

# Aseguramiento de la Calidad en el Proceso de Desarrollo de Software utilizando CMMI, TSP y PSP

Alex Espejo Chavarría<sup>1</sup>, Sussy Bayona Oré<sup>1</sup>, Carlos Pastor<sup>1</sup>

[alex.espejo.c@gmail.com](mailto:alex.espejo.c@gmail.com), [sbayonao@hotmail.com](mailto:sbayonao@hotmail.com), [cpastor@gmail.com](mailto:cpastor@gmail.com)

<sup>1</sup> Unidad de Posgrado de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos, Cercado de Lima, Lima 01, Lima, Perú

DOI: 10.17013/risti.20.62-77

**Resumen:** En la actualidad las empresas que desarrollan 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. Sin embargo, aún existen problemas en los proyectos de desarrollo de software como la calidad. En este artículo se describe los resultados de la implementación de un modelo de aseguramiento de la calidad que integra CMMI (Capability Maturity Model Integration), TSP (Team Software Process) y PSP (Personal Software Process) en una organización desarrolladora de software. El propósito es mejorar la calidad del proceso de desarrollo y del producto software. La metodología de investigación es cuantitativa y el diseño de investigación es cuasi-experimental de pre test y post test. Un total de 22 proyectos software fueron seleccionados para evaluar las mejoras del modelo. Los resultados muestran que existen mejoras significativas en la reducción de costes de calidad y la cantidad de defectos. Asimismo, la rentabilidad del proyecto y la satisfacción del cliente se incrementan.

**Palabras-clave:** cmmi; tsp; psp; aseguramiento de calidad; software.

## *Quality Assurance in the Software Development Process using CMMI, TSP and PSP*

**Abstract:** Nowadays, the enterprises that develop software have become strategic partners for their clients through value generation, by delivering products that support different business processes. However, there are still problems in software developing projects, like quality. This paper describes the results of the implementation of a quality assurance model that integrates CMMI (Capability Maturity Model Integration), TSP (Team Software Process) and PSP (Personal Software Process), with the objective to improve the quality of the process and of the software product delivered. The research methodology is a quantitative one, and the investigation design is quasi-experimental about pre-test and post-test. A total number of 22 software projects have been selected to evaluate the model improvements. The results show that there are important improvements in reducing the quality costs and the number of defects. At the same time, the project profitability and customer satisfaction is also increased.

**Keywords:** cmmi; tsp; psp; calidad, software.

## 1. Introducción

Hoy en día son cada vez más las organizaciones que incorporan Tecnologías de información como soporte a las actividades de sus unidades de negocio, en el deseo que les permita crear valor y mejorar su competitividad (Laureano & Rocha, 2015). Por un lado, demandan productos software de calidad (Piattini et al., 2014) y por otro exigen el uso de estándares internacionales que aseguren la entrega de productos software de calidad. Para cumplir con estas exigencias las organizaciones desarrolladoras de software no solo deben asegurar la calidad del producto software, sino que además deben asegurar la calidad del producto software (Rocha & Freixo, 2015).

Diversos organismos especializados como International Standardization Organization (ISO), Software Engineering Institute (SEI) han desarrollado estándares, buenas prácticas y metodologías. Podemos mencionar a ISO/IEC 9126 (ISO 9126, 1991) orientado al producto software, CMMI (Capability Maturity Model Integration) (Chrissis, Konrad & Shrum, 2011), TSP (Team Software Process) (Software Engineering Institute, 2010b) y PSP (Personal Software Process) (Software Engineering Institute, 2010a). CMMI para desarrollo consiste de las mejores prácticas que dirige las actividades de desarrollo aplicado a productos y servicios (Chrissis, Konrad & Shrum, 2011), TSP se enfoca en equipos de alto rendimiento para desarrollar software de calidad y PSP ayuda en la gestión del tiempo y mejora de la productividad personal. Sin embargo, a pesar de que se han desarrollado modelos, estándares y métodos para mejorar la calidad del proceso software, los proyectos de desarrollo de software no alcanzan los objetivos planificados respecto a su línea base de alcance, coste, tiempo, calidad y satisfacción de los clientes. Según el informe Chaos (The Standish Group Report, 2015) sobre proyectos software, solo un 29% de los proyectos fueron exitosos, 52% de los proyectos fueron cuestionados o sufrieron cambios, y el 19% de proyectos fracasaron (cancelados antes de su finalización o entregados y nunca se usaron). Entre las causas mencionadas de la baja tasa de proyectos software exitosos se encuentra en primer lugar el haber tratado a la programación por su enfoque científico y a veces artístico, y en segundo lugar los cambios de los requerimientos (Sommerville, 2010). Además, las propuestas de métodos, metodologías y estándares solo se enfocan en una parte del negocio y no se presenta como una propuesta integral.

Según Shem et al. (2015) generalmente existe una correlación entre la implementación de estándares y calidad de producto. El concepto de calidad está relacionado con un conjunto de características del producto software y la satisfacción de las necesidades de los usuarios (ISO 9126, 1991). Según Fenton & Bieman (2014) la calidad de software se puede medir en términos de densidad de defectos. Por ello las empresas dedicadas a la producción de software deben establecer los mecanismos de control que permitan determinar que su producto cumpla con normas, procesos y estándares de calidad, que garanticen que su producto esté libre de errores (Puello et al., 2016).

En este artículo se presenta un caso de estudio de la aplicación de un modelo para mejorar la calidad del proceso de desarrollo software en una empresa desarrolladora de software que ha sido certificada en CMMI. Con un enfoque en la mejora continua, la importancia

del talento humano, los procesos y tecnología en el desarrollo de software, la empresa ha considerado oportuno integrar las metodologías PSP y TSP y el modelo CMMI. El modelo propuesto tiene como propósito asegurar la calidad del proceso de desarrollo de software mediante la integración de PSP y TSP. Algunas razones para adoptar TSP y PSP e integrarlo a CMMI son: (1) TSP se basa en PSP mediante la aplicación de procesos similares para el equipo, (2) PSP y TSP ayudan a construir una cultura organizacional de calidad, (3) TSP reduce la resistencia al cambio, (4) los roles de Proyecto de TSP se ajustan a los roles de CMMI (Cedillo, 2005), (5) cuando se utiliza TSP los programadores y diseñadores tienen la intención de monitorear lo que están haciendo, identificar sus debilidades y mejorar ellos mismos (Britton, 2015) y (6) TSP está enfocado en las personas (Bayona et al., 2012). Adicionalmente, al integrar CMMI, TSP y PSP se mejora los procesos software (Zhang, 2010). Para validar el modelo propuesto se plantearon tres actividades importantes como son la formación, lanzamiento y seguimiento del modelo propuesto. Se brindó capacitación sobre los fundamentos de las metodologías PSP, TSP e integración con CMMI donde resaltan los siguientes temas: fundamentos de TSP, análisis y mejora, métricas de planeamiento y calidad, formación de equipos, integración TSP, PSP con CMMI y explicación de las herramientas. Se definieron objetivos grupales, individuales y por roles, se asignaron los roles. Asimismo, se ejecutó la primera reunión de grupo, el establecimiento de cronograma para las reuniones semanales y la especificación del cuaderno del proyecto.

Este artículo ha sido estructurado en cinco secciones. En la sección 2 se presenta la fundamentación teórica, En la sección 3 se presenta la propuesta del modelo de aseguramiento de calidad que integra CMMI, PSP y TSP. En la sección 4 se presenta la metodología usada para validar el modelo. En la sección 5 se presenta los resultados de la implementación y se comprueba las hipótesis planteadas. Finalmente, en la sección 6 se presenta las conclusiones.

## 2. Marco Teórico

En la presente sección se detallan los componentes de un proceso de desarrollo de software para asegurar la calidad del proceso. En la Figura 1 se muestra los tres componentes “Proceso, Tecnología y Personas” del proceso de desarrollo y como una adecuada integración soportada en modelos y metodologías aportará calidad al proceso, generando valor y permitiendo alcanzar los objetivos.



Figura 1 – Integración de los componentes de un procesos de desarrollo de software

## **2.1. Capability Maturity Model Integration - CMMI**

CMMI 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 Product Team, 2010). Su enfoque es la mejora de procesos, que provee a las organizaciones de los elementos esenciales para un proceso efectivo y provee guía para la calidad de los procesos (Altwies & Preston, 2016). CMMI fue desarrollado por el Software Engineering Institute (SEI).

El objetivo de CMMI es crear productos de calidad que permita obtener resultados medibles y repetibles (Conrad, Misenar & Feldman, 2016). CMMI mide la madurez del proceso de desarrollo del software en una escala del 1 al 5 e 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.

El modelo CMMI consta de 22 áreas de proceso. Las áreas de proceso de Nivel 2 y 3 son las siguientes: 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). CMMI es utilizado en más de 5000 empresas (SEI, 2015) de las cuales 24 son empresas ubicadas en el Perú. En su gran mayoría las empresas peruanas se encuentran en nivel 2 y 3.

## **2.3. Personal Software Process - 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 (Humphrey, 2005). 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. Tiene como 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. PSP ayuda al ingeniero a estimar y planificar su trabajo, lograr sus compromisos y responder de la mejor manera ante un trabajo bajo presión.

Los principales objetivos de PSP son: maximizar calidad del software, lograr una disciplina de mejora continua en el proceso de desarrollo, mejorar la calidad del proceso de desarrollo e incrementar la productividad. Considerando que calidad se basa en la medición (Britton, 2016). PSP integra un conjunto de formularios, que también son usados por PSP para recoger las métricas necesarias.

Según Humphrey (2000) son tres las métricas básicas que establece PSP tales como tamaño tiempo y defectos y las demás son medidas derivadas de estas métricas básicas. Defectos es una métrica relacionado a la calidad de software y al utilizar PSP se puede reducir los defectos (Wesslén, 2000).

## 2.2. Team Software Process - TSP

TSP es un proceso de desarrollo que enfatiza en calidad y métricas. Un proyecto de software de TSP se desarrolla a través de una serie de ciclos de desarrollo, donde cada ciclo comienza con un proceso de planificación llamado lanzamiento y termina con un proceso de cierre llamado postmortem (Kazman & Cervantes, 2016). El uso de TSP en los proyectos de desarrollo software permite que los proyectos sean entregados a tiempo, dentro del presupuesto y con calidad (Conrad, Misener & Feldman, 2016).

TSP ayuda a planificar y gestionar el equipo de un proyecto software y está basado en cuatro principios básicos: (1) aprender es más efectivo cuando sigues un proceso definido, (2) el trabajo en equipo requiere de una combinación de metas específicas, un ambiente de trabajo de apoyo, liderazgo y entrenamiento capaces, (3) aprecia los beneficios de las prácticas de desarrollo y (4) la formación es más eficaz cuando se dispone de conocimiento previo (Humphrey, 1999).

Los principales objetivos de TSP son: maximizar calidad Software, integrar equipos independientes de alto rendimiento que planifiquen, registren su trabajo, establezcan metas, monitorear y motivar a sus equipos de trabajo, acelerar la mejora continua de procesos y proveer de una guía para el mejoramiento en organizaciones maduras.

Diversos estudios demuestran que aplicación de TSP mejora la estimación de tamaño y esfuerzo que son métricas fundamentales en el proyecto software. Asimismo, se reducen los defectos encontrados en las diferentes fases del ciclo de vida del proyecto software (Davis & Mullaney, 2003; Bayona et al, 2008; Shurei, 2009)

## 3. Modelo de Aseguramiento de Calidad

El modelo propuesto tiene como propósito asegurar la calidad en el proceso de desarrollo de software con un enfoque en el talento humano. Considera que las personas que realizan las actividades deben comprometerse con el trabajo y con la calidad. Resalta



Figura 2 – Modelo de aseguramiento de la calidad alto nivel

la importancia de la formación de los miembros del equipo como instrumento para sensibilizar y seguir un proceso de desarrollo con criterios de calidad para producir productos de calidad. El modelo propuesto consta de tres componentes a mencionar: procesos, prácticas y herramientas PSP, Procesos y Prácticas TSP y Procesos CMMI. En la Figura 2 se presenta el modelo de aseguramiento de la Calidad Alto Nivel.

### 3.1. Procesos, prácticas y herramientas PSP

Los procesos, prácticas y herramientas PSP se diseñaron 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. Proporcionando 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.

- **Proceso de medición personal:** Proceso que permite recoger sobre el trabajo (tiempo gastado, cantidad de defectos, etc.), establecer una base que incluye mediciones y formatos de reportes para medir progreso y definir los cimientos de mejora.
- **Proceso de planificación personal:** Proceso que facilita las estimaciones de tamaño y recursos, cronograma y seguimiento del proyecto, y enseña a realizar compromisos que se pueden cumplir.
- **Administración de calidad personal:** Proceso para mejorar la habilidad del ingeniero para producir software de calidad de manera natural y consciente y explotar habilidades inherentes y capacidades.
- **Herramientas:** Formato de estimación semanal de actividades, formato de registro de tiempos, formato de registro de defectos y formato resumen plan de proyecto.

### 3.2. Procesos y prácticas TSP

- **Proceso de software personal:** Procesos que manejan la mejora continua de las habilidades de los ingenieros de software (primer componente).
- **Proceso para construir equipos:** Permite identificar condiciones mínimas de un equipo, construir equipos eficaces y cohesionados en entornos compartidos y comunes, estimar el tamaño del producto en función a la metodología y complejidad, y negociar los compromisos finales.
- **Proceso para trabajar en equipos:** Permite recolectar datos precisos y exactos del trabajo, analizar datos para identificar problemas con anticipación, encontrar soluciones en equipo, comunicación del equipo y análisis de riesgos.
- **Herramientas:** Formato de registro de intereses y experiencias, formato de registro de decisiones estratégicas, formato de reporte semanal y formato de evaluación de equipo y roles.

### 3.3. Procesos CMMI

- **Procesos:** Para la administración básica de proyectos y para la estandarización de procesos. Las áreas de procesos considerados fueron los de nivel 2 y 3 de madurez.

- **Herramientas:** 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 y checklist de entrega.

#### 4. Metodología

La selección del caso de estudio se realizó en función a las facilidades de acceso a la información y respaldo de los gestores de proyecto en la integración y seguimiento del modelo de aseguramiento de la calidad.

El objetivo del caso de estudio fue aplicar el modelo de aseguramiento de calidad propuesto en la sección 3, que integra las mejores prácticas, procesos y métricas de PSP, TSP y CMMI. La empresa en que se aplicó el modelo de aseguramiento de calidad es una empresa global de consultoría en gestión, servicios de tecnología y outsourcing.

La unidad de análisis se encuentra representada por los proyectos de desarrollo de software de la empresa donde se integró el modelo de aseguramiento de la calidad al proceso de desarrollo. Se seleccionaron 22 proyectos software correspondiente a la consultora, de los cuales 11 proyectos (grupo pos-test) integraron el modelo de aseguramiento de calidad a su proceso de desarrollo y los 11 proyectos restantes se desarrollaron según su proceso actual. En la Tabla 1 se presenta los criterios de selección de los proyectos software.

<b>Criterio</b>	<b>Descripción del Criterio</b>
<i>Unidad de negocio</i>	Seguros generales (EPS, SOAT, AUTOS)
<i>Plataforma tecnologica</i>	Microsoft .NET y Oracle
<i>Duración del proyecto</i>	3 meses en promedio
<i>Ciclo de vida</i>	Reporte de Especificaciones Funcionales, Reporte de Especificaciones Técnica, Arquitectura y Construcción, Certificación, Implantación y Seguimiento.
<i>Entidad certificadora del producto</i>	Consultora Itera Process

Tabla 1 – Criterios de selección de los proyectos software

El tipo de investigación es cuantitativo y diseño pre-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 (O1), luego el grupo se expone al tratamiento (X) y finalmente se toma una medición posterior (O2). El efecto del tratamiento se calcula como  $O2 - O1$ . (Lerma, 2009).

##### 4.1. Hipótesis y Variables

La Hipótesis General que se plantea para validar el modelo es:

*HG: La integración del modelo de aseguramiento de la calidad basado en CMMI, TSP y PSP permite asegurar la calidad en el proceso de desarrollo de software.*



La hipótesis general será contrastada a través de sus hipótesis específicas asociadas a cada dimensión e indicador. Las Hipótesis Específicas son:

*HE.1.: La integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software reduce el costo de calidad en los proyectos.*

*HE2.: La integración del modelo de aseguramiento de la calidad en el proceso de desarrollo de software reduce los defectos en los proyectos.*

Variable	Tipo de Variable	Dimensión	Indicador
Modelo de aseguramiento de la calidad	Independiente	-	-
Calidad en el proceso de desarrollo de software	Dependiente	Costo de calidad	% de Costo de Calidad (Costo de Conformidad + Costo de No Conformidad)
		Defectos en el Proyecto	Cantidad de Defectos en el Proceso de Desarrollo de Software
		Rentabilidad del Proyecto	% de desviación de costo respecto a la línea base.
		Satisfacción del Cliente	% de clientes satisfechos.

Tabla 2 – Definición de variables

*HE.3.: La integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software incrementa la rentabilidad de los proyectos.*

Indicador	Unidad de Medida	Instrumento de Medición	Formula
Porcentaje de costo de calidad	Porcentaje	Ficha de registro de datos	$PCC = CC/CR * 100$ Donde: PCC = Porcentaje de Costo de Calidad CC = Costo de Calidad CR = Costo Real del Proyecto
Cantidad de defectos en el proceso de desarrollo de software	Cantidad	Ficha de registro de datos	$CDP = \sum(DE)$ Donde: CDP: Cantidad defectos en el proyecto DE: Defecto
Porcentaje de desviación de costo respecto a la línea base	Porcentaje	Ficha de registro de datos	$PVC = (1 - (CR/CP)) * 100$ Donde: PVC = Porcentaje de Variación de Costo CR = Costo Real CP = Costo Planificado
Porcentaje de clientes satisfechos	Porcentaje	Cuestionario	$PCS = (TC-CI) / TC * 100$ Donde: PCS = Porcentaje de Clientes Satisfechos TC = Total Clientes Proyecto CI = Clientes que presentaron inconformidad

Tabla 3 – Definición de indicadores de la variable dependiente



*HE.4.: La integración del modelo de aseguramiento de calidad en el proceso de desarrollo de software aumenta el grado de satisfacción de los clientes.*

Las variables, tipo de variable, dimensión e indicador se detallan en la Tabla 2.

En la Tabla 3 se detalla los atributos de los indicadores de la variable dependiente (indicador, unidad de medida, instrumento de medición y la fórmula de cálculo) los cuales nos ayudan en la recolección de datos y en contrastar la hipótesis.

#### 4.2. Recolección de Datos

Para la recolección de datos se usaron las técnicas de análisis documental y encuesta. Los instrumentos asociados a cada indicador fueron la ficha de registro de datos y el cuestionario respectivamente. Los datos recolectados fueron procesados y analizados. Los resultados se presentan en la sección 5.

Variable	Dimensión	Indicador	Unidad de Medida	Técnica	Instrumento
<i>Calidad en el proceso de desarrollo de software</i>	Costo de Calidad por Proyecto	Porcentaje de costo de calidad	Porcentaje	Análisis documental	Ficha de registro de datos
	Defectos por Proyecto	Cantidad de defectos	Cantidad	Análisis documental	Ficha de registro de datos
	Rentabilidad del Proyecto	Porcentaje de desviación de coste	Porcentaje	Análisis documental	Ficha de registro de datos
	Satisfacción del cliente	Porcentaje de clientes satisfechos	Porcentaje	Encuesta	Cuestionario

Tabla 4 – Instrumentos de medición asociados al indicador

## 5. Resultados

En esta sección se presenta un análisis comparativo entre dos grupos de proyectos cada uno integrado por 11 proyectos. El grupo pre-test integrado por proyectos con su proceso de desarrollo sin utilizar el modelo y el grupo post-test integrado por los proyectos que adoptaron el modelo de aseguramiento de la calidad en su proceso de desarrollo. El comparativo se realiza en función a los indicadores mostrados en la Tabla 3.

### 5.1. Indicador Porcentaje de Costo de Calidad

En la Figura 3 se muestra la comparación entre los dos grupos de proyectos.

Se observa que los proyectos con mayor porcentaje de costo de calidad están en el grupo pre-test con un máximo de 19.85% versus el grupo post-test donde se obtiene un máximo de 7.41% de costo de calidad, respecto al costo real del proyecto.

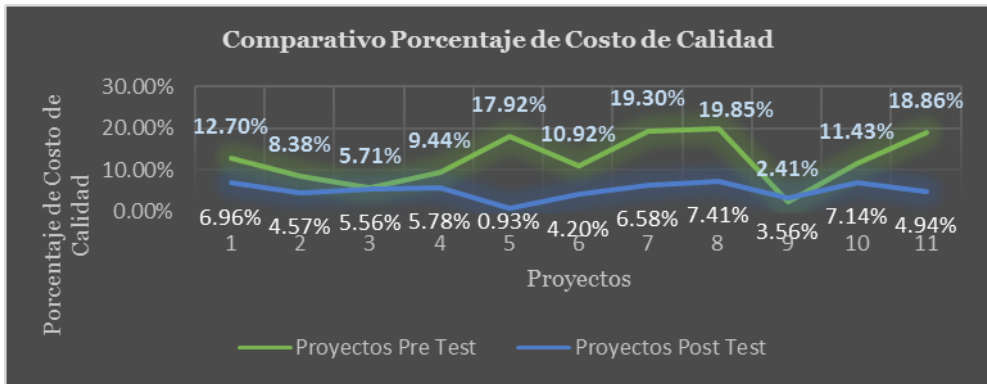


Figura 3 – Comparativo indicador porcentaje de costo de calidad

### 5.2. Indicador Cantidad de Defectos en el Proceso de Desarrollo

En la Figura 4 se muestra la comparación entre los dos grupos de proyectos. Se observa que los proyectos con mayor cantidad de defectos están en el grupo pre-test con una cantidad máxima de 249 defectos versus el grupo post-test, donde se obtiene una cantidad máxima de 95 defectos. Es preciso indicar que la cantidad de defectos 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 e implantación y seguimiento). Es decir, cuando se enfoca en la calidad durante el proceso de desarrollo de software se puede detectar los defectos en etapas tempranas, evitando el retrabajo. Uno de los principios de TSP, es que para maximizar la productividad, el primer foco de atención debe ser la calidad, y la calidad se debe hacer en todas las fases del desarrollo del proyecto para minimizar la cantidad de defectos en el proyecto.

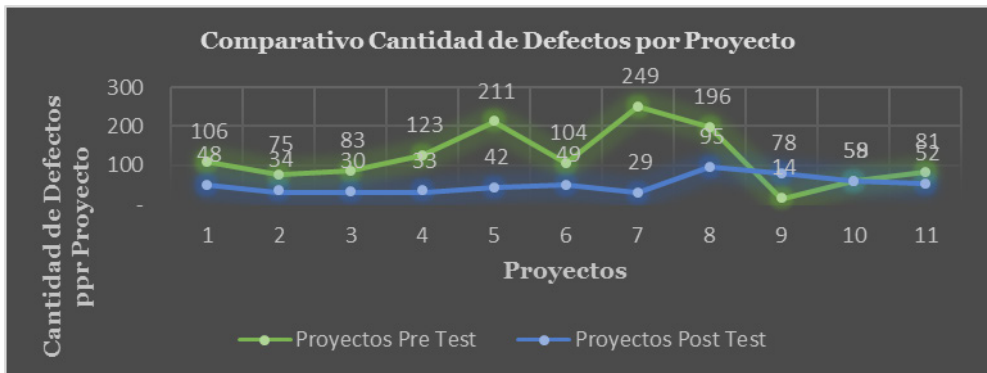


Figura 4 – Comparativo indicador cantidad de defectos por proyecto

### 5.3. Indicador Porcentaje de Desviación de Costo

En la Figura 5 se presenta la comparación entre los dos grupos de proyectos; en el análisis descriptivo se observa que los proyectos con mayor porcentaje de desviación de

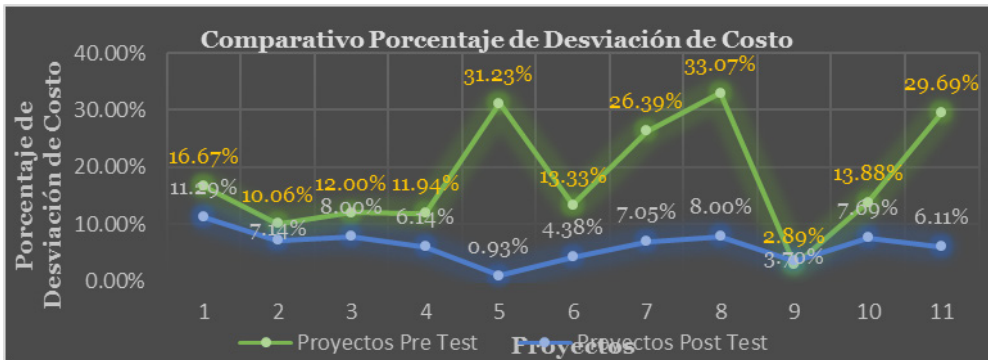


Figura 5 – Comparativo indicador porcentaje de desviación de costo

costo están en el grupo (pre-test) con una porcentaje máximo de 33.07% versus el grupo (post-test) donde se obtiene una porcentaje máximo de 11.29%.

Según la filosofía de Deming, cuando las organizaciones se concentran en calidad, en un periodo de tiempo los costos tienden a bajar. Al respecto, con la aplicación del modelo se ha logrado reducir la desviación de los costos.

#### 5.4. Indicador Porcentaje de Satisfacción del Cliente

En la Figura 6 se presenta la comparación entre los dos grupos de proyectos respecto al grado de satisfacción de cliente. En el análisis descriptivo se observa que el 46.97% de clientes de los proyectos se encuentran insatisfechos correspondiente a los proyectos del grupo (pre-test) versus un 48.48% de clientes satisfechos que corresponden al grupo (post-test).

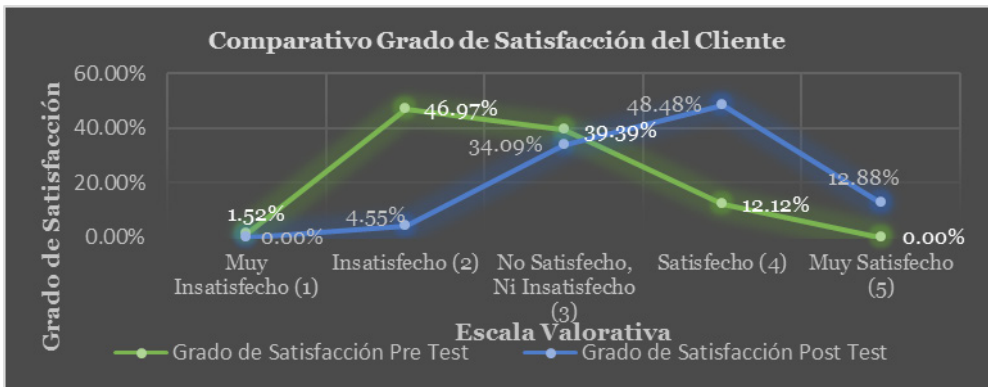


Figura 6 – Comparativo indicador grado de satisfacción del cliente

La calidad está relacionada con dos términos “requerimientos” y “satisfacción del cliente”, por lo tanto entregar productos software de calidad incrementa la satisfacción del cliente.

### 5.5. Prueba de hipótesis

En la presente sección se resumen los resultados obtenidos luego de contrastar la hipótesis formulada. Para una mejor comprensión de discuten los resultados por cada indicador. Se utiliza la distribución t-student con muestras independientes para realizar la prueba de hipótesis.

#### 5.5.1. Indicador Porcentaje de Costo de Calidad

En la Figura 6 se presentan los porcentajes promedios de costo de calidad correspondientes a los grupos de pre-test y post-test, donde nos ilustra una reducción en promedio de 7.19% de costo de calidad respecto al costo real del proyecto, logrando reducir de 12.47% obtenido en el grupo (pre-test) a 5.25% obtenido en el grupo (post-test).

Al realizar la prueba t-student se concluye que 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-VALOR  $\leq \alpha = 0.002 \leq 0.05$ ).

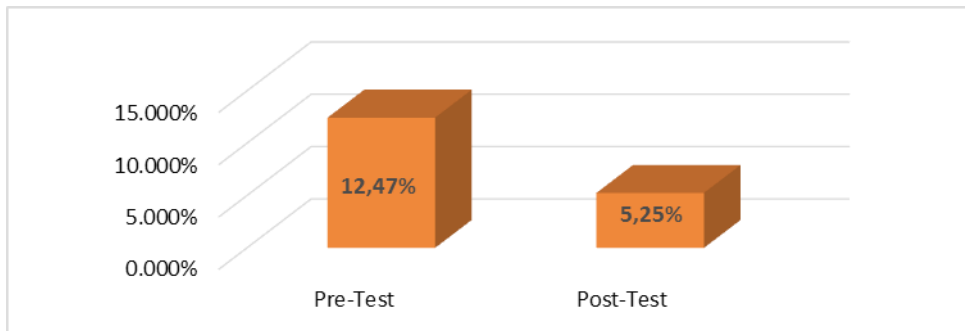


Figura 6 – Porcentaje promedio de costo de calidad

#### 5.5.2. Indicador Cantidad de Defectos por Proyecto

En la Figura 7 se presentan las cantidades promedios de defectos por proyecto correspondientes a los grupos de pre-test y post-test, donde se observa una reducción en

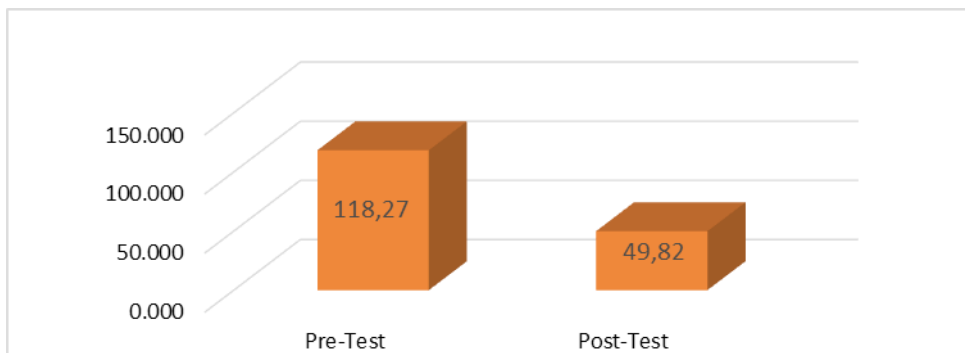


Figura 7 – Cantidad promedio de defectos por proyecto

promedio de 68.45 defectos por proyecto. Se logra reducir de 118.27 defectos obtenido en el grupo (pre-test) a 49.82 defectos obtenido en el grupo (post-test).

Al realizar la prueba t-student se concluye que 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-VALOR  $\leq \alpha = 0.01 \leq 0.05$ ).

### 5.5.3 Indicador Porcentaje de Desviación de Costo

En la Figura 8 se presentan los porcentajes promedios de desviación de costo correspondientes a los grupos de pre-test y post-test, donde se observa una importante reducción de 11.88% (en promedio) en el porcentaje de desviación de costo, logrando disminuir de 18.286% obtenido en el grupo (pre-test) a 6.40% obtenido en el grupo (post-test).

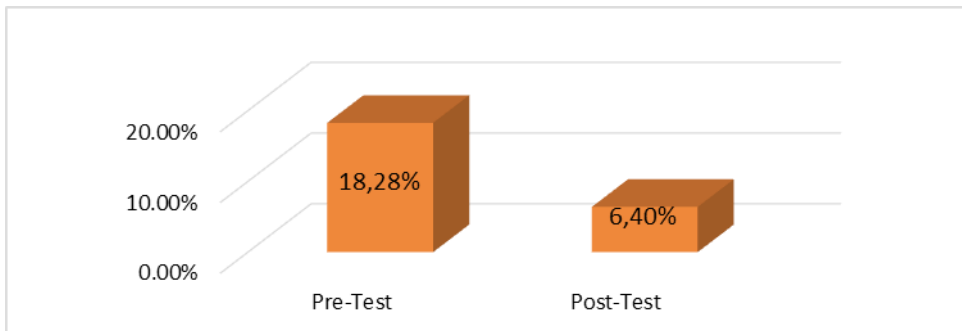


Figura 8 – Porcentaje promedio de desviación de costo

Al realizar la prueba t-student se concluye que 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-VALOR  $\leq \alpha = 0.03 \leq 0.05$ ).

### 5.5.4 Indicador Porcentaje de Clientes Satisfechos

En la Figura 9 se presentan los porcentajes promedios de satisfacción del cliente correspondientes a los grupos de pre-test y post-test, donde se observa un considerable incremento de 22.423% (en promedio) de cliente satisfechos, logrando incrementa de

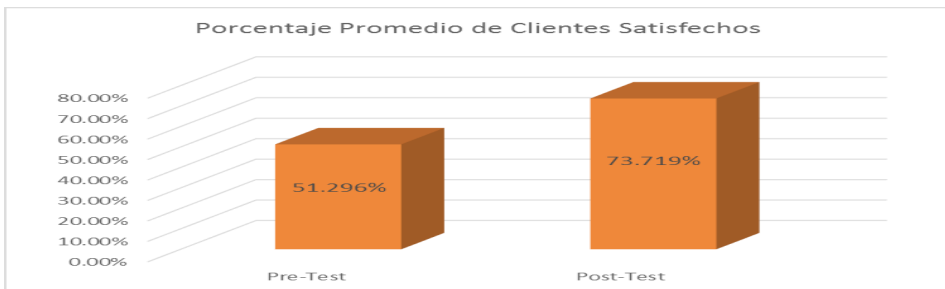


Figura 9 – Porcentaje promedio de clientes satisfechos

51.296% obtenido en el grupo (pre-test) a 73.719% obtenido en el grupo (post-test). Al realizar la prueba t-student se concluye que existe una diferencia significativa en el porcentaje de clientes satisfechos con la integración del modelo de aseguramiento de la calidad ( $P\text{-VALOR} \leq \alpha = 0.00 \leq 0.05$ ).

## 6. Conclusiones

Las organizaciones invierten dinero para incorporar TI como soporte a sus unidades de negocio. Cuando requieren de un producto software, calidad del producto software y uso de los estándares internacionales en el desarrollo del producto, son exigencias que la empresa desarrolladora de software debe tomar en cuenta. Existen diversas propuestas de metodologías como PSP, TSP y modelos como CMMI orientadas a obtener productos de calidad, pero estas no están integradas. En este artículo se presenta un caso de estudio de la propuesta y aplicación de un modelo de aseguramiento de calidad que integra CMMI, TSP y PSP. El modelo integra las buenas prácticas, procedimientos, formularios con el propósito de obtener productos de calidad. El modelo ha sido implementado en una empresa desarrolladora de software peruana. Un total de 22 proyectos fueron seleccionados, 11 corresponden al pre-test y 11 al post-test. Los resultados fueron evaluados en cuatro dimensiones: costo de calidad, defectos, rentabilidad y satisfacción del cliente. Con la implementación del modelo de aseguramiento de la calidad se logra reducir el coste de calidad, reducir los defectos, la desviación de coste y se incrementa la satisfacción del usuario. Como resultado de la reducción significativa en la desviación de costo de los proyectos se incrementa la rentabilidad en cada uno de los proyectos. En este artículo se ha revisado el impacto en los proyectos y como trabajo futuro se plantea evaluar la mejora en los miembros de los equipos de proyectos en aspectos relacionados a la comunicación, participación, trabajo en equipo, conocimientos entre otros.

## Referencias

- Altwies, D. & Preston, J. (2016). *Achieve PMP® Exam Success*, 5th Edition, J. Ross Publishing
- Bayona, S., Calvo Manzano, J., Gonzalo, C., & San Feliu, T. (2008). Teaching Team Software Process in Graduate Courses to Increase Productivity and Improve Software Quality. 32nd Annual IEEE International Computer Software and Applications Conference (pp. 440–446). Turku: Ieee. doi:10.1109/COMPSAC.2008.135
- Bayona, S., Calvo-Manzano, J., Cuevas, G., San Feliu, T. (2012) *Method for selecting a reference model for software process deployment*, Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on, pp. 1-6
- Britton, C. (2015). *Designing the Requirements: Building Applications that the User Wants and Needs*, Addison-Wesley Professional, pp. 76
- Cedillo, K. (2005). Accelerating CMMI Implementation with PSP and TSP in a Small Organization, SEPG 2005 [https://resources.sei.cmu.edu/asset\\_files/Presentation/2005\\_017\\_001\\_22688.pdf](https://resources.sei.cmu.edu/asset_files/Presentation/2005_017_001_22688.pdf)

- Chrissis, M., Konrad, M & Shrum, S. (2011). *CMMI for Development®: Guidelines for Process Integration and Product Improvement*, Third Edition. Addison-Wesley Professional.
- CMMI Product Team (2010). *CMMI for Development, Version 1.3*. CMU/SEI-2010-TR-033. Software Engineering Institute, Carnegie Mellon University. 2010. [http://resources.sei.cmu.edu/asset\\_files/TechnicalReport/2010\\_005\\_001\\_15287.pdf](http://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf)
- Conrad, E., Misenaar, S. & Feldman, J. (2016) *Eleventh Hour CISSP®*, Syngress
- Davis, N. & Mullaney, J. (2003). The Team Software Process (TSP) in Practice: A Summary of Recent Results. *CMU/SEI- 2003-TR-014, Software Engineering Institute, Carnegie Mellon University, Pittsburgh*
- Fenton, N. & Bieman, J. (2014). *Software metrics: a rigorous and practical approach*, CRC Press.
- Humphrey W. & Over J. (1999) *Introduction to the Team Software Process(SM)*. Addison-Wesley Professional
- Humphrey, W. (2000). The personal software processSM (PSPSM), *Technical Report CMU/SEI-2000-TR-022*
- Humphrey, W. (2005). *PSP: A Self-Improvement Process for Software Engineers*. Reading, MA, Addison Wesley
- ISO/IEC International Standard 9126 (1991). *Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for their use*, International Standards Organization.
- Kazman, R. & Cervantes, H. (2016). *Designing Software Architectures: A Practical Approach*. Addison-Wesley Professional.
- Laureano, R. & Rocha, A. (2015) Information Technology governance and informatics tools for audit. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (17), pp. 26-40. DOI: 10.17013/risti.15.vii-ix
- Lerma, H. (2009). *Metodología de la investigación. Propuesta, anteproyecto y proyecto*. (4th ed.) Bogotá: ECOE Ediciones.
- Piattini, M., García, F., García Rodríguez, I. & Pino, F. (2014). *Calidad de Sistemas de Información*. Segunda edición, Madrid: Rama, 2014.
- Puello, J., San Feliu, T. & Calvo-Manzano, J. (2016). Una aproximación basada en metamodelado del área de proceso de Validación del CMMI: Un caso de estudio. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (17), 26–40.
- Rocha, Á., & Freixo, J. (2015). Information Architecture for Quality Management Support in Hospitals. *Journal of Medical Systems*, 39(10), 1–11.
- SEI (2015). Published Appraisal Results, [fecha de consulta: 16 noviembre 2016]. Disponible en <https://sas.cmmiinstitute.com/pars/>.



- She, S, Eddie, L and Kellys, S (2015). *Assessing the role of standards in enhancing the competitiveness of locally manufactured products in Zambia*. Merit Research J. Bus. Manag. 3(2): 16–28.
- Shurei, T. (2009). *Integrating CMMI and TSP/PSP: Using TSP Data to Create Process Performance Models*, [fecha de consulta: 30 julio 2016]. Disponible en <http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=9063>
- Software Engineering Institute (2010a). *The Personal Software Process (PSP) Body of Knowledge* [en línea]. Ver. 2.0. [fecha de consulta: 2 agosto 2016]. Disponible en <http://www.sei.cmu.edu/reports/o9sr018.pdf>
- Software Engineering Institute (2010b). *The Team Software Process (TSP) Body of Knowledge* [en línea]. [fecha de consulta: 3 marzo 2016]. Disponible en <http://www.sei.cmu.edu/reports/10tro20.pdf>
- Sommerville, I. (2010). *Software Engineering*, (9th ed).
- The Standish Group Report (2015). *Chaos Report 2015* [fecha de consulta: 14 febrero 2016]. Disponible en <https://www.infoq.com/articles/standish-chaos-2015>
- Wesslén, A. (2000). A replicated empirical study of the impact of the methods in the PSP on individual engineers. *Empirical Software Engineering* 5 (2) pp. 93–123
- Zhang, L (2010). *Software Process Improvement for Small Organizations Based on CMMI/TSP/PSP*. Recuperado de <http://www.seiofbluemountain.com/upload/product/201001/1264727480s3xvkd4g.pdf>