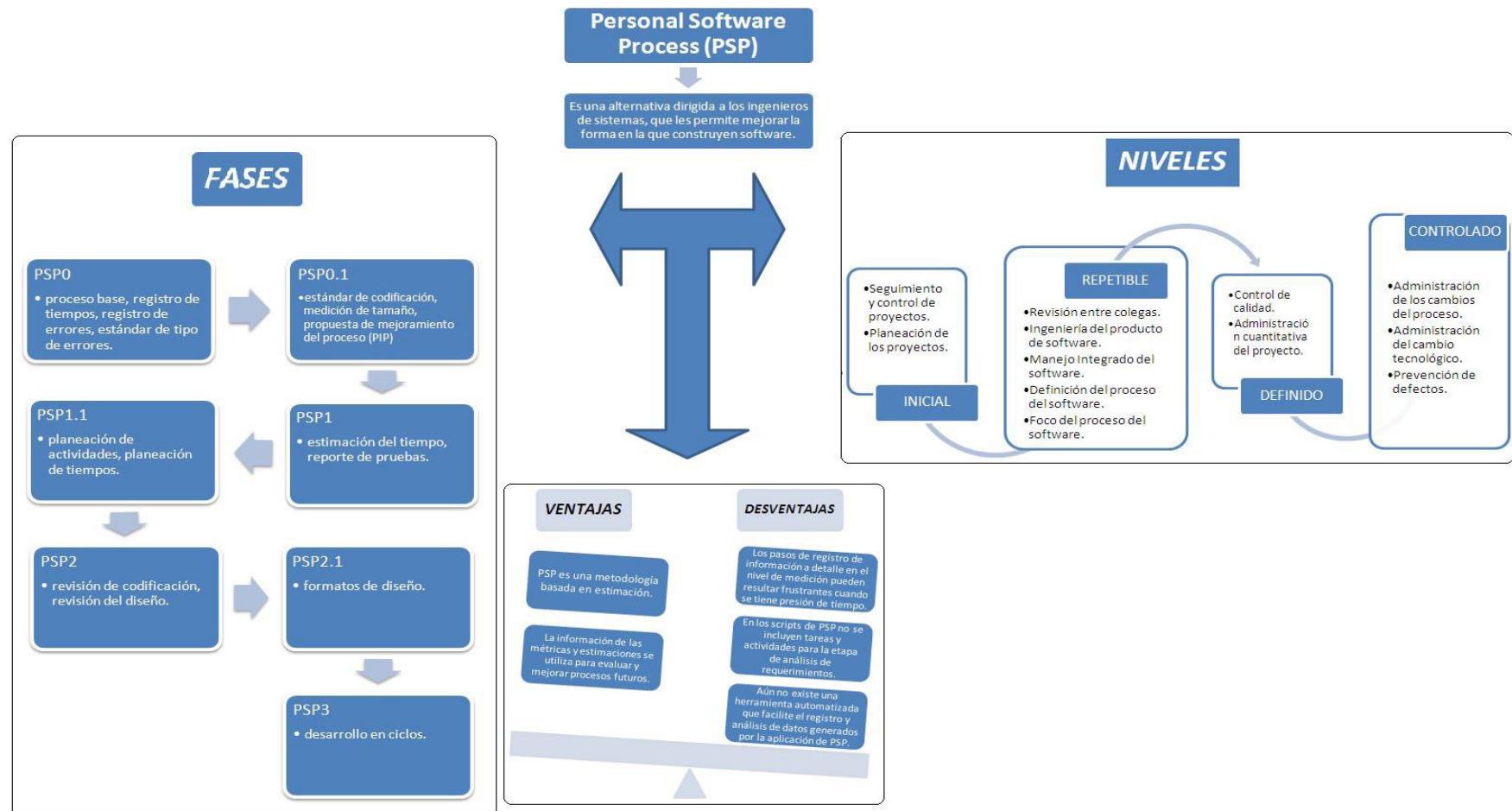


Figura 2.10 Mapa Conceptual PSP

Fuente. Elaboración Propia

2.3.1.7. Proceso de Software de Equipo TSP

El modelo TSP es un conjunto de disciplinas que orientan el desarrollo del sistema de forma organizada dentro de procesos estructurados, con el propósito de obtener productos (sistemas) de la más alta calidad. (Software Engineering Institute, 2010b).

Conjunto de procesos estructurados que indican qué hacer en cada fase del desarrollo del proyecto y muestra cómo conectar cada fase para construir un producto completo. (Bayona, 2007).

TSP extiende y refina los métodos CMM y PSP, para guiar a los miembros de los equipos en el trabajo de mantenimiento y desarrollo.

TSP muestra cómo construir un equipo auto dirigido y cómo ser un efectivo miembro del equipo.

Objetivos de TSP:

- Maximizar calidad Software, Minimizar costos.
- Integrar equipos independientes de alto rendimiento que planeen y registren su trabajo, establezcan metas, y sean dueños de sus procesos y planes.
- Mostrar a los gerentes como monitorear y motivar a sus equipos de trabajo y como ayudarlos a alcanzar su máxima productividad.
- Acelerar la mejora continua de procesos.
- Proveer de una guía para el mejoramiento en organizaciones maduras.

Fases del Ciclo de Vida de TSP

- **Implementación:**

- ✓ Se usa PSP para implementar módulos y unidades.
- ✓ Se crea el diseño detallado de los módulos y unidades.
- ✓ Se revisa el diseño.
- ✓ Se convierte el diseño al código.
- ✓ Se inspecciona el código
- ✓ Se compilan y prueban los módulos y unidades.
- ✓ Se analiza la calidad de los módulos/unidades.

- **Lanzamiento:**

- ✓ Revisión de objetivos a perseguir.
- ✓ Asignación de equipos y roles al personal.
- ✓ Se describen las necesidades del cliente.
- ✓ Se establece las metas individuales y del equipo.

Adicionalmente se detallan los Checklist para la planeación:

- ✓ Establecer productos y objetivos de empresa
- ✓ Establecer roles y objetivos de equipo
- ✓ Definir estrategia de desarrollo
- ✓ Hacer un plan general
- ✓ Hacer un plan de calidad
- ✓ Balancear el plan (cargas de trabajo)
- ✓ Proyecto de riesgos
- ✓ Diseñar reporte para administración

- ✓ Revisión del plan con administración
- ✓ Análisis Postmortem, nuevo equipo revisa proceso.

Productos/Entregables de la fase.

- ✓ Objetivos de equipo por escrito
- ✓ Roles definidos
- ✓ Plan de desarrollo
- ✓ Plan de calidad
- ✓ Plan de soporte al proyecto
- ✓ Desarrollo en conjunto de planes y programas.
- ✓ Plan detallado para cada ingeniero.
- ✓ Plan contra riesgos.
- ✓ Reporte del estado del proyecto.

- **Estrategia:**

- ✓ Crear un diseño conceptual para el producto.
- ✓ Se establece la estrategia de desarrollo: se decide
- ✓ que será producido en cada ciclo.
- ✓ Se hacen estimaciones iniciales de esfuerzos y tamaño.
- ✓ Se establece un plan de administración de la configuración.
- ✓ Se reutiliza el plan anterior.
- ✓ Se establecen riesgos de administración.

- **Planeamiento:**

- ✓ Estima el tamaño de cada artefacto a ser desarrollado.

- ✓ Se identifican las tareas: se estima el tiempo para completar cada tarea; se asignan tareas a los miembros del equipo.
- ✓ Hacer un cronograma semanal para tareas terminadas.
- ✓ Hacer un plan de calidad.

- **Requerimientos:**

- ✓ Se analizan las necesidades del cliente y se entrevistan.
- ✓ Se especifican los requerimientos.
- ✓ Se hace inspección de los requerimientos.
- ✓ Se diseña un plan de pruebas del sistema.

- **Diseño:**

- ✓ Se crea un diseño de alto nivel.
- ✓ Se especifica el diseño.
- ✓ Se inspecciona el diseño.
- ✓ Se desarrolla un plan de pruebas de integración.

- **Pruebas:**

- ✓ Se construye e integra el sistema.
- ✓ Se llevan a cabo las pruebas del sistema.
- ✓ Se produce la documentación de usuario.

- **Postmorten:**

- ✓ Análisis de resultados.
- ✓ Se escribe el reporte del ciclo.
- ✓ Se producen evaluaciones de pares y equipo.

Ventajas de TSP

- Mejora la productividad de las personas.
- Mejora en los hábitos de programación.
- Se puede lograr una detección temprana de defectos y riesgos lo que deriva en una disminución de los defectos.
- Una mejora en la calidad.
- Una reducción en el ciclo de vida.

Desventajas de TSP

- Es necesario que cada uno de los miembros tiene que tener el compromiso y la disciplina de seguir el plan.
- Debe de llenar toda la documentación requerida que incluye sus registros, planificación, las plantillas o formularios.
- Se debe de contar con un buen conjunto de métricas y parámetros de calidad, lo cual, para algunas organizaciones, puede ser difícil de definir.

En la figura 2.11 se muestra un mapa conceptual del modelo de Proceso de Software para Equipos donde se representa todos los componentes del modelo (metodología y ciclo de vida) y su contribución en el aseguramiento de la calidad en el proceso de desarrollo de software.

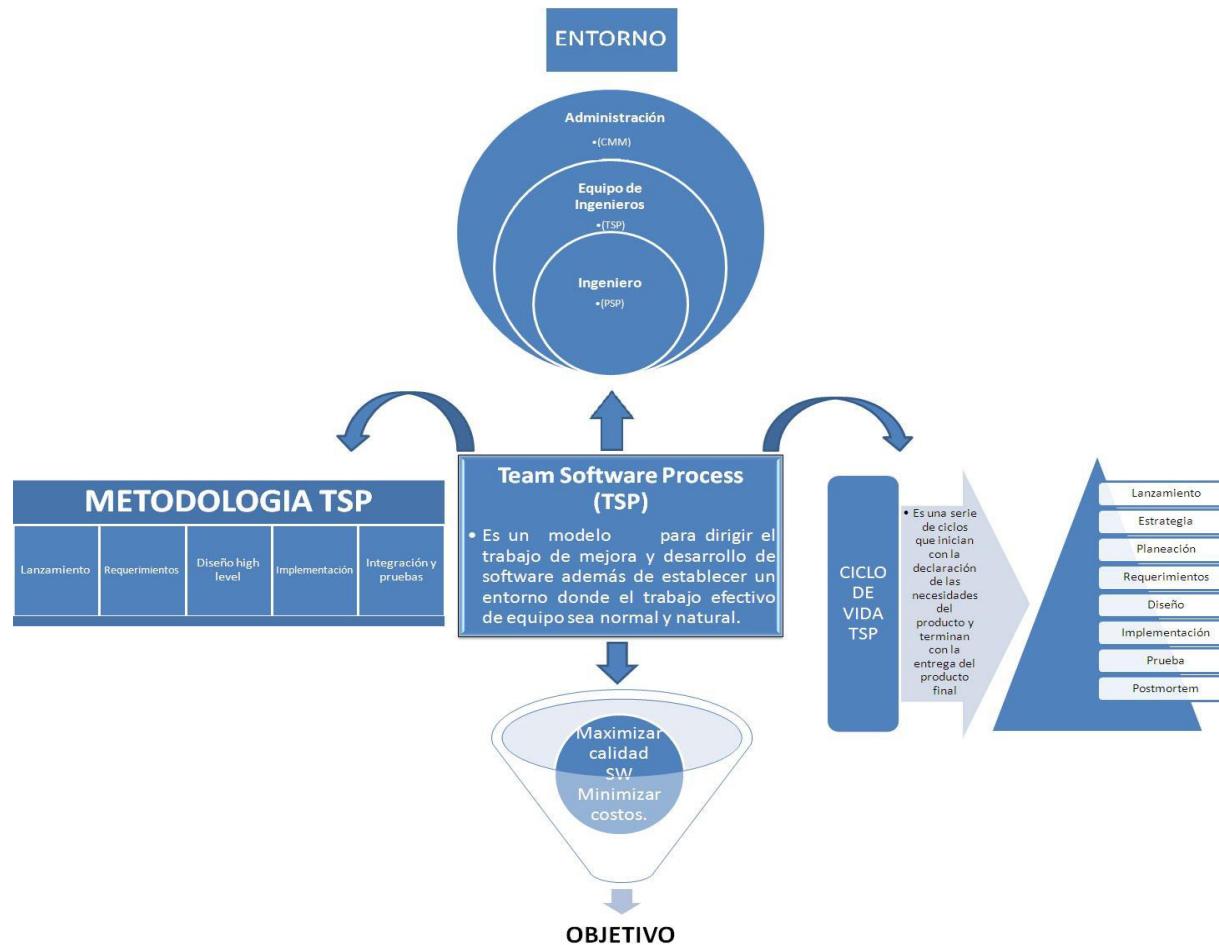


Figura 2.11 Mapa Conceptual TSP

Fuente. Elaboración Propia

2.3.1.8. Integración de los Modelos CMMI, TSP y PSP

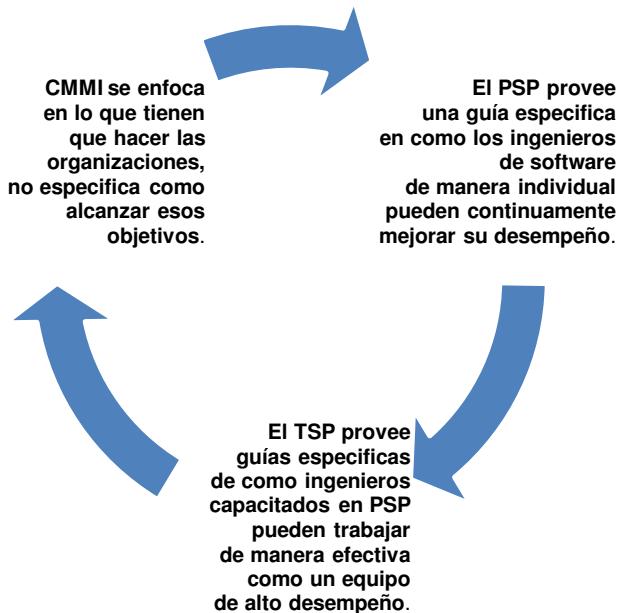


Figura 2.12 Integración Modelos CMMI, TSP y PSP

Fuente. Elaboración Propia

CMMI, PSP y TSP proporcionan un marco tridimensional para la mejora de los procesos.

PSP provee una guía de como los ingenieros de software pueden mejorar su desempeño individual y de forma sostenible.

TSP provee guías de como ingenieros formados en PSP pueden trabajar de forma efectiva en equipos de alto desempeño.

CMMI se enfoca en lo que tienen que hacer las organizaciones, sin embargo no especifica cómo alcanzar esos objetivos. Este vacío es cubierto por PSP y TSP.

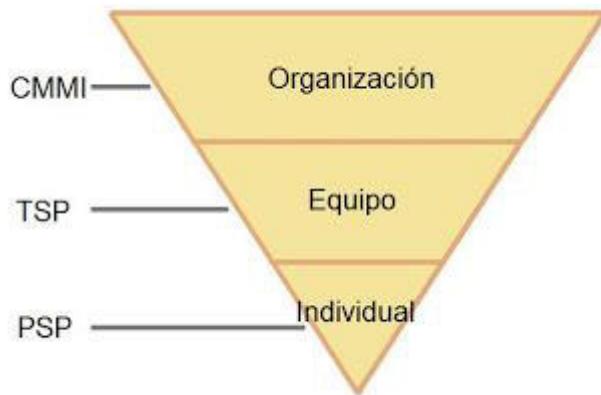


Figura 2.13 Jerarquía Conceptual Modelos CMMi, TSP y PSP

Fuente: <http://cmmiinstitute.com/cmmi-solutions/translations/cmmi-dev-spanish/>

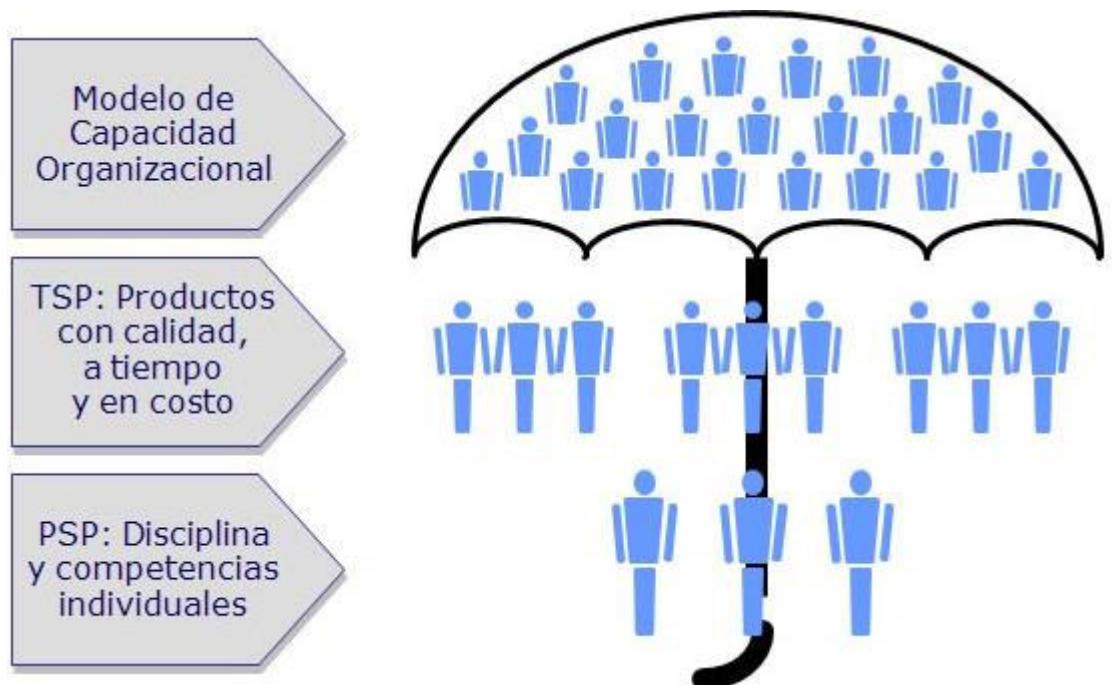


Figura 2.14 Vista Conceptual Integración Modelos CMMi, TSP y PSP

Fuente. Datos tomados de <http://www.sei.cmu.edu/process/index.cfm>

TSP y CMMI son dos modelos del SEI enfocadas a la mejora de procesos de las organizaciones. Ambas comparten las mismas metas: ayudar a las empresas a desarrollar productos de calidad en los tiempos, alcance y presupuestos asignados.

Por un lado, CMMI es un modelo de referencia y por otro lado, TSP es una implementación de procesos de alto rendimiento que es utilizada por equipos auto-dirigidos. Mientras CMMI considera aspectos e infraestructura organizacional, TSP se enfoca en desarrollar disciplina de procesos y una cultura de calidad a nivel personal (a través de PSP) así mismo construye equipos de alto desempeño.

El CMMI es un cómo, mientras el TSP es un producto con elementos de procesos, materiales de entrenamiento y un marco de trabajo para métricas, planeación y calidad. Las debilidades de TSP se pueden resolver con el enfoque organizacional de CMMI. El riesgo de implementar procesos burocráticos que no aportan valor agregado a la empresa se mitiga usando TSP.

Adicionalmente combinar TSP y CMMI ayuda a reducir el tiempo para alcanzar CMMI nivel 3 (de 4 años a 18 meses) obteniendo la institucionalización de procesos definidos que son usados en equipos auto-dirigidos comprometidos con los planes, con un fuerte enfoque personal en la calidad, con ciclos de prueba cortos, teniendo alto desempeño, y haciendo uso de una infraestructura para administrar la mejora continua. (Software Guru, 2011)

2.3.2. Conceptos de Calidad

La calidad es el conjunto de propiedades inherentes a una entidad, que permiten juzgar su valor. Está cuantificada por el valor que se le da al conjunto de propiedades seleccionadas. De esta manera la calidad es subjetiva y circunstancial. Es subjetiva porque depende de los atributos elegidos para medirla y es circunstancial porque el conjunto de atributos elegidos puede variar en situaciones diferentes. (Roger Pressman, 2010)

2.3.3. Calidad en el Proceso de Desarrollo de Software

2.3.3.1. Definición de Proceso de Software

Existen varias definiciones para procesos de software, entre las que tenemos:

- Un proceso de software es la combinación de métodos, información, materiales, maquinas, gente, medio ambiente y mediciones que se utilizan de manera conjunta para obtener un servicio o producto de software con valor agregado para un cliente (Gonzalo Cuevas Agustín, 2002).
- Un proceso de software es un conjunto de actividades para obtener un producto o servicio que satisface a un cliente externo o interno (Coral Calero Muñoz, 2010).
- Un proceso de software se define como un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad (Daniel Edgardo Riesco, 2012).
- El proceso de desarrollo de software define el qué, quién, cuándo y cómo del desarrollo de software. (Roger Pressman, 2010).

De las definiciones presentadas podemos concluir que los procesos de software comprende una serie de actividades tanto técnicas y administrativas que son indispensables para la creación del producto software. Estas actividades comprenden el ciclo de vida del producto, mantenimiento del producto, aseguramiento de la calidad, y demás actividades involucradas en el proceso de desarrollo.

2.3.3.2. Importancia del Proceso de Desarrollo de Software

Para el desarrollo de productos de software es necesario la combinación de 3 componentes:

- **Personal:** Incluye el conocimiento y la experiencia del capital humano que crea y sostiene la evolución del producto. Sin el personal competente y experimentado, es imposible crear productos competitivos que satisfagan las necesidades de los clientes.
- **Tecnología:** Incluye la posesión de la tecnologías que sustentan el producto y las herramientas utilizadas en su desarrollo.
- **Proceso:** Es el saber cómo utilizar el conocimiento personal y la tecnología en forma eficiente para lograr productos de alta calidad que satisfagan las necesidades de los clientes, y son producidos dentro de costos y plazos aceptables (Wilson Pantoja, 2013).

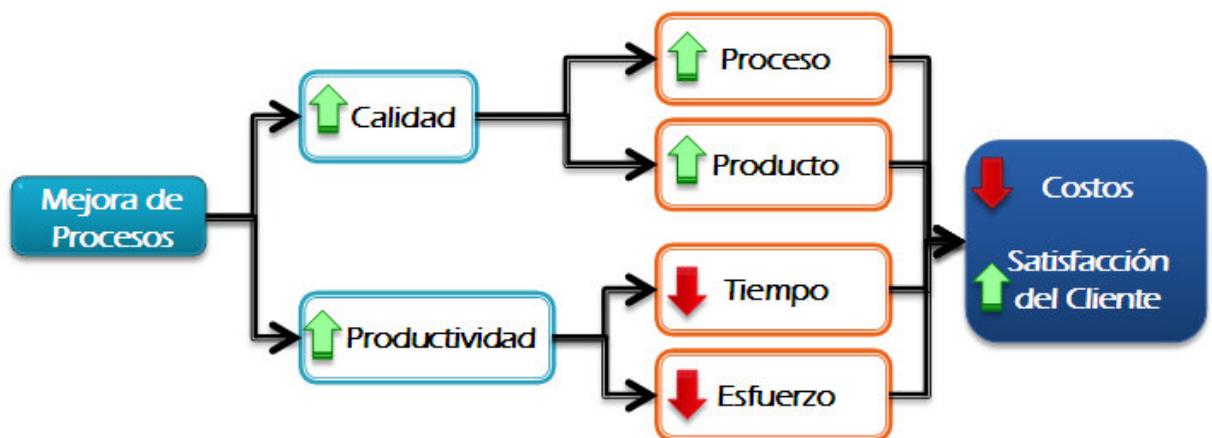


Figura 2.15 Importancia de la mejora de procesos de software

Fuente. Datos tomados de
univirtual.unicauca.edu.co/moodle/mod/resource/view.php?id=30186

2.3.3.3. Calidad del producto

Para definir la calidad de un producto de software, es esencial conocer el concepto de software, el cual es un conjunto de programas, documentos, procedimientos y rutinas en un sistema informático que realizan una función o tarea para obtener un resultado determinado.

Entonces la calidad de un producto de software está relacionada a que éste satisfaga las necesidades y expectativas razonables del cliente, tomando en cuenta los siguientes aspectos: Portabilidad, Interoperabilidad, Funcionalidad, Escalabilidad, Confiabilidad, Seguridad, eficiencia, usabilidad y Mantenibilidad (Luis Fernández, 2012).

Hay diferentes puntos de vista para definir calidad de software. Desde el punto de vista del cumplimiento de los requerimientos Roger Pressman define la calidad de software como:

“El cumplimiento de los requerimientos funcionales y de performance explícitamente definidos, de los estándares de desarrollo explícitamente documentados y de las características implícitas esperadas del desarrollo de software profesional.”

Desde el punto de vista del cliente o usuario Watts Humphrey dice:

“El foco principal de cualquier definición de calidad de software debería ser las necesidades del cliente. Crosby al igual que Pressman define la calidad como conformidad con los requerimientos. Mientras uno puede discutir la diferencia entre requerimientos, necesidades y deseos, la definición de calidad debe considerar la perspectiva de los usuarios. Entonces las preguntas claves son ¿Quiénes son los usuarios?, ¿Qué es importante para ellos? Y ¿Cómo sus prioridades se relacionan con la manera en que se construye, empaqueta y se da soporte al producto?”

Al Davis define calidad del software como:

“La calidad no se trata de tener cero defectos o una mejora medible de la proporción de defectos, no se trata de tener los requerimientos documentados. No es más ni menos que satisfacer las necesidades del

cliente (por más que las necesidades estén o no correctamente documentadas)"

Finalmente, desde estas dos perspectivas el glosario de la IEEE para la ingeniería de software define la calidad del software como:

"El grado con el cual un sistema, componente o proceso cumple con los requerimientos y con las necesidades y expectativas del usuario."

Más allá de cómo definamos la calidad del software, para que la definición tenga sentido esta debe ser medible. Para poder controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición, ya que, como bien plantea Tom De Marco, "no se puede controlar lo que no se puede medir".

Para poder identificar los costos y beneficios del software se definieron los atributos de calidad. La intención es separar el software en atributos que puedan ser medidos o cuantificados (en términos de costo beneficio). Ejemplos de estos atributos son confiabilidad, adaptabilidad, usabilidad y funcionalidad.

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

Los principios básicos del concepto de calidad del software son:

- No alcanza en pensar en calidad a la hora hacer las revisiones y pruebas sino que debe ser una preocupación durante todo el ciclo de vida del software.
- Sólo se alcanza con la contribución de todas las personas involucradas.
- La calidad debe ser planificada y gestionada con eficacia.
- Dirigir esfuerzos a prevención de defectos.
- Reforzar los sistemas de detección y eliminación de defectos durante las primeras fases.
- La calidad es un parámetro importante del proyecto al mismo nivel que los plazos de entrega, costo y productividad.
- Es esencial la participación de la dirección, que ha de propiciar la calidad.

2.3.3.4. Calidad del proceso

Partiendo de la premisa que el producto de calidad surge de lo eficaz y eficiente que sea el proceso de desarrollo, se tiene la necesidad de extender el concepto de calidad tanto a procesos como a productos y servicios, de una organización, o a una combinación de ellos.

Tener un proceso de calidad significa, primero, que está definido y, segundo, que sirve para lo que se especificó, es decir se puede verificar que los objetivos para los que fue definido se satisfacen (M. Teresa Villalba de Benito, 2012).

Definir un proceso de software implicar precisar objetivos, las personas involucradas, las entradas y salidas del proceso, los criterios de entrada y salida, las actividades, los métodos, las herramientas y modelos que se utilizarán, la manera como se medirán elementos dentro del proceso

que permitan verificar resultados, de tal manera que se obtenga un producto de calidad.

2.3.3.5. Aseguramiento de la calidad

Al igual que ocurrió con la definición de calidad hay varios puntos de vista desde donde se puede definir el aseguramiento de la calidad del software.

Desde el punto de vista de la evidencia, la IEEE define el aseguramiento de la calidad como “Una guía planificada y sistemática de todas las acciones necesarias para proveer la evidencia adecuada de que un producto cumple los requerimientos técnicos establecidos. Un conjunto de actividades diseñadas para evaluar el proceso por el cual un producto es desarrollado o construido.”

Desde el punto de vista de la visibilidad, el SEI define SQA como “El aseguramiento de la calidad del software provee claro control del proceso que está siendo usado por el proyecto y del producto que se está construyendo.”

Funciones generales:

Describir los diferentes roles que puede jugar el equipo de SQA en una organización nos dará una visión clara de las funciones que puede llevar a cabo.

“Como policía del proceso”: el trabajo del equipo de SQA es asegurar que el desarrollo sigue el proceso establecido. Entre sus funciones en este rol se encuentran:

- Auditar los productos del trabajo para identificar deficiencias.

- Determinar el cumplimiento del plan de desarrollo del proyecto y del proceso de desarrollo de software.
- Juzgar el proceso y no el producto.

“Como abogado del cliente”: el trabajo del equipo de SQA es representar al cliente. Entre sus funciones en este rol se encuentran:

- Identificar la funcionalidad que al cliente le gustaría encontrar.
- Ayudar a la organización a sensibilizarse con las necesidades del cliente.
- Actuar como un cliente de prueba para obtener una alta satisfacción del cliente.

“Como analista” el trabajo del equipo de SQA es recabar información. Entre sus funciones en este rol se encuentran:

- Juntar muchos datos sobre todos los aspectos del producto y del proceso.
- Con esta información ayudar a mejorar los procesos y los productos.

“Como proveedor de información” el trabajo del equipo de SQA es revisar qué es lo que esté hecho y decir cuáles objetivos técnicos realmente están cumplidos para que la gerencia pueda tomar mejores decisiones de negocios. Entre sus funciones en este rol se encuentran:

- Proveer información técnica objetiva para que la gerencia pueda usarla para tomar mejores decisiones.
- Proveer información apropiada de las clases de productos y de los riesgos asociados con estos.
- Concentrarse más en la reducción de los riesgos que en el cumplimiento del proceso.

“Como responsable de la elaboración del proceso” el trabajo del equipo de SQA es participar en la definición de los planes, procesos, estándares y procedimientos para asegurar que se ajustan a las necesidades del proyecto y que pueden ser usados para realizar las evaluaciones de QA y cumplir los requerimientos del proyecto y las políticas de la organización. Para cumplir este rol el aseguramiento de la calidad debería comenzar en las fases tempranas del proyecto.

2.3.3.6. Proceso disciplinado

El Proceso es Disciplinado si y solo si satisface todas las siguientes condiciones (Wilson Pantoja, 2013):

- El proceso es documentado.
- Se da entrenamiento sobre el proceso y se aplica.
- La gente acepta el proceso como la forma normal de realizar sus actividades.
- El proceso es monitoreado.
- El proceso está sujeto a la mejora continua.

2.3.3.7. Productividad en el Proceso de Desarrollo de Software

Productividad puede definirse como la relación entre la cantidad de bienes y servicios producidos y la cantidad de recursos utilizados.

Productividad en términos de empleados es sinónimo de rendimiento. En un enfoque sistemático decimos que algo o alguien es productivo cuando con una cantidad de recursos (Insumos) en un periodo de tiempo dado obtiene el máximo de productos.

La productividad representa la relación entre el valor del valor generado o agregado y los insumos utilizados en la producción. La medición de esta variable es una valiosa herramienta de gestión para los empresarios, que tienen la función de decidir la combinación de insumos y factores más adecuada para producir los bienes y servicios que los clientes desean (Wilson Pantoja, 2013).

2.3.4. Cuadro Comparativo de Modelos de Aseguramiento de la Calidad

2.3.4.1. Cuadro Comparativo de Modelos a Nivel de Proceso

El cuadro comparativo contribuye en la selección del modelo(s) o estándar(es) de aseguramiento de la calidad apropiado(s) acorde al tipo de empresa y objetivos de la misma. De esta manera se puede lograr una selección e implementación óptima del modelo o marco de referencia que aseguran la calidad en el proceso de desarrollo de software.

El presente trabajo de investigación plantea el siguiente cuadro comparativo donde se hace referencia a los principales modelos / marcos de referencia que tienen relación directa con la presente investigación.

*Cuadro 2.1***Comparativo entre CMMI, PSP, TSP, ISO 9001-2000, ISO/IEC 15504, PMBOK**

ORGANISMO	NORMA /ESTANDAR	APLICABLE	VENTAJAS	DESVENTAJAS
CMMI	<p>CMMI es un conjunto de modelos integrados, más consistentes entre ellos. Proporciona una guía para desarrollar procesos, que además ayuda a evaluar la madurez de la organización o capacidad de un área de procesos. Incluye los procesos de ingeniería de software e ingeniería de sistemas.</p>	<p>Empresas en general (Accenture Perú, GMD SA, Everis Perú, Hildebrando, Synopsis SA, Oracle, IBM Perú, TeamSoft, Telefónica, etc.)</p>	<ul style="list-style-type: none"> ✓ Mejor alineación a objetivos de negocio. ✓ Mejor visibilidad hacia las actividades de ingeniería, con el objetivo de asegurar que el producto o servicio cumple las expectativas del cliente. ✓ Aprovechamiento de mejores prácticas adicionales (medición, riesgo, administración, y manejo de proveedores). ✓ Acoplamiento más estrecho con estándares de ISO. ✓ Integración con los modelos PSP y TSP. ✓ Incremento de la productividad. ✓ Reducción del coste de calidad. 	<ul style="list-style-type: none"> ✓ El proceso de evaluación es costoso en tiempo y esfuerzo ✓ La complejidad de la evaluación continua puede atentar contra la definición de objetivos concretos de madurez. ✓ Puede llegar a ser excesivamente detallado para algunas organizaciones.

PSP	<p>Es un proceso diseñado para ayudar a los ingenieros de software a controlar, manejar y mejorar su trabajo. PSP está basado en una motivación: La calidad de software depende del trabajo de cada uno de los ingenieros de software.</p>	<p>Se implementa a nivel personal, para posteriormente integrarse al modelo TSP.</p>	<ul style="list-style-type: none"> ✓ Los datos y su análisis permitirán determinar las fortalezas y debilidades. ✓ Estos mismos conducirán hacia nuevas ideas para la mejora del proceso. ✓ Se tendrá control sobre el calendario de compromisos. ✓ Tomar control del propio trabajo ✓ Una estructura de trabajo de mejoramiento personal. ✓ La estimulación por nuevas ideas. ✓ Integración con procesos (ej. CMMi). 	<ul style="list-style-type: none"> ✓ El tiempo requerido para conocerlo ✓ El costo emocional por mantener una disciplina ✓ El ego del cambio en las costumbres.
TSP	<p>Se fundamenta en que el software, en su mayoría, es desarrollado por equipos, por lo que los ingenieros de software deben primero saber controlar su trabajo,</p>	<p>Se implementa a nivel personal, para después integrarse las actividades en el equipo de trabajo TSP, que a su vez se puede</p>	<ul style="list-style-type: none"> ✓ Mejora la productividad de las personas. ✓ Mejora en los hábitos de programación. ✓ Detección temprana de defectos y riesgos. 	<ul style="list-style-type: none"> ✓ Los miembros tiene que tener el compromiso y la disciplina de seguir el plan. ✓ Debe de llenar toda la documentación requerida.

	y después saber trabajar en equipo. TSP le enseña a los ingenieros a construir equipos auto-dirigidos y desempeñarse como un miembro efectivo del equipo.	integrar al modelo de calidad de la organización como CMMi.	<ul style="list-style-type: none"> ✓ Mejora en la calidad. ✓ Integra con PSP. 	<ul style="list-style-type: none"> ✓ Se debe de contar con un buen conjunto de métricas y parámetros de calidad. ✓ Cada miembro debe de estar entrenado en el PSP.
ISO 9001-2000	Conjunto de estándares que establecen los requerimientos para la gestión de los sistemas de calidad	Empresas en general (Abengoa, ABB, Adecco Perú, AFP Horizonte, Ajeper, BS Grupo, Cosapi Data, Electro Perú, etc.).	<ul style="list-style-type: none"> ✓ La confianza reforzada entre los actuales y potenciales clientes en la capacidad de la empresa para suministrar en forma consistente los productos y/o servicios acordados. ✓ La verificación de esta capacidad es efectuada por un organismo independiente. ✓ Existe una mejor posición competitiva. ✓ La auditoría externa que implica la certificación según ISO 9000, permite identificar oportunidades 	<ul style="list-style-type: none"> ✓ Requiere gran esfuerzo y tiempo para lograr el objetivo. ✓ El sistema origina cierta burocracia.

			<p>de mejoramiento del sistema de calidad.</p> <ul style="list-style-type: none"> ✓ Es posible sustituir las auditorías de calidad de cada uno de los clientes por la efectuada por un solo organismo idóneo, imparcial. ✓ Ayuda a su vez en los procesos de mejoramiento de la calidad iniciados por los clientes. ✓ Produce un mejoramiento en la motivación y en el trabajo en equipo ya que la certificación es la resultante del esfuerzo colectivo de la empresa. 	
ISO/IEC 15504	<p>Estándar internacional que ofrece un marco para la evaluación de procesos.</p> <p>Marco de trabajo para evaluación de múltiples modelos (de software o no). Su dirección actual es</p>		<ul style="list-style-type: none"> ✓ Primer modelo de procesos de dos dimensiones, es decir, dimensiones independientes para los procesos y la capacidad. ✓ El resultado de una evaluación de proceso puede ser representado por un perfil de proceso. 	<ul style="list-style-type: none"> ✓ ISO/IEC TR-15504 permite que el dominio de procesos sea tan amplio para abarcar todos los posibles ciclos de vida, de tal manera que hace difícil que todos los atributos de proceso sean universales, provocando dificultad y confusión durante la evaluación.

	<p>poder aplicarse a múltiples disciplinas y permitir a las diferentes comunidades definir sus propios procesos específicos, modelos de referencia o buenas prácticas</p>		<ul style="list-style-type: none"> ✓ Es el modelo más consensuado y probado. ✓ Mayor reconocimiento en el mercado Europeo. ✓ Coherencia con otros modelos de calidad ya implementados en la organización ISO9001- ISO 20000- ISO 27000. ✓ Menores costos de la certificación. 	<ul style="list-style-type: none"> ✓ Poco reconocimiento en el mercado Norteamericano.
PMBOK	<p>Es una guía de fundamentos para la Dirección de Proyectos, sus principales objetivos son describir el conocimiento y las prácticas aplicables a la mayoría de los proyectos de la época.</p> <p>Provee una base formal para fundar proyectos.</p>	<p>IBM, SAP, Software Enterprise Services, Everis Perú, GMD SA, CosapiSoft, TeamSoft, Accenture Perú, Telefónica, etc.</p>	<ul style="list-style-type: none"> ✓ Es precisa en la definición de conceptos. ✓ Su enfoque es sistemático y presenta la información con esa misma visión. ✓ Es simple para la aplicación. ✓ Es más operativo aunque parte de una visión estratégica. ✓ Define claramente los procesos de gestión de calidad. ✓ Brinda herramientas que generan utilidad. 	<ul style="list-style-type: none"> ✓ Deja por fuera temas importantes o los aborda de manera simplista. ✓ Aborda sola la gestión del proyecto no así la del producto. ✓ Puede confundir con algunas acciones o productos que son entradas en un proceso luego no son salidas en otros procesos, de acuerdo a la lógica de sistemas.

	Guía y Orienta a gestores de proyectos sobre la forma de conducir la construcción de resultados.		✓ Presenta una lógica en la utilización de las acciones de cada proceso, si se utilizan como salidas luego son entradas de otros procesos.	
--	--	--	--	--

Fuente. Elaboración Propia

2.3.4.2. Criterios de Selección del Modelo

Los criterios que fueron evaluados en la selección de los modelos son los siguientes:

- Alcance de la evaluación.
- Tamaño de la organización.
- Ámbito de aplicación.
- Experiencia en el mejoramiento.
- Factores clave para asegurar la calidad.
- Plazos de los resultados.
- Recursos de la organización para la implementación.

*Cuadro 2.2***Formulario de criterios de selección del modelo**

CRITERIO	DETALLE	Modelo					
		ISO/IEC 15504	ISO 9001: 2015	PMBOK	CMMi	TSP	PSP
Alcance de la Evaluación	¿Alcance de la evaluación, abarca todos los procesos de negocio?	X	X	X			
Tamaño de la Organización	¿Es aplicable a cualquier organización?	X	X	X			
Ámbito de Aplicación	¿Es específica para procesos de desarrollo de software?				X	X	X
Experiencia en el mejoramiento	¿Tiene como principio el mejoramiento continuo?	X	X	X	X	X	X
Factores claves para asegurar la calidad	¿Factor Humano es clave es el proceso de aseguramiento?				X	X	X
Plazos de los resultados	¿Los resultados son palpables a corto y mediano plazo?			X	X	X	X
Recursos de la organización para la implementación	¿Se incurre en gastos adicionales?	X	X				

Fuente. Elaboración Propia

Luego de revisar los principales marcos de referencia de aseguramiento de calidad en general, se puede indicar que los modelos que comparten o tienen similitudes con el presente trabajo de investigación son los modelos CMMi, TSP, PSP; los cuales consideran el **Factor Humano** (de manera individual o grupal) como pilar fundamental en el proceso de aseguramiento de la calidad.

Generar un modelo de aseguramiento a partir de CMMi, TSP y PSP es importante y viable porque sus procesos se complementan e integran de manera sencilla sin que existe duplicidad o algún proceso se solape, adicionalmente son modelos que fueron creados para brindar soporte a la industria del software.

2.4. Hipótesis y Variables

2.4.1. Hipótesis general

La integración del modelo de aseguramiento de la calidad basado en los modelos de madurez de capacidades CMMi, TSP y PSP permite asegurar la calidad en el proceso de desarrollo de software.

2.4.2. Hipótesis específicas

Las hipótesis específicas que se plantea probar son las siguientes:

- H.1.: La integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software reduce el costo de calidad en los proyectos.
- H.2.: La integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software reduce los defectos en los proyectos.
- H.3.: La integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software incrementa la rentabilidad de los proyectos.
- H.4.: La integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software aumenta el grado de satisfacción de los clientes.

2.4.3. Identificación de variables

Las variables que se han determinado son las siguientes:

Variable Independiente (VI):

VI: Modelo de aseguramiento de la calidad.

Variable Dependiente (VD):

VD: Calidad en el proceso de desarrollo de software.

2.4.4. Operacionalización de variables

Cuadro 2.3

Operacionalización de Variables

Variable	Tipo de Variable	Definición Conceptual	Dimensión	Indicador	Descripción del Indicador
Modelo de aseguramiento de la calidad en el proceso de desarrollo de software.	Independiente	Una guía planificada y sistemática de todas las acciones necesarias para proveer la evidencia adecuada de que un producto cumple los requerimientos técnicos establecidos (IEEE, 2012)	-	-	-
Calidad en el proceso de desarrollo de software.	Dependiente	Proceso de calidad significa, primero, que está definido y, segundo, que sirve para lo que se especificó, es decir se puede verificar que los objetivos para los que fue definido se satisfacen (M. Teresa Villalba de Benito, 2012).	Costo de Calidad	% de Costo de Calidad (Costo de Conformidad + Costo de No Conformidad)	Porcentaje de Costo de calidad respecto al costo real del proyecto.
			Defectos en el Proyecto	Cantidad de Defectos en el Proceso de Desarrollo de Software.	Cantidad de Defectos en los proyectos incurridos en el proceso de desarrollo de software.
			Rentabilidad del Proyecto	% de desviación de costo respecto a la línea base.	Porcentaje de desviación de costo del proyecto respecto al presupuesto planificado (línea base).
			Satisfacción del Cliente	% de clientes satisfechos.	Grado de satisfacción del cliente respecto a las restricciones del proyecto (Costo, Tiempo, Alcance, Calidad y Satisfacción del Cliente).

Fuente. Elaboración Propia

Cuadro 2.4

Indicadores de la variable dependiente

Indicador	Descripción	Unidad de Medida	Instrumento de Medición	Formula	Observación
Porcentaje de Costo de Calidad	Porcentaje de Costo de calidad respecto al presupuesto del proyecto.	Porcentaje	Ficha de Registro de Datos	$PCC = CC/CR * 100$ <p>Donde: PCC = Porcentaje de Costo de Calidad CC = Costo de Calidad CR = Costo Real del Proyecto</p>	-
Cantidad de Defectos en el Proceso de Desarrollo de Software.	Cantidad de Defectos en los proyectos incurridos en el proceso de desarrollo de software.	Cantidad	Ficha de Registro de Datos	$CDP = \sum(DE)$ <p>Donde: CDP: Cantidad defectos en el proyecto DE: Defecto</p>	-
Porcentaje de Desviación de Costo respecto a la Línea Base.	Porcentaje de desviación de costo del proyecto respecto al presupuesto planificado (línea base).	Porcentaje	Ficha de Registro de Datos	$PVC = (1 - (CR/CP)) * 100$ <p>Donde: PVC = Porcentaje de Variación de Costo CR = Costo Real CP = Costo Planificado</p>	-
Porcentaje de Clientes Satisfechos.	Grado de satisfacción del cliente respecto a las restricciones del proyecto (Costo, Tiempo, Alcance, Calidad y Satisfacción del Cliente).	Porcentaje	Cuestionario	$PCS = (TC-Cl) / TC * 100$ <p>Donde: PCS = Porcentaje de Clientes Satisfechos TC = Total Clientes Proyecto Cl = Clientes que presentaron inconformidad</p>	-

Fuente. Elaboración Propia

CAPITULO III: METODOLOGIA

3.1. Tipo y Diseño de Investigación

El tipo de investigación utilizado en el presente trabajo es cuantitativo **“Experimental”**, para su selección se tuvo en cuenta los criterios de análisis y alcance de sus resultados, lo cual nos permite:

- Establecer con precisión relaciones de causa-efecto
- Conocer cómo se comporta una variable, conociendo el comportamiento de otra.

El objetivo del diseño Experimental, es aplicar la relación causa-efecto entre dos o más variables. En este tipo de diseño el investigador modifica intencionalmente el estado de algunos sujetos de estudio y manipula un tratamiento que desea estudiar o evaluar. (Lerma, 2009).

Debido a que en esta investigación se observará la relación causa-efecto entre la variable independiente (Modelo de aseguramiento de la calidad) y la variable dependiente (Proceso de desarrollo de software), por tanto nuestro diseño de investigación será Experimental.

Diseño Pre-Experimental de la prueba previa y la posterior a un solo grupo: En este diseño, un grupo de unidades de prueba se mide dos veces. No existe un grupo de control. Primero, se toma una medición previa al tratamiento (O_1), luego el grupo se expone al tratamiento (X) y finalmente se toma una medición posterior (O_2). El efecto del tratamiento se calcula como $O_2 - O_1$. (Lerma, 2009).

El diseño de la presente investigación es un diseño pre-experimental, el cuál fue nuestro plan y estrategia que nos permitió responder a nuestra pregunta de investigación, así como alcanzar los objetivos de estudio y llegar a contrastar nuestra hipótesis.

En la presente investigación debido a que se tendrá a un mismo grupo de control, se realizó una prueba al grupo pre-prueba en un intervalo de tiempo específico; posteriormente se aplicará un estímulo (Modelo de Aseguramiento de la Calidad) al grupo post-prueba durante un intervalo de tiempo, y por último, ambos resultados serán comparados. En la siguiente figura se observa la simbología del diseño Pre Experimental.

GRUPO	ASIGNACIÓN	PRE-PRUEBA	EXPERIMENTO	POST-PRUEBA	DIFERENCIA
GE		O1	X	O2	O2 – O1 = d1

Figura 3.1. Simbología del diseño pre-experimental

Fuente. Datos tomados de Lerma (2009)

En la siguiente tabla se muestra una breve descripción de la simbología del diseño Pre Experimental.

Cuadro 3.1

Descripción de la simbología del diseño Pre Experimental

Símbolo	Descripción
GE	Grupo Experimental
	Asignación (No aleatoria).
O1	Medición 1 de los sujetos de un grupo, que consiste en una pre-prueba (previa al tratamiento).
X	Tratamiento, estímulo o condición experimental (presencia de algún nivel o modalidad de la variable independiente).
O2	Medición 2 de los sujetos de un grupo, que consiste en una post-prueba (posterior al tratamiento).

Fuente. Elaboración Propia

Los objetivos del diseño pre experimental son: (Lerma, 2009).

- Comparar resultados encontrados en un mismo grupo de estudio.
- Ilustrar la forma en que las variables extrañas pueden influir en la validez interna de un diseño.

Las ventajas del diseño pre experimental son: (Lerma, 2009).

- Permite saber los beneficios o cambios que se lograron con el tratamiento.
- En ocasiones es el único recurso para llegar a inferencias sobre variables que es imposible manipular.
- Pocos requisitos de aplicación.

Los datos se obtuvieron a través de procedimientos válidos y aceptados tales como reportes, encuestas, entrevistas e informes estructurados de las oficinas de gestión de proyectos. Para analizar la información se hizo uso de procedimientos estadísticos de diversa complejidad.

3.2. Unidad de análisis

La unidad de análisis de la presente investigación son los proyectos de desarrollo de software de la consultora donde se aplicó el modelo de aseguramiento de la calidad en el proceso de desarrollo de software.

Las unidades de análisis indicadas nos permitieron obtener los datos de las variables e indicadores de la investigación y sobre el cual se realizaron las diversas mediciones obteniendo información esencial que nos permitió dar respuesta a nuestra hipótesis planteada.

3.3. Población de estudio

Proyectos de desarrollo de software que ejecutan las Consultoras y Empresas (in house) en el departamento de Lima – Perú, cuyo proceso de desarrollo de software se encuentra certificado por CMMI Institute en el Nivel de Madurez 2 o 3.

La información de las consultoras y empresas se obtuvo desde el Portal Oficial de CMMI Institute, donde se encuentran publicadas las empresas certificadas. Ver anexo C.

3.4. Tamaño de muestra

El muestreo es el proceso de seleccionar un número suficiente de elementos a partir de una población. En el caso de la presente tesis, las consultoras y empresas que desarrollan software para uso interno tienen características similares porque su proceso de desarrollo se basa en el modelo de madurez de capacidades CMMi. Por tanto se puede generalizar los resultados a toda la población. Para ello se utilizó un muestreo no probabilístico (dirigido) sobre nuestra población establecida.

El muestreo permite superar las dificultades de recopilar datos de toda la población que a menudo es imposible o prohibitivo en términos de tiempo, costo y recursos humanos.

La presente investigación tiene una muestra de 22 proyectos de software correspondientes a la consultora “**Accenture**”, 11 proyectos formaran parte del grupo de pre-test y los 11 restantes del grupo post-test. (Anexo G).

Dado que un proyecto es único y comprende actividades de carácter grupal temporal para producir un producto, se van a comparar dos grupos de proyectos, con características comunes las cuales se especificación en el punto 3.5 correspondiente a la selección de la muestra.

3.5. Selección de muestra

La selección de la muestra se realizó en función a la factibilidad en el acceso a la información, respaldo de la fábrica de software en la integración del modelo y criterios de negocio y técnicos que se detallan en el cuadro 3.2.

*Cuadro 3.2***Criterios de selección de la muestra**

Nº	Criterio	Proyectos Muestra Pre-Test	Proyectos Muestra Post-Test
1	Consultora	Accenture	Accenture
2	Cliente	Pacifico Seguros (PPS)	Pacifico Seguros (PPS)
3	Unidad de Negocio	Seguros Generales (SOAT, Autos, EPS, Siniestros)	Seguros Generales (SOAT, Autos, EPS)
4	Lenguaje de Programación	Visual Studio.Net C# (2008, 2010), Oracle Reports	Visual Studio.Net C# (2008, 2010)
5	Base de Datos	Oracle 11g R2	Oracle 11g R2
6	Duración del Proyecto	3,4 meses aprox.	3 meses aprox.
7	Ciclo de Vida (Metodología)	REF, RET, Arquitectura & Construcción, Certificación e Implantación & Seguimiento	REF, RET, Arquitectura & Construcción, Certificación e Implantación & Seguimiento
8	Entidad Certificadora del Producto	Itera Process	Itera Process
9	Constitución de Equipos	Mixtos (Senior, Junior, Asistente)	Mixtos (Senior, Junior, Asistente)
10	Número de Proyectos	11	11

Fuente. Elaboración Propia

3.6. Técnicas de recolección de Datos

Para la recolección de datos se usaron las siguientes técnicas e instrumentos:

Se recopiló datos a través de tesis, direcciones de internet, encuestas diseñados para este propósito, documentación de proyectos (Líneas base, Trabajo real, etc.), resultados de evaluaciones a sus procesos y reportes de la oficina de gestión de proyectos (PMO) de las empresas/consultoras seleccionados en la muestra, ésta última se realizó de forma presencial.

Las técnicas de recolección de datos mencionadas anteriormente nos permitieron obtener información detallada que luego fueron analizadas e interpretadas en el capítulo siguiente.

3.6.1. Técnicas

- **Análisis Documental**

Es una técnica que permite acumular datos, recoger ideas y organizarlo todo en un fichero, siendo una constante fuente de información, creciente y flexible. (Lerma, 2009).

En la presente investigación esta técnica nos permite recopilar, organizar y registrar la información que se observa en el proceso de desarrollo de software de los proyectos.

- **Encuesta**

Técnica que permite al investigador observar la problemática que existe en un campo específico, sin que se haga ninguna intervención para que los organismos se comporten de un modo diferente a que usualmente lo hacen. (Lerma, 2009).

En la presente investigación esta técnica nos permite observar la problemática en el proceso de desarrollo de software en la consultora Accenture.

3.6.2. Instrumentos

- **Ficha de Registro de Datos**

Es un documento que se ha elaborado con el fin de registrar datos que se generan como el resultado del contacto directo entre el observador y la realidad que se observa. (Lerma, 2009).

En esta investigación las fichas de observación permitieron registrar los datos en porcentaje de: “**costo de calidad, número de defectos y rentabilidad del Proyecto**”.

Estas fichas fueron colocadas en el **Anexo G: Ficha de registro de datos**.

- **Cuestionario**

El presente instrumento en la presente investigación nos permitió obtener datos en porcentaje de: “**Satisfacción del Cliente**”.

Estos cuestionarios fueron colocadas en el **Anexo H: Cuestionario**.

En el cuadro N° 3.3 se especifica que instrumento de medición será usado para medir los indicadores utilizados.

Cuadro 3.3

Instrumentos de medición asociados al indicador

Variable	Dimensión	Indicador	Unidad de Medida	Técnica	Instrumento
Calidad en el proceso de desarrollo de software.	Costo de Calidad por Proyecto	% de Costo de Calidad (Costo de Conformidad + Costo de No Conformidad)	Porcentaje	Análisis Documental	Ficha de Registro de Datos
	Defectos por Proyecto	Cantidad de Defectos en el proceso de desarrollo de software	Cantidad	Análisis Documental	Ficha de Registro de Datos
	Rentabilidad del Proyecto	% de variación respecto a la línea base de costo.	Porcentaje	Análisis Documental	Ficha de Registro de Datos
	Satisfacción del Cliente	% de clientes satisfechos.	Porcentaje	Encuesta	Cuestionario

Fuente. Elaboración Propia

3.7. Métodos y técnicas de procesamiento y análisis de datos

Dado que las variables se formularon en valores numéricos, por consiguiente se realizó un análisis cuantitativo.

Por ello se deben ordenar los valores y darles forma de una matriz de tabulación, para luego procesarlos mediante el programa estadístico SPSS v.22.0, y realizar el respectivo análisis de datos y obtener los resultados correspondientes.

En los datos ingresados, cada columna representa el indicador antes y después de adoptar el modelo de aseguramiento de la calidad en el proceso de desarrollo de software, y cada fila representa un sujeto específico de la muestra.

Para analizar los datos colocados en la matriz de tabulación se utilizará el análisis cuantitativo, que consiste en comparar diferencias entre grupos de la muestra.

Dado que la muestra es menor a 30 elementos, entonces se utilizará la distribución t-student con muestras independientes.

La prueba t se utiliza para comparar los resultados de una pre-prueba y de una post-prueba en un contexto experimental, donde se compararan las medias y varianzas del grupo en dos momentos diferentes.

La media se calcula aplicando una fórmula estadística, más conocida como la fórmula del promedio, dado que la pre-prueba y post-prueba fueron en momentos diferentes, ambos promedios pueden ser comparados para determinar las conclusiones de la investigación.

CAPITULO IV: RESULTADOS Y DISCUSIÓN

4.1. Diagnóstico inicial

En esta sección se presenta datos obtenidos de un diagnóstico inicial realizado a consultoras certificadas en CMMi. Dicha información fue recopilada a través de un cuestionario dirigido a expertos de las diversas entidades.

4.1.1. Tasa de respuesta

En la presente investigación, se enviaron 15 cuestionarios comprendidos entre consultoras y empresas que desarrollan software para uso interno, de los cuales 12 respondieron a la petición de manera eficaz, lo que representa una tasa de respuesta del 80.00%, la cual se considerada apropiada para el presente estudio.

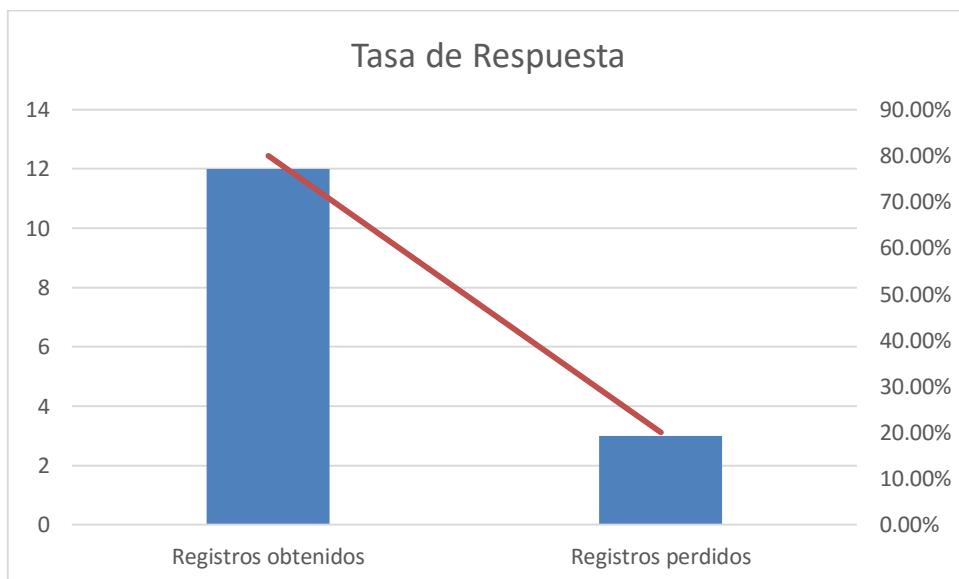


Gráfico 4.1. Tasa de respuesta

Fuente. Elaboración propia

4.1.2. Información de la encuesta

En esta sección se muestran los detalles estadísticas de los datos recogidos en el diagnóstico inicial.

A continuación se presenta el análisis descriptivo de la información obtenida por cada pregunta contenida en la encuesta.

En el gráfico N° 4.2., se muestra las metodologías más utilizadas en el proceso de desarrollo de software en consultoras certificadas en CMMI nivel 2 y 3. Como se muestra en el gráfico 8 consultoras utilizan metodologías ágiles, 3 utilizan RUP y 1 utiliza una metodología diseñado a medida.

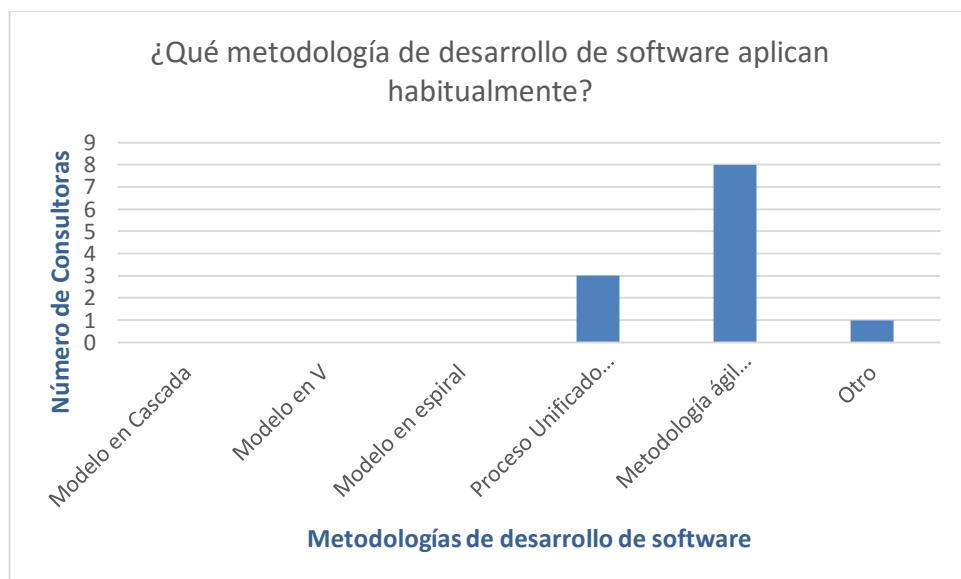


Gráfico 4.2. Qué metodología de desarrollo de software aplican habitualmente

Fuente: Elaboración propia

En el gráfico N°4.3., se presenta la duración promedio dedicado a cada fase del proyecto, donde la fase que mayor duración es “Pruebas e Integración” con un 35% de la duración total del proyecto.

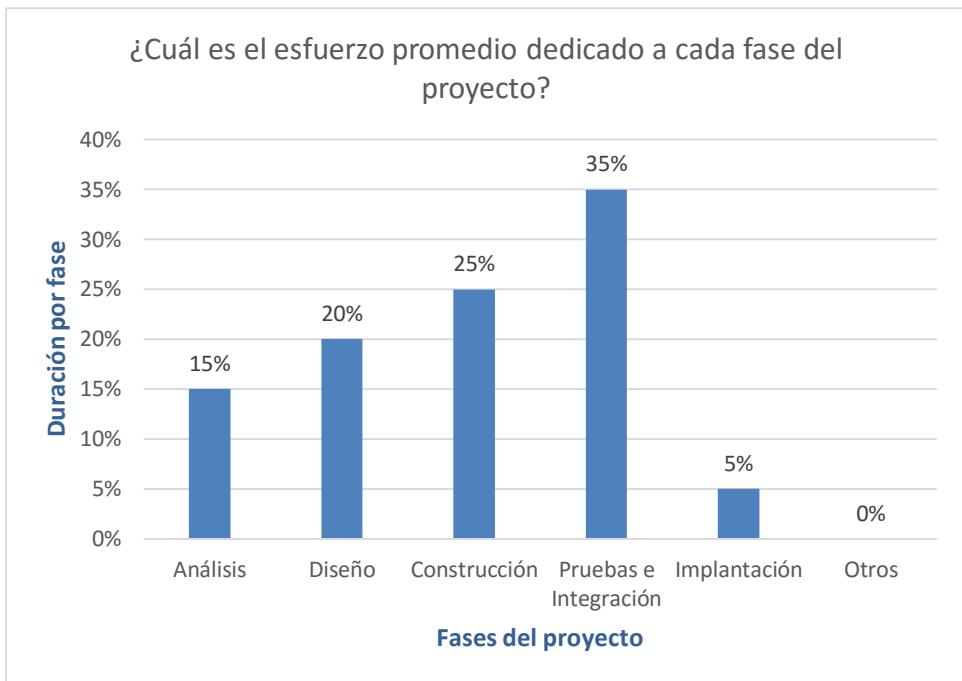


Gráfico 4.3. Cuál es el esfuerzo promedio dedicado a cada fase del proyecto

Fuente. Elaboración propia

En el gráfico N° 4.4., se presenta el esfuerzo adicional que se realiza al finalizar la fase de construcción, donde las actividades que generan mayor esfuerzo adicional son Corrección de errores, Mejoras de eficiencia y Estandarización con un 30%, 25%, y 15% respectivamente.

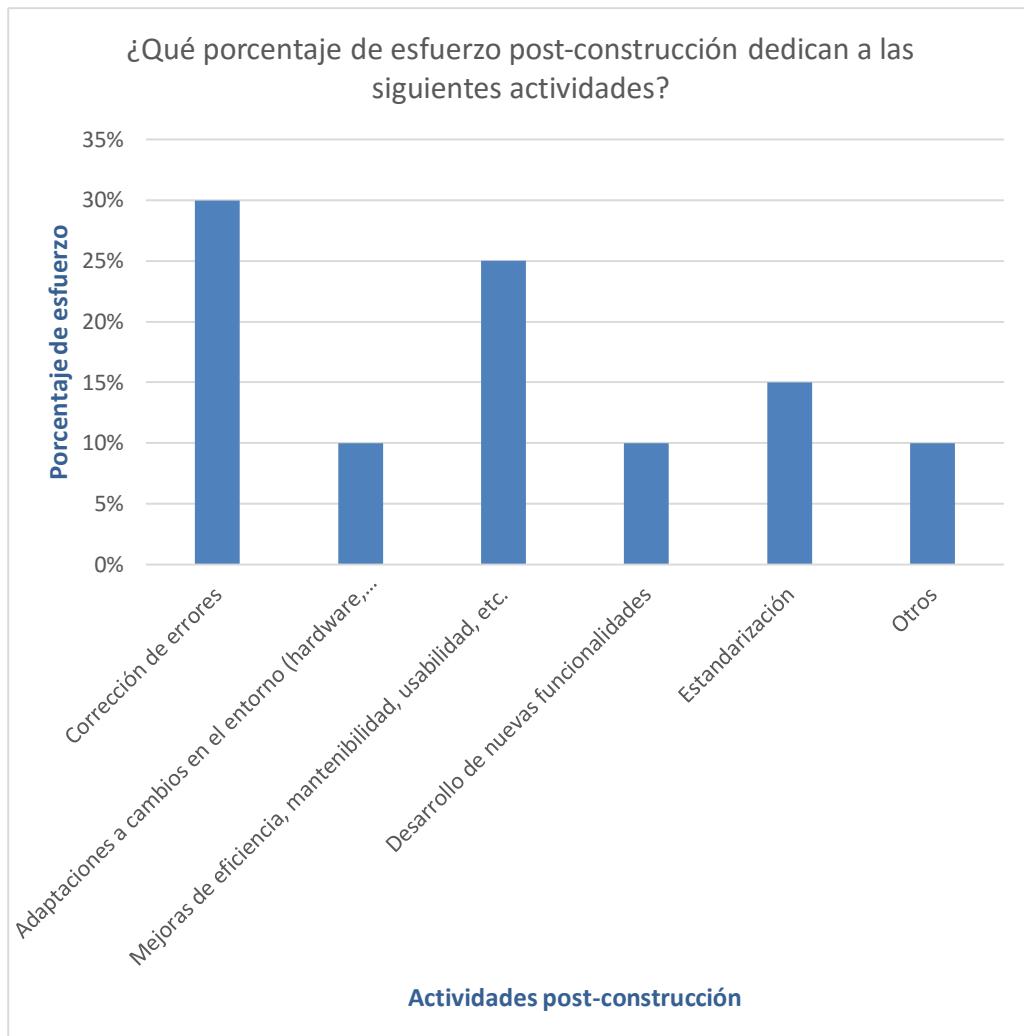


Gráfico 4.4. Qué porcentaje de esfuerzo post-construcción dedican a las siguientes actividades

Fuente. Elaboración propia

En el gráfico N° 4.5., se presentan las herramientas que soportan el proceso de desarrollo de software en las consultoras certificadas en CMMI nivel 2 y 3. Las herramientas más utilizadas son aquellas que permiten entornos de desarrollo integrados (Team Foundation Server, Subversión, Git, etc.)

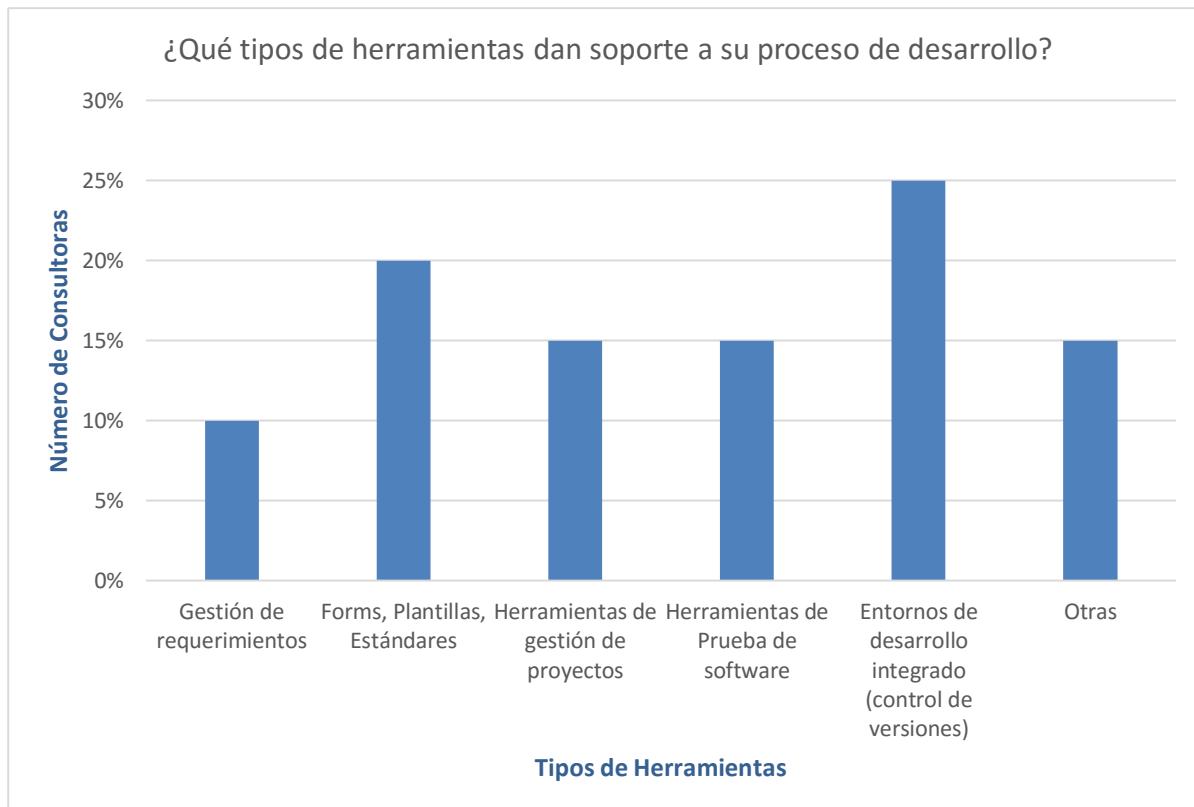
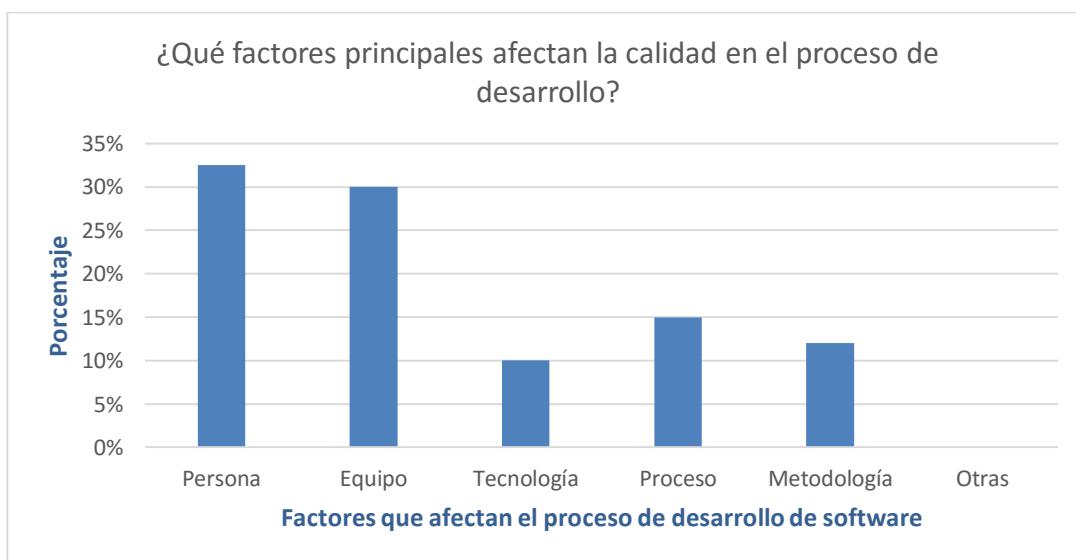


Gráfico 4.5. Qué tipos de herramientas dan soporte a su proceso de desarrollo

Fuente. Elaboración propia

En el gráfico N° 4.6., se muestran los principales factores que afectan la calidad en el proceso de desarrollo de software. El factor humano (Persona, Equipo) es el principal factor que afecta la calidad en el proceso con un 33% y 30% respectivamente.



Fuente. Elaboración propia

Gráfico 4.6. Qué factores principales afectan la calidad en el proceso de desarrollo

En el gráfico N° 4.7., se determina las políticas de calidad en las consultoras de software, donde un 25% de las consultoras encuestadas poseen alguna certificación de calidad y un porcentaje igual tiene planeado obtener una certificación de calidad.

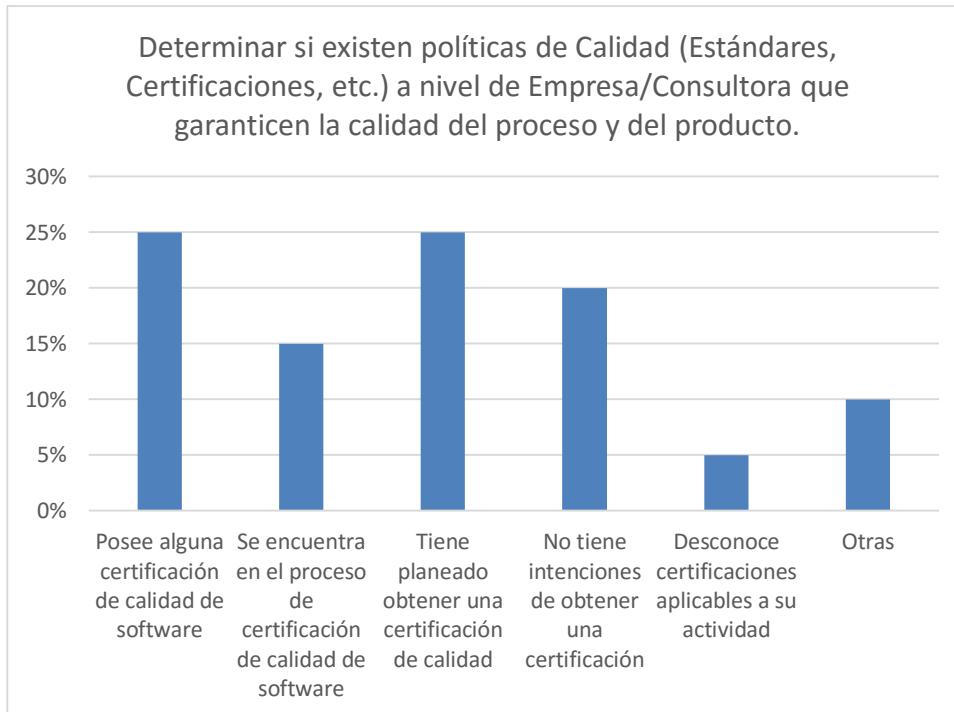


Gráfico 4.7. Determinar si existen políticas de Calidad (Estándares, Certificaciones, etc.) a nivel de Empresa/Consultora que garanticen la calidad del proceso y del producto

Fuente. Elaboración propia

En el gráfico N° 4.8., nos muestra que 8 consultoras indican que su proceso de desarrollo asegura la calidad y 4 consultoras indican que su proceso actual no garantiza la calidad en su proceso de desarrollo.

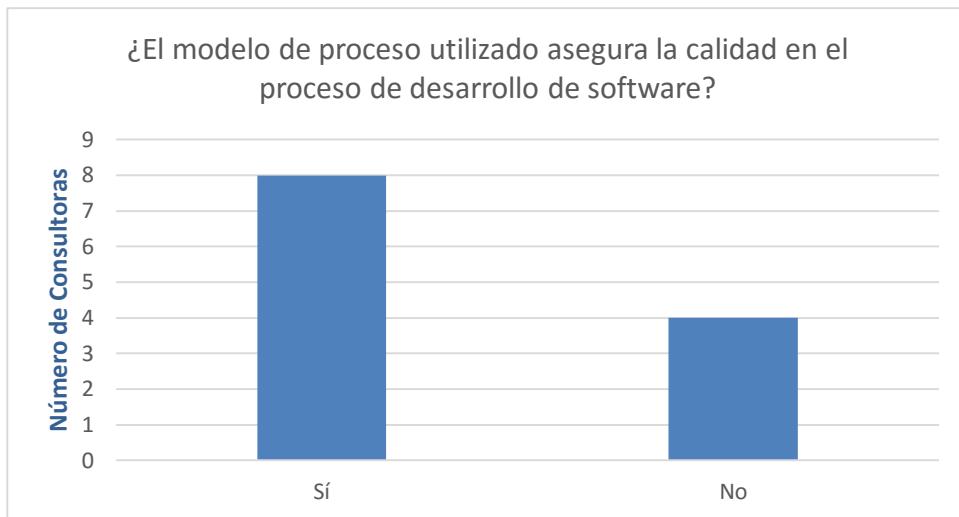


Gráfico 4.8. El modelo de proceso utilizado asegura la calidad en el proceso de desarrollo de software

Fuente. Elaboración propia

En el gráfico N° 4.9., nos muestra que 8 consultoras indican que la metodología utilizada en su proceso de desarrollo de software asegura la calidad de sus productos, y 4 consultoras indican que su metodología actual no asegura la calidad del producto.

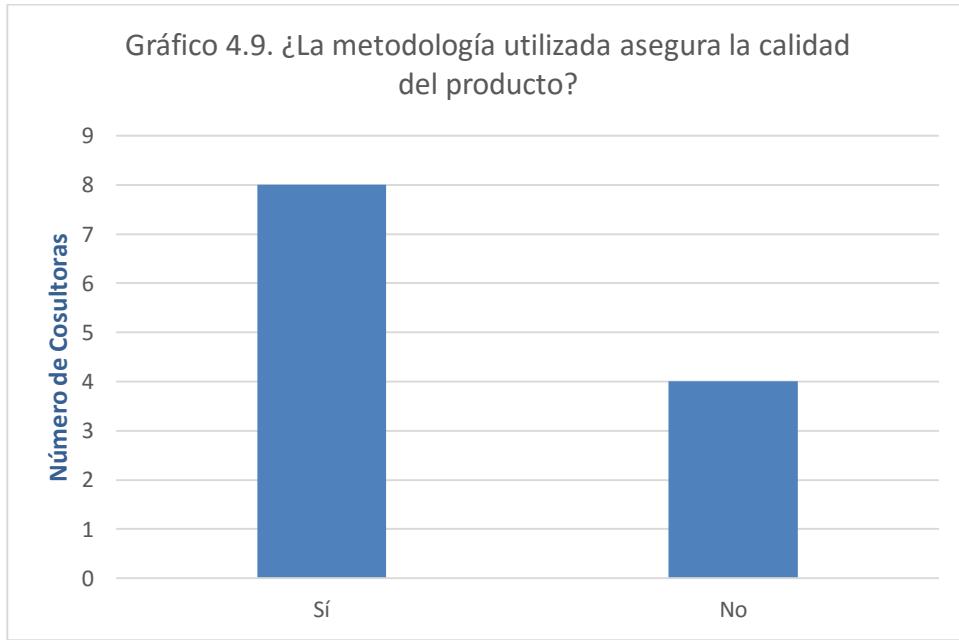


Gráfico 4.9. La metodología utilizada asegura la calidad del producto

Fuente. Elaboración propia

En el gráfico N° 4.10., se presenta las respuestas de las consultoras encuestadas a la pregunta de los posibles factores que asegurarían la calidad en el proceso de desarrollo de software. 5 consultoras indicaron que el factor principal que posiblemente aseguré la calidad en el proceso son los ingenieros de software y 4 consultoras indicaron que posiblemente la calidad en el proceso de software sería introducido por el equipo.

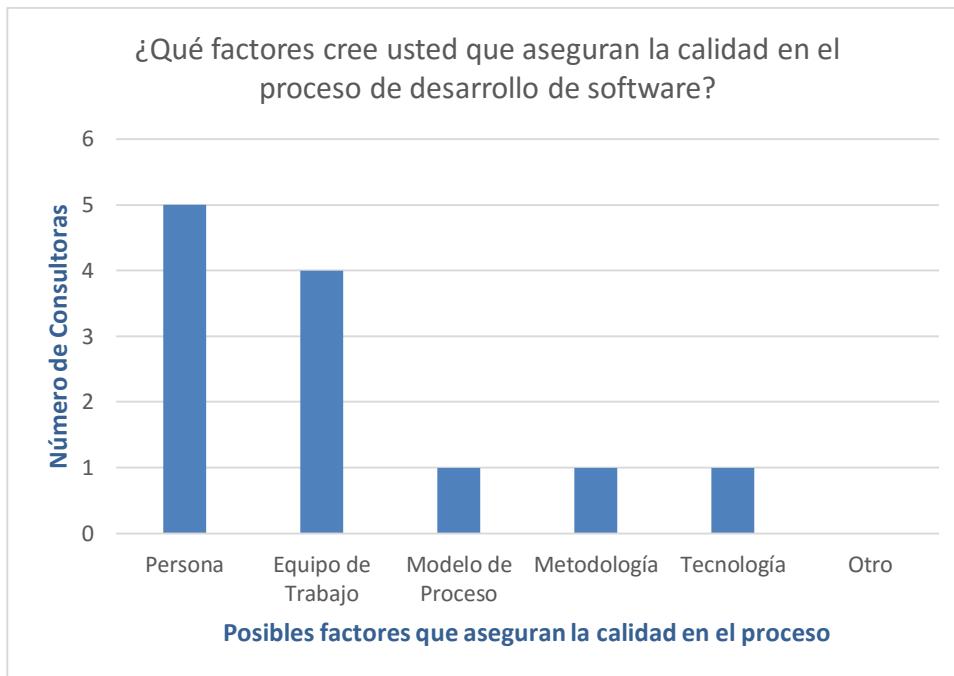


Gráfico 4.10. Qué factores cree usted que aseguran la calidad en el proceso de desarrollo de software

Fuente. Elaboración propia

El gráfico N° 4.11., muestra la actual gestión de defectos de las consultoras. 9 consultoras indican que sí poseen una gestión de defectos, soportándose en documentos, 2 indican que sí poseen gestión de defectos basados en una herramienta especializada y una consultora indica que no realiza gestión de defectos.

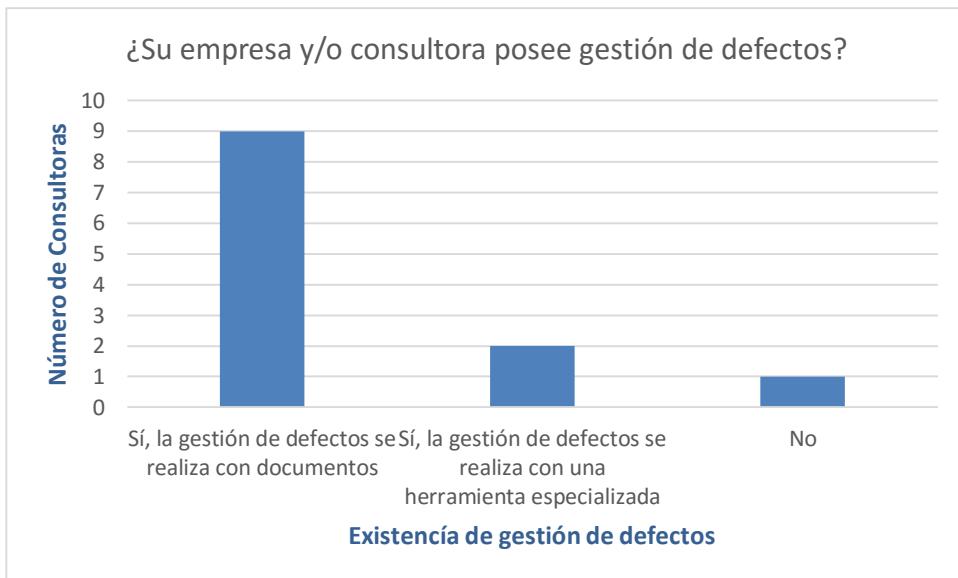


Gráfico 4.11. Su empresa y/o consultora posee gestión de defectos.

Fuente. Elaboración propia

En el gráfico N° 4.12., se presenta la información sobre los procesos de testing de las consultoras encuestadas. 7 consultoras indicaron que si poseen procesos de testing como una estrategia global, 4 indicaron que cada proyecto define su propio proceso de testing y una consultora indicó que realizan testing ad-hoc.

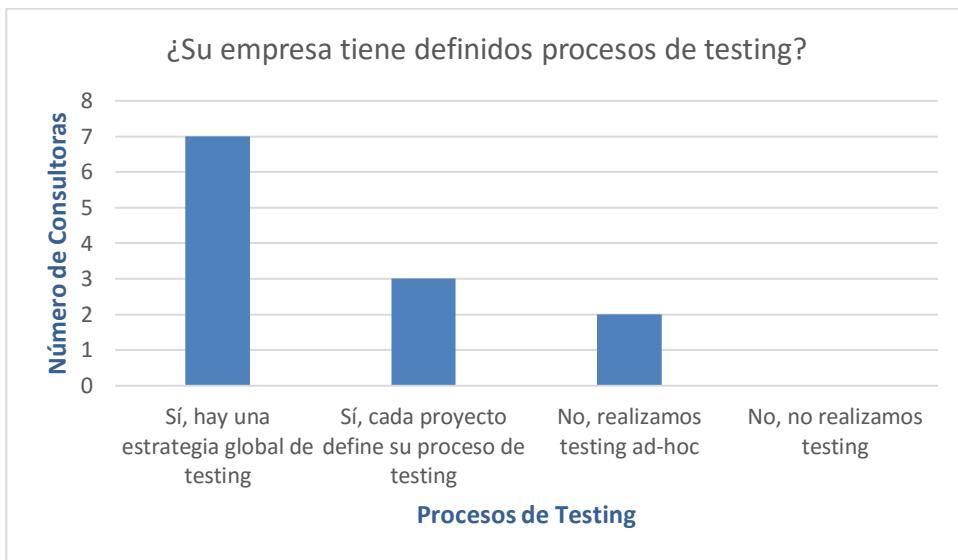


Gráfico 4.12. Su empresa tiene definidos procesos de testing.

Fuente. Elaboración propia

En el gráfico 4.13., se muestra el porcentaje de esfuerzo de las actividades que involucra el proceso de testing. La ejecución de las pruebas es la actividad que tiene más del 50% del esfuerzo total.

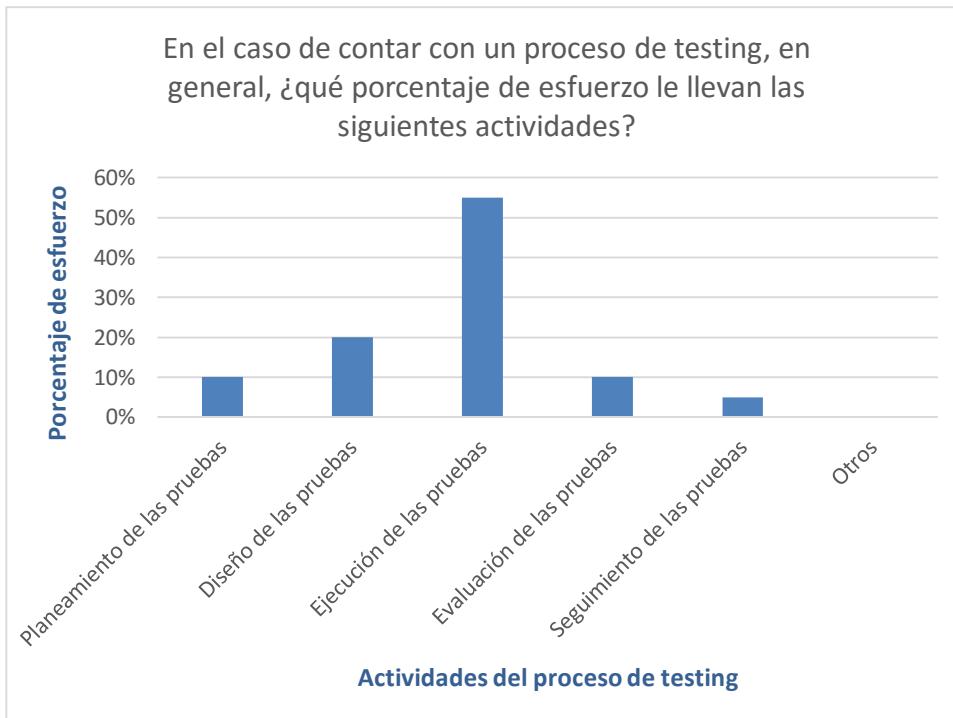


Gráfico 4.13. En el caso de contar con un proceso de testing, en general, ¿qué porcentaje de esfuerzo le llevan las siguientes actividades?

Fuente. Elaboración propia

En el gráfico 4.14., se presenta la información sobre quien realizan las pruebas de software. 5 Consultoras indicaron que las pruebas de testing lo realiza una empresa externa (outsourcing), 4 Consultoras indicaron que lo realiza su área de testing, 2 Consultoras indicaron el área que solicita el requerimiento y 1 Consultora indica que lo ejecuta el usuario final.

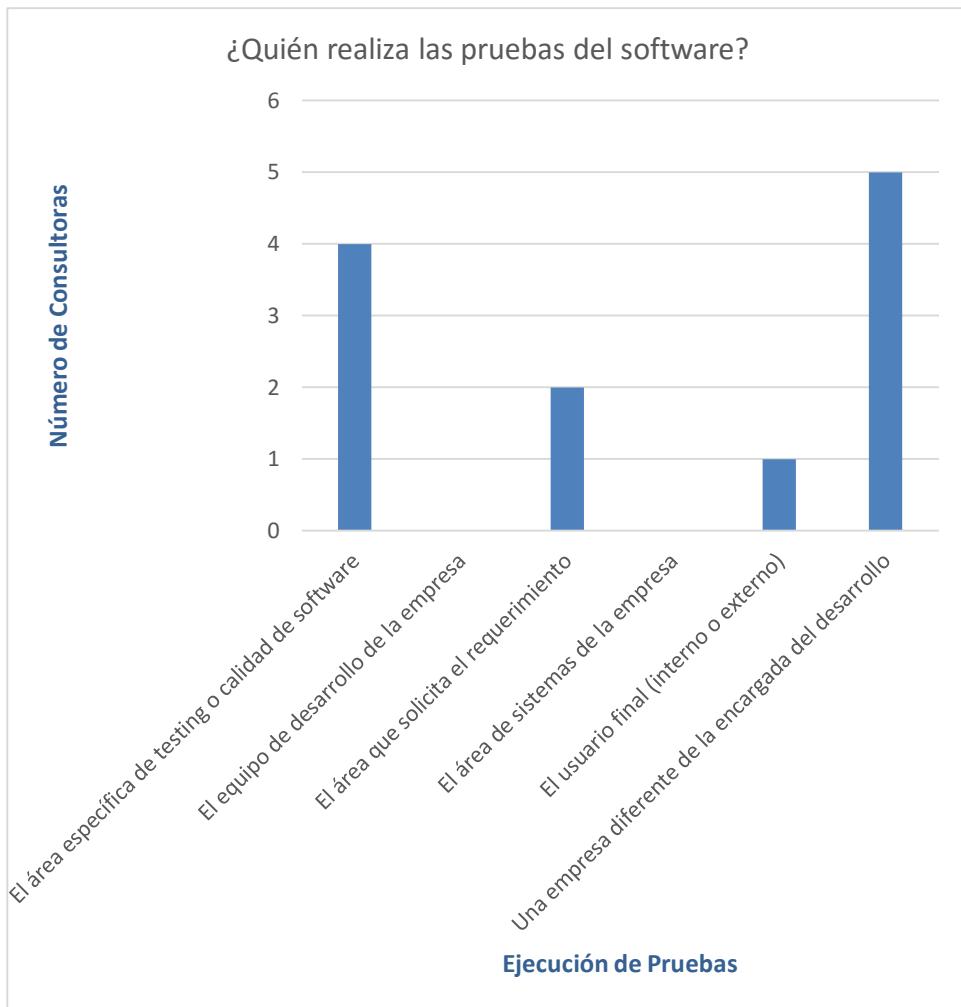


Gráfico 4.14. Quién realiza las pruebas del software.

Fuente. Elaboración propia

4.2. Análisis, interpretación y discusión de resultados

En el presente capítulo se describe el análisis, interpretación y la discusión de resultados de la presente investigación; mediante los indicadores propuestos observaremos si la integración del modelo asegura la calidad en el proceso de desarrollo de software.

El diseño de la investigación es Pre-Experimental, de Pre y Post prueba. Por ello inicialmente se realizó un Pre Test, una medición de los indicadores sin el modelo de aseguramiento de calidad en el proceso de desarrollo de software, para luego ser comparado con un Post Test, una medición de los indicadores con el modelo de aseguramiento de calidad integrado en el proceso de desarrollo de software.

Finalmente se realiza una comparación entre ambos datos obtenidos en diferentes tiempos, para la misma área de estudio.

A continuación se detalla la información obtenida desde los **proyectos de aplicación** donde se incorporó el modelo de aseguramiento de la calidad versus los proyectos con características similares (alcance, tiempo y costo) (prueba previa) basado en su actual proceso de desarrollo. Los proyectos de aplicación se ejecutaron en la consultora **Accenture** por las razones expuestas en el punto 3.5 de la metodología.

En el gráfico 4.15 se muestra un análisis descriptivo de la comparación entre los proyectos cuyo proceso de desarrollo no fue afectado (Pre Test) versus los proyectos que integraron el modelo de aseguramiento de la calidad en su proceso de desarrollo (Post Test). En el análisis podemos observar que el proyecto con mayor porcentaje de costo de calidad donde no se afectó el proceso de desarrollo es de 19.85% respecto al costo real del proyecto versus un máximo de 7.41% respecto al costo real en los proyectos donde se integró el modelo de aseguramiento de la calidad.

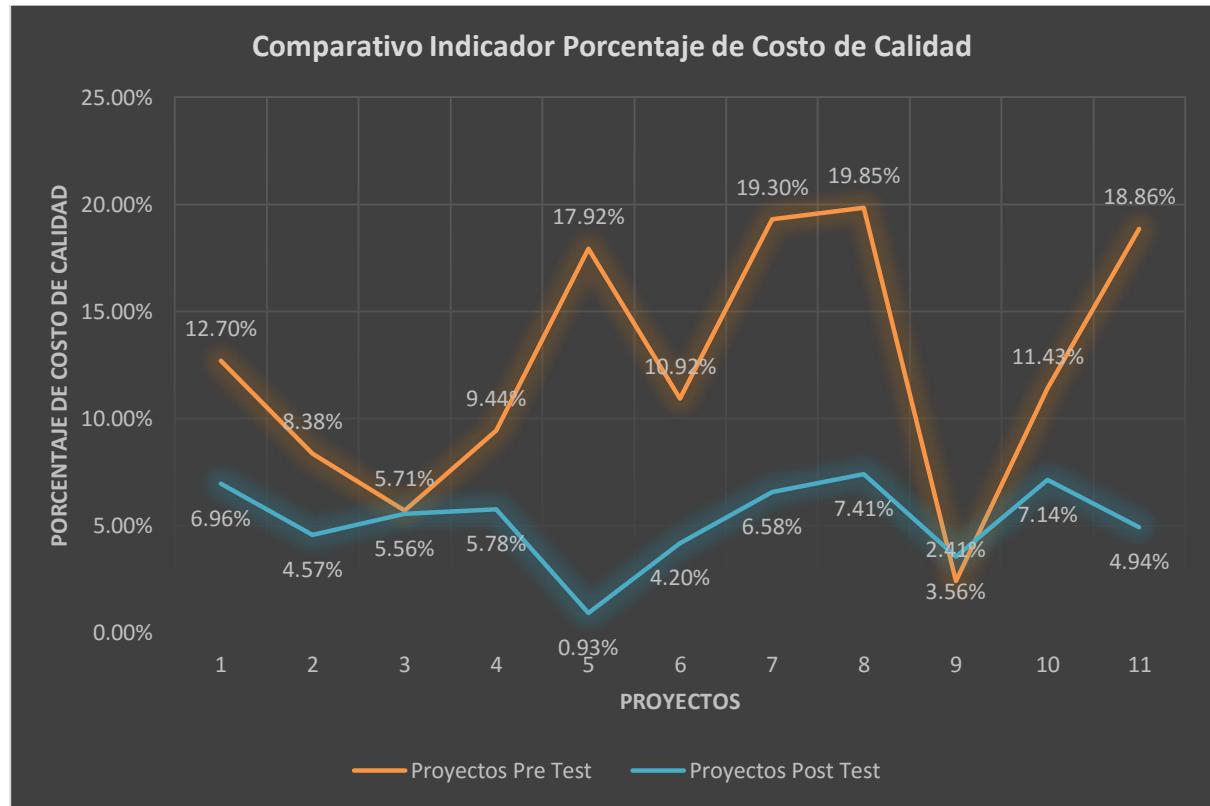


Gráfico 4.15. Comparativo Indicador Porcentaje de Costo de Calidad

Fuente. Elaboración propia

En el gráfico N° 4.16, se muestra un análisis descriptivo de la comparación del indicador Cantidad de Defectos por Proyecto. Los proyectos cuyo proceso de desarrollo no fue modificado (Pre Test) muestran un máximo de 249 defectos introducidos durante su ciclo de vida, mientras que el máximo de defectos entre los proyectos que integraron el modelo de aseguramiento (Post Test) es de 95 errores introducidos durante el ciclo de vida.

La cantidad de defectos por proyecto es la sumatoria de los defectos de las fases del ciclo de vida:

- Reporte de Especificaciones Funcionales.
- Reporte de Especificaciones Técnicas.
- Arquitectura y Construcción.
- Certificación.
- Implementación y Seguimiento.

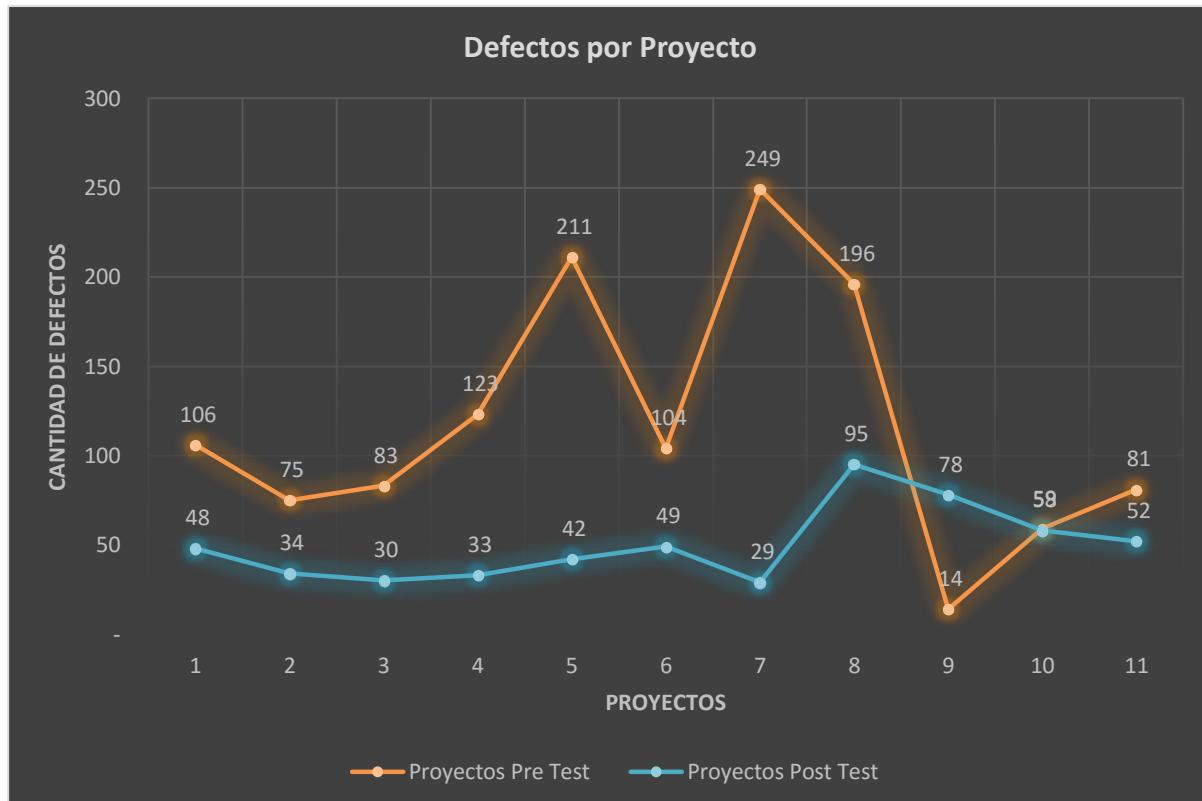


Gráfico 4.16. Comparativo Indicador Cantidad de Defectos por Proyecto

Fuente. Elaboración propia

En el gráfico N° 4.17, se muestra un análisis descriptivo de la comparación del indicador Porcentaje de Desviación de Costo respecto a la Línea Base de los Proyectos. Los proyectos cuyo proceso de desarrollo no fue modificado (Pre Test) muestran un máximo de desviación de 33.07% versus un máximo de desviación de costo de 11.29% entre los proyectos que integraron el modelo de aseguramiento (Post Test).

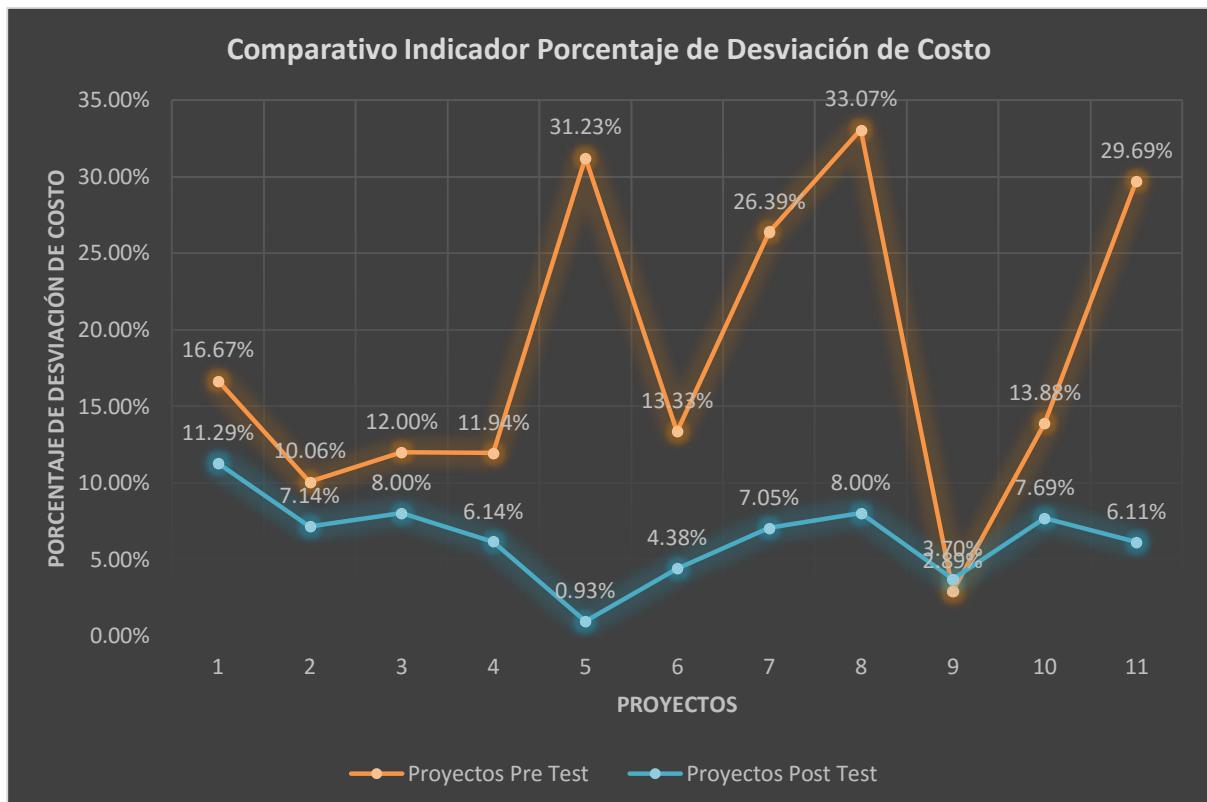


Gráfico 4.17 Comparativo Indicador Porcentaje de Desviación de Costo

Fuente. Elaboración propia

En el gráfico N° 4.18, se muestra el análisis descriptivo de la comparación del indicador Grado de Satisfacción del Cliente respecto de los Proyectos. Los proyectos cuyo proceso de desarrollo no fue modificado (Pre Test) muestra que la valoración más alta corresponde a los Clientes Insatisfechos con un 46.97%; mientras que la valoración más alta entre los proyectos que integraron el modelo de aseguramiento (Post Test) es de 48.48% de Clientes Satisfechos.

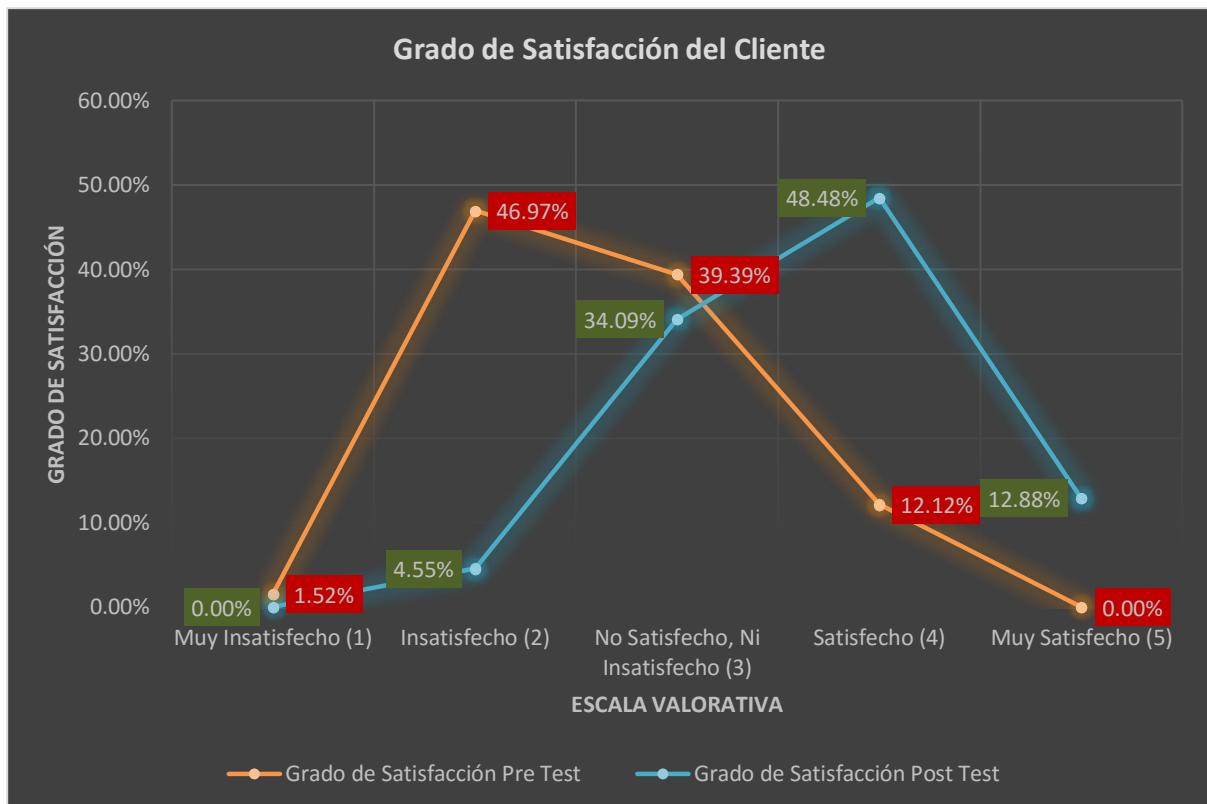


Gráfico 4.18. Comparativo Indicador Grado de Satisfacción del Cliente

Fuente. Elaboración propia

4.3. Pruebas de hipótesis

La integración del modelo de aseguramiento de la calidad basado en los modelos de madurez de capacidades CMMi, TSP y PSP permite asegurar la calidad en el proceso de desarrollo de software.

4.3.1. Indicador Porcentaje de Costo de Calidad

a) Prueba de Normalidad

Como las muestras obtenidas son menores a 30, la prueba de normalidad para comprobar si los resultados son normales es la de Shapiro-Wilk, los resultados que se obtuvieron a través del software SPSS v22.0 se muestran en la cuadro 4.1.

Cuadro 4.1

Prueba de normalidad indicador porcentaje de costo de calidad

Shapiro-Wilk				
Indicador	Grupo	Tamaño Muestra	Nivel de Significancia	Nivel de Significación α
Porcentaje de Costo de Calidad	Modelo Actual (Prueba Previa)	11	0.401	0.05
	Modelo de Aseguramiento (Post Prueba)	11	0.227	0.05

Fuente. Elaboración Propia

En el cuadro 4.1 se observan los resultados de la prueba de normalidad, la cual permite indicar que la distribución de la muestra de ambos grupos es normal (nivel significancia $> \alpha$).

Gráfico Q-Q normal de Porcentaje de Costo de Calidad

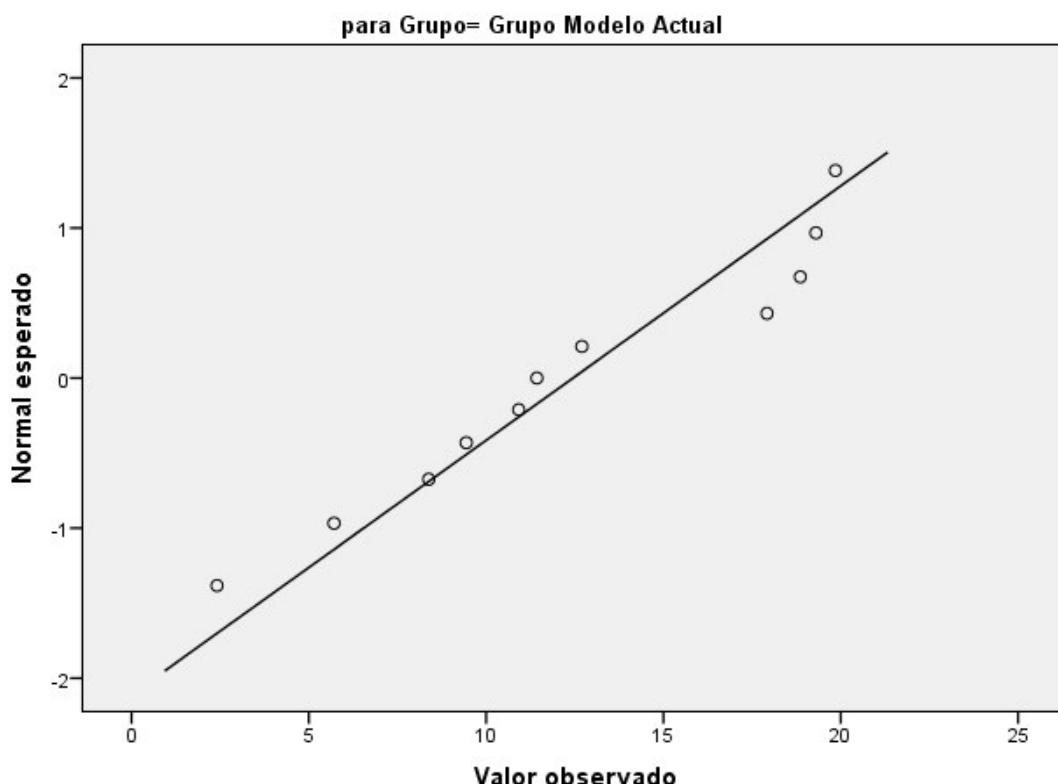


Gráfico 4.19. Q-Q de normalidad para datos del indicador porcentaje de costo de calidad – Prueba Previa

Fuente. Elaboración propia

En el gráfico 4.19 se observa que los puntos se acercan a la línea recta, esto significa que los datos de prueba coinciden con los datos normales, siendo así, se puede decir que los datos de porcentaje de costo de calidad son normales para el grupo de la prueba previa.

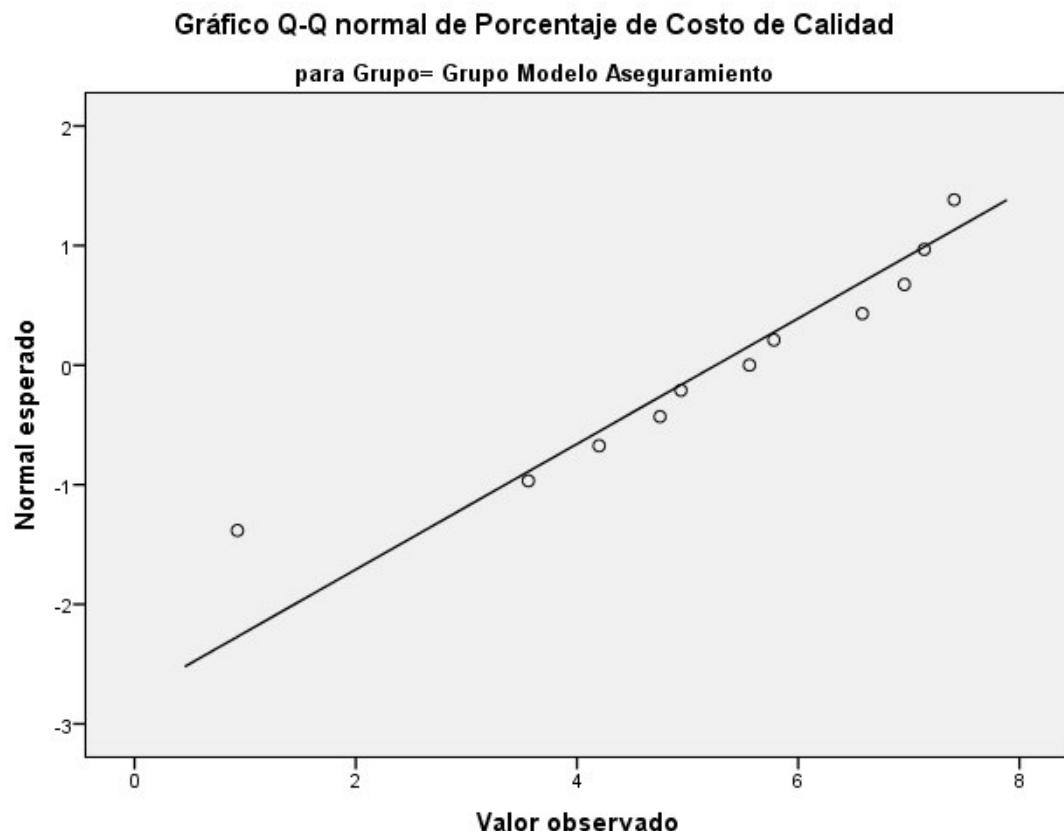


Gráfico 4.20. Q-Q de normalidad para datos del indicador porcentaje de costo de calidad – Post Prueba

Fuente. Elaboración propia

En el gráfico 4.20 se observa que los puntos se acercan a la línea recta, esto significa que los datos de prueba coinciden con los datos normales, siendo así, se puede decir que los datos de porcentaje de costo de calidad son normales para el grupo de la post prueba.

b) Hipótesis Estadística

H1: La integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software reduce el costo de calidad en los proyectos.

H1₁: Existe una diferencia significativa en la reducción del costo de calidad con la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software.

H1₀: NO Existe una diferencia significativa en la reducción del costo de calidad con la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software.

Criterio para decidir es:

Si la probabilidad obtenida (t) P-VALOR $\leq \alpha$, rechace H_{1₀} (Se acepta H_{1₁}).

Si la probabilidad obtenida (t) P-VALOR $> \alpha$, no rechace H_{1₀} (Se acepta H_{1₀}).

c) Nivel de Significancia

Se escoge un nivel de significancia del 5% o 0.05, quiere decir que de cien valores, hay cinco valores entre los cien de rechazar la hipótesis, entonces tenemos un 95% de confianza de haber aceptado la hipótesis.

X = 5% (ERROR)

Nivel de confiabilidad ((1-X)=0.95)

d) Resultados

A continuación se presenta los resultados obtenidos luego de la aplicación de la prueba T a los grupos de la prueba previa y post prueba a través del software SPSS v22.0 (comparación de medias).

Cuadro 4.2

Resultados de la prueba de T del indicador porcentaje de costo de calidad

	Prueba de muestras independientes							
	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	prueba t para la igualdad de medias		
						Inferior	Superior	
Porcentaje de Costo de Calidad	Se asumen varianzas iguales	3,848	20	,001	7,19182	1,86904	3,29307	11,09057
	No se asumen varianzas iguales	3,848	12,062	,002	7,19182	1,86904	3,12187	11,26177

Fuente. Elaboración Propia

Por lo tanto:

De acuerdo al criterio de decisión especificado en la hipótesis estadística del indicador Porcentaje de Costo de Calidad se concluye:

Existe una diferencia significativa en la reducción del costo de calidad con la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software ($P\text{-VALOR} \leq \alpha = 0.002 \leq 0.05$).

4.3.2. Indicador Cantidad de Defectos en el Proceso de Desarrollo de Software

a) Prueba de Normalidad

Como las muestras obtenidas son menores a 30, la prueba de normalidad para comprobar si los resultados son normales es la de Shapiro-Wilk, los resultados que se obtuvieron a través del software SPSS v22.0 se muestran en la cuadro 4.3.

Cuadro 4.3

Prueba de normalidad indicador cantidad de defectos

Shapiro-Wilk				
Indicador	Grupo	Tamaño Muestra	Nivel de Significancia	Nivel de Significación α
Cantidad de Defectos	Modelo Actual (Prueba Previa)	11	0.345	0.05
	Modelo de Aseguramiento (Post Prueba)	11	0.092	0.05

Fuente. Elaboración Propia

En el cuadro 4.3 se observan los resultados de la prueba de normalidad, la cual permite indicar que la distribución de la muestra es normal (nivel significancia $> \alpha$).

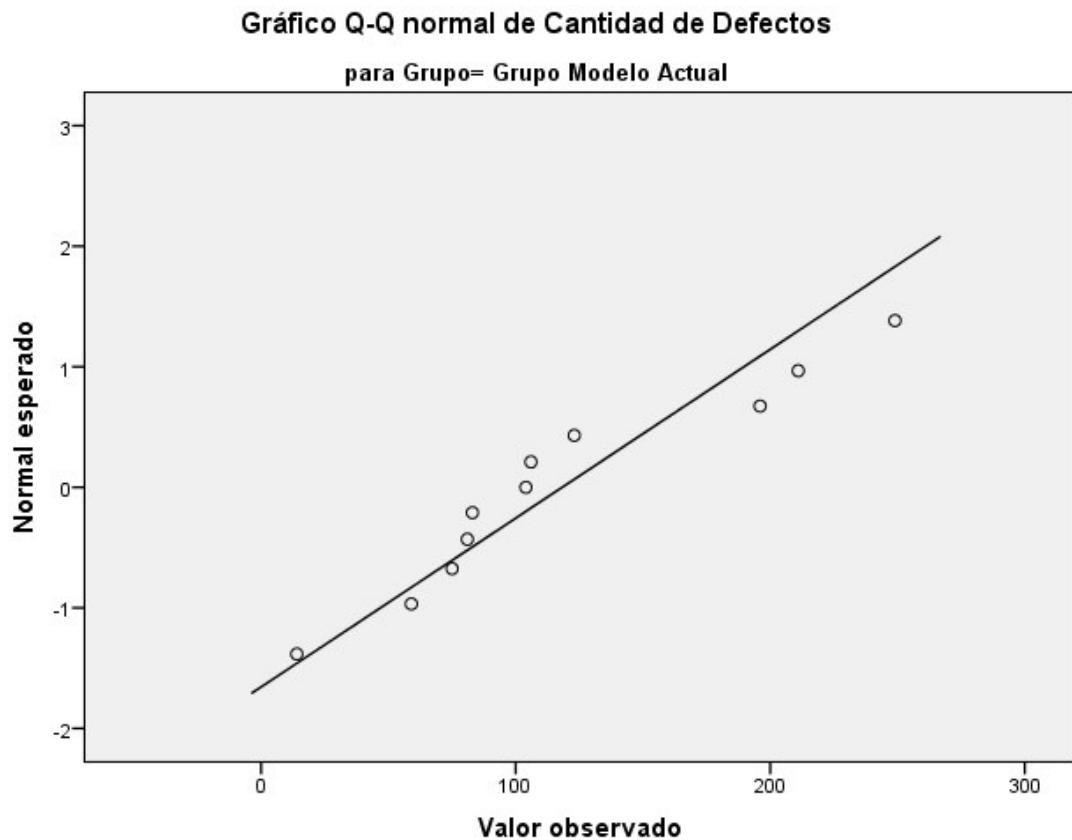


Gráfico 4.21. Q-Q de normalidad para datos del indicador cantidad de defectos – Prueba Previa

Fuente. Elaboración propia

En el gráfico 4.21 se observa que los puntos se acercan a la línea recta, esto significa que los datos de prueba coinciden con los datos normales, siendo así, se puede decir que los datos de cantidad de defectos en el proceso de desarrollo de software son normales para el grupo de la prueba previa (pre-test).

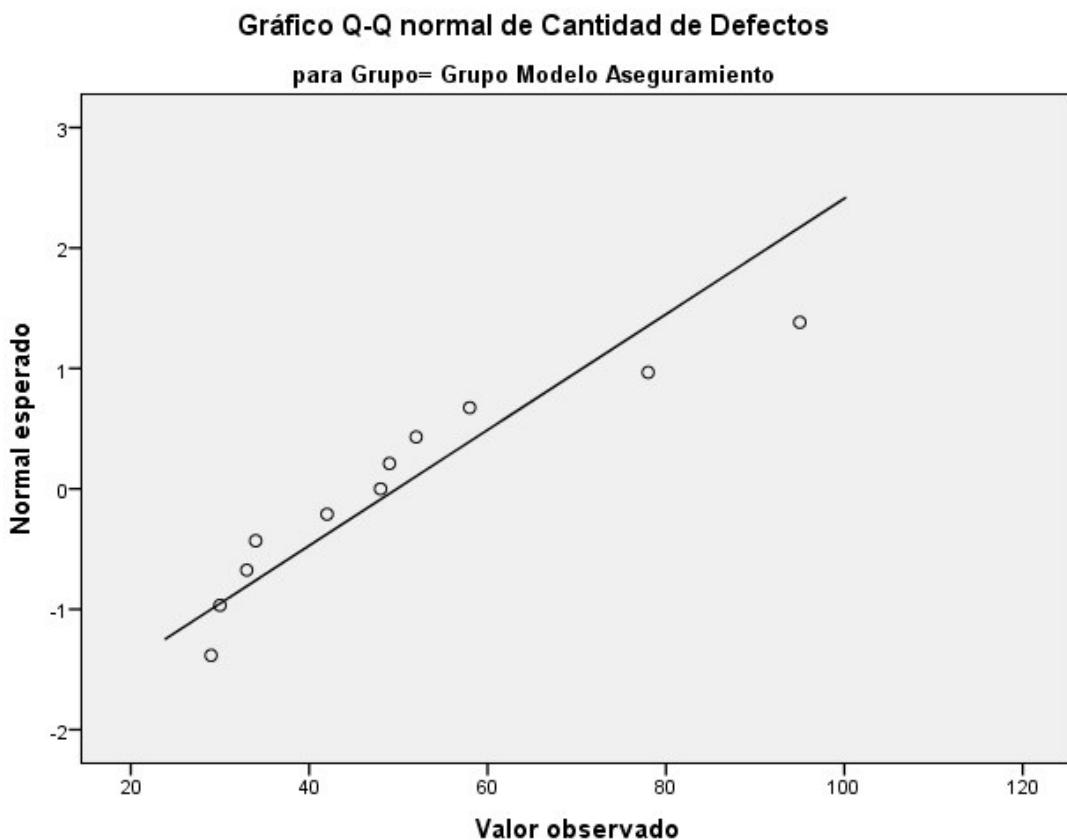


Gráfico 4.22. Q-Q de normalidad para datos del indicador cantidad de defectos – Post Prueba

Fuente. Elaboración propia

En el gráfico 4.22 se observa que los puntos se acercan a la línea recta, esto significa que los datos de prueba coinciden con los datos normales, siendo así, se puede decir que los datos de cantidad de defectos en el proceso de desarrollo de software son normales para el grupo de la post prueba (post-test).

b) Hipótesis Estadística

H2: La integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software reduce los defectos en los proyectos.

H2₁: Existe una diferencia significativa en la reducción de los defectos en los proyectos con la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software.

H₂₀: NO Existe una diferencia significativa en la reducción de los defectos en los proyectos con la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software.

Criterio para decidir es:

Si la probabilidad obtenida (t) P-VALOR $\leq \alpha$, rechace H₂₀ (Se acepta H₂₁).

Si la probabilidad obtenida (t) P-VALOR $> \alpha$, no rechace H₂₀ (Se acepta H₂₀).

c) Nivel de Significancia

Se escoge un nivel de significancia del 5% o 0.05, quiere decir que de cien valores, hay cinco valores entre los cien de rechazar la hipótesis, entonces tenemos un 95% de confianza de haber aceptado la hipótesis.

X = 5% (ERROR)

Nivel de confiabilidad ((1-X)=0.95)

d) Resultados

A continuación se presenta los resultados obtenidos luego de la aplicación de la prueba T a los grupos de la prueba previa y post prueba a través del software SPSS v22.0 (comparación de medias).

Cuadro 4.4

Resultados de la prueba de T del indicador cantidad de defectos

Prueba de muestras independientes							
	prueba t para la igualdad de medias						95% de intervalo de confianza de la diferencia
	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	Inferior	
Cantidad de Defectos	Se asumen varianzas iguales	3,052	20	,006	68,45455	22,42762	21,67134
	No se asumen varianzas iguales	3,052	11,687	,010	68,45455	22,42762	19,44320

Fuente. Elaboración Propia

Por lo tanto:

De acuerdo al criterio de decisión especificado en la hipótesis estadística del indicador Cantidad de Defectos se concluye:

Existe una diferencia significativa en la reducción de cantidad de defectos en el proceso de desarrollo de software con la integración del modelo de aseguramiento de la calidad ($P\text{-VALOR} \leq \alpha = 0.01 \leq 0.05$).

4.3.3. Indicador Porcentaje de Desviación de Costo respecto a la Línea Base

a) Prueba de Normalidad

Como las muestras obtenidas son menores a 30, la prueba de normalidad para comprobar si los resultados son normales es la de Shapiro-Wilk, los resultados que se obtuvieron a través del software SPSS v22.0 se muestran en la cuadro 4.5.

Cuadro 4.5

Prueba de normalidad indicador porcentaje de desviación de costo

Shapiro-Wilk				
Indicador	Grupo	Tamaño Muestra	Nivel de Significancia	Nivel de Significación α
Porcentaje de desviación de costo	Modelo Actual (Prueba Previa)	11	0.195	0.05
	Modelo de Aseguramiento (Post Prueba)	11	0.667	0.05

Fuente. Elaboración Propia

En el cuadro 4.5 se observan los resultados de la prueba de normalidad, la cual permite indicar que la distribución de la muestra es normal (nivel significancia $> \alpha$).

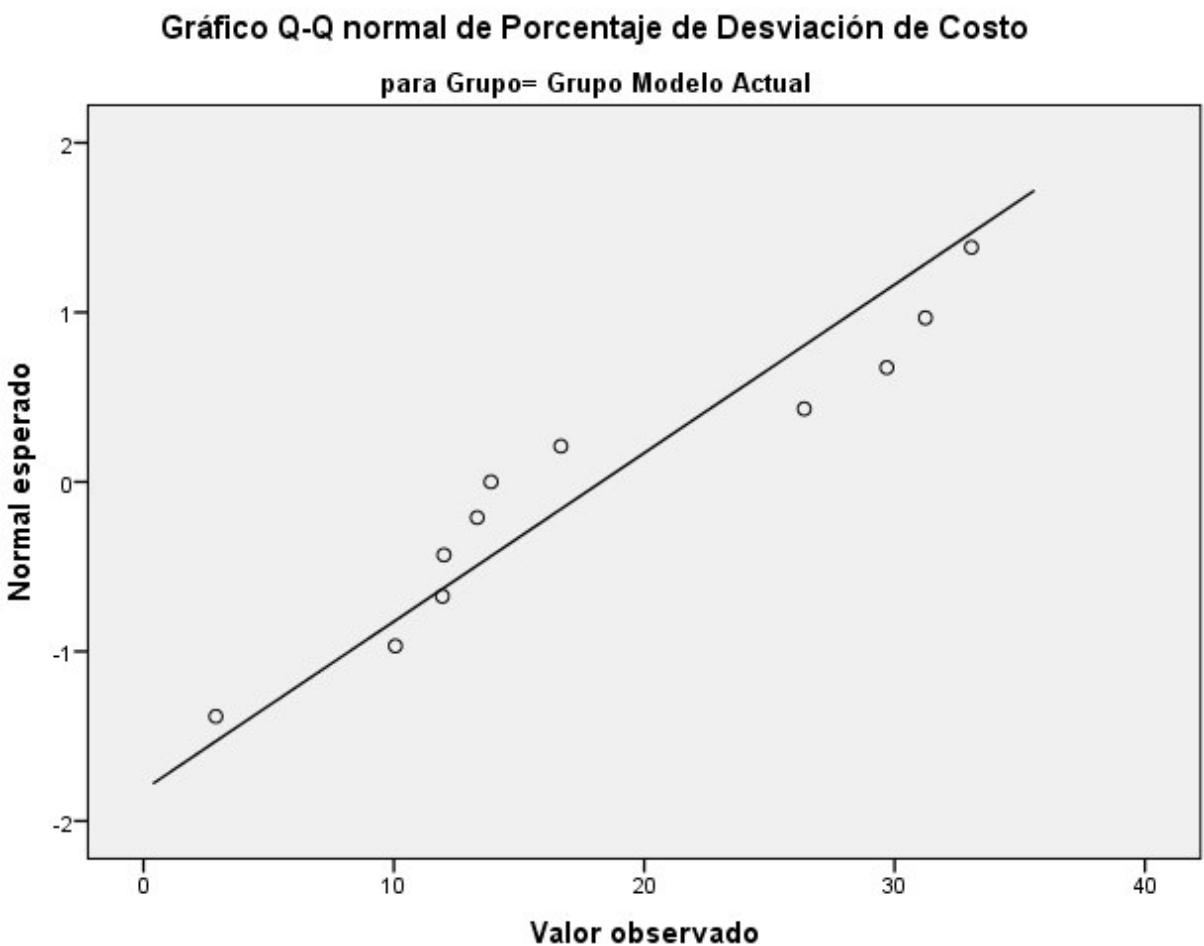


Gráfico 4.23. Q-Q de normalidad para datos del indicador porcentaje de desviación de costo – Prueba Previa

Fuente. Elaboración propia

En el gráfico 4.23 se observa que los puntos se acercan a la línea recta, esto significa que los datos de prueba coinciden con los datos normales, siendo así, se puede decir que los datos de porcentaje de desviación de costo son normales para el grupo de la prueba previa (pre-test).

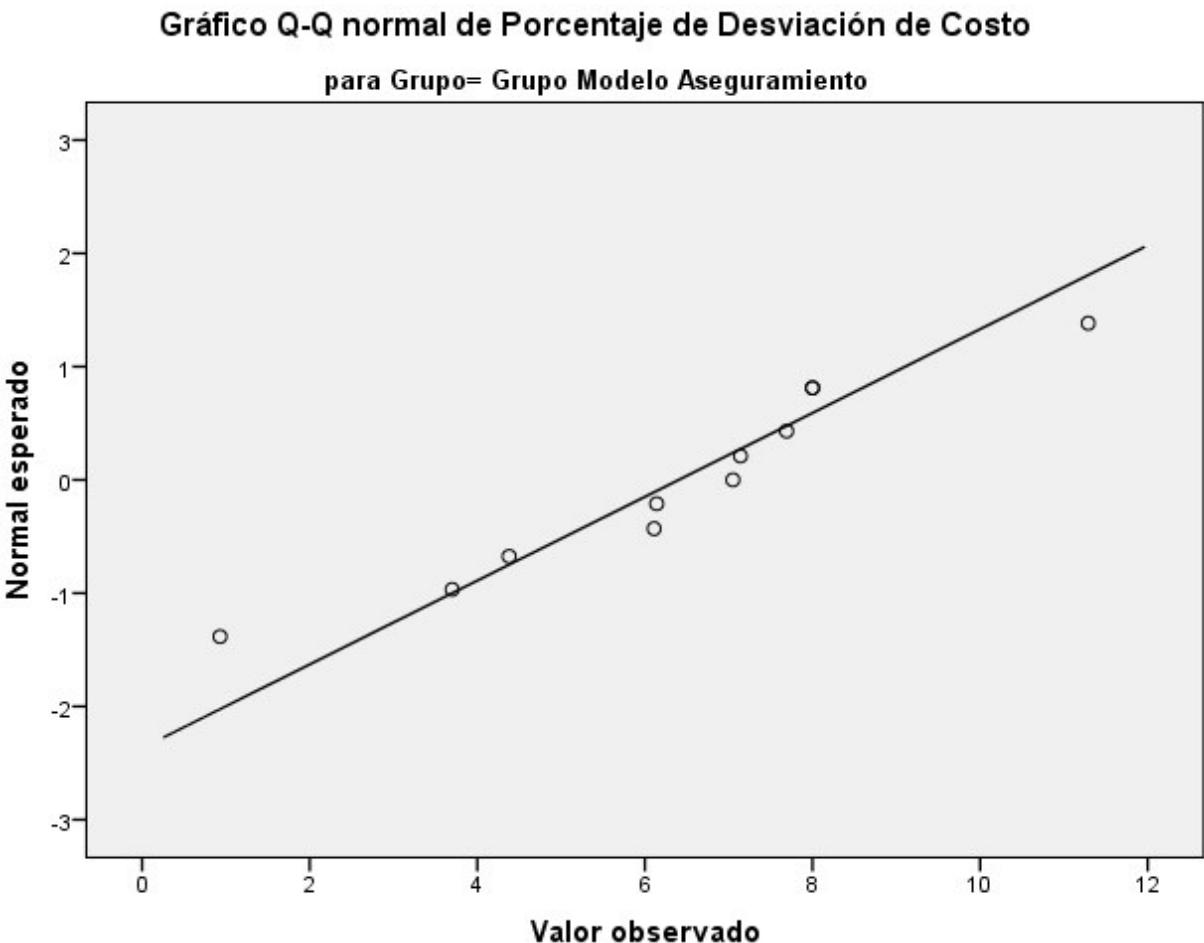


Gráfico 4.24. Q-Q de normalidad para datos del indicador porcentaje de desviación de costo – Post Prueba

Fuente. Elaboración propia

En el gráfico 4.24 se observa que los puntos se acercan a la línea recta, esto significa que los datos de prueba coinciden con los datos normales, siendo así, se puede decir que los datos de porcentaje de desviación de costo son normales para el grupo de la post prueba (post-test).

b) Hipótesis Estadística

H3: La integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software incrementa la rentabilidad de los proyectos.

H3₁: Existe una diferencia significativa en el incremento de la rentabilidad de los proyectos con la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software.

H3₀: NO Existe una diferencia significativa en el incremento de la rentabilidad en los proyectos con la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software.

Criterio para decidir es:

Si la probabilidad obtenida (t) P-VALOR $\leq \alpha$, rechace H_{3₀} (Se acepta H_{3₁}).

Si la probabilidad obtenida (t) P-VALOR $> \alpha$, no rechace H_{3₀} (Se acepta H_{3₀}).

c) Nivel de Significancia

Se escoge un nivel de significancia del 5% o 0.05, quiere decir que de cien valores, hay cinco valores entre los cien de rechazar la hipótesis, entonces tenemos un 95% de confianza de haber aceptado la hipótesis.

X = 5% (ERROR)

Nivel de confiabilidad ((1-X)=0.95)

d) Resultados

A continuación se presenta los resultados obtenidos luego de la aplicación de la prueba T a los grupos de la prueba previa y post prueba a través del software SPSS v22.0 (comparación de medias).

Cuadro 4.6

Resultados de la prueba de T del indicador porcentaje de desviación de costo

		Prueba de muestras independientes						
		prueba t para la igualdad de medias						
		t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	95% de intervalo de confianza de la diferencia	
Porcentaje de Desviación de Costo	Se asumen varianzas iguales	3,782	20	,001	11,88364	3,14213	5,32928	18,43799
	No se asumen varianzas iguales	3,782	11,436	,003	11,88364	3,14213	4,99990	18,76737

Fuente. Elaboración Propia

Por lo tanto:

De acuerdo al criterio de decisión especificado en la hipótesis estadística del indicador Porcentaje de Desviación de Costo se concluye:

Existe una diferencia significativa en el incremento de la rentabilidad de los proyectos de software con la integración del modelo de aseguramiento de la calidad ($P\text{-VALOR} \leq \alpha = 0.03 \leq 0.05$).

4.3.4. Indicador Porcentaje de Clientes Satisfechos

a) Prueba de Normalidad

Como las muestras obtenidas son menores a 30, la prueba de normalidad para comprobar si los resultados son normales es la de Shapiro-Wilk, los resultados que se obtuvieron a través del software SPSS v22.0 se muestran en la cuadro 4.7.

Cuadro 4.7

Prueba de normalidad indicador porcentaje de clientes satisfechos

Shapiro-Wilk				
Indicador	Grupo	Tamaño Muestra	Nivel de Significancia	Nivel de Significación α
Porcentaje de clientes satisfechos	Modelo Actual (Prueba Previa)	11	0.115	0.05
	Modelo de Aseguramiento (Post Prueba)	11	0.805	0.05

Fuente. Elaboración Propia

En el cuadro 4.7 se observan los resultados de la prueba de normalidad, la cual permite indicar que la distribución de la muestra es normal (nivel significancia $> \alpha$).

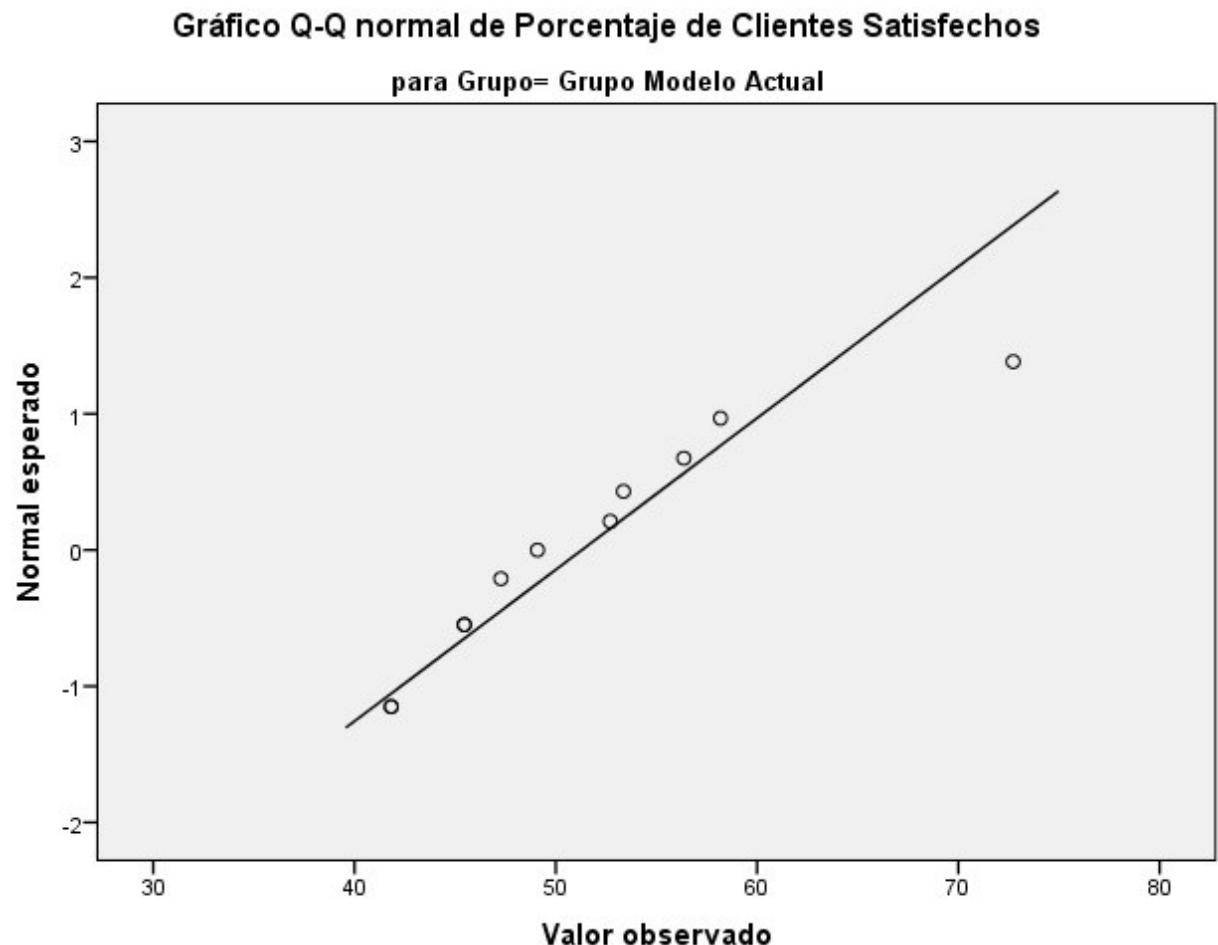


Gráfico 4.25. Q-Q de normalidad para datos del indicador porcentaje de clientes satisfechos – Prueba Previa

Fuente. Elaboración propia

En el gráfico 4.25 se observa que los puntos se acercan a la línea recta, esto significa que los datos de prueba coinciden con los datos normales, siendo así, se puede decir que los datos de porcentaje de clientes satisfechos son normales para el grupo de la prueba previa (pre-test).



Gráfico 4.26. Q-Q de normalidad para datos del indicador porcentaje de clientes satisfechos – Post Prueba

Fuente. Elaboración propia

En el gráfico 4.26 se observa que los puntos se acercan a la línea recta, esto significa que los datos de prueba coinciden con los datos normales, siendo así, se puede decir que los datos de porcentaje de clientes satisfechos son normales para el grupo de la post prueba (post-test).

b) Hipótesis Estadística

H4: La integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software aumenta el grado de satisfacción de los clientes.

H4₁: Existe una diferencia significativa en el grado de satisfacción de los clientes con la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software.

H4₀: NO Existe una diferencia significativa en el grado de satisfacción de los clientes con la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software.

Criterio para decidir es:

Si la probabilidad obtenida (t) P-VALOR $\leq \alpha$, rechace H_{4₀} (Se acepta H_{4₁}).

Si la probabilidad obtenida (t) P-VALOR $> \alpha$, no rechace H_{4₀} (Se acepta H_{4₀}).

c) Nivel de Significancia

Se escoge un nivel de significancia del 5% o 0.05, quiere decir que de cien valores, hay cinco valores entre los cien de rechazar la hipótesis, entonces tenemos un 95% de confianza de haber aceptado la hipótesis.

X = 5% (ERROR)

Nivel de confiabilidad ((1-X)=0.95)

d) Resultados

A continuación se presenta los resultados obtenidos luego de la aplicación de la prueba T a los grupos de la prueba previa y post prueba a través del software SPSS v22.0 (comparación de medias).

Cuadro 4.8

Resultados de la prueba de T del indicador porcentaje de desviación de costo

		Prueba de muestras independientes						
		prueba t para la igualdad de medias						95% de intervalo de confianza de la diferencia
		t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	Inferior	Superior
Porcentaje de Clientes Satisfechos	Se asumen varianzas iguales	-6,349	20	,000	-22,42309	3,53163	-29,78995	-15,05624
	No se asumen varianzas iguales	-6,349	19,391	,000	-22,42309	3,53163	-29,80482	-15,04136

Fuente. Elaboración Propia

Por lo tanto:

De acuerdo al criterio de decisión especificado en la hipótesis estadística del indicador Porcentaje de Clientes Satisfechos se concluye:

Existe una diferencia significativa en el porcentaje de clientes satisfechos con la integración del modelo de aseguramiento de la calidad ($P\text{-VALOR} \leq \alpha = 0.00 \leq 0.05$).

4.4. Presentación de resultados

A continuación se presentan los resultados obtenidos en el punto 4.3., referente a las pruebas de hipótesis:

4.4.1. Indicador Porcentaje de Costo de Calidad

a) Análisis de Resultados

Dado que el resultado hallado ($P\text{-VALOR} \leq \alpha = 0.002 \leq 0.05$), se puede determinar que existe una diferencia significativa en las medias antes y después de la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software; por lo tanto se rechaza la hipótesis nula H_{10} y se acepta la hipótesis alternativa H_{11} , con un nivel de confianza del 95%, siendo la integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software una solución para el problema de investigación.

b) Análisis Descriptivos

En el gráfico 4.27 se observa los porcentajes promedios del costo de calidad en las pruebas antes (pre-test) y después (post-test) de la incorporación el modelo de aseguramiento de la calidad en el proceso de desarrollo de software, encontrándose una reducción de 7.19%, pasando del 12.4% en el pre-test al 5.2% en el post-test.

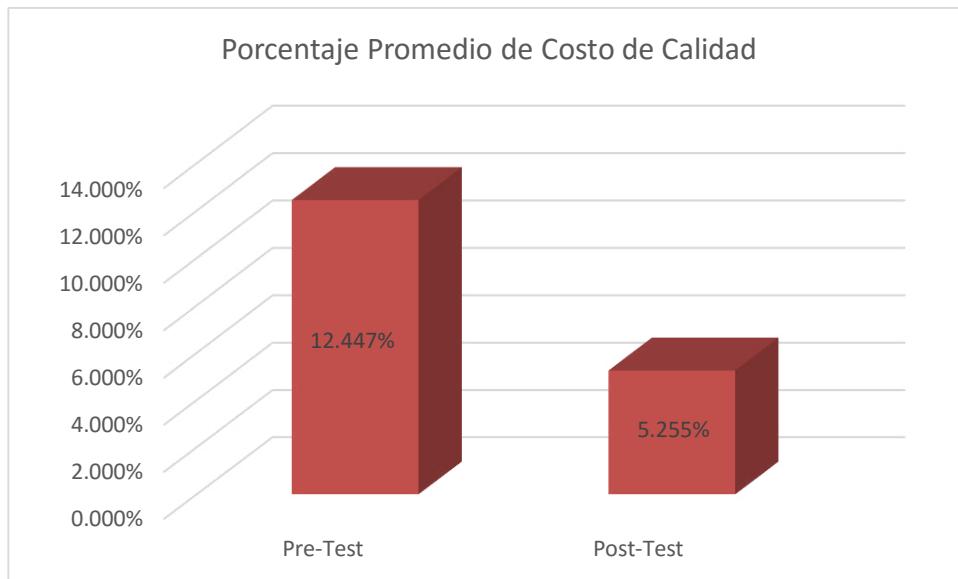


Gráfico 4.27. Porcentaje promedio de costo de calidad

Fuente. Elaboración propia

4.4.2. Indicador Cantidad de Defectos en el Proceso de Desarrollo de Software

a) Análisis de Resultados

Dado que el resultado hallado ($P\text{-VALOR} \leq \alpha = 0.010 \leq 0.05$), se puede determinar que existe una diferencia significativa en las medias antes y después de la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software; por lo tanto se rechaza la hipótesis nula H_20 y se acepta la hipótesis alternativa H_{21} , con un nivel de confianza del 95%, siendo la integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software una solución para el problema de investigación.

b) Análisis Descriptivos

En el gráfico 4.28 se observa la cantidad promedio de defectos en las pruebas antes (pre-test) y después (post-test) de la incorporación el modelo de aseguramiento de la calidad en el proceso de desarrollo de software, encontrándose una reducción de 68.45 defectos, pasando del 118.27 en el pre-test al 49.82 en el post-test.

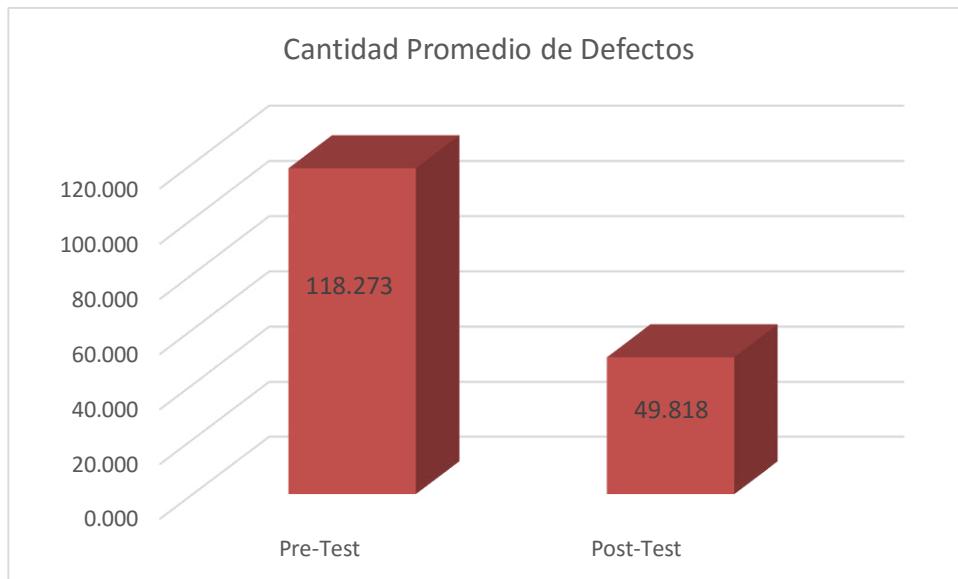


Gráfico 4.28. Cantidad promedio de defectos

Fuente. Elaboración propia

4.4.3. Indicador Porcentaje de Desviación de Costo respecto a la Línea Base

a) Análisis de Resultados

Dado que el resultado hallado ($P\text{-VALOR} \leq \alpha = 0.003 \leq 0.05$), se puede determinar que existe una diferencia significativa en las medias antes y después de la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software; por lo tanto se rechaza la hipótesis nula H_{30} y se acepta la hipótesis alternativa H_{31} , con un nivel de confianza del 95%, siendo la integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software una solución para el problema de investigación.

b) Análisis Descriptivos

En el gráfico 4.29 se observa el porcentaje promedio de desviación de costo respecto a la línea en las pruebas antes (pre-test) y después (post-test) de la incorporación el modelo de aseguramiento de la calidad en el proceso de desarrollo de software, encontrándose una reducción de 11.883%, pasando del 18.286% en el pre-test al 6.403 en el post-test. Por tanto se puede concluir que la rentabilidad del proyecto se incrementa porque existe una reducción en la desviación de costo de los proyectos.

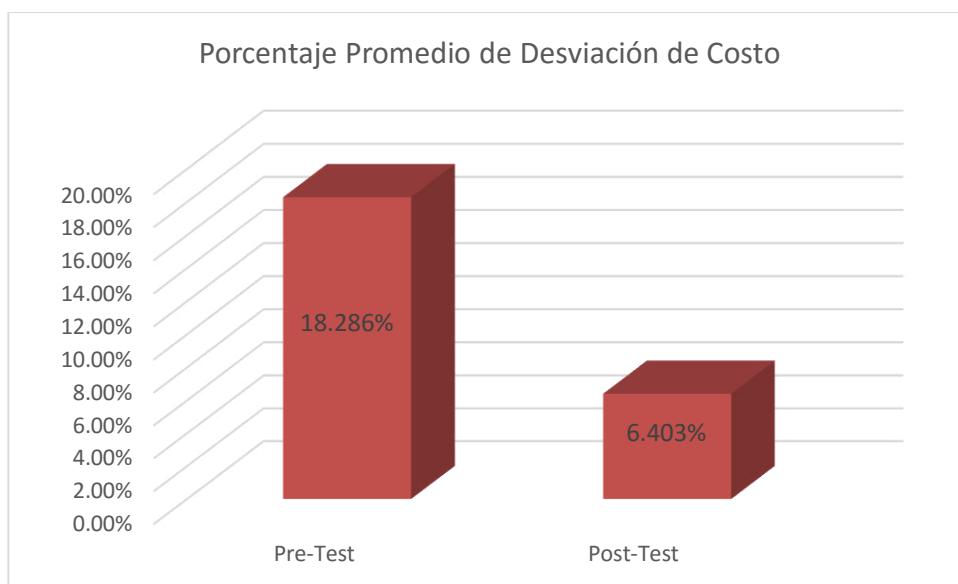


Gráfico 4.29. Porcentaje promedio de desviación de costo

Fuente. Elaboración propia

4.4.4. Indicador Porcentaje de Clientes Satisfechos

a) Análisis de Resultados

Dado que el resultado hallado ($P\text{-VALOR} \leq \alpha = 0.000 \leq 0.05$), se puede determinar que existe una diferencia significativa en las medias antes y después de la integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software; por lo tanto se rechaza la hipótesis nula H_0 y se acepta la hipótesis alternativa H_1 , con un nivel de confianza del 95%, siendo la integración del modelo de aseguramiento de calidad en el

proceso de desarrollo de software una solución para el problema de investigación.

b) Análisis Descriptivos

En el gráfico 4.30 se observa el porcentaje promedio de clientes obtenidas de las pruebas antes (pre-test) y después (post-test) de la incorporación el modelo de aseguramiento de la calidad en el proceso de desarrollo de software, encontrándose un incremento de 22.423% de clientes satisfechos, pasando del 51.296% en el pre-test al 73.719 en el post-test. Por tanto se puede concluir que se incrementó el porcentaje de clientes satisfechos.

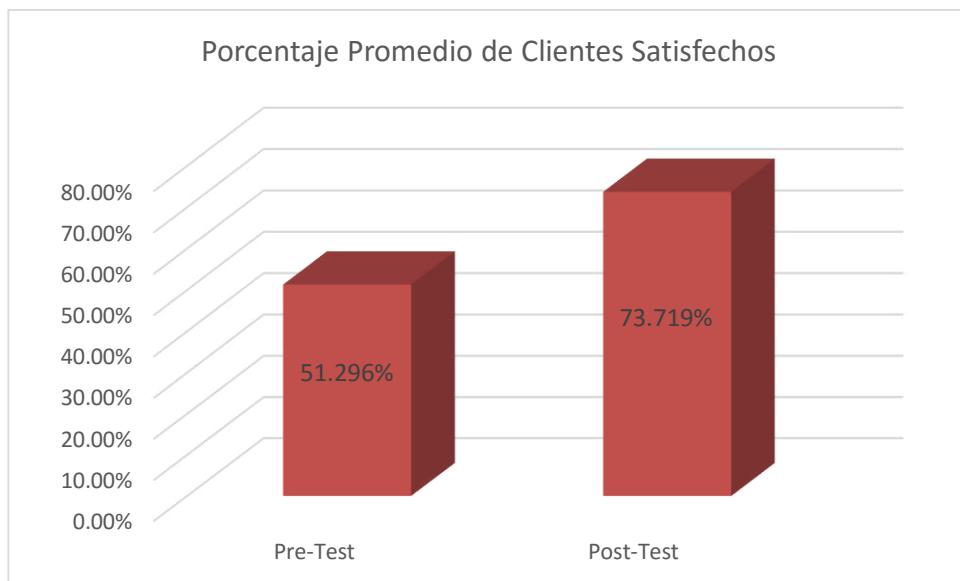


Gráfico 4.30. Porcentaje promedio de clientes satisfechos

Fuente. Elaboración propia

CAPITULO V: IMPACTOS

5.1. Modelo Propuesto

5.1.1. Fundamentos del Aporte

De acuerdo a la revisión de la literatura, los modelos CMMI, TSP y PSP cuentan con diversos procesos que contribuyen al aseguramiento de la calidad en el proceso de desarrollo de software, básicamente por lo siguiente:

Modelo de madurez de capacidades CMMI:

- Inclusión de las prácticas de institucionalización, que permiten asegurar que los procesos asociados con cada área de proceso sean efectivos, repetibles y duraderos.
- Guía paso a paso para la mejora, a través de niveles de madurez y capacidad.
- Transición del aprendizaje individual al aprendizaje de la organización por mejora continua, lecciones aprendidas y uso de bibliotecas y bases de datos de proyectos mejorados (SEI, 2012).

Proceso de Software para Equipos:

- Combinación de PSP (Personal Software Process) con el manejo de trabajo en equipo.

- Extiende y refina los métodos CMMI y PSP, para guiar a los miembros de los equipos en el trabajo de desarrollo.
- Muestra cómo construir un equipo auto dirigido y cómo ser un efectivo miembro del equipo.
- Muestra a los ingenieros de software cómo producir productos de calidad por medio de una planificación de costes.
- Proporciona equipos de proyectos con guías explícitas sobre cómo alcanzar sus objetivos.
- Construir equipos autosuficientes que planifiquen y documenten su trabajo, estableciendo metas además de sus progresos y planificaciones.
- Ayudar a los líderes de proyecto a dirigir y motivar a los grupos.
- Proporcionar una guía para que las empresas alcancen el más alto nivel de madurez. (SIE, 2012).

Proceso de Software para Personas:

- Está basado en prácticas encontradas en el modelo CMMI para el mejoramiento de procesos.
- Orientada a manejar la mejora continua de las habilidades de los ingenieros de software.
- Guía la planeación y el desarrollo de productos de software a lo largo de todo el proceso incluyendo análisis, definición de requerimientos, construcción del producto, documentación, pruebas y mantenimiento.

- Los ingenieros de software producen software usando un enfoque estructurado y disciplinado.
- Los ingenieros de software administran la calidad de los productos y aplican una retroalimentación (feedback) cuantitativa para mejorar sus procesos personales de trabajo, obteniendo así: mejores estimaciones, mejora planificación y seguimiento, compromiso personal hacia la calidad e involucrarse en un proceso de mejoramiento continuo (SIE, 2013)

5.1.2. Descripción del Aporte

El modelo propuesto para lograr el aseguramiento de la calidad en el proceso de desarrollo de software se basa en los modelos de madurez de capacidades integrado CMMi, modelo de proceso para equipos de software TSP y modelo de proceso para ingenieros de software PSP.

El presente modelo incorpora al factor humano de manera grupal e individual como los actores principales para introducir y administrar la calidad en el proceso de desarrollo de software.

El modelo integra procesos de TSP y PSP al modelo de mejora de procesos CMMi, permitiendo de este modo al ingeniero y equipo de ingenieros ser parte fundamental del proceso de desarrollo de software.

La imagen que se presenta a continuación describe la integración en un alto nivel de los modelos antes mencionados orientados a conseguir un objetivo común, lograr el aseguramiento de la calidad en el proceso de desarrollo de software.

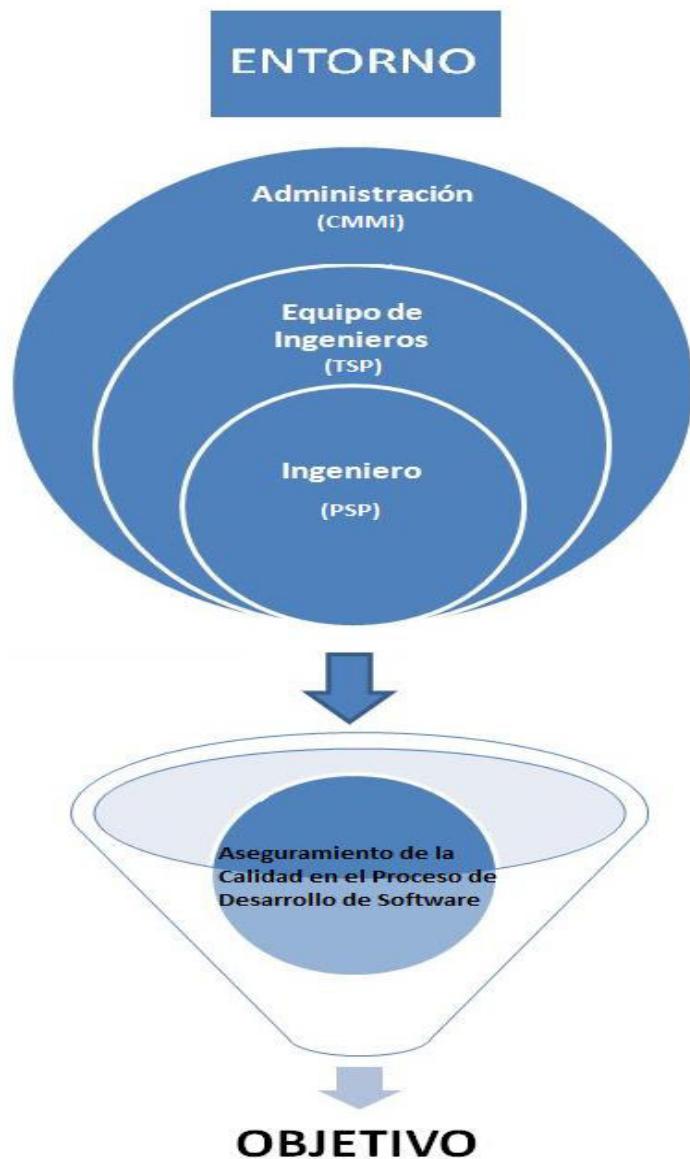


Figura 5.1 **Modelo de aseguramiento de la calidad alto nivel**

Fuente. Elaboración propia

La figura 5.2 muestra la representación gráfica del modelo de aseguramiento de la calidad, en el cual apreciamos que en los niveles de madurez 2 y 3 se están adicionando procesos de los modelos TSP y PSP con la finalidad de integrar al recurso humano como parte del fundamental en el aseguramiento de la calidad en el proceso de desarrollo de software.

En el punto 5.1.3 del presente capítulo se detallan cada uno de los procesos TSP y PSP incorporados al modelo de mejora de procesos CMMi, con sus respectivas especificaciones, los cuales forman parte del modelo de aseguramiento de la calidad.

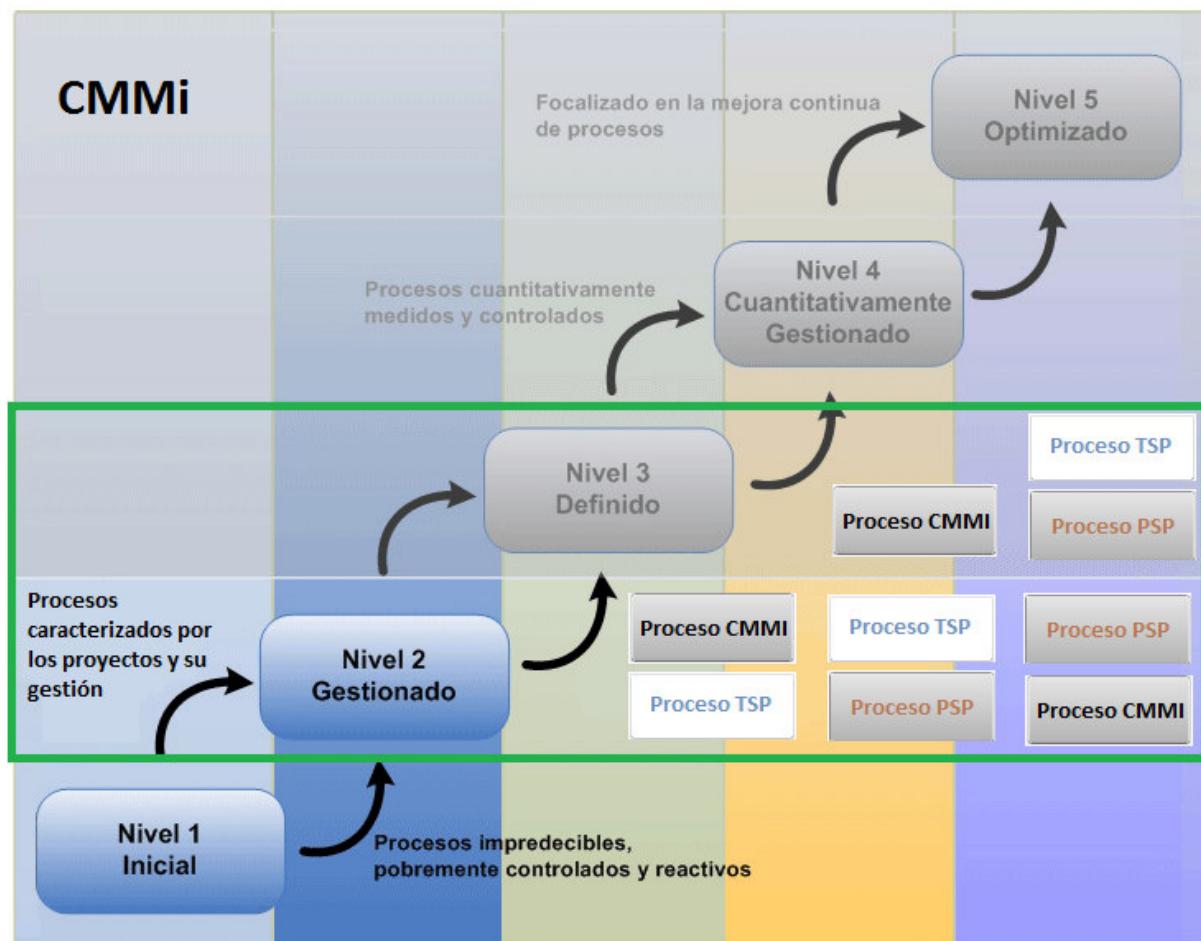


Figura 5.2. Modelo de aseguramiento de la calidad del proceso de desarrollo de software
Fuente. Elaboración propia

5.1.3. Descripción Detallada del Aporte

En la presente sección se detallan los procesos, prácticas y principios de los modelos PSP, TSP y CMMi que integran el modelo de aseguramiento de calidad en el proceso de desarrollo de software.

5.1.3.1. Introducción

El objetivo principal del modelo de aseguramiento de la calidad es asegurar la calidad en el proceso de desarrollo de software en consultoras o empresas que cuenten con procesos gestionados y definidos (CMMi nivel 2,3).

El modelo proporcionará una referencia para la aplicación y puesta en marcha de planes específicos de aseguramiento de calidad aplicables a proyectos concretos.

En el presente modelo de aseguramiento de la calidad se reflejan las actividades de calidad a realizar, los estándares a aplicar, los productos a revisar, los procedimientos a seguir, y realizar el seguimiento de los mismos.

5.1.3.2. Encuadramiento del modelo propuesto

El modelo propuesto tiene como objetivo principal asegurar la calidad en el proceso de desarrollo de software a través de la integración adecuada de procesos, tecnología y personas.

En el contexto del modelo los ingenieros de software son los principales factores y actores en el aseguramiento de la calidad, ya sea de manera individual o como parte del equipo de trabajo.

5.1.3.3. Modelo propuesto

Los componentes que articulan el presente modelo cubren tres áreas fundamentales, las cuales pueden conceptualizarse como los componentes del modelo. Estos son:

- Componente 1. Procesos, Prácticas y Herramientas PSP
- Componente 2. Procesos y Prácticas TSP
- Componente 3. Procesos CMMi

5.1.3.4. Componente 1. Procesos, Prácticas y Herramientas PSP

El Proceso Software Personal (PSP) fue diseñado para ayudar a los ingenieros del software a hacer bien su trabajo, por tanto nos permite aplicar métodos avanzados de ingeniería a nuestras tareas diarias. Proporciona métodos detallados de planificación y estimación, muestra a los ingenieros

cómo controlar su rendimiento frente a estos planes y explica cómo los procesos definidos guían su trabajo.

Los procesos del modelo PSP que integran el componente 1 del modelo de aseguramiento son los siguientes:

- **Proceso de medición Personal:** Punto de partida de PSP, es un proceso sencillo, definido y personal. Las actividades principales son:
 - Recoger datos sobre el trabajo del Ingeniero de Software (Tiempo gastado por fase, cantidad de defectos en compilación y pruebas, etc.)
 - Proporcionar un informe resumen.
 - Establecer una base que incluya mediciones y un formato de reportes para medir el progreso y definir los cimientos para mejorar.
 - Establecer estándares de código y mediciones de tamaño.
- **Proceso de Planeación Personal:** Complementa al proceso anterior, adicionando actividades de planeamiento y seguimiento, contempla lo siguiente:
 - Estimaciones de tamaño, recursos y reporte de prueba.
 - Cronograma y seguimiento del proyecto por parte de los Ingenieros.
 - Aprender a realizar compromisos que se puedan cumplir.
 - Establecer una base para realizar el seguimiento de su trabajo.
- **Administración de Calidad Personal:** El objetivo principal de este proceso es ayudar a los Ingenieros de Software a lidiar de manera realista y objetiva con los defectos que introducen, comprende lo siguiente:
 - Mejorar la habilidad del Ingeniero para producir programas de calidad.
 - Realizar trabajar de calidad más natural y consistente.
 - Explotar habilidades inherentes y capacidades.
 - Revisiones de código (Checklist).
 - Revisiones de diseño (Checklist).

La Figura 5.3 nos detalla las actividades principales correspondientes a cada proceso del modelo PSP.

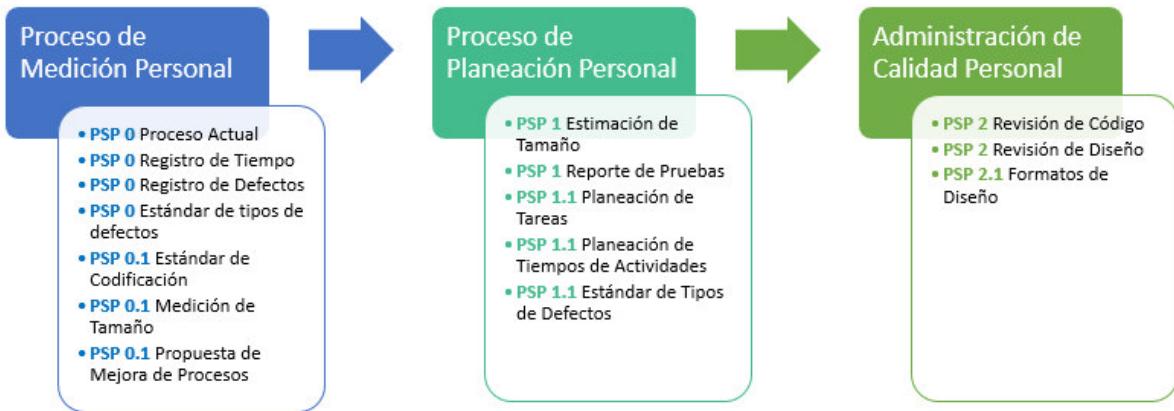


Figura 5.3 Procesos PSP que integran el modelo de Aseguramiento de la Calidad en el Proceso de Desarrollo de Software

Fuente. Elaboración propia

Los formatos sobre los cuales trabaja el siguiente componente son los siguientes:

- **Formato de estimación semanal de actividades:** Permite registrar la cantidad de horas estimadas a cada actividad durante la semana, las horas se deben registrar en función a la fase o ciclo de vida del producto. El formato de registro de estimación de actividades se encuentra en el anexo K.
- **Formato de registro de tiempos:** Permite registrar el tiempo destinado a cada actividad durante el día. La información principal a registrar son la fecha, hora inicio, hora fin y tiempo de interrupción. El cuaderno de registro de tiempo se encuentra en el anexo K.
- **Formato de registro de defectos:** Permite registrar los defectos asociados a la fase o ciclo de vida del producto. La principal información que se registra son Identificar de defecto, fecha, tipo, defecto introducido, defecto eliminado y tiempo de corrección. En el anexo K se encuentra el cuaderno de registro de defectos.

- **Formato resumen plan de proyecto:** Permite registrar información relevante de forma resumida del plan de proyecto. El formato consta de 5 partes (resumen, tamaño del proyecto, tiempo por fase, defectos introducidos y defectos eliminados). En el anexo K se encuentra en formato de resumen de plan de proyecto.

5.1.3.5. Componente 2. Procesos y Prácticas TSP

Los procesos del modelo TSP que integran el componente 2 del modelo de aseguramiento de calidad son los siguientes:

- **Proceso de Software para Personas (PSP):** Procesos que manejan la mejora continua de las habilidades de los ingenieros de software (Componente 1).
- **Proceso para Construir Equipos:** Comprende identificar condiciones mínimas de un equipo, construcción de equipos eficaces, cohesionados con entornos compartidos y comunes.
 - Define metas y asigna roles (gestor de calidad, líder de equipo, etc.).
 - Estima el tamaño del producto que va a desarrollar en función a la metodología y la complejidad.
 - Define la estrategia, los procesos y los planes detallados para realizar el trabajo.
 - Negocia los compromisos finales (Usuarios / Clientes Finales).
- **Proceso para Trabajar en Equipo:** Destinado a ayudar a cada miembro a adquirir su pleno potencial, aceptan metas y normas establecidas, ayuda mutua, creatividad, motivación, flexibles y adaptables.
 - Sigue los procesos y planes que definió.
 - Recolecta datos precisos y exactos de su trabajo.
 - Analiza los datos para identificar problemas con anticipación.
 - Encuentra soluciones en equipo y las ejecuta.
 - Comunicación del equipo.
 - Coordinación del equipo.

- Seguimiento del equipo.
- Análisis de riesgos.

La Figura 5.4 nos detalla las actividades principales correspondientes a cada proceso del modelo TSP.

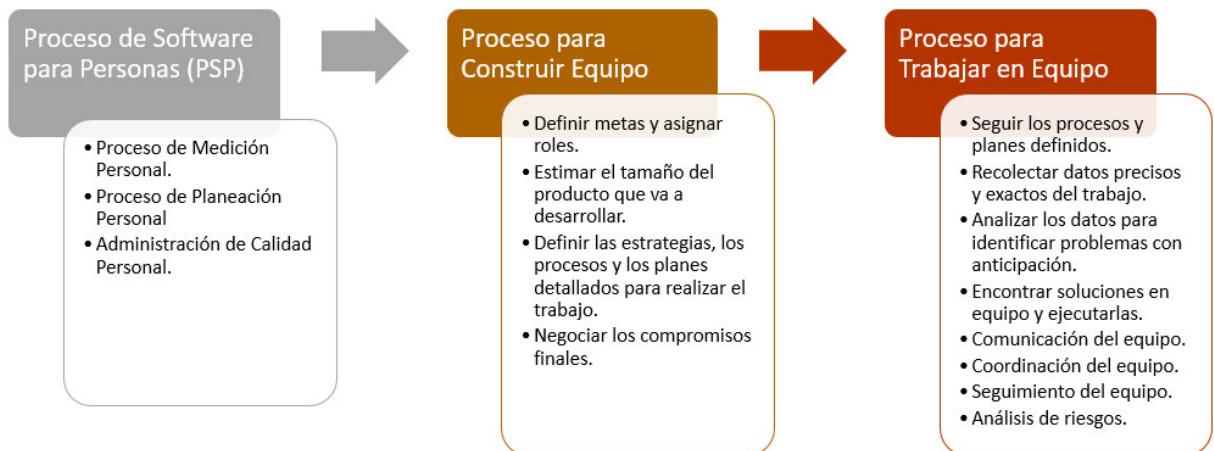


Figura 5.4 Procesos TSP que integran el modelo de Aseguramiento de la Calidad en el Proceso de Desarrollo de Software

Fuente. Elaboración propia

Los formatos sobre los cuales se soporta el componente 2 son los siguientes:

- **Formato de registro de intereses y experiencias:** Permite obtener información de los integrantes del equipo como sus intereses y experiencias, lo cual permite conformar un equipo de alto rendimiento. La información que se registra hace referencia a su experiencia e intereses relevantes, proyectos en otros equipos, liderazgo, preferencias de equipo y roles. Las especificaciones del formato se encuentra en el anexo L.
- **Formato de registro de decisiones estratégicas:** Permite registrar las decisiones estratégicas durante las fases del proyecto, asignar funcionalidades del producto, etc. La información que se registra va referida a funcionalidad, LOC en ciclos, referencias, etc. La especificación del formato se encuentra en el anexo L.

- **Formato de reporte semanal:** Permite preparar el reporte semanal, con información resumida de las ocurrencias de la semana como datos semanales por rol, tareas terminadas, seguimiento de problemas o riesgos. La especificación del modelo se encuentra en el anexo L.
- **Formato de evaluación del equipo y roles:** Facilita la evaluación y/o desempeño del equipo y de los roles. La información a registrar hace referencia contribuciones de cada rol, colaboración de equipo, desempeño, etc. La especificación del formato se encuentra en el anexo L.

5.1.3.6. Componente 3. Procesos CMMi

Los procesos CMMi que se integraran el componente 3 del modelo de aseguramiento son los siguientes:

- **Planificación de Proyecto (PP):** Desarrollar y mantener el plan de proyecto, integrar e implicar a los participantes y obtener compromiso con el plan.
- **Seguimiento y Control de Proyectos (PMC):** Monitoriza las actividades y toma decisiones correctivas incluyendo la re-planificación.
- **Desarrollo de Equipos Integrado (IT):** Identifica y organiza a los participantes en equipos colaborativos y desarrolla la visión compartida alineada con la visión compartida del proyecto y la organización.
- **Gestión de Riesgos (RSKM):** Desarrolla e implementa una estrategia proactiva para identificar, evaluar, priorizar, actuar y realizar seguimiento a los riesgos.
- **Desarrollo de Requisitos (RD):** Recopila y armoniza las necesidades de los participantes y las traduce en requisitos del producto.
- **Gestión de Requisitos (RM):** Asegura que los requisitos acordados son comprendidos y gestionados.

- **Solución Técnica (TS):** Convierte requisitos en arquitectura del producto, diseño y desarrollo.
- **Integración del Producto (PI):** Combina los componentes del producto y asegura las interfaces.
- **Verificación (VER):** Asegura que el producto cumpla con las especificaciones.
- **Validación (VAL):** Asegura que el producto cumple con el uso propuesto cuando se sitúa en el entorno propuesto.
- **Medición y Análisis (MA):** Establece un programa de métricas para proveer resultados objetivos que puedan ser usados para tomar decisiones informadas y tomar acciones correctivas.
- **Gestión de la Configuración (CM):** Establece y mantiene la integridad de los productos de trabajo.
- **Aseguramiento de Calidad de Proceso y Producto (PPQA):** Proporciona prácticas para evaluar objetivamente procesos, productos y servicios.

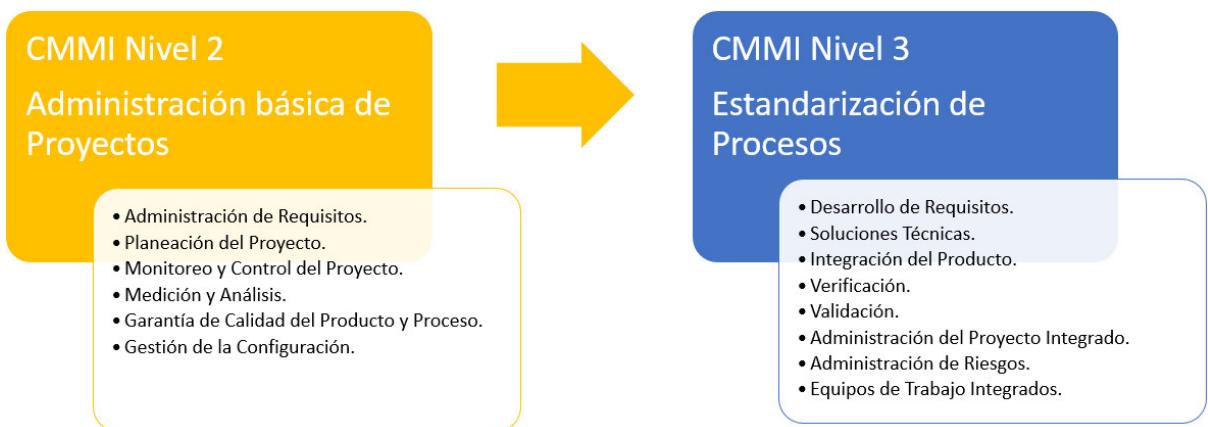


Figura 5.5 Procesos CMMI que integran el modelo de Aseguramiento de la Calidad en el Proceso de Desarrollo de Software

Fuente. Elaboración propia

Las especificaciones a detalle del modelo de aseguramiento de la calidad en el proceso de desarrollo de software se encuentran en el Anexo J.

Los checklist sobre los cuales se soporta el componente 3 son los siguientes:

- **Checklist de recepción de petición de estimación:** Permite revisar y dar conformidad a la documentación de entrada, procedimientos, dependencias, restricciones y otros contenidos. La especificación del checklist se encuentra en el anexo M.
- **Checklist de revisión de pares de análisis de impacto y estimación:** Permite revisar y dar conformidad a través de la revisión en pares sobre control de la estimación, análisis de impacto de las peticiones de estimación, entre otros. La especificación del checklist se encuentra en el anexo M.
- **Checklist de diseño funcional:** Permite revisar y dar conformidad sobre modelo lógico de procesos, modelos de alto nivel, modelos de detalle, modelo conceptual de datos, enfoque de prueba, diseño de arquitectura impactado, entre otros. La especificación del checklist se encuentra en el anexo M.
- **Checklist de diseño técnico:** Permite revisar y dar conformidad sobre el diseño orientado a objetos, diseño de procedimientos, definición de interfaces, transacciones, modelo de datos, pruebas de producto y otros criterios. La especificación del checklist se encuentra en el anexo M.
- **Checklist de construcción:** Permite revisar y dar conformidad respecto a estándares de programación, estándares de base de datos, integración del producto, entre otros. La especificación del checklist se encuentra en el anexo M.
- **Checklist de pruebas, sistemas e integración:** Permite revisar y dar conformidad respecto a casos de prueba, datos de prueba, calidad,

documentación de la prueba, resolución de errores, entre otros. La especificación se encuentra en el anexo M.

- **Checklist de entrega:** Permite validar y dar conformidad respecto a las actividades de cierre, documentación, manuales, entre otros entregables. La especificación del checklist se encuentra en el anexo M.

5.1.3.7. Consideraciones sobre la implementación de modelo

La empresa y/o consultara que desee integrar o adoptar el modelo debería tener en cuenta las siguientes consideraciones:

- Se cuenta con el apoyo de la alta dirección.
- El recurso humano entiende el significado de Aseguramiento de la Calidad y es consciente de la importancia de su implementación y cumplimiento.
- El recurso humano en general conoce la importancia de satisfacer tanto los requisitos del cliente, como los legales y reglamentarios.
- Se cuenta con presupuesto para poder implementar, actualizar y/o reemplazar los procesos o procedimientos que sean necesarios.
- Se cuenta con el apoyo de facilitadores y/o profesionales con experiencia en implementación de modelos de aseguramiento de calidad en procesos de desarrollo de software.

5.1.4. Supuestos

- El proceso de desarrollo donde se pretende integrar el modelo de aseguramiento de la calidad debe tener procesos gestionados y/o definidos para un acoplamiento eficiente.
- Los integrantes de los equipos deben tener conocimiento sobre los procesos, prácticas y herramientas de los modelos CMMi, TSP y PSP.

5.1.5. Limitaciones

El presente modelo solo tiene aplicación en empresas y/o consultoras que desarrollen software cuyos procesos de desarrollo de software cuenten con certificación CMMi nivel 2 o 3.

5.1.6. Alcance

El presente modelo será aplicado al proceso de desarrollo de software de empresas que desarrollan software in house y consultoras de la región Lima, cuyo proceso de desarrollo se encuentre certificado en CMMi nivel 2 o 3 con la finalidad de asegurar la calidad en el proceso de desarrollo.

5.2. Metodología de Implementación

5.2.1. Alcance

El alcance de la metodología de implementación se resume en los siguientes puntos:

- Puede ser utilizado en procesos de desarrollo de software en empresas o consultoras cuyo proceso de desarrollo se encuentre certificado en CMMi nivel 2 o 3.
- La metodología integra el modelo propuesto a los procesos de desarrollo basados en CMMi de las empresas y consultoras.

5.2.2. Descripción General

La presente metodología de implementación tiene una implicancia práctica, debido a que especifica cada uno de los pasos que se siguieron para integrar el modelo propuesto al proceso de desarrollo de software en la empresa indicado en la sección 4.1.3. Información del Proyecto de Aplicación:

La implementación tuvo dos aspectos importantes:

- Capacitación.
- Lanzamiento y seguimiento del modelo.

5.2.3. Descripción Específica

A continuación se detalla la metodología utilizada en la implementación del modelo de aseguramiento de calidad en el proceso de desarrollo de software.

5.2.3.1. Capacitación

Se brindó capacitación sobre los modelos CMMi, PSP, TSP, donde resalta los siguientes temarios:

a) Capacitación PSP para ingenieros

Se brindó capacitación sobre los fundamentos de PSP y PSP para ingenieros e integración con CMMi donde resalta los siguientes contenidos:

- Introducción a PSP
- Definición y uso de un proceso personal.
- Conceptos de planeamiento.
- Métricas de proceso y tamaño y calidad.
- Estimaciones de tamaño y esfuerzo.
- Gestión de defectos.
- Integración con modelo de mejora de procesos CMMi.

Se revisó y explico los formatos a utilizar correspondientes a PSP, cuyas especificaciones se encuentran en el anexo K:

- Estimación semanal de actividades.
- Cuaderno de registro de tiempos.
- Cuaderno de registro de defectos.

- Resumen de plan de proyecto.

b) Capacitación TSP para ingenieros

Se brindó capacitación sobre los fundamentos de PSP y PSP para ingenieros e integración con CMMi donde resalta los siguientes contenidos:

- Fundamentos TSP
- Análisis y mejora.
- Métricas de planeamiento y calidad.
- Formación de equipos.
- Integración con PSP y CMMi.

Se revisó y explico los formatos a utilizar correspondientes a TSP, cuyas especificaciones se encuentran en el anexo L:

- Información inicial de intereses y experiencias.
- Decisiones estratégicas.
- Reporte semanal.
- Evaluación de equipos y roles.

c) Modelo de Madurez de Capacidades CMMi

Se brindó capacitación sobre los procesos de CMMi orientados al aseguramiento de la calidad:

- Planificación de Proyecto (PP).
- Seguimiento y Control de Proyectos (PMC).
- Desarrollo de Equipos Integrado (IT).
- Gestión de Riesgos (RSKM).
- Desarrollo de Requisitos (RD).
- Gestión de Requisitos (RM).
- Solución Técnica (TS).
- Integración del Producto (SI).
- Verificación (VER).
- Validación (VAL).

- Medición y Análisis (MA).
- Gestión de la Configuración (CM).
- Aseguramiento de Calidad de Proceso y Producto (PPQA).

Se revisó y explico los checklist a utilizar correspondientes a CMMi para la validación y verificación en el proceso de desarrollo. Las especificaciones de los checklist se encuentran en el anexo M:

- Checklist de recepción de petición de estimación.
- Checklist de revisión de pares de análisis de impacto y estimación
- Checklist de diseño funcional.
- Checklist de diseño técnico.
- Checklist de construcción.
- Checklist de pruebas, sistemas e integración.
- Checklist de entrega.

5.2.3.2. Lanzamiento y Seguimiento del Modelo

Luego de la capacitación a los integrantes del equipo de proyecto, se prosiguió con el lanzamiento y seguimiento del modelo, el cual contenía las siguientes actividades:

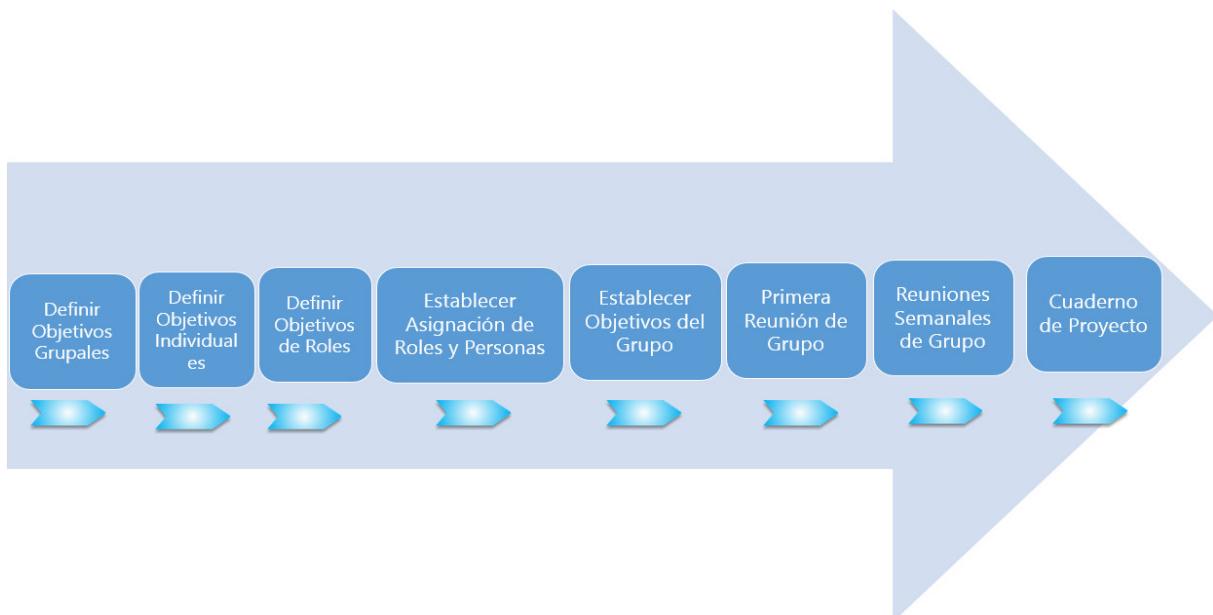


Figura 5.6 Lanzamiento y seguimiento del modelo

Fuente. Elaboración propia

a) Definir objetivos grupales

Etapa en la cual se establecieron métricas a nivel de equipo de trabajo, lo cual incluye:

- Producir un producto de calidad.
- Ejecutar un proyecto productivo y bien gestionado.
- Finalizar el proyecto dentro de las líneas base (tiempo, costo, alcance, calidad y recursos).

b) Definir objetivos individuales

Etapa en la cual se definió lo siguiente:

- Ser un miembro efectivo del equipo.
- Producir productos de calidad.

c) Definir objetivos de los roles

Etapa en la cual se definieron los objetivos de los diferentes roles que comprende el modelo TSP:

- **Líder de equipo:**
 - Responsabilidad:
 - ✓ Dirección y aseguramiento de las actividades realizadas por el resto del equipo.
 - Expectativas:
 - ✓ Motivar al equipo.
 - ✓ Controlar el seguimiento de actividades.
 - ✓ Verificar la entrega puntual y completa de tareas.
 - ✓ Identificar tareas semanales.
 - ✓ Desarrolla y Dirige al equipo.
- **Administrador de desarrollo:**

- Responsabilidad:
 - ✓ Dirección en requerimientos, diseño, construcción y pruebas.
- Expectativas:
 - ✓ Proponer la estrategia de desarrollo.
 - ✓ Auxiliar en las estimaciones de tamaño y tiempo.
 - ✓ Dirigir el análisis de alto nivel, diseño, construcción y pruebas.

- **Administrador de planeación:**

- Responsabilidad:
 - ✓ Apoyar y guiar la planeación y seguimiento.
- Expectativas:
 - ✓ Dirigir la generación del plan para cada ciclo o iteración.
 - ✓ Define productos y tamaños.
 - ✓ Registra tiempos individuales y del equipo.
 - ✓ Balancea cargas de trabajo para cada miembro.
 - ✓ Analiza y reporta estadísticas semanales de desarrollo del equipo.

- **Administrador de calidad:**

- Responsabilidad:
 - ✓ Determina necesidades en el proceso de calidad, mantiene y sigue la calidad del producto.
- Expectativas:
 - ✓ Dirigir y generar el plan de calidad
 - ✓ Alertar al equipo en problemas relativos a la calidad.
 - ✓ Mantener estándares de desarrollo.
 - ✓ Dirigir las inspecciones
 - ✓ Registrar las reuniones

- **Administrador de configuración:**

- Responsabilidad:

- ✓ Determinar, obtener y mantener las herramientas para realizar tareas administrativas.
- Expectativas:
 - ✓ Administrar el control de cambios
 - ✓ Administrar el sistema de control de cambios
 - ✓ Recomendar que cambios hacer o implementar.
 - ✓ Verificar y evaluar la pertinencia de los cambios.
 - ✓ Mantener versiones del sistema
 - ✓ Fomenta la reutilización, etc.

d) Establecer la asignación de roles y personas

En esta etapa se realizó la asignación de los diversos roles a los miembros del equipo de proyecto.

e) Establecimiento de los objetivos del grupo

En esta etapa se estableció los objetivos al equipo de proyecto.

f) Primera reunión de grupo

Se definieron las plantillas a utilizar por los integrantes del equipo de proyecto.

g) Reuniones semanales de grupo

Reuniones en las cuales se discuten riesgos, problemas, avances, pronósticos en base a información registrada en las plantillas durante la semana de trabajo.

h) Cuaderno de proyecto

Elemento físico que contiene el conjunto de elementos que describen y conforman el proyecto.

CONCLUSIONES

Las conclusiones que se han logrado obtener luego de finalizar todas las etapas de la investigación son las siguientes:

- Se concluyó que el promedio de costo de calidad antes de la integración del modelo de aseguramiento de la calidad es de **12.45%** del costo real del proyecto, mientras que el promedio de costo de calidad después de la integración del modelo de aseguramiento de la calidad es de **5.25%** del costo real del proyecto; obteniendo una reducción de **7.19%** en promedio en el costo de calidad en los proyectos de desarrollo de software.
- Se concluyó que el promedio de defectos por proyecto antes de la integración del modelo de aseguramiento de la calidad es de **118.27** defectos, mientras que el promedio de defectos por proyecto después de la integración del modelo de aseguramiento de la calidad es de **49.82** defectos por proyecto; obteniendo una reducción de **68.45** defectos en promedio en los proyectos de software.
- Se concluyó que el promedio de desviación de costo por proyecto antes de la integración del modelo de aseguramiento de la calidad es de **18.29%** respecto a su línea base, mientras que el promedio de desviación de costo por proyecto después de la integración del modelo de aseguramiento de la calidad es de **6.40%** respecto de la línea base; obteniendo una reducción significativa de **11.89%** en la desviación de costo del proyecto en promedio. Como resultado de esta reducción significativa en la desviación de costo de proyecto se incrementa la rentabilidad del proyecto.
- Se concluyó que el promedio de clientes satisfechos antes de la integración del modelo de aseguramiento de la calidad es de **51.30%**, mientras que el

promedio de clientes satisfechos después de la integración del modelo de aseguramiento de la calidad es de **73.72%**; obteniendo un incremento significativo de **22.42%** en promedio de clientes satisfechos.

- Finalmente se concluye que el modelo de aseguramiento de calidad tuvo un impacto positivo en el proceso de desarrollo de software porque contribuye en el aseguramiento de la calidad en el proceso de desarrollo, lo cual se ve reflejado en la reducción del costo de calidad, reducción en la desviación de costo, reducción de defectos e incremento de satisfacción de los clientes en cada uno de los proyectos que integraron el modelo en su proceso de desarrollo.

RECOMENDACIONES

- Luego de adoptar el modelo de aseguramiento de calidad en el proceso de desarrollo de software, las empresas y consultoras deberá a través de controles y auditorias, efectuar un proceso de mejoramiento continuo que le permita mantener y mejorar sus niveles de calidad.
- Considerar aspectos adicionales para asegurar la calidad en el proceso de desarrollo de software, aspectos como el clima laboral, línea de carrera, la motivación de los miembros del equipo, liderazgo y desarrollo de habilidades interpersonales.
- Recomendar a los ingenieros de software obtener la certificación PSP Developer como parte de su crecimiento profesional y para mejorar la calidad en su trabajo diario.
- Es necesario que el uso de modelos o estándares sea política de la empresa, de esta manera se garantiza que los procedimientos y estándares se mantengan vigentes y evitar que caigan en desuso.

FUTURAS INVESTIGACIONES

El presente estudio de investigación identificó cuatro indicadores (costo de calidad, cantidad de defectos, desviación de costo y satisfacción del cliente) en el proceso de desarrollo de software y su ámbito de aplicación en empresas que cuenten con procesos de desarrollo gestionados y/o definidos, por todo lo anterior se plantean los siguientes estudios futuros que complementarían la presente investigación:

- El presente trabajo de investigación se constituya como una base para futuras investigaciones, donde se debería considerar dimensiones e indicadores adicionales, como estimaciones de tamaño, estimación de esfuerzo, medición de tamaño, productividad, análisis cuantitativo de riesgos que permitan determinar la calidad en el proceso de desarrollo de software.
- Ampliar el ámbito de aplicación del modelo de aseguramiento de calidad, para que se pueda integrar en empresas y/o consultoras de software que aún no cuenten con procesos maduros.

REFERENCIAS BIBLIOGRÁFICAS

- Bayona, S., Calvo Manzano, J., Cuevas, G. & San Feliu, T. (2007). *Team Software Process (TSP): Mejoras en la estimación, calidad y productividad de los equipos en la gestión de software*, pp. 85-120.
- Blanco Llano, Francisco Javier Rodríguez Hernández, Aida (2010). *Diseño de procesos claves para el mejoramiento de la calidad en proyectos de software*, pp. 50-60.
- Carlos Mauricio Echeverría Goyes & Cynthia Denisse Echeverría Goyes (2006). *Implementación de un Sistema integrado de Control de Costos de Producción, Órdenes de Trabajo, Presupuesto de Obras, Bodega y Control de Inventario utilizando PSP (Personal Software Process) y TSP (Team Software Process)*. Recuperado de <http://www.dspace.espol.edu.ec/handle/123456789/5006>
- CMMI Institute (2012) *Guía para la Integración de Procesos y la Mejora de Productos*. Recuperado de <http://cmmiinstitute.com/resources/spanish-language-translation-cmmi-development-version-13>
- CMMI Institute Published Appraisal Results. Recuperado de <https://sas.cmmiinstitute.com/pars/pars.aspx>
- Coral Calero Muñoz, Mario G. Piattini Velthuis & María Ángeles Moraga de la Rubia (2010) *Calidad del producto y proceso software*, pp. 80-102.
- Daniel Edgardo Riesco (2012) *Automatización del Proceso de Desarrollo de Software: Automatización en RSL de la Administración del Proceso de Desarrollo de Software en SPEM usando Flujo de Trabajo*, pp. 305-321.
- Fernanda Scalone (2006). *Tesis: Estudio Comparativo de los Modelos y Estándares de Calidad del Software*. Recuperado de <http://laboratorios.fi.uba.ar/lsi/scalone-thesis-maestria-ingenieria-en-calidad.pdf>

Giancarlo Juan Nakashima Chávez (2009). *Mejora del proceso de software de una Empresa desarrolladora de software: Caso COMPETISOFT - PERÚ DELTA.* Recuperado de <http://tesis.pucp.edu.pe/repositorio/handle/123456789/355>

Gonzalo Cuevas Agustín, Antonio De Amescua Seco, Tomás San Feliu Gilabert, José A. Calvo-Manzano Villalón, Magdalena Arcilla Cobián, Miguel García Cordero & José Antonio Cerrada Somolinos (2002). *Gestión Del Proceso Software*, pp. 205-215.

Hernán Berinsky, Francisco Facioni, Pablo Sobrino & Verónica Sznek (2008). *The Personal Software Process. Watts Humphrey. Technical Report. CMU/SEI-2000-TR-022.* Recuperado de http://www-2.dc.uba.ar/materias/isoft2/2008_01/clases/PSP.pdf

Ian Sommerville (2010), Software Engineering, (9th ed).

ISO/IEC 15504-1 (2004). *Information technology - Process assessment - Part 1: Concepts and vocabulary*

Javier Saldaña Ramos (2010), *Un marco metodológico para la mejora en la gestión de los equipos de desarrollo de software global.* Recuperado de http://e-archivo.uc3m.es/bitstream/handle/10016/10286/Tesis_Javier_Saldana.pdf

Jesús Araque & Keyla Soteldo (2015). *ISO 9001:2015 y el Pensamiento Basado en Riesgo.*

Lerma González, Héctor Daniel (2009). *Metodología de la investigación. Propuesta, anteproyecto y proyecto. (4th ed.)* Bogotá: ECOE Ediciones.

M.Teresa Villalba de Benito, Luis Fernández (2012) *Modelos de Calidad del Software: Un modelo práctico para medir la calidad de los productos de Seguridad Informática*, España: Editorial Académica Española, pp. 178-190.

Pantaleo, Guillermo (2011) *Calidad en el desarrollo de software*, pp. 25-32.

Project Management Institute (2014). *Guía de los Fundamentos Para la Dirección de Proyectos (Guía del PMBOK) (5th ed.)*

Rangel Bocardo, Marco A. (2011). Bug Manager: Introduciendo calidad a las organizaciones de software (tesis de maestría). Instituto Tecnológico y de estudios superiores de Monterrey, Guadalajara.

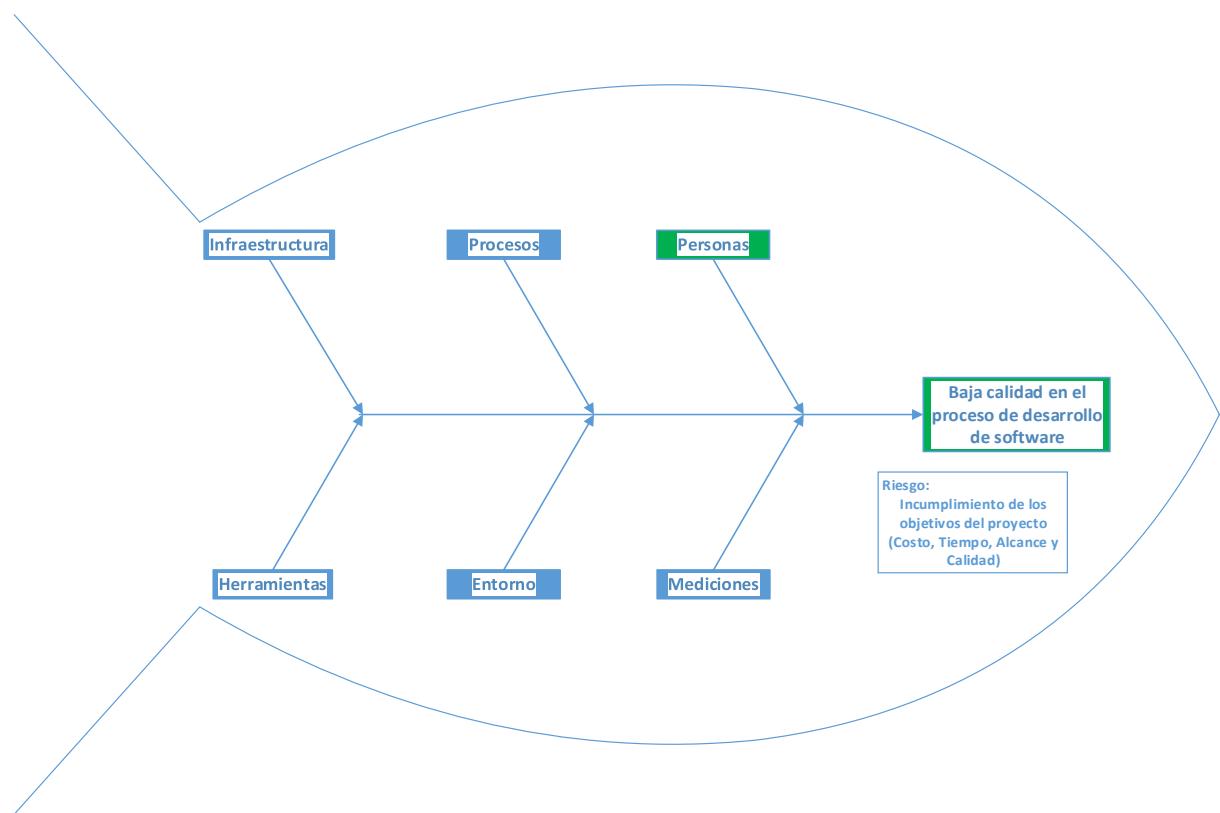
- Roger Pressman (2010). *Ingeniería del software – un enfoque práctico*. México: Mc Graw Hill.
- Shurei Tamura (2009) *Integrating CMMI and TSP/PSP: Using TSP Data to Create Process Performance Models*. Recuperado de <http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=9063>
- Software Engineering Institute (2010a). *The Personal Software Process (PSP) Body of Knowledge Version 2.0*. Recuperado de <http://www.sei.cmu.edu/reports/09sr018.pdf>
- Software Engineering Institute (2010b). *The Team Software Process (TSP) Body of Knowledge*. Recuperado de <http://www.sei.cmu.edu/reports/10tr020.pdf>
- Software Guru (2011). *Gestión de Procesos: Integrando TSP y CMMI*. Recuperado de <http://sg.com.mx/content/view/1070>
- The Standish Group Report (2015). Chaos. Recuperado de <https://www.infoq.com/articles/standish-chaos-2015>
- WATTS S. HUMPHREY (2006) *TSP(SM) Coaching Development Teams, Part of the SEI Series in Software Engineering series*.
- WATTS S. HUMPHREY (2007). *Introduction to the Personal Software Process*.
- Wilson Pantoja (2013). *Mejora de procesos de software – Universidad del Cauca-Colombia*. Recuperado de univirtual.unicauc.edu.co/moodle/mod/resource/view.php?id=30186
- Young-Suk Ahn Park (2012). *Adapting the Team Software ProcessSM (TSPSM) an Academic (Master of Software Engineering) Studio Project Experience Report*. Recuperado de http://www.academia.edu/1163365/Adapting_the_Team_Software_Process_SM_TSP_SM_
- Yu, Angus G. (2009). *Software Crisis, What Software Crisis*. Recuperado de https://www.researchgate.net/publication/224609375_Software_Crisis_What_Software_Crisis

Zhang Lina (2010) *Software Process Improvement for Small Organizations Based on CMMI/TSP/PSP*. Recuperado de
<http://www.seiofbluemountain.com/upload/product/201001/1264727480s3xvkd4g.pdf>

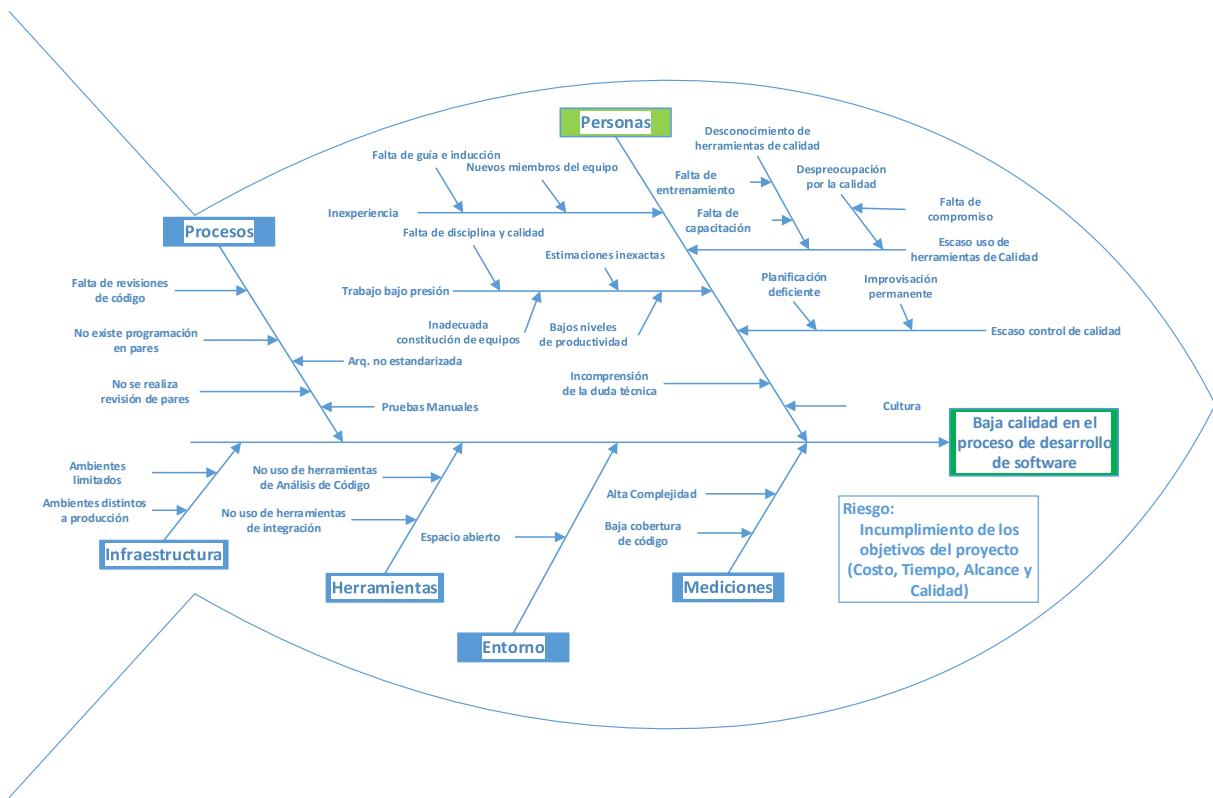
ANEXOS

ANEXO A: Diagrama de Ishikawa

El proceso de análisis de las principales causas al problema de investigación son representadas a través de un modelo de causa efecto, permitiéndonos identificar los problemas críticos y los riesgos asociados al proceso de desarrollo de software.



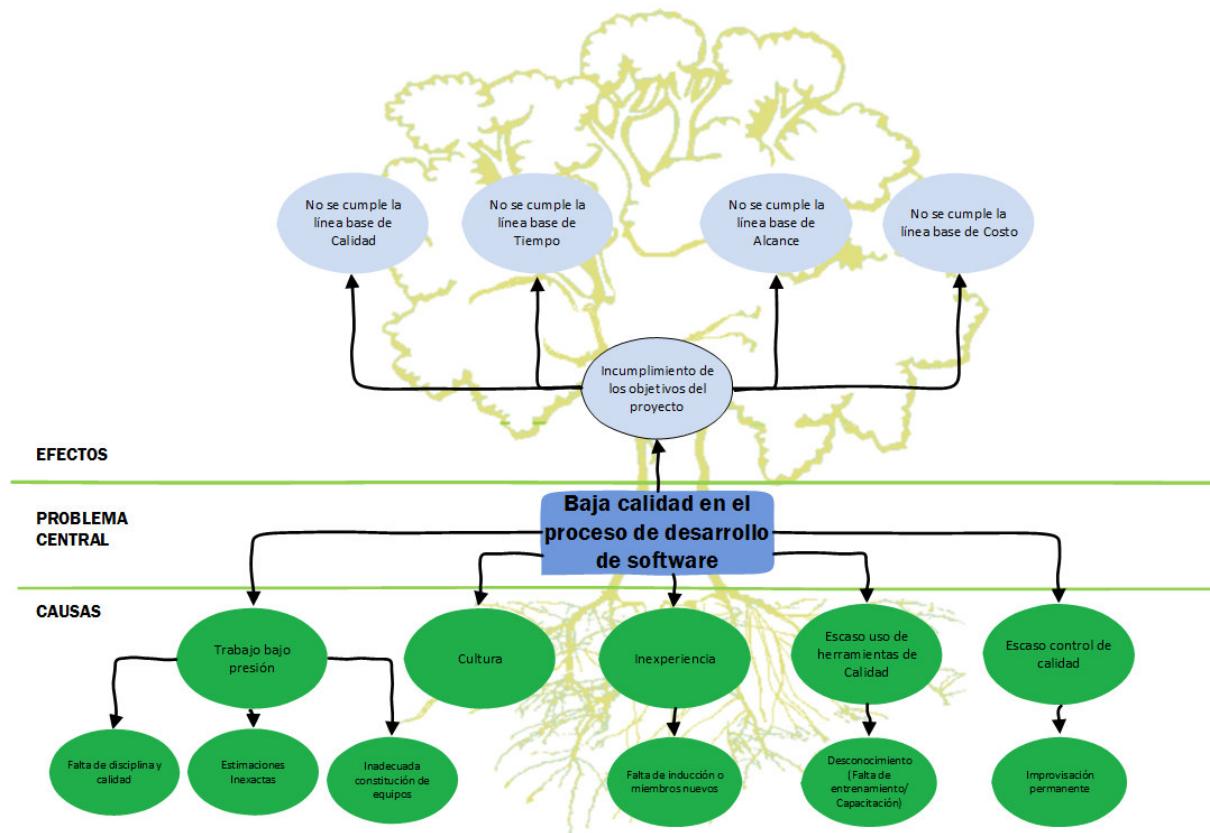
Fuente. Elaboración Propia



Fuente. Elaboración Propia

ANEXO B: Árbol de problemas

El análisis de las principales causas y efectos del problema de investigación se presentan en un diagrama de árbol de problemas, permitiéndonos identificar las principales causas y efectos asociados del problema de la baja calidad en el proceso de desarrollo de software en empresas que desarrollan software in house y consultoras certificadas en CMMI nivel 2 y 3.



Fuente. Elaboración Propia

ANEXO C: Población de estudio

La información que se presenta a continuación representa a nuestra población de estudio, obtenida desde el portal de CMMI Institute. La información obtenida es de las empresas que cuentan con certificación vigente en CMMI para desarrollo nivel 2 y 3.

Organization Organizational Unit	Team Leader Sponsor	Appraisal End Date	Model (Representation): Maturity Level
Banco Central de Reserva del Perú Gerencia de Tecnologías de Información - Subgerencia de Soluciones de Tecnologías de Información.	David Artega Gil FELIPE ERNESTO ROEL MONTELANOS	08/29/2012	CMMI-DEV v1.3(Staged):Maturity Level 2
SIVSA, Soluciones Informáticas, S.A. Software Development Projects and HOSIX Software Maintenance Service.	Ramiro Carballo Jose Antonio Román Jiménez	04/23/2014	CMMI-DEV v1.3(Continuous):Maturity Level 2 CMMI-SVC v1.3(Continuous):Maturity Level 2
UNIVERSIDAD PRIVADA ANTENOR ORREGO Software Development Projects at Office: Systems and Information's Engineering - (Oficina de Sistemas e Ingeniería de Información - Proyectos de Desarrollo de Software)	David Artega Gil MARCELINO WALDEMAR CARRETERO OBANDO	12/09/2013	CMMI-DEV v1.3(Staged):Maturity Level 2

Fuente. <https://sas.cmmiinstitute.com/pars/pars.aspx>

Organization Organizational Unit	Team Leader Sponsor	Appraisal End Date	Model (Representation): Maturity Level
Accenture LATAM: Argentina DC; Brazil DC; Colombia & Peru (Telefonica TGP)	John Voss Marcio Theme	07/25/2012	CMMI-DEV v1.3(Staged):Maturity Level 3
CosapiSoft S.A. CosapiSoft	David Artega Gil EMILIO FERNANDEZ DE CORDOVA	01/27/2012	CMMI-DEV v1.3(Staged):Maturity Level 3
Delaware Consultoría Perú SAC Delaware Peru - Software Services	David Artega Gil Alberto Centeno	06/21/2013	CMMI-DEV v1.3(Staged):Maturity Level 3
everis Peru everis Peru - Outsourcing	Carmen Cecilia Rodriguez Pascual Ruiz	02/08/2013	CMMI-DEV v1.3(Continuous):Maturity Level 3
GMD S.A. Application Outsourcing - Maintenance, Development and Testing Services	David Artega Gil ALDO MARTIN GALLI ALVAREZ Luis Mercado	04/19/2013	CMMI-DEV v1.3(Staged):Maturity Level 3
Hildebrando Hildebrando Mexico, Peru, Colombia	JuanJo Cukier Rafael Lopez Escalera Ruben Guerrero JORGE VELEZ	05/31/2013	CMMI-DEV v1.3(Staged):Maturity Level 3
Novatronic SAC Gerencia de Operaciones (Operations Management)	David Artega Gil Guillermo Pacheco	02/07/2013	CMMI-DEV v1.3(Staged):Maturity Level 3
SOFTWARE ENTERPRISE SERVICES SAC SES – Software Factory	David Artega Gil JUAN HUAPAYA	01/11/2013	CMMI-DEV v1.3(Staged):Maturity Level 3
Synopsis S.A. Gerencia de Operaciones	David Artega Gil Ricardo Palma	09/13/2013	CMMI-DEV v1.3(Staged):Maturity Level 3
TeamSoft S.A.C. Software Development Unit	David Artega Gil ALBERTO OLAECHEA	10/18/2012	CMMI-DEV v1.3(Staged):Maturity Level 3
Telefónica Gestión de Servicios Compartidos Perú S.A.C. Fabricas de Software Interna y Externa en Gerencia de Sistemas de Negocio y Servicio de Testing en Gerencia de TI	David Artega Gil Roger Bernedo	07/16/2013	CMMI-DEV v1.3(Staged):Maturity Level 3
Trans Solutions Systems S.A. Trans Solutions Systems	David Artega Gil Dante Rebagliati	05/17/2012	CMMI-DEV v1.3(Staged):Maturity Level 3

Fuente. <https://sas.cmmiinstitute.com/pars/pars.aspx>

ANEXO D: Ficha de validación de instrumento de investigación

FICHA DE VALIDEZ DE INSTRUMENTO DE INVESTIGACIÓN JUICIO DE EXPERTO

Modelo de aseguramiento de la calidad en el proceso de desarrollo de software basado en los modelos de madurez de capacidades (CMMi), proceso de software para equipos (TSP) y personas (PSP)"

Responsable: ALEX JOSÉ ESPEJO CHAVARRÍA

Instrucción: Luego de analizar y cotejar el instrumento de investigación “**CUESTIONARIO DE ENCUESTA**” con la matriz de consistencia de la presente, le solicitamos que en base a su criterio y experiencia profesional, valide dicho instrumento para su aplicación.

NOTA: Para cada criterio considere la escala de 1 a 5 donde:

1 – Muy Poco	2 – Poco	3 – Regular	4 – Aceptable	5 – Muy Aceptable
---------------------	-----------------	--------------------	----------------------	--------------------------

Criterio de Validez	Puntuación					Argumento	Observaciones y/o sugerencias
	1	2	3	4	5		
Validez de contenido							
Validez de criterio metodológico							
Validez de intención y objetividad de medición y observación							
Presentación y formalidad del Instrumento							
Total Parcial							
TOTAL							

Puntuación:

De 4 a 11: No válida, reformular

De 12 a 14: No válida, modificar

De 15 a 17: Validó, mejorar

De 18 a 20: Válido, aplicar

Apellidos y Nombres	
Grado Académico	
Institución	
Firma	

ANEXO E: Carta de presentación



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
 (Universidad del Perú, DECANA DE AMÉRICA)

FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA
UNIDAD DE POSGRADO

**Decenio de las Personas con Discapacidad en el Perú”
 “Año de la Consolidación del Mar de Grau”**

Lima, 04 de abril de 2016.

Sr.:.....

Presente.-

Por la presente, reciba usted el saludo cordial y fraternal a nombre de la Facultad de Ingeniería de Sistemas e Informática - Unidad de Posgrado de la Universidad Nacional Mayor de San Marcos; luego para manifestarle, que estoy desarrollando la tesis titulada **“Modelo de aseguramiento de la calidad en el proceso de desarrollo de software basado en los modelos de madurez de capacidades (CMMi), proceso de software para equipos (TSP) y personas (PSP)”**; por lo que conocedores de su excelente trayectoria profesional y estrecha vinculación con el ámbito de investigación, le solicito su colaboración en emitir su **JUICIO DE EXPERTO**, para la validación del instrumento **“Cuestionario de encuesta sobre modelo de aseguramiento de calidad en el proceso de desarrollo de software”** de la presente investigación.

Agradeciéndole por anticipado su gentil colaboración como experto, me despido de usted.

Atentamente

.....
 Alex José Espejo Chavarría
 Ingeniero de Sistemas

Adjunto:

1. Matriz de consistencia.
2. Operacionalización de variables.
3. Instrumento de investigación
4. Ficha de juicio de experto.

ANEXO F: Cuestionario de diagnóstico

La encuesta que se presenta a continuación permite obtener y analizar la problemática en el proceso de desarrollo de software de las Consultoras y Empresas que desarrollan software para uso interno. En la cual se pretende identificar qué factores influyen directamente en la calidad del proceso de desarrollo.

DATOS DE LA EMPRESA/CONSULTORA	
Razón Social	
Puesto	
Nombres y Apellidos	
Correo de Contacto	
Teléfono de Contacto	

PROCESO DE DESARROLLO	
OBJETIVO: Observar las características del proceso de desarrollo de software y analizar su diversa problemática referente a la calidad del proceso y producto.	
1. ¿Qué metodología de desarrollo de software aplican habitualmente?	
<input type="checkbox"/>	Modelo en Cascada
<input type="checkbox"/>	Modelo en V
<input type="checkbox"/>	Modelo en espiral
<input type="checkbox"/>	Proceso Unificado (Rational Unified Process)
<input type="checkbox"/>	Metodología ágil (SCRUM, eXtreme Programming, etc.)
<input type="checkbox"/>	Otro
Especifique	Haga clic aquí para escribir texto.
2. ¿Cuál es el esfuerzo promedio dedicado a cada fase? (Responda en porcentaje)	
<input type="checkbox"/>	Planificación
<input type="checkbox"/>	Análisis
<input type="checkbox"/>	Diseño
<input type="checkbox"/>	Codificación
<input type="checkbox"/>	Pruebas e Integración

<input type="checkbox"/>	Implantación
<input type="checkbox"/>	Otros
Especifique	Haga clic aquí para escribir texto.
3. ¿Qué porcentaje de esfuerzo post-construcción dedican a las siguientes actividades?	
<input type="checkbox"/>	Corrección de errores
<input type="checkbox"/>	Adaptaciones a cambios en el entorno (hardware, sistema operativo, frameworks)
<input type="checkbox"/>	Mejoras de eficiencia, mantenibilidad, usabilidad, etc.
<input type="checkbox"/>	Desarrollo de nuevas funcionalidades
<input type="checkbox"/>	Estandarización
<input type="checkbox"/>	Otra
Especifique	Haga clic aquí para escribir texto.
4. ¿Qué tipos de herramientas dan soporte a su proceso de desarrollo?	
<input type="checkbox"/>	Gestión de requerimientos
<input type="checkbox"/>	Forms, Plantillas, Estándares
<input type="checkbox"/>	Herramientas de gestión de proyectos
<input type="checkbox"/>	Herramientas de Prueba de software
<input type="checkbox"/>	Entornos de desarrollo integrado (control de versiones)
<input type="checkbox"/>	Otras
Especifique	Haga clic aquí para escribir texto.
5. ¿Qué factores principales afectan la calidad en el proceso de desarrollo?	
<input type="checkbox"/>	Persona
<input type="checkbox"/>	Equipo
<input type="checkbox"/>	Tecnología
<input type="checkbox"/>	Proceso
<input type="checkbox"/>	Metodología
<input type="checkbox"/>	Otras
Especifique	Haga clic aquí para escribir texto.

POLITICAS DE CALIDAD	
OBJETIVO: Determinar si existen políticas de Calidad (Estándares, Certificaciones, etc.) a nivel de Empresa/Consultora que garanticen la calidad del proceso y del producto.	
6. Su empresa	
<input type="checkbox"/>	¿Posee alguna certificación de calidad de software?
¿Cuál o cuáles?	Haga clic aquí para escribir texto.
<input type="checkbox"/>	¿Se encuentra en el proceso de certificación de calidad de software?
¿Cuál o cuáles?	Haga clic aquí para escribir texto.
<input type="checkbox"/>	¿Tiene planeado obtener una certificación de calidad?
¿Cuál o cuáles?	Haga clic aquí para escribir texto.
<input type="checkbox"/>	No tiene intenciones de obtener una certificación
<input type="checkbox"/>	Desconoce certificaciones aplicables a su actividad
7. ¿El modelo de proceso utilizado asegura la calidad en el proceso de desarrollo de software?	
<input type="checkbox"/>	Sí
¿Cómo?	Haga clic aquí para escribir texto.
<input type="checkbox"/>	No
¿Cómo?	Haga clic aquí para escribir texto.
8. ¿La metodología utilizada asegura la calidad del producto?	
<input type="checkbox"/>	Sí
¿Cómo?	Haga clic aquí para escribir texto.
<input type="checkbox"/>	No
¿Cómo?	Haga clic aquí para escribir texto.
9. ¿Qué factores cree usted que aseguran la calidad en el proceso de desarrollo de software?	
<input type="checkbox"/>	Persona
<input type="checkbox"/>	Equipo de Trabajo
<input type="checkbox"/>	Modelo de Proceso
<input type="checkbox"/>	Metodología
<input type="checkbox"/>	Tecnología
<input type="checkbox"/>	Otro

Especifique	Haga clic aquí para escribir texto.
TESTING	
OBJETIVO: Determinar la forma como validan y verifican la calidad del proceso de desarrollo y del producto.	
10. ¿Su empresa/consultora posee gestión de defectos?	
<input type="checkbox"/>	Sí, la gestión de defectos se realiza con documentos
<input type="checkbox"/>	Sí, la gestión de defectos se realiza con una herramienta especializada
<input type="checkbox"/>	No
11. ¿Su empresa tiene definidos procesos de testing?	
<input type="checkbox"/>	Sí, hay una estrategia global de testing
<input type="checkbox"/>	Sí, cada proyecto define su proceso de testing
<input type="checkbox"/>	No, realizamos testing ad-hoc
<input type="checkbox"/>	No, no realizamos testing
12. En el caso de definir un proceso de testing, en general, ¿qué porcentaje de esfuerzo le llevan las siguientes actividades?	
<input type="checkbox"/>	Planeamiento de las pruebas
<input type="checkbox"/>	Diseño de las pruebas
<input type="checkbox"/>	Ejecución de las pruebas
<input type="checkbox"/>	Evaluación de las pruebas
<input type="checkbox"/>	Seguimiento de las pruebas
<input type="checkbox"/>	Otros
Especifique	Haga clic aquí para escribir texto.
13. ¿Quién realiza las pruebas del software?	
<input type="checkbox"/>	El área específica de testing o calidad de software
<input type="checkbox"/>	El equipo de desarrollo de la empresa
<input type="checkbox"/>	El área que solicita el requerimiento
<input type="checkbox"/>	El área de sistemas de la empresa
<input type="checkbox"/>	El usuario final (interno o externo)
<input type="checkbox"/>	Una empresa diferente de la encargada del desarrollo

ANEXO G: Ficha de registro de datos

Ficha de observación: Indicadores Costo de Calidad y Rentabilidad del Proyecto (Pre Prueba)

Investigador	Espejo Chavarría, Alex
Institución donde investiga	
Dirección	Av. Nicolás de Piérola N° 589 INT. 6PS Lima - Lima - Lima
Indicadores observados	% Costo de calidad y % Desviación de Costo

ID Proyecto	Descripción	Fecha Cierre	Duración Planificada (Horas)	Duración Real (Horas)	% Costo de Calidad	% Desviación de Costo
SOL 9367	Modificación reporte RRGG – ELPA	Oct-2015	540	630	12.70%	16.67%
SOL 9477	VTAR_BCP - WILLIS Fase II	Nov-2015	477	524	8.38%	10.06%
SOL 9571	Clientes de FFW de Negocios Directos	Dic-2015	500	560	5.71%	12.00%
SOL 9216	Información EPS en los Estados Financieros	Dic-2015	871	975	9.44%	11.94%
SOL 9266	Renovación Seguro Contra Trabajo Riesgoso SCTR	Ene-2016	1063	1395	17.92%	31.23%
SOL 9465	Redondeo de Facturas Globales y de Comisiones 1.5	Ene-2016	525	595	10.92%	13.33%
SOL 9502	Vulnerabilidad CG Web	Feb-2016	902	1140	19.30%	26.39%
SOL 9546	Problemas en el Servidor de Impresión Emisión - Optimización	Feb-2016	511	680	19.85%	33.08%
SOL 9554	Cuenta Corriente de Coaseguro	Feb-2016	484	498	2.41%	2.89%
SOL 9570	Automatización Tramas MLPE-PH01-PRE1-VDES	Mar-2016	461	525	11.43%	13.88%
SOL 9636	Nuevas Cuentas Resolución 7037SB	Mar-2016	879	1140	18.86%	29.69%

**Ficha de observación: Indicadores Costo de Calidad y Rentabilidad del Proyecto
(Post Prueba)**

Investigador	Espejo Chavarría, Alex
Institución donde investiga	 High performance. Delivered.
Dirección	Av. Nicolás de Piérola N° 589 INT. 6PS Lima - Lima - Lima
Indicadores observados	% Costo de calidad y % Desviación de Costo

ID Proyecto	Descripción	Fecha Cierre	Duración Planificada (Horas)	Duración Real (Horas)	% Costo de Calidad	% Desviación de Costo
SOL 9175	Portal Único de Clientes Fase IV	Nov-2015	620	690	6.96%	10.32%
SOL 9227	Validación Código Vendedor	Nov-2015	490	525	4.57%	7.14%
SOL 9366	Conexión Autos Equifax Fase I	Nov-2015	400	432	7.99%	5.56%
SOL 9370	Conexión Autos Equifax Fase II	Dic-2015	391	415	5.378%	6.14%
SOL 9471	Implementación eSIC Fase 1	Dic-2015	428	432	0.93%	0.93%
SOL 9472	SITEDS Servicios	Ene-2016	548	572	4.20%	4.36%
SOL 9437	Campaña de Asesores Vida	Ene-2016	511	547	6.58%	7.05%
SOL 9711	Upgrade Sistema Integral de Plataforma SIP Fase 1	Feb-2016	800	864	6.25%	6.75%
SOL 9712	Upgrade Sistema Integral de Plataforma SIP Fase 2	Feb-2016	974	1010	3.56%	3.70%
SOL 9742	SOAT Web Primax	Mar-2016	520	560	7.14%	7.69%
SOL 9848	ECommerce SOAT	Mar-2016	458	486	4.94%	6.11%

Ficha de observación: Cantidad de Defectos (Pre Prueba)

Investigador	Espejo Chavarría, Alex
Institución donde investiga	 High performance. Delivered.
Dirección	Av. Nicolás de Piérola N° 589 INT. 6PS Lima - Lima - Lima
Indicadores observados	Cantidad de Defectos

ID Proyecto	Nombre Proyecto	Reporte de Especificaciones Funcionales REF	Reporte de Especificaciones Técnicas RET	Arquitectura & Construcción	Certificación	Implantación & Seguimiento	TOTAL
SOL 9367	Modificación reporte RRGG – ELPA	5	10	79	10	2	106
SOL 9477	VTAR_BCP - WILLIS Fase II	2	2	61	6	4	75
SOL 9571	Clientes de FFW de Negocios Directos	2	5	73	2	1	83
SOL 9216	Información EPS en los Estados Financieros	8	11	97	4	3	123
SOL 9266	Renovación SCTR	17	21	145	21	7	211
SOL 9465	Redondeo de Facturas Globales y de Comisiones 1.5	2	2	91	5	4	104
SOL 9502	Vulnerabilidad CG Web	4	16	215	8	6	249
SOL 9546	Problemas en el Servidor de Impresión Emisión - Optimización	2	7	164	21	2	196
SOL 9554	Cuenta Corriente de Coaseguro	1	2	8	2	1	14
SOL 9570	Automatización Tramas MLPE-PH01-PRE1-VDES	2	3	39	9	6	59
SOL 9636	Nuevas Cuentas Resolución 7037SB	7	4	59	7	4	81

Ficha de observación: Cantidad de Defectos (Post Prueba)

Investigador	Espejo Chavarría, Alex
Institución donde investiga	 High performance. Delivered.
Dirección	Av. Nicolás de Piérola N° 589 INT. 6PS Lima - Lima - Lima
Indicadores observados	Cantidad de Defectos

ID Proyecto	Nombre Proyecto	Reporte de Especificaciones Funcionales REF	Reporte de Especificaciones Técnicas RET	Arquitectura & Construcción	Certificación	Implantación & Seguimiento	TOTAL
SOL 9175	Portal Único de Clientes Fase IV	4	2	37	2	3	48
SOL 9227	Validación Código Vendedor	2	6	23	1	2	34
SOL 9366	Conexión Autos Equifax Fase I	3	4	18	3	2	30
SOL 9370	Conexión Autos Equifax Fase II	2	6	21	2	2	33
SOL 9471	Implementación eSIC Fase 1	1	4	31	4	2	42
SOL 9472	SITEDS Servicios	6	7	28	6	2	49
SOL 9437	Campaña de Asesores Vida	1	4	19	3	2	29
SOL 9711	Upgrade Sistema Integral de Plataforma SIP Fase 1	12	11	61	7	4	95
SOL 9712	Upgrade Sistema Integral de Plataforma SIP Fase 2	9	7	54	5	3	74
SOL 9742	SOAT Web Primax	4	6	43	3	2	58
SOL 9848	ECommerce SOAT	1	5	39	5	2	59

ANEXO H: Cuestionario

Cuestionario: Indicador Satisfacción del Cliente (Pre Prueba)

Código de Proyecto		
Nombre del Proyecto		
Nombre y Apellidos		
Cargo		

OBJETIVO	Recopilar información sobre su nivel de satisfacción en la gestión de los proyectos respecto a las principales restricciones (Alcance, Tiempo, Costo, Calidad, Riesgo y Satisfacción del Cliente).
-----------------	--

ESCALA VALORATIVA				
MUY INSATISFECHO (1)	INSATISFECHO (2)	NO SATISFECHO, NI INSTISFECHO (3)	SATISFECHO (4)	MUY SATISFECHO (5)

ASPECTOS A EVALUAR		GRADO DE IMPORTANCIA				
		1	2	3	4	5
ALCANCE						
P1	Documentación de requisitos y Matriz de trazabilidad de requisitos.					
P2	Enunciado del alcance, EDT, Diccionario EDT.					
TIEMPO						
P3	Lista de Actividades, Atributos de las Actividades y Lista de hitos.					
P4	Ejecución del proyecto según cronograma.					
COSTO						
P5	Umbrales de control, Reglas de valor ganado, formatos de informe.					
P6	Ejecución del proyecto según cronograma.					
CALIDAD						
P7	Métricas de calidad, Listas de control y Plan de mejora de procesos.					
P8	Ejecución del proyecto según plan de calidad.					
RIESGO						
P9	Plan de gestión de riesgos.					
P10	Informes de desempeño de trabajo.					
SATISFACCIÓN DE CLIENTE						
P11	Comunicaciones del proyecto e Informes de desempeño del trabajo.					
P12	Plan de gestión de proyecto y solicitudes de cambio.					

¡SUS APORTES, FORTALECEN NUESTRO CRECIMIENTO!

ANEXO I: Matriz de consistencia

Problema	Objetivos	Hipótesis	Variables	Indicadores	Instrumentos de recopilación de información
¿De qué manera la adopción de un modelo basado en los modelos de Madurez de Capacidades Integrado, Procesos de Software de Equipos y Proceso de Software de Personas permite asegurar la calidad en el proceso de desarrollo de software?	Generar un modelo que permita asegurar la calidad en el proceso de desarrollo de software basado en los modelos de madurez de capacidades, Proceso de Software de Equipos y Proceso de Software de Personas.	La adopción del modelo de aseguramiento de la calidad basado en los modelos de madurez de capacidades CMMi, TSP y PSP permite asegurar la calidad en el proceso de desarrollo de software.	<p>Variable X = Variable Independiente: Modelo de Aseguramiento de la Calidad en el Proceso de Desarrollo de Software.</p> <p>Indicador:</p> <p>Variable Y = Variable Dependiente: Aseguramiento de la Calidad en el proceso de desarrollo de software.</p>	<ul style="list-style-type: none"> - % de Costo de Calidad (Costo de Conformidad + Costo de No Conformidad). - % de variación respecto a la línea base de costo. - % de Clientes Satisfechos. 	Análisis Documental (Ficha de Registro de Datos). Encuesta (Cuestionario).
Problema Específico	Objetivo Específico	Hipótesis Subsidiaria	Variables	Indicadores	Instrumentos de recopilación de información
¿De qué manera la adopción del modelo de aseguramiento de calidad en el proceso de desarrollo de software reduce el costo de calidad?	Reducir el costo de calidad en los proyectos a través del aseguramiento de la calidad en el proceso de desarrollo de software.	H.1.: La integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software reduce el costo de calidad.	<p>Variable X = Variable Independiente: Modelo de Aseguramiento de la Calidad</p> <p>Indicador:</p> <p>Variable Y = Variable Dependiente: Reducción del Costo de Calidad</p>	<ul style="list-style-type: none"> - % de Costo de Calidad (Costo de Conformidad + Costo de No Conformidad) 	Análisis Documental (Ficha de Registro de Datos).
¿De qué manera la integración del modelo de aseguramiento de calidad reduce los defectos en los proyectos incurridos en el proceso de desarrollo de software?	¿De qué manera la integración del modelo de aseguramiento de calidad reduce los defectos en los proyectos incurridos en el proceso de desarrollo de software?	H2.: La integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software reduce los defectos en los proyectos.	<p>Variable X = Variable Independiente: Modelo de Aseguramiento de la Calidad</p> <p>Indicador:</p> <p>Variable Y = Variable Dependiente: Reducción de los defectos</p>	<ul style="list-style-type: none"> - Cantidad de Defectos en el Proceso de Desarrollo de Software. 	Análisis Documental (Ficha de Registro de Datos).

<p>¿De qué manera la adopción del modelo de aseguramiento de calidad en el proceso de desarrollo de software incrementa la rentabilidad de los proyectos?</p>	<p>Incrementar la rentabilidad de los proyectos a través del aseguramiento de la calidad en el proceso de desarrollo de software.</p>	<p>H.3.: La integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software incrementa la rentabilidad de los proyectos.</p>	<p>Variable X = Variable Independiente: Modelo de Aseguramiento de la Calidad Indicador: Variable Y = Variable Dependiente: Incrementar Rentabilidad de los Proyectos.</p>	<ul style="list-style-type: none"> - % de variación respecto a la línea base de costo. 	<p>Análisis Documental (Ficha de Registro de Datos).</p>
<p>¿De qué manera la adopción del modelo de aseguramiento mejora la satisfacción de los clientes?</p>	<p>Aumentar el grado de satisfacción de los clientes a través del aseguramiento de la calidad en el proceso de desarrollo de software.</p>	<p>H.4.: La integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software aumenta el grado de satisfacción de los clientes.</p>	<p>Variable X = Variable Independiente: Modelo de Aseguramiento de la Calidad. Indicador: Variable Y = Variable Dependiente: Mejorar la Satisfacción de los Clientes.</p>	<ul style="list-style-type: none"> - % de Clientes Satisfechos. 	<p>Encuesta (Cuestionario)</p>

Fuente. Elaboración Propia

ANEXO J: Modelo de aseguramiento de la calidad

El siguiente cuadro integra los procesos del Modelo de Madurez de Capacidades CMMI, Modelo TSP y PSP; el cual representa a nuestra propuesta de la creación de un modelo de aseguramiento de la calidad en el proceso de desarrollo de software para empresas y consultoras que cuenten con certificación CMMI Nivel 2 o 3.

Nivel de Madurez	Áreas de Proceso por Nivel	Proceso de TSP/PSP	Herramientas
Nivel 3: Definido	RD – Desarrollo de Requerimientos	TSP – Proceso para Trabajar en Equipos PSP – Proceso de Planeación Personal	- Checklist - Logs y Forms - Estándares - Checklist de Planeación
	TS – Solución Técnica	TSP – Proceso para Trabajar en Equipos PSP – Proceso de Planeación Personal PSP – Administración de Calidad Personal	- Estándares - Checklist - Logs y Forms - Evaluación de Pares - Registro de defectos
	PI – Integración del Producto	TSP – Proceso para Trabajar en Equipos PSP – Proceso de Planeación Personal PSP – Administración de Calidad Personal	- Estándares - Checklist - Evaluación de Pares

	VER – Verificación	TSP – Proceso para Trabajar en Equipos	- Estándares - Checklist - Evaluación de Pares - Logs y Forms
	VAL – Validación	TSP – Proceso para Trabajar en Equipos	- Checklist - Especificaciones - Evaluación de Pares - Logs y Forms
	IT – Equipos de Trabajo Integrados	TSP – Proceso para Trabajar en Equipos PSP – Proceso de Planeación Personal PSP – Administración de Calidad Personal	- Evaluación de Pares - Logs y Forms - Scripts
	RSKM – Gestión de Riesgos	TSP – Proceso para Trabajar en Equipos PSP – Administración de Calidad Personal	- Scripts - Logs y Forms - Estándares - Formación / Capacitación
	DAR – Análisis de Decisiones y Soluciones	TSP – Proceso para Trabajar en Equipos PSP – Administración de Calidad Personal	- Análisis de Resultados - Evaluación de Equipos - Logs y Forms
Nivel 2: Gestionado	REQM – Gestión de Requisitos	TSP – Proceso para Trabajar en Equipos PSP – Proceso de Planeación Personal	- Scripts - Logs y Forms.

			<ul style="list-style-type: none"> - Checklist - Medición de Tamaño
	PP – Planificación de Proyecto	<p>TSP – Proceso para Construir Equipos</p> <p>TSP – Proceso para Trabajar en Equipo</p> <p>PSP – Proceso de Planeación Personal</p>	<ul style="list-style-type: none"> - Scripts - Checklist de Planeación - Formulario Resumen de Plan de proyecto.
	PMC – Seguimiento y Control de Proyecto	<p>TSP – Proceso para Trabajar en Equipo</p> <p>PSP – Administración de Calidad Personal</p>	<ul style="list-style-type: none"> - Evaluación de Equipos - Análisis de Resultados - Estándares - Informes Post Mortem
	MA – Medición y Análisis	<p>TSP – Proceso para Trabajar en Equipo</p> <p>PSP – Administración de Calidad Personal</p>	<ul style="list-style-type: none"> - Evaluación de Equipos - Análisis de Resultados - Checklist - Logs y Forms - Informes Post Mortem
	PPQA – Aseguramiento de la Calidad del Proceso y Producto	<p>TSP – Proceso para Construir Equipos</p> <p>TSP – Proceso para Trabajar en Equipo</p> <p>PSP – Administración de Calidad Personal</p>	<ul style="list-style-type: none"> - Logs y Forms. - Forms - Checklist

			<ul style="list-style-type: none">- Análisis de Resultados- Estándares- Evaluación de Pares- Formación / Capacitación- Informes Post Mortem
	CM - Gestión de la Configuración	TSP – Proceso para Trabajar en Equipo PSP – Proceso de Planeación Personal PSP – Administración de Calidad Personal	<ul style="list-style-type: none">- Estándares- Scripts- Checklist

Fuente. Elaboración Propia

ANEXO K: Formatos PSP

Plantillas utilizadas para documentar estimaciones, tiempos, defectos y plan de proyecto.

Formato: Estimación Semanal de Actividades

Proyecto:	Fecha:						
Nombre:	 accenture High performance. Delivered.						
Estimación de Semana:							
Observación:							
TAREA	Reporte de Especificaciones Funcionales	Reporte de Especificaciones Técnicas	Construcción & Arquitectura	Certificación	Implantación & Seguimiento	-	TOTAL
L							
M							
X							
J							
V							
S							
D							
TOTAL							

Formato: Cuaderno de Registro de Tiempos

Formato: Cuaderno de Registro de Defectos

Proyecto:	Fecha:					
Nombre:						
Observación:						
Tipos de defectos:						
10 Reporte de Especificaciones Funcionales 20 Reportes de Especificaciones Técnicas 30 Construcción & Arquitectura 40 Certificación 50 Implantación & Seguimiento						
Fecha	Nº	Tipo	Introducido	Eliminado	T. Corrección	Defecto Corregido
Descripción:						
Fecha	Nº	Tipo	Introducido	Eliminado	T. Corrección	Defecto Corregido
Descripción:						
Fecha	Nº	Tipo	Introducido	Eliminado	T. Corrección	Defecto Corregido
Descripción:						
Fecha	Nº	Tipo	Introducido	Eliminado	T. Corrección	Defecto Corregido
Descripción:						

Formato: Resumen Plan de Proyecto

Proyecto:	Fecha:	> accenture High performance. Delivered.		
Nombre:	Lenguaje:			
Observación:				
Resumen	Plan	Real	Hasta la fecha	
Minutos/LOC				
LOC/Hora				
Defectos/KLOC				
Rendimiento				
V/F				
Tamaño Proyecto	Plan	Real	Hasta la fecha	%Hasta la fecha
Total Nuevo & Cambiado				
Tamaño Máximo				
Tamaño Mínimo				
Tiempo por fase (min)	Plan	Real	Hasta la fecha	%Hasta la fecha
R. Especif. Funcionales				
R. Especif. Técnicas				
Construcción & Arquitectura				
Certificación				
Implantación & Seguimiento				
Total				
Defectos introducidos	Plan	Real	Hasta la fecha	%Hasta la fecha
R. Especif. Funcionales				
R. Especif. Técnicas				
Construcción & Arquitectura				
Certificación				
Implantación & Seguimiento				
Total				
Defectos eliminados	Plan	Real	Hasta la fecha	%Hasta la fecha
R. Especif. Funcionales				
R. Especif. Técnicas				
Construcción & Arquitectura				
Certificación				
Implantación & Seguimiento				
Total				

ANEXO L: Formatos TSP

Formato: Intereses y Experiencias

Objetivo	Use este formato para describir los intereses y experiencias de los integrantes del proyecto.
General	<ul style="list-style-type: none"> ✓ Se usará para las asignaciones de equipo y de rol. ✓ La información de horario es necesaria, para que los equipos puedan formarse y reunirse de acuerdo a un tiempo establecido. ✓ Se lee la descripción de roles en la sección correspondiente. ✓ Se usan páginas adicionales, en caso necesario.
Encabezado	<ul style="list-style-type: none"> ✓ Su nombre, el nombre del proyecto y la fecha. ✓ Campo de estudio.
Experiencia e intereses relevantes	<ul style="list-style-type: none"> ✓ Se indica cualquier experiencia e intereses que puedan ser útiles al momento de formar los equipos y hacer la asignación de roles. Los ejemplos serían: facilidad en lenguajes, experiencia en PSP, experiencia en diseño y desarrollo de base de datos, etc.
Proyectos en otros equipos	<ul style="list-style-type: none"> ✓ Se indica cualquier experiencia e intereses que puedan ser útiles al momento de formar los equipos y hacer la asignación de roles. Los ejemplos incluirían: tipos de proyectos en los que ha participado, roles desempeñados, herramientas o métodos utilizados, etc.
Liderazgo o administración	<ul style="list-style-type: none"> ✓ Se indica cualquier experiencia en liderazgo o en administración que puedan ser útiles al momento de formar los equipos y hacer la asignación de roles. Ejemplos: administrador de alguna organización, asistente de maestro, tiempo permanecido en la oficina, etc.
Preferencias del equipo	<ul style="list-style-type: none"> ✓ Se indican las preferencias al trabajar con grupos particulares.
Horario	<ul style="list-style-type: none"> ✓ Se indica el tiempo dedicado a clases o a otras actividades. ✓ Si el tiempo no es compatible con el horario registrado, se definen períodos aproximados y se anota debajo el tiempo preciso.
Preferencias en roles	<ul style="list-style-type: none"> ✓ Se indica el rango de acuerdo a la preferencia de roles. ✓ Se pueden marcar varios roles.

Nombre: _____ Proyecto: _____
Fecha: _____
Especialización: _____

Describa brevemente su experiencia e intereses relevantes

Describa brevemente su trabajo en otros proyectos de equipo

Describa brevemente cualquier experiencia en liderazgo o en administración (ya sea en el área laboral o en organizaciones)

Si existen preferencias en equipo indíquelas

Liste su horario de clases y de otras actividades semanales, tales como trabajo, organizaciones, equipos de deporte, etc.

Marque los roles de su preferencia utilizando un rango desde 1 (menor) hasta 5 (mayor)

Líder de equipo	1	2	3	4	5
Administrador de Desarrollo	1	2	3	4	5
Administrador de Planeación	1	2	3	4	5
Administrador de Calidad/Proceso	1	2	3	4	5
Administrador de Apoyo	1	2	3	4	5

Formato: Estrategia

Objetivo	<ul style="list-style-type: none"> ✓ Este formato se usa para registrar las decisiones estratégicas. ✓ Se usa durante el desarrollo de la estrategia para asignar las funcionalidades del producto en cada ciclo. ✓ También se usa durante el diseño de alto nivel para asignar las funcionalidades de Especificaciones de Requisitos de Software (ERS) a los componentes.
General	<ul style="list-style-type: none"> ✓ Esta forma sugiere una manera de registrar las decisiones de estrategia. ✓ Utilice este u otro formato que contenga los mismos datos.
Encabezado	<ul style="list-style-type: none"> ✓ Se registra el nombre del rol, fecha, nombre del equipo y del instructor. ✓ Se indica el nombre de la parte y su nivel. ✓ Se registra el número de ciclo.
Referencia	<ul style="list-style-type: none"> ✓ Use esta columna para listar la definición de necesidades o el párrafo de ERS, o el número de funcionalidad de cada ciclo.
Funcionalidades	<ul style="list-style-type: none"> ✓ En esta columna liste todas las funcionalidades que serán incluidas en todos los ciclos del producto.
LOC en ciclos	<ul style="list-style-type: none"> ✓ Se registran las LOC estimadas para cada funcionalidad bajo el número del ciclo correspondiente. ✓ Si planea implementar una funcionalidad en dos o tres ciclos, se registran las LOC nuevas y cambiadas en cada ciclo. ✓ Si una funcionalidad es incluida en las LOC de otra, se indica con una X.
Horas en ciclos	<ul style="list-style-type: none"> ✓ Se registra el tiempo estimado para cada funcionalidad bajo el número del ciclo planeado. ✓ Si planea implementar una funcionalidad en dos o tres ciclos, se registra el tiempo de desarrollo estimado en cada ciclo. ✓ Si una funcionalidad se incluye en las horas de otra, indíquelo con una X.

Nombre _____ Fecha _____
Proyecto _____ Equipo _____
Parte / Nivel _____ Fase _____

Formato: Reporte Semanal

Objetivo	Use el formato para preparar el reporte semanal
General	<ul style="list-style-type: none"> ✓ Cada integrante del equipo realiza esta forma semanalmente indicando el trabajo terminado durante la última semana y los planes para la siguiente. ✓ Cada semana, el AP prepara una copia de la forma SEMANA con un resumen del estado del equipo y del trabajo realizado en la semana.
Encabezado	<ul style="list-style-type: none"> ✓ Se indica el nombre del rol y el del responsable. ✓ Se indica el nombre del equipo, número de ciclo, fecha y número de semana.
Datos semanales	<ul style="list-style-type: none"> ✓ Se indica el total de horas utilizadas y las planeadas en la semana. ✓ Se indica el total de horas acumuladas planeadas y actuales durante el ciclo de desarrollo. ✓ Se indica el valor planeado y el valor actual ganado en la semana. ✓ Se indica el valor planeado acumulado y el valor ganado acumulado durante el ciclo de desarrollo a la fecha. ✓ Se indican las horas totales planeadas y actuales para las tareas terminadas durante el ciclo de desarrollo a la fecha.
Datos semanales de cada integrante del equipo	<p>Para cada integrante reporte:</p> <ul style="list-style-type: none"> ✓ El tiempo total actual y planeado por cada ingeniero. ✓ El valor planeado y ganado de cada ingeniero en la semana. ✓ Las horas totales planeadas y trabajadas por cada ingeniero.
Datos semanales del equipo	<p>Para el reporte semanal del equipo registre:</p> <ul style="list-style-type: none"> ✓ El tiempo total actual y planeado por el equipo. ✓ El valor planeado y ganado por el equipo. ✓ Las horas totales planeadas y trabajadas del equipo.
Tareas de desarrollo terminadas	<p>Para las tareas terminadas en la semana indique:</p> <ul style="list-style-type: none"> ✓ El nombre de cada tarea. ✓ El tiempo total actual y planeado para cada tarea. ✓ Número de semana en que se planeó la tarea. ✓ Valor ganado para la tarea.
Seguimientos de asuntos y riesgos	<ul style="list-style-type: none"> ✓ Resuma el estado de asuntos y riesgos, así como cualquier cambio importante acontecido en la semana.
Otros elementos importantes	<ul style="list-style-type: none"> ✓ Liste cualquier tarea realizada o evento ocurrido en la semana. ✓ Ejemplos de roles que incluyan el estándar de código terminado, procedimiento de control de cambios aprobado, etc. ✓ Ejemplos de desarrollo que incluyan diseño, codificación, inspección o pruebas de varios elementos del producto.

Nombre _____ Proyecto _____ Equipo _____
 Fecha _____ Ciclo _____ Semana # _____

Datos semanales	Planeado	Actual
Horas del proyecto para esta semana		
Horas del proyecto de este ciclo a la fecha		
Valor ganado para esta semana		
Valor ganado en este ciclo a la fecha		
Horas totales para las tareas terminadas en esta fase a la fecha		

Datos semanales por rol	Horas planeadas	Horas actuales	Valor planeado	Valor ganado
LE				
AD				
AP				
ACP				
AA				
Totales				

Tareas de desarrollo terminadas	Horas planeadas	Horas actuales	Valor ganado	Semana planeada
Totales				

Seguimiento de asuntos o riesgos	Estatus

Formato: Evaluación Equipos y Roles

Propósito	Use este formato para evaluar al equipo y los roles
General	<ul style="list-style-type: none"> ✓ Los equipos entregan los formatos y los reportes de postmortem. ✓ Se evalúa el desempeño personal y por equipo. ✓ Se añaden comentarios o sugerencias al reverso de la forma o en hojas aparte.
Encabezado	<ul style="list-style-type: none"> ✓ Se registra su nombre y del proyecto. ✓ Se registra el nombre del equipo, el número de ciclo, la fecha y el número de semana.
Trabajo y dificultad requeridos en el rol	<p>Para cada rol:</p> <ul style="list-style-type: none"> ✓ Se estima el trabajo requerido por cada rol. ✓ Se estima la dificultad de cada rol. <p>Haga las estimaciones en términos de porcentaje, hasta sumar un 100%.</p> <ul style="list-style-type: none"> ✓ Por ejemplo, si fueron relativamente iguales, asigne 20, 20, 20, 20 y 20.
Evaluación completa del equipo	<p>Con una escala desde 1(bajo) hasta 5(alto), evalúe:</p> <ul style="list-style-type: none"> ✓ Espíritu de equipo, efectividad del equipo, experiencia adquirida, productividad, calidad del proceso y del producto.
Contribución completa de cada rol	<ul style="list-style-type: none"> ✓ Se evalúa cada rol de acuerdo a la contribución que el ingeniero aportó al proyecto durante el ciclo de desarrollo. ✓ Se encierra un número, desde el 1 (considerado la contribución más baja) hasta el 5 (contribución más alta).
Apoyo y ayuda proporcionado por cada rol	<ul style="list-style-type: none"> ✓ Se evalúa cada rol de acuerdo a la ayuda aportada por el ingeniero que lo desempeñó, durante el ciclo de desarrollo del proyecto. ✓ Se encierra un número, desde el 1 (considerado mínima ayuda) hasta el 5 (bastante ayuda).
Desempeño del rol	<ul style="list-style-type: none"> ✓ Se evalúa cada rol de acuerdo al desempeño. ✓ La evaluación debe basarse en la manera de haber manejado las tareas y responsabilidades definidas. ✓ Se encierra un número, desde el 1 (considerando desempeño más bajo) hasta el 5 (bastante alto).

Nombre: _____ Proyecto: _____ Equipo: _____
 Fecha: _____ No. ciclo: _____ No. semana: _____

Para cada rol, evalúe el por ciento de trabajo requerido y la dificultad durante este ciclo

Rol	Trabajo requerido	Dificultad en el rol
Líder de Equipo		
Administrador de Desarrollo		
Administrador de Planeación		
Administrador de Calidad y Proceso		
Administrador de Apoyo		
Contribución total (100%)		

Evaluación del equipo sobre los siguientes criterios. Encierre un número desde el 1 (bajo) hasta el 5 (alto)

Espíritu de equipo	1	2	3	4	5
Efectividad total	1	2	3	4	5
Experiencia adquirida	1	2	3	4	5
Productividad del equipo	1	2	3	4	5
Calidad del proceso	1	2	3	4	5
Calidad del producto	1	2	3	4	5

Evalué la contribución de cada rol. Encierre un número desde el 1 (bajo) hasta el 5 (alto)

Líder de Equipo	1	2	3	4	5
Administrador de Desarrollo	1	2	3	4	5
Administrador de Planeación	1	2	3	4	5
Administrador de Calidad y Proceso	1	2	3	4	5
Administrador de Apoyo	1	2	3	4	5

Evalué cada rol de acuerdo al apoyo y ayuda proporcionada. Encierre un número desde el 1 (bajo) hasta el 5 (alto)

Líder de Equipo	1	2	3	4	5
Administrador de Desarrollo	1	2	3	4	5
Administrador de Planeación	1	2	3	4	5
Administrador de Calidad y Proceso	1	2	3	4	5
Administrador de Apoyo	1	2	3	4	5

Evalué cada rol de acuerdo a su desempeño. Encierre un número desde el 1 (bajo) hasta el 5 (alto)

Líder de Equipo	1	2	3	4	5
Administrador de Desarrollo	1	2	3	4	5
Administrador de Planeación	1	2	3	4	5
Administrador de Calidad y Proceso	1	2	3	4	5
Administrador de Apoyo	1	2	3	4	5

Notas:

ANEXO M: Checklist CMMI

Especificación Checklist de Recepción de Petición de Estimación

1	2	A	B	C	D	E
Checklists de Recepción de Petición Estimación						
-	1			Cliente: PACIFICO		
-	2			Número		
-	3			Nombre		
-	4			Nombre del Archivo:		
-	5			Autor del Producto de Trabajo:		
-	6			Revisado por:		
-	7			Fecha de Revisión:		
-	8			Tamaño del Producto de Trabajo:		
-	9			Coste de Revisión (Horas):		
-	10			Status de la Checklist:	FAIL	
-	11					
-	12					
13		#	Criticó	Criterio	Estado	Comentarios/Acción*
14						
-	15	1. CONTROL DE DOCUMENTACIÓN DE ENTRADA			APPLY	<- Change value if required
-	16	1.a. Requerimientos de Entrada			APPLY	<- Cambiar el valor en caso necesario
-	17	1	X	¿Se ha identificado si el requerimiento recibido es del tipo SOL, SOL Rapido, Correctivo Planificado o Documentación		
-	18	2	X	La ubicación de los requerimientos recibidos cumple las normas aplicables a la gestión de configuración del proyecto.		
-	19	3	X	Teniendo en cuenta el entorno, expectativas y necesidades del usuario, los requerimientos recibidos son factibles, implementables, completos, y consistentes con otros ya existentes.		
-	20	4	X	¿Lo requerido en la petición está de acuerdo con el servicio acordado con el cliente?		
-	21	5		¿Cada petición tiene una única interpretación sin posibilidad de que se haya descrito en términos que se preste a múltiples significados?		

Especificación Checklist de Revisión de Pares de Análisis de Impacto y Estimación

1	2	A	B	C	D	E
Peer Review de Análisis de Impacto y Estimación						
-	1			Cliente: PACIFICO		
-	2			Número		
-	3			Nombre		
-	4			Nombre del Documento Análisis Impacto Revisado:	IMPACTO_	
-	5			Nombre del Documento Estimador Revisado:	ESTIMATOR_	
-	6			Autor del Producto de Trabajo:		
-	7			Revisado por:		
-	8			Fecha de Revisión:		
-	9			Tamaño del Producto de Trabajo:		
-	10			Coste de Revisión (Horas):		
-	11			Status de la Checklist:	Falta Completar	
-	12					
-	13					
14		#	Criticó	Criterio	Estado	Comentarios/Acción*
15						
-	16	2		Se han identificado en el documento los responsables de la validación por parte del cliente. Sección "Identificación del Requerimiento", campo EDS	OK	
-	38	3		Se han completados las fechas solicitadas en la sección "Identificación del Requerimiento" y verificado si se cumplen con los compromisos de tiempo. Se es mas de una entrega se debe especificar	OK	
-	39	4			N/A	
-	40	5			N/A	
-	41	2.2. Impacto sobre Aplicativo y Alternativas			APPLY	<- Cambiar el valor en caso necesario
-	42	1		Se hace referencia en la columna REF al numero de requerimiento especificado en el SOL	OK	
-	43	2		Se comproba por trazabilidad de que todos los requerimientos especificados en el SOL esten contemplados dentro del análisis de impacto.	OK	
-	44					

Especificación Checklist de Diseño Funcional

1	2	3	4	5	A	B	C	D	E
1	2	3	4	5			Checklist de Diseño Funcional		
6	7	8	9	10			<Código> - <Descripción>		
11	12	13	14	15			Cliente:	PACIFICO	
16	17	18	19	20			Operadora:		
21	22	23	24	25			Tipo de Revisión:	FDR	
26							Producto de Trabajo:	<Código>	
							Descripción de Producto de Trabajo:	<Descripción>	
							Autor de Producto de Trabajo:		
							Revisado por:		
							Fecha de Revisión:		
							Tamaño de Producto de Trabajo:		
							Coste de Revisión (Horas):		
							Status Checklist:	OK	
							#	Critical	Comments/Actions*
								APPLY	<- Change value if required
							1.	GENERAL	
							1	X	La ubicación y nomenclatura del Diseño Funcional cumple las normas aplicables a la gestión de configuración del proyecto.
							2	X	El Diseño Funcional está realizado sobre la última versión de la plantilla establecida para el proyecto.
							1	X	Están definidos los objetivos, alcances y definiciones de acuerdo al requisito funcional
							2	X	Desde el Diseño Funcional se puede realizar trazabilidad hacia atrás, que permita identificar los requisitos funcionales que originaron la petición
							3	X	El nivel de claridad y detalle de la información es adecuado para realizar el diseño.
							4		Se han seguido los estándares de documentación definidos.

Especificación Checklist de Diseño Técnico

1	2	3	4	A	B	C	D	E
1	2	3	4					
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19	#	Critical Y		Criteria		Status		Comments/Actions*
20	1.	GENERAL				APPLY		<- Change value if required
31	2.	DISEÑO ORIENTADO A OBJETOS				APPLY		<- Change value if required
32	2.a -	DIAGRAMA DE TRES CAPAS				APPLY		<- Change value if required
37	2.b -	DIAGRAMA DE SECUENCIA				APPLY		<- Change value if required
43	2.c -	DIAGRAMA DE CASOS DE USO				APPLY		<- Change value if required
50	2.d -	DIAGRAMA DE CAPA DE PRESENTACIÓN				APPLY		<- Change value if required
56	2.e -	DIAGRAMA DE CAPA LÓGICA				APPLY		<- Change value if required
67	2.f -	DIAGRAMA DE LÓGICA DEL NEGOCIO				APPLY		<- Change value if required
73	2.f -	DIAGRAMA DE SCRIPTS				APPLY		<- Change value if required
82	3.	PROCEDURAL DESIGN				APPLY		<- Change value if required
174	4.	DEFINICIÓN DE INTERFAZES O LLAMADAS A OTROS ANAGRAMAS				APPLY		<- Change value if required

Checklist de Diseño Técnico

<Código> - <Descripción>

Cliente: PACIFICO
Operadora:
Tipo de Revisión: TDR
Producto de Trabajo: <Código>
Descripción de Producto de Trabajo: <Descripción>
Autor de Producto de Trabajo:
Revisado por:
Fecha de Revisión:
Tamaño de Producto de Trabajo:
Coste de Revisión (Horas):
Status Checklist: OK

Especificación Checklist de Construcción

1	2	3	A	B	C	D	E
							Revisión de Construcción
				<input type="button" value="Submit"/>			
-	1	2	3				
.	4						
.	5						
.	6						
.	7						
.	8						
.	9						
.	10						
.	11						
.	12						
.	13						
.	14						
.	15						
	16						
	17						
	18						
	19	#	Critical ^Y	Criteria	Status	Comments/Actions*	
+	20	1. GENERAL			N/A	<- Change value if required	
+	39	2. HTML/JSP STANDARDS			N/A	<- Change value if required	
+	55	3. SCRIPTS STANDARDS			N/A	<- Change value if required	
+	70	4. JAVA STANDARDS			N/A	<- Change value if required	
+	93	5. PL/SQL STANDARDS			N/A	<- Change value if required	
+	110	6..NET STANDARDS			N/A	<- Change value if required	
+	132	7. SQL OTHERS			N/A	<- Change value if required	
+	140	8. C, C++ OTHERS			N/A	<- Change value if required	
+	177	9. NATURAL			N/A	<- Change value if required	
+	224	10. COBOL			N/A	<- Change value if required	
+	340	11. JCLs			N/A	<- Change value if required	
+	355	12. CODIFICACIÓN POR TABLAS			N/A	<- Change value if required	
+	358	13. VISUAL BASIC			N/A	<- Change value if required	
+	426	14. POWER BUILDER			N/A	<- Change value if required	

Especificación Checklist de Pruebas, Sistemas e Integración

1	2	A	B	C	D	E	
			<input type="button" value="Submit"/>				
-	1						
.	2						
.	3						
.	4						
.	5						
.	6						
.	7						
.	8						
.	9						
.	10						
.	11						
.	12						
.	13						
.	14						
.	15						
	16						
	17	#	Critical ^Y	Criteria	Status	Comments/Actions*	
+	18	1. GENERAL			APPLY	<- Change value if required	
+	32	2. CASOS DE PRUEBA				<- Change value if required	
+	41	3. DATOS DE PRUEBA				<- Change value if required	
+	46	4. CALIDAD				<- Change value if required	
+	50	5. DOCUMENTACION DE LA PRUEBA				<- Change value if required	
+	57	6. RESOLUCION DE ERRORES				<- Change value if required	
+	59	7. OTROS					
	60						
	61						
	62						
	63						
	64						
	65						

Especificación Checklist de Entrega

	A	B	C	D	E
	1	Submit	Revisión de Entrega		
-	2		Cliente:		
.	3		Operadora:		
.	4		Tipo de Revisión: CRW		
.	5		Producto de Trabajo:		
.	6		Descripción del Producto de Trabajo:		
.	7		Autor del Producto de Trabajo:		
.	8		Revisado por:		
.	9		Fecha de Revisión:		
.	10		Tamaño del Producto de Trabajo:		
.	11		Coste de Revisión (Horas):		
.	12		Status de Checklist: OK		
.	13				
.	14				
	#	Critical ^y	Criterio	Status	Comments/Actions*
-	15				
-	16		1. GENERAL	APPLY	<- Change value if required
.	17	X	Se ha generado el paquete de entrega con lista de componentes		
.	18	2	Se realizó la entrega de componentes mediante la herramienta de administración de la configuración / entregas utilizadas		
.	19	3	Se realizó la actualización del estado de la entrega en la herramienta de entregas (Project Server)		
.	20	4	Se realizó la actualización del estado de la entrega en la herramienta de Gestión de peticiones quedando en estado "Entregada"		
-	21		2. BBDD	APPLY	<- Change value if required
.	22	1	Se copiaron las estadísticas de las tablas antes del cambio a las nuevas tablas generadas o modificadas		
-	23		4. Manual de Usuario	N/A	<- Change value if required
.	24	1	El documento o Manual de Usuario se genero dentro de los patrones establecidos		
-	25		5. Documentación	N/A	<- Change value if required
.	26	1	Se encuentra almacenada toda la documentación en el Sharepoint de Apolo (estimación, diseño, pruebas), dentro de carpeta definida para cada anagrama		
.	27	2	La documentación que se generó durante el ciclo de vida del requerimiento está alineada al template acordado con el cliente		
-	28		6. Entregable	N/A	<- Change value if required
.	29	1	Se ejecutaron todos los checklists asociados al ciclo de vida de desarrollo de las peticiones		
.	30	2	El código fuente está incluido en el paquete entregable		
.	31	3	Se incluye todo script sql utilizado en el desarrollo (create y replace packages, create y alter tables, insert y updates de configuración) en el paquete entregable		
.	32		7. OTROS	N/A	<- Change value if required

ANEXO N: Glosario de Términos

Modelo de Referencia: Un Modelo de Referencia en ingeniería de sistemas y en ingeniería de software es un modelo de algo que contiene un objetivo o idea básica de algo, y que se puede establecer como una referencia para múltiples propósitos.

Modelo: Representación gráfica, matemática (simbólica) o verbal; o versión simplificada de un concepto, fenómeno, relación, estructura, sistema o algún aspecto del mundo real.

Modelo de madurez de Capacidad Integrado (CMMi): Es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. (SEI, 2012)

Proceso de Software Personal (PSP): Proceso Operativo Personal para desarrollar software. (SEI, 2012)

Proceso de Software de Equipos (TSP): Marco de trabajo enfocada a procesos, para ayudar a los equipos de software, a mejorar su habilidad para producir software de alta calidad en los tiempos y costos comprometidos. (SEI, 2012)

Calidad: El grado con el cual un sistema, componente o proceso satisface las expectativas del usuario o cliente. (IEEE, 2012)

Calidad de Software: Capacidad de un producto de software para satisfacer las necesidades explícitas e implícitas cuando se usa en determinadas condiciones. (ISO/IEC 25000, 2005)

Control de Calidad: Un conjunto de actividades diseñadas para evaluar la calidad de los productos desarrollados o fabricados. (ISO 9001, 2015)

Aseguramiento de la Calidad: Un conjunto de actividades diseñadas para evaluar el proceso por el cual los productos son desarrollados o fabricados. (ISO/IEC 12207, 2008)

Sistema de Gestión de la Calidad: Sistema para dirigir y controlar una organización con respecto a la calidad. (ISO 9001, 2015)

Actividad: Cualquier paso dado o función realizada, tanto mental como física, para lograr algún objetivo. (SEI, 2012)

Proceso: Sucesión de pasos realizados con un propósito dado. Conjunto de actividades, técnicas, herramientas, notación y participantes, utilizados para generar un modelo o producto preestablecido. (IEEE, 2012)

Proceso de desarrollo de software: (1) Proceso a través del cual los requerimientos de usuario son traducidos en especificaciones funcionales, las especificaciones funcionales en especificaciones de diseño, las especificaciones de diseño en código, el cual es testeado, documentado y liberado para ser usado por el usuario. (IEEE). (2) Conjunto de actividades, métodos, prácticas y transformaciones para desarrollar y mantener software y sus productos asociados, tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requisitos del cliente. (SEI, 2012).

Ciclo de vida del Software: Período de tiempo que comienza cuando un producto de software es concebido y termina cuando el software no está más disponible para su uso. (IEEE, 2012)

Desarrollo: Conjunto de procesos que permiten la construcción e instalación de un sistema.

Estándar: Requerimientos obligatorios empleados e impuestos para establecer un enfoque disciplinado y uniforme al desarrollo de software. (IEEE, 2012)

Herramienta: Instrumento utilizado para facilitar una tarea o actividad.

Mantenimiento: Proceso de modificar un sistema de software o componente después de su liberación, para corrección de errores, mejora de rendimiento u otros atributos o adaptación a un entorno cambiante. (IEEE, 2012)

Mapa de actividades: Especifica aquellas actividades seleccionadas para su ejecución, de una metodología estándar, según las particularidades de un proyecto.

Método: Conjunto de reglas y criterios razonables que establecen una forma precisa y repetible de ejecutar una tarea y llegar al resultado deseado. (SEI, 2012)

Metodología: Colección de métodos, procedimientos y estándares que definen una síntesis integrada de enfoques de ingeniería para el desarrollo de un producto. (SEI, 2012)

Notación: Representación formal utilizada para modelar información.

Participante: Individuo responsable o corresponsable de llevar a cabo una actividad o tarea.

Procedimiento: Descripción escrita de un curso de acción a ser tomado para ejecutar una tarea dada. (SEI, 2012).

Productividad: La productividad es la relación entre la cantidad de productos obtenida por un sistema productivo y los recursos utilizados para obtener dicha producción.

Producto de software: Conjunto completo, o cualquiera de los ítems individuales del conjunto, de programas de computadora, procedimientos y documentación asociada y datos, designados para liberar a un cliente o usuario final. (SEI, 2012)

Producto de trabajo de software: Cualquier artefacto creado como parte de la definición, mantenimiento, o uso de un proceso de software. Pueden incluir descripciones de procesos, planes, procedimientos, programas de computadora y documentación asociada, los cuales pueden ser liberados o no a un cliente o usuario final. (SEI, 2012)

Proyecto: Emprendimiento que requiere un esfuerzo convenido que se enfoca en desarrollar, y/o mantener un producto específico. El producto puede incluir hardware,

software, y otros componentes. Típicamente un proyecto tiene su propio financiamiento, contabilidad de costos y cronograma de productos entregables. (SEI, 2012)

Proyecto de software: Emprendimiento que requiere un esfuerzo convenido y enfocado a analizar, especificar, diseñar, desarrollar, testear y/o mantener los componentes de software y la documentación asociada de un sistema. (SEI, 2012)

Postmorten: Es la revisión final de los datos obtenidos, el análisis y tratamiento de los PIP (Processes Improvement Proposal), la revisión de las lecciones aprendidas, la retroalimentación obtenida de usar los procesos de TSP, y demás elementos de cierre de un proyecto TSP.

Tareas: (1) Sucesión de instrucciones tratadas como una unidad básica de trabajo. [IEEE]. (2) Unidad bien definida de trabajo en el proceso de software que proporciona un punto de control visible a la dirección en el estado del proyecto. Las tareas tienen criterios de comienzo (precondiciones) y criterios de finalización (post-condiciones). (SEI, 2012)

Técnica: Método o procedimiento específico para resolver un aspecto del desarrollo o mantenimiento.