Universidad Nacional Federico Villarreal

VRIN | VICERRECTORADO DE INVESTIGACIÓN

ESCUELA UNIVERSITARIA DE POSGRADO

EL NUEVO MODELO LACS BASADO EN LEAN IT Y DESARROLLO ÁGIL PARA MEJORAR LA EJECUCIÓN DE PROYECTOS DE IMPLEMENTACIÓN DE SISTEMAS SAP

Línea de investigación:

Sistemas de información y optimización

Tesis para optar el Grado Académico de Doctor en Ingeniería de Sistemas

Autor:

Contreras Sagástegui, Luis Angel

Asesor:

Huamán Samaniego, Héctor

(ORCID: 0000-0003-0761-5000)

Jurado:

Manrique Suárez, Luis Humberto

Mujica Ruiz, Oscar Hugo

Rodriguez Rodriguez, Ciro

Lima - Perú

2023

Reporte de Análisis de Similitud

Archivo:	1A_Contreras_Sagastegui_Luis_Doctorado_2021.docx
Fecha del Análisis:	3/08/2021
Analizado por:	Namo Garcia, Robert Leonel
Correo del analista:	rnamo@unfv.edu.pe
Porcentaje:	10 %
Título:	EL NUEVO MODELO LACS BASADO EN LEAN IT Y DESARROLLO ÁGIL PARA MEJORAR LA EJECUCIÓN DE PROYECTOS DE IMPLEMENTACIÓN DE SISTEMAS SAP
Enlace:	https://secure.urkund.com/old/view/105761461-602127-275757#FcyxCgJBDATQf9l6kCSbZDf3K2Ihh8oWXnOl+O+OMI8wCeTT3mfbrgJVp6RBBTWlgEbBhPlTMvBiyqqdnIISxgs3PtHZenAyBRe4wg0+wGc5MVFxQzvX61jPtd+P/dE2uUhJlxi9MtN8zPr+AA==

UNIVERSIDAD NACIONAL FEDERICO VILLARREAL
OFICINA DE GRADOS Y GESTIÓN DEL EGRESADO
ESCUELA UNIVERSITARIA DE POSGRADO

DRA. MIRIAM LILIANA FLORES CORONADO
JEFA DE GRADOS Y GESTIÓN DEL EGRESADO

ESCUELA UNIVERSITARIA DE POSGRADO

EL NUEVO MODELO LACS BASADO EN LEAN IT Y DESARROLLO ÁGIL PARA MEJORAR LA EJECUCIÓN DE PROYECTOS DE IMPLEMENTACIÓN DE SISTEMAS SAP

Línea de Investigación: Ingenierías y Arquitectura

Para optar el Grado Académico de Doctor en Ingeniería de Sistemas

Autor

Contreras Sagástegui, Luis Angel

Asesor:

Huamán Samaniego, Héctor

ORCID: 0000-0003-0761-5000

Jurado:

Manrique Suárez, Luis Humberto

Mujica Ruiz, Oscar Hugo

Rodriguez Rodriguez, Ciro

Lima – Perú

2023

TÍTULO

EL NUEVO MODELO LACS BASADO EN LEAN IT Y DESARROLLO ÁGIL PARA MEJORAR LA EJECUCIÓN DE PROYECTOS DE IMPLEMENTACIÓN DE SISTEMAS SAP.

AUTOR

LUIS ANGEL CONTRERAS SAGÁSTEGUI

ASESOR:

DR. HÉCTOR HUAMÁN SAMANIEGO

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El propósito de la investigación es determinar en qué medida el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software mejora la ejecución de proyectos de implementación del sistema SAP. La investigación es de diseño cuasi experimental con una estrategia longitudinal que permite estudiar el efecto de cambio del Pre-Test y Pos-Test, que por la naturaleza del trabajo se utilizó una población no aleatoria, comprendida por 10 integrantes entre analistas funcionales, consultores, desarrolladores, director del proyecto y jefe del proyecto. Las variables utilizadas fueron la ejecución de proyecto del Sistema SAP y el nuevo modelo LACS. Los resultados indican que el nuevo modelo LACS mejora la ejecución de proyectos de implementación del sistema SAP. El primer cambio que se obtuvo fue que aumento la cantidad de pruebas unitarias validas en 40%, en segundo lugar, se incrementó la capacitación al personal en 30% y en tercer lugar se incrementó la cantidad de documentación técnica valida en 40%, conforme el avance de las atenciones. Se puede concluir que la hipótesis general ha sido aceptada porque existe la evidencia suficiente para indicarnos que el nuevo modelo LACS mejora significativamente la ejecución de proyectos de implementación del sistema SAP, en las pruebas estadísticas se utilizó chi-cuadrado para demostrar las diferencias en la ejecución del proyecto de implementación antes y después de la aplicación del nuevo modelo LACS.

Palabras claves: metodologías agiles, Modelo LACS, ejecución de proyectos de implementación, desarrollo de software.

ABSTRACT

The purpose of the research is to determine to what extent the new LACS model based on LEAN IT and Agile Software Development will improve the execution of SAP system implementation projects. The research is of a quasi-experimental design with a longitudinal strategy that allows studying the effect of change of the Pre-Test and Pos-Test, which due to the nature of the work, a non-random population was used, comprised of 10 members among functional analysts, consultants, developers, director of the project and project manager. The variables used were the execution of the SAP System project and the new LACS model. The results indicate that the new LACS model improves the execution of SAP system implementation projects. The first change that was obtained was that the number of valid unit tests increased by 40%, secondly, the training of personnel was increased by 30% and thirdly, the amount of valid technical documentation was increased by 40%, according to the progress in care. Therefore, it can be concluded that the general hypothesis has been accepted because there is sufficient evidence to indicate that the new LACS model significantly improves the execution of SAP system implementation projects, in the statistical tests chi-square was used to demonstrate the differences in the execution of the implementation project before and after the application of the new LACS model.

Keywords: agile methodologies, Lacs Model, implementation project execution, software development.

1. INTRODUCCIÓN

1.1 Planteamiento Del Problema

Hasta algunas décadas, un proyecto de Tecnología de la Información mal desarrollado tenía un impacto menor en una empresa, pero hoy en día muchos proyectos de TI impactan directamente en la supervivencia de una empresa. La demora en la implementación de un ERP SAP puede significar la pérdida de una posición competitiva, no realizar las compras de materiales, no atender el despacho de productos, no cumplir con las obligaciones tributarias, el impacto negativo seria en todos los módulos que se deseen implementar en la empresa.

Existen muchos proyectos que terminan siendo cancelados antes del tiempo previsto, inclusive antes de ser terminados, se pierden todos los recursos invertidos, ocasionando un daño a la Gerencia de Sistemas y al prestigio de la empresa.

Un proyecto de TI con una deficiente gestión ocasiona una mala relación con los clientes y proveedores, se refleja en la pérdida de ingresos e inclusive penalidades por incumplimiento.

La descripción del planteamiento del problema del trabajo de investigación presenta el **problema, causa, consecuencias y el aporte.** Explicadas cada una a continuación:

El problema se aparece cuando el consultor desarrollador de la implementación no muestra su avance al equipo cliente, tanto al consultor cliente responsable de las pruebas unitarias y al consultor desarrollador cliente para que vaya documentando las modificaciones de los programas Z y configuraciones en los diferentes módulos SAP.

Cuando el consultor desarrollador responsable de la implementación termina todas las correcciones a las funciones y métodos del SAP, recién libera todas las ordenes de transporte con el código al ambiente de Calidad, a partir de ese momento el consultor cliente recién puede ingresar al ambiente de Calidad y empezar con sus pruebas unitarias, ocasionando que al

encontrar observaciones en la funcionalidad, el consultor desarrollador responsable de la implementación vuelva a revisar las funciones y métodos desde el inicio, que en ocasiones deberá modificar una o varias funciones, que ya dio por cerradas y deberá volver a abrir el código anterior, perdiendo demasiado tiempo, en lugar de corregir la función en la primera oportunidad, en ese momento debe liberar el código en el ambiente de Calidad y permitir que el consultor cliente realice su prueba unitaria y de ser correcta, dejar la función cerrada y continuar con la siguiente función por revisar, de esta manera se revisa, se modifica y se realiza la prueba unitaria de la función una sola vez y no esperar hasta el final y revisar toda la prueba funcional, ocasionando tener que revisar y corregir el código varias veces.

El consultor desarrollador de la implementación no muestra las modificaciones que realiza en las funciones y métodos en el código SAP al consultor desarrollador del cliente, ocasionando que el traspaso del conocimiento sea muy lento e incompleto, no entregan la documentación técnica completa, que será de gran utilidad para el futuro mantenimiento de los programas Z y configuraciones que se realizaran en los módulos de SAP después de su implementación.

Las causas pueden ser múltiples para el problema, pero elegimos una causa la cual indica que no están utilizando el método correcto que les permita analizar, modificar y realizar las pruebas unitarias de manera sincronizada en cada función y método que sea necesario, aprovechar el tiempo para realizar una correcta implementación de los módulos de SAP.

La consecuencia de continuar con el mismo problema y no realizar ninguna acción, las empresas clientes que decidan implementar algún módulo de SAP estarán en riesgo de que el proyecto de implementación aumente sus costos iniciales, que el tiempo de implementación sea mucho mayor, que sus procesos más importantes dejen de funcionar por algún tiempo más, ocasionando que la empresa no cumplan con sus respectivos clientes, despachos, ventas,

compras y además con alguna responsabilidad tributaria, colocando en riesgo la imagen de la empresa y el bienestar de muchos colaboradores y sus respectivas familias.

El aporte que se propone es una nueva metodología, en la cual el consultor desarrollador implementador y el consultor del cliente responsable de las pruebas unitarias y funcionales, también el desarrollador del cliente responsable del mantenimiento futuro de los módulos de SAP, los tres trabajen de manera coordinada y sincronizada, para empezar a analizar, modificar, probar y documentar cada una las funciones y métodos que sean necesarios para una correcta implementación del sistema SAP.

1.2 Descripción Del Problema

La descripción de realidad problemática encontrada es la siguiente:

- La consultora envía al usuario responsable de las pruebas funcionales de la empresa todas las correcciones de los programas para que inicie las pruebas unitarias y funcionales, muchas veces lo envía en bloque y ya no queda mucho tiempo y el usuario no puede realizar correctamente todas sus pruebas.
- Las pruebas unitarias se realizan muy rápidamente por el tiempo y la presión que tiene el usuario del cliente para cumplir con el cronograma establecido por la consultora, cuando le entregan de pronto muchos casos para probar en el sistema.
- Las pruebas unitarias deben tener un margen de tiempo para su correcta prueba unitaria.
- Las pruebas funcionales también deben tener un determinado tiempo y no dejar sus pruebas para el final de la etapa, cuando todos están terminando sus actividades para cumplir el cronograma.
- El desarrollador consultor implantador no capacita al desarrollador del cliente porque está dedicado a terminar todas sus modificaciones de métodos y funciones.

- La consultora no entrega una documentación técnica completa de la ubicación donde se encuentras las funciones, métodos, nombres de tablas secundarias y tablas maestras para una futura revisión y mantenimiento.
- El consultor implantador encargado de la configuración no entrega una documentación completa para un futuro mantenimiento dé cada módulo.
- El desarrollador del cliente no es capacitado correctamente en ninguna etapa de la implementación del proyecto, mucho menos al final por los tiempos tan cortos que tendrá el consultor desarrollador de la implementación.
- El proveedor no entregó toda la documentación necesaria para las respectivas pruebas funcionales, como son los documentos de análisis y diseño, los cuales son necesarias para la revisión del producto.
- Cuando queda muy poco tiempo para realizar todas las pruebas, el usuario por presión realiza sus pruebas unitarias rápidamente, pero el problema viene después del pase a producción, cuando el usuario de la empresa empieza a utilizar el sistema en vivo y empiezan las incidencias, a causa de realizar una prueba rápida e incorrecta.
- Los usuarios del sistema aceptan el nuevo sistema con menores funcionalidades que el sistema anterior, por la falta de tiempo aceptan lo mínimo para poder salir a producción en la fecha de corte establecida.
- La fecha de salida en productivo es muy complicada de mover, porque para esa fecha ya se realizó una copia de respaldo de la data del sistema actual, la cual deberá trasladarse al nuevo servidor para continuar con las operaciones, también la configuración realizada en el sistema SAP que moverla tomará mucho tiempo y consumo de más recursos.

- Por no tener una buena coordinación y reducir los tiempos en la etapa del desarrollo, pruebas y capacitación al personal, no se cumplen con los plazos de entrega de alguna funcionalidad importante.
- La empresa se perjudica a nivel social, económico y de imagen frente a sus usuarios por las deficiencias del nuevo sistema.
- Se confió en la experiencia y nombre de la consultora que implemento la solución SAP, pero el resultado no es el esperado.
- La consultora que implementa la solución SAP debe contar con un equipo de trabajo, como mínimo 2 años atrás, al menos cubrir que el 70% del personal humano sea el mismo para este nuevo proyecto.
- El desarrollador de la consultora cuando termine con toda su corrección, entregara al usuario del cliente el programa para que inicie con todas sus pruebas unitarias y funcionales juntas, la consultora se puede beneficiar en que algún error no sea detectado por el usuario quien debe realizar las pruebas, porque el usuario del cliente estará presionado en realizar todas las pruebas unitarias que envió la consultora y el tiempo en que deberá terminarlas, así como realizar la documentación de su prueba con la observación bien especificada, para que el desarrollador consultor de la implementación pueda realizar el seguimiento y la corrección del programa.
- Así empieza una revisión por parte del consultor del cliente y la corrección del desarrollador de la consultora, siendo un punto crítico, muchas veces la presión del tiempo hace que no realice una verificación como es debido.
- La consultora no realiza un informe diario del avance en programación y en pruebas funcionales al director del proyecto del cliente, para el monitoreo y seguimiento del avance del proyecto.

- Se debe contratar una segunda empresa que supervise el avance de la consultora quién está realizando el proyecto de implementación de SAP.
- Las reuniones no se realizan una vez por semana, entre la empresa supervisora, la consultora y el cliente que está recibiendo la implementación de SAP, para revisar los riesgos del proyecto.
- No se realiza una capacitación en el Lenguaje de programación ABAP desde el primer día al personal técnico de la empresa, quienes se encargarán del mantenimiento de sistema SAP después de la implementación.
- No se realiza una capacitación en la Configuración del módulo SAP desde el primer día al personal técnico de la empresa, quienes se encargarán del mantenimiento en la nueva configuración después de la implementación.
- El equipo del proyecto quienes desarrollan y crean las funcionalidades que hacen falta, dejan para el final las pruebas unitarias con el usuario, porque esperan terminar todos los desarrollos y después invitar al usuario para que realice la comprobación y las observaciones que encuentre.
- Si existen correcciones el desarrollador deberá corregir una o varias funciones donde se encontró el problema, volviendo al inicio de las correcciones, es cuando se pierde tiempo valioso, la prueba se debería realizar justo cuando el desarrollador realizo el cambio por primera vez.
- Se debe realizar la modificación del programa, la prueba de usuario, volver a corregir si es necesario, probar nuevamente y cerrar el caso, continuar con la siguiente actividad.
- No tienen una mejora continua, es necesario utilizar el CICLO DE DEMMING como la estrategia basada en mejora de la calidad. Aunque pueda parecer muy sencillo de entender, la puesta en práctica no es tan sencilla. En muchos equipos, las empresas se olvidan de las dos

últimas fases de comprobación y actuar, estando en un continuo: planificar y hacer, “Plan-Do-Plan-Do-Plan-Do...” donde la comprobación y mejora se realiza al final y de una manera poco explícita y lenta.

- Demasiadas entradas y salidas de personas al proyecto impactan en la calidad del resultado.
- En la implementación de la metodología ASAP no teníamos reuniones de 15 min. Diarios para revisar los avances y/o problemas que tienen los integrantes del equipo, esta reunión se debe realizar de pie y es muy rápida no más de 15 minutos, ayuda para empezar el día conociendo el problema, según indica las metodologías agiles en el desarrollo de software.
- Nos apoyamos en las metodologías agiles, el poder aligerar los procesos, mediante principios y buenas prácticas, así como darle mucha importancia a la mejora continua.
- Es interesante esta aproximación ya que da especial importancia a la fase de feedback (inspección) y a la de adaptación, lo que nos permitirá de una forma continua validar los resultados obtenidos para, posteriormente, adaptarnos en función del resultado.
- No se deben cerrar todos los Business Blue Print en una fecha, sino de manera escalonada, porque con la presión de cerrarlos todos en una fecha se puede obviar alguna actividad importante, sino ir cerrando conforme se va avanzando y necesitando entregar al equipo de programación y al equipo funcional, el cerrar los Business Blue Print es una ventaja para la consultora y desventaja para la empresa porque tendrá que pagar las horas adicionales para modificar los programas, en el caso que olvido agregar alguna actividad en el proceso que se definió en el Business Blue Print, que por su poca experiencia pueden cometer errores.
- Los usuarios finales de la empresa no deben esperar a la fase 4 de preparación final para ser capacitados y conocer por primera vez el SAP, según la metodología ASAP, el tiempo es corto, los consultores están terminando de configurar el sistema y las prueba no se realizan de

manera completa, por ese motivo se propone que los usuarios finales de la empresa deben conocer el sistema SAP desde la Fase de Preparación porque la consultora puede entregar un IDE SAP para mostrar el funcionamiento del módulo a los usuarios finales y que estos puedan interactuar desde el primer día con el SAP, y cuando los consultores de los procesos terminen con la configuración del SAP de la empresa, en ese momento los usuarios ya tendrán con mínimo 6 o 8 meses utilizando el IDE SAP. De esta manera los usuarios finales estarán con mayor experiencia en su modulo SAP cuando tengan que atender a los clientes en la salida en vivo.

1.2.1 Ámbito Global:

A menudo para justificar que los proyectos no se terminan a tiempo, a pesar de los esfuerzos gestores, Los que realizan las pruebas pueden leer y analizar el código, ayudando a los desarrolladores a determinar los errores de forma más rápida, el trabajo en paralelo es eficiente, reduciendo el sobre esfuerzo en la comunicación, Menos confianza en métodos de gestión tradicionales para reducir esfuerzos de duplicados. (Brooks, 1995)

Las maneras en que las tecnologías de la información posibilitan formas extensivas de colaboración que pueden tener consecuencias transformativas para la economía y la sociedad, acuñó el término de envidia y altruismo, para describir la dinámica en la producción entre pares basada en el bien común. (Benkler, 2012)

Dado un número suficientemente elevado de ojos, todos los errores se vuelven obvios, fue bautizada como la ley de Linus, todo problema deberá ser transparente para alguien. Las personas que entienden y las que resuelven un problema no deben ser necesariamente las mismas, ni siquiera en la mayor parte de los casos, Alguien encuentra el problema y otro lo resuelve. Pero el punto está en que ambas cosas suelen suceder muy rápidamente. (Raymond, 2001)

1.2.2 Ámbito Local:

En su estudio “Aplicación de una metodología ágil en el desarrollo de un sistema de información”: Las aplicaciones informáticas han contribuido enormemente en el escenario de nuestras vidas, están directas o indirectamente presentes en nuestro día a día, y somos consumidores o desarrolladores de ellas. Cuando cumplimos un papel de desarrolladores, se presentan diversas metodologías al momento de empezar un proyecto de software, dentro de ellas nos ofrecen un punto de vista alternativo a las clásicas y duras, las metodologías ágiles. El presente proyecto aplica una de estas metodologías ágiles, Programación Extrema (Extreme Programming), en un pequeño proyecto de software, utilizando herramientas de software libre como Java, y como repositorio de datos el estándar XML. El resultado de esta investigación aporta una guía del uso de la metodología ágil en un pequeño proyecto de software que tiene aplicabilidad dentro del ciclo de inteligencia de la información. (Samamé et al., 2013)

1.3 Formulación Del Problema

Problema General

¿De qué manera el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software mejorará la ejecución de proyectos de implementación del sistema SAP?

Problemas Específicos

- ¿De qué manera un nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software aumentará la cantidad de pruebas unitarias válidas en la ejecución de proyectos de implementación del sistema SAP?

- ¿De qué manera un nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará la capacitación del personal actual en la ejecución de proyectos de implementación del sistema SAP?
- ¿De qué manera un nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará la cantidad de documentación técnica en la ejecución de proyectos de implementación del sistema SAP?

1.4 Antecedentes

1.4.1 Antecedentes Internacionales:

En su estudio "Metodologías ágiles y desarrollo basado en conocimiento": En este proyecto de investigación se ha planteado realizar una investigación tendiente a exponer los fundamentos de diferentes metodologías ágiles propuestas para el desarrollo de sistemas y sobre el desarrollo basado en conocimiento. A su vez, se ha pretendido realizar un intento por esbozar relaciones que pudieran hacer posible trabajar con metodologías ágiles y desarrollo basado en conocimiento al mismo tiempo. En función de los objetivos a alcanzar, se está realizando una investigación bibliográfica para profundizar el conocimiento sobre las metodologías ágiles actuales y sobre las bases de datos del conocimiento que se combinará con estudio de campo de la utilización o no de las mismas en sectores públicos y privados. Además, también se buscará poder definir pautas metodológicas para combinar el uso de las metodologías ágiles y del desarrollo basado en conocimiento. (Saravia, 2012)

En su estudio: "desarrollo ágil de Software con Arquitecturas Dirigidas por Modelos": La especificación Model Driven Architecture, es una especialización del desarrollo dirigido por modelos que separa la lógica del negocio del software y las plataformas tecnológicas. Para ello se define tres tipos de modelos. Los Computation Independent Model, asociados al

dominio del negocio, los Platform Independent Model, asociados a modelos abstractos del software, y los Platform Specific Model, relacionados con modelos de software específicos de plataformas tecnológicas. Sin embargo, no se detalla cómo deben ser los modelos y tampoco describe cómo deben ser transformados a modelos. Como solución a dicho problema, esta tesis presenta una recomendación que propone un proceso de desarrollo de software basado en la creación de modelos de procesos del negocio, clasificados que son asociados a los modelos iniciales del software. Partiendo de una interpretación válida, la recomendación propuesta se apoya además en la aplicación de otras disciplinas de gran actualidad. Entre ellas destacamos el uso de desarrollo ágil de software, para la definición adecuada de los procesos del negocio. (García-Bustelo et al., 2007).

En su estudio: "Selección de metodologías ágiles e integración de arquitecturas de software en el desarrollo de sistemas de información": La Ingeniería de Software se ha convertido imprescindible en el ámbito organizacional, su desenvolvimiento y gestión depende en gran medida de los Sistemas de información y de las Tecnologías de la información.

En los últimos años, se ha impuesto el uso de las Metodologías Ágiles, marcando una tendencia en su adopción al desarrollo de proyectos de software. La causa principal es que en ambientes donde las necesidades de las organizaciones y la tecnología cambian rápidamente, las metodologías tradicionales predictivas han demostrado ser poco eficientes para atender los requerimientos de clientes y usuarios, limitando la competitividad y a la obtención de mayores beneficios en la producción de bienes o en la prestación de servicios, en el menor tiempo posible, y es en ese escenario donde han ganado bastante popularidad siendo una muy buena solución para proyectos a corto plazo, en especial, aquellos proyectos en donde los requisitos están cambiando constantemente.

La Arquitectura de Software comprende elementos de software, las propiedades externamente visibles de aquellos elementos y su interrelación para satisfacer la funcionalidad y requerimientos deseados. Si bien en los últimos años, ha comenzado a cobrar una mayor importancia dentro de los estudios propiciados, su desarrollo es una práctica poco común para algunos desarrolladores, en especial si se utilizan metodologías de desarrollo ágiles, en las que esas actividades no se consideran relevantes.

En este trabajo se presenta un avance de la exploración, comparación y selección bajo distintos criterios de dos metodologías ágiles, con el propósito de incluir actividades de diseño de arquitecturas de software, que permita generar un modelo genérico de integración, que pueda ser aplicado a otras metodologías bajo el dominio agil, sin dejar de lado las consideraciones de calidad, riesgo y costos asociados a la integración pretendida. (Navarro et al, 2017)

En su estudio: “Implementación de la Metodología SCRUM en un Ambiente Bancario”: El objetivo principal de este trabajo es implementar la metodología SCRUM en un grupo de trabajo de un ambiente bancario, que no ha usado esta metodología previamente. El equipo desarrollador posee conocimiento del marco de trabajo, más no experiencia en el mismo.

La metodología se implementará en el desarrollo de un proyecto con la duración de cuatro iteraciones, de las cuales se pudieron ejecutar únicamente tres, debido a diversos inconvenientes que se irán detallando a lo largo de este documento. La aplicación ya se ha desarrollado previamente, sin embargo, debido a la deuda técnica y a la necesidad de migración a nuevas tecnologías, se desarrolló nuevamente. Para la implementación, se realiza una capacitación previa y los integrantes se preparan para ejecutar de manera correcta el desarrollo. Se realizan tres iteraciones que contienen el grueso de la aplicación y queda pendiente una

cuarta iteración que es de migración de información y ejecución de procesos batch. Se diseña y se desarrolla un plan de trabajo para la implementación de la metodología ágil. Se analizan las variables del ambiente de trabajo para determinar las necesidades específicas. Se logran usar las herramientas a disposición para la implementación satisfactoria, en actividades que son inherentes a la entidad, que pertenecen a las iteraciones y que deben de ser desarrolladas por los integrantes (Forero, 2018)

En su estudio: "Metodologías ágiles para desarrollo de proyectos en la gestión de las organizaciones públicas en Colombia": En el presente documento que se mostró, contiene una temática muy específica, la cual es la implementación de metodologías agiles de desarrollo en proyectos para las organizaciones del sector público en Colombia. El tema presentado ha resultado de ser de mucho interés, ya que hoy en día, medir el desarrollo y desempeño de un país, está directamente ligado con la capacidad que tiene un Gobierno o una administración para poder gestionar, planificar y ejecutar los proyectos que se puedan llevar a cabo en todas las regiones de Colombia y con ello poder evaluar cuales son los beneficios y ventajas que conlleva implementar estas nuevas metodologías de desarrollo de proyectos en Colombia. (Erazo, 2017).

En su estudio "Integración de arquitectura de software en el ciclo de vida de las metodologías ágiles": Las metodologías Ágiles se centran en el trabajo en equipo, la adaptabilidad y colaboración dentro del grupo de software y también entre los miembros del grupo y los usuarios finales. El uso de las Metodologías Ágiles (MA), ha marcado una tendencia en su adopción al desarrollo de proyectos de software dado las necesidades cambiantes y la espera de beneficios en el menor tiempo posible.

En general, pero también desde la perspectiva de los requisitos, esto hace que las MA, típicamente eviten un trabajo inicial sustancial, suponiendo que los requisitos siempre cambian y continúan cambiando a lo largo del ciclo de vida del proyecto.

La Arquitectura de Software, en tanto, es una manifestación de decisiones de etapas muy tempranas del diseño sobre un sistema.

Estas decisiones tempranas llevan un peso importante con respecto al desarrollo del resto de un sistema, ya que condicionan otras decisiones que siguen, y en el caso de cambios, implican ramificaciones posteriores. Esto supone una captura de requisitos que no tengan cambios sustanciales en las etapas intermedias y finales del desarrollo de un proyecto.

Este tratamiento con enfoques diferentes en las primeras etapas (y también en otros aspectos), ha sido uno de los factores que ha causado la sensación en las llamadas Metodologías Agiles.(Navarro et al., 2017)

El proceso de desarrollo de software a través de los años se ha venido implementando una serie de metodologías que facilitan a programación. La presente investigación realiza una revisión de publicaciones sobre las metodologías llamadas ágiles, sus principios y fundamentos, estableciendo definiciones y explicaciones detalladas de las más relevantes en la actualidad (Scrum y XP), convirtiéndose en la más acertada para el desarrollo de los procesos de ingeniería de software. (Molina et al., 2018)

1.4.2 Antecedentes Nacionales:

En su estudio “Metodologías ágiles en la implementación de una aplicación móvil para la gestión de citas en la clínica dental Perio Dent - Huancayo”: fue desarrollada en el área de Ingeniería de Sistemas y tiene como objetivo determinar la influencia de la implementación de una aplicación móvil con las metodologías ágiles en la Gestión de Citas en la Clínica Dental “PERIO DENT” – Huancayo, con el desarrollo de este trabajo de investigación se obtuvo una

mejora en la Gestión de Citas el cual fue innovador, capaz de permitir que los pacientes de la Clínica Dental PERIO DENT tengan la facilidad de reservar una cita en cualquier momento del día y en cualquier lugar; gestionándose mediante un sistema administrador que asegura la integridad y consistencia de los datos ingresados; además de presentar los horarios de atención disponibles de los profesionales médicos, evitando acudir físicamente y en muchos de los casos en vano a la Clínica Dental. (Huaylinos, 2017).

En su estudio: "Desarrollo e implementación de un sistema web para generar valor en una pyme aplicando una metodología ágil. Caso de estudio: Manufibras Perez SRL": Las empresas en la actualidad se apoyan cada vez más en la tecnología para la mejora de sus procesos y productos. Por lo que la adopción de un sistema web que automatice procesos del negocio, está dejando de ser una alternativa para pasar a ser un requerimiento en las pymes, debido a que tienen que estar adaptándose rápidamente a los cambios que puedan presentarse en su entorno por causa de la alta competencia de los productos que elaboran y el poder competir dentro del mercado. En este contexto, es viable mejorar la situación actual para la pyme de caso de estudio, pues al momento todo es un proceso manual, el cual trae como consecuencia pérdidas económicas por errores manuales y la alta inversión de tiempo en sus actividades. Por lo que el objetivo del presente trabajo es la generación de valor para la pyme, debido a que es importante mejorar la situación económica de las pymes ya que investigaciones previas señalan que son las que aportan un mayor crecimiento al país y son generadoras de empleo. Al finalizar el proyecto se demuestra como con la consecución del sistema para la promoción de productos, gestión de pedidos y registro de ventas, se genera valor para la pyme con la reducción de tiempo, costos operativos y el mejorar el servicio a los clientes, los cuales permitirán que los beneficios sean mayores a la inversión del proyecto. Además, también se comprueba con la revisión de la literatura que estudios previos sobre el desarrollo web inciden

en el uso de las metodologías ágiles, las cuales referencian a la Extreme Programming (XP) y Scrum como las más destacadas metodologías ágiles para el desarrollo de software. (Castillo, 2016).

1.5 Justificación De La Investigación

El estudio se encuentra en el marco de la línea de investigación, se propone utilizar una nueva metodología ágil para la implementación del Sistema SAP, es conveniente mejorar el procedimiento en la etapa de diseño cuando el consultor desarrollador de la implementación empieza a modificar la funciones y métodos, es allí cuando se debe sincronizar los esfuerzos, entre el desarrollador implementador, el responsable de las pruebas unitarias del cliente y el desarrollador del cliente, para que puedan revisar, modificar y probar en el ambiente de Calidad en el mismo momento de la atención, una por una las funciones y métodos que son necesarios para la completa implementación del sistema SAP en la empresa.

Esta investigación servirá para comprobar que los tiempos entre corregir una función o método y las pruebas unitarias, se van a reducir considerablemente, con el fin de llegar a cumplir los tiempos del proyecto.

La investigación aportara con evidencias que el trabajo coordinado en esta etapa va a aumentar la capacidad del personal del cliente, al momento de realizar las pruebas y al desarrollador para obtener la información necesaria para preparar un documento técnico completo.

Los beneficios del presente trabajo será que el responsable de las pruebas unitarias del cliente tendrá más tiempo para realizar una mejor prueba unitaria y evitar futuras incidencias, al detectar un mal funcionamiento en sus pruebas unitarias, ya no recibirá todo el bloque para realizar sus pruebas unitarias al final de la etapa, teniendo que quedarse hasta muy tarde, no pudiendo realizar todas las pruebas que desea realizar, muchas veces por el corto

tiempo o por la presión de los tiempos del proyecto. También será beneficiado el desarrollador del cliente quien podrá observar las modificaciones que realizará el consultor desarrollador implementador en cada función y método, así podrá anotar con más detalle todos los cambios y podrá documentar lo observado que le será muy útil para los futuros mantenimientos.

El beneficio de este trabajo de investigación será para todo el proyecto, porque no será parte del porcentaje de proyectos que fracasan por exceder los tiempos de implementación, también se reducirá las incidencias futuras por haber realizado pruebas unitarias completas, también directamente los beneficiarios serán el responsable de las pruebas unitarias quien realizara sus pruebas con mayor tiempo y mejor calidad, también el desarrollador del cliente quien podrá documentar todos los detalles que le serán necesarios cuando este solo frente al SAP y le tenga que realizar el mantenimiento, cuando la consultora responsable de la implementación ya haya terminado su trabajo y se haya retirado de la empresa.

Otros beneficiarios serán desde la alta dirección del proyecto, el director del proyecto implementador, director del proyecto cliente, gerente del proyecto implementador, gerente del proyecto cliente, los consultores de la implementación, los consultores del cliente, los desarrolladores de la implementación, los desarrolladores del cliente y los Key user.

El trabajo de investigación busca cambiar la forma en que se viene llevando esta etapa, en la que el consultor desarrollador implementador modifica las funciones y métodos, pero cuando termina todas sus modificaciones recién avisa al consultor responsable de las pruebas unitarias de la empresa para que empiece con sus pruebas, muchas veces le envía el bloque con todas las pruebas unitarias por realizar, provocando que las pruebas se realicen de manera deficiente por el poco tiempo y por la presión del momento, dejando pasar muchas veces código incorrecto que será la causa de un mal funcionamiento futuro y el motivo de muchas incidencias, provocando que la empresa atienda de manera deficiente a sus clientes,

proveedores, puede afectar la imagen de la empresa, así como tener problemas legales, tributarios, penalidades, es por ello que esta etapa es muy crítica, por ese motivo se propone en esta investigación corregir y que se trabaje de manera coordinada, se analice el código, se realice la corrección del código, se realice la prueba unitaria y se vaya tomando nota de los cambios para crear un documento técnico en el mismo momento, para futuras modificaciones.

1.6 Limitaciones De La Investigación

Actualmente en el Perú existen muy pocos estudios sobre la implementación de proyectos SAP en las empresas utilizando metodologías ágiles, la investigación plantea realizar pruebas unitarias en la etapa de diseño cuando se realiza la modificación de los métodos y funciones del código fuente SAP, de esta manera se busca minimizar los tiempos de reprocesos para cumplir con los tiempos establecidos del proyecto, sin aumentar los costos y mantener la calidad del producto, es la razón por la cual la mayor parte de la bibliografía es internacional, la investigación se realiza con un financiamiento personal, respecto a la encuesta a expertos de metodologías ágiles se tiene poca referencia a nivel nacional, por ese motivo se solicitó el apoyo de expertos extranjeros que participaron en la implementación del sistema SAP, de una empresa global de consultoría de negocios y TI, especializada en servicios SAP y transformación digital.

1.7 Objetivos

Objetivo General

Determinar en qué medida el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software mejorará la ejecución de proyectos de implementación del sistema SAP.

Objetivos Específicos

- Determinar en qué medida el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software aumentará la cantidad de pruebas unitarias válidas en la ejecución de proyectos de implementación del sistema SAP.
- Determinar en qué medida el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará la capacitación del personal actual en la ejecución de proyectos de implementación del sistema SAP.
- Determinar en qué medida el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará la cantidad de documentación técnica en la ejecución de proyectos de implementación del sistema SAP.

1.8 Hipótesis

Hipótesis General

Si se aplica el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software entonces mejorará de manera sustantiva la ejecución de proyectos de implementación del sistema SAP.

Hipótesis Especificas

H1.- Si se aplica el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software aumentará la cantidad de pruebas unitarias válidas en la ejecución de proyectos de implementación del sistema SAP.

H2.- Si se aplica el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará la capacitación del personal actual en la ejecución de proyectos de implementación del sistema SAP.

H3.- Si se aplica el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará la cantidad de documentación técnica valida en la ejecución de proyectos de implementación del sistema SAP.

II. MARCO TEÓRICO

2.1 Marco Conceptual

2.1.1 Gestión del riesgo – pmbok

La guía se basa en el estándar para la Dirección de Proyectos, un proyecto es un esfuerzo que se lleva a cabo para crear un producto, servicio o resultado único, y tiene la característica de ser naturalmente temporal, es decir, que tiene un inicio y un final establecidos, y que el final se alcanza cuando se logran los objetivos del proyecto o cuando se termina el proyecto porque sus objetivos no se cumplirán o no pueden ser cumplidos, o cuando ya no existe la necesidad que dio origen al proyecto.

Sobre la Gestión de Riesgos del Proyecto incluye los procesos para llevar a cabo la planificación de la gestión, identificación, análisis, planificación de respuesta, implementación de respuesta y monitoreo de los riesgos de un proyecto. Los objetivos de la gestión de los riesgos del proyecto son aumentar la probabilidad y/o el impacto de los riesgos positivos y disminuir la probabilidad y/o el impacto de los riesgos negativos, a fin de optimizar las probabilidades de éxito del proyecto.

Los procesos de Gestión de los Riesgos del Proyecto son:

1. Planificar la Gestión de los Riesgos: El proceso de definir como realizar las actividades de gestión de riesgos de un proyecto.

2. Identificar los Riesgos: El proceso de identificar los riesgos individuales del proyecto, así como las fuentes de riesgo general del proyecto y documentar sus características.

3. Realizar el Análisis Cualitativo de Riesgos: El proceso de priorizar los riesgos individuales del proyecto para análisis o acción posterior, evaluando la probabilidad de ocurrencia e impacto de dichos riesgos, así como otras características.

4. Realizar el Análisis Cuantitativo de Riesgos: El proceso de analizar numéricamente el efecto combinado de los riesgos individuales del proyecto identificados y otras fuentes de incertidumbre sobre los objetivos generales del proyecto.

5. Planificar la Respuesta a los Riesgos: El proceso de desarrollar opciones, seleccionar estrategias y acordar acciones para abordar la exposición al riesgo del proyecto en general, así como para tratar los riesgos individuales del proyecto.

6. Implementar la Respuesta a los Riesgos: El proceso de implementar planes acordados de respuesta a los riesgos.

7. Monitorear los Riesgos: El proceso de monitorear la implementación de los planes acordados de respuesta a los riesgos, hacer seguimiento a los riesgos identificados, identificar y analizar nuevos riesgos y evaluar la efectividad del proceso de gestión de los riesgos a lo largo del proyecto. (Institute, 2017).

2.1.2 Business Blue Print (Bbp)

Considera que esta función documenta los procesos de negocio de su empresa que desea implementar en el sistema. En un Plan de Negocios para Proyectos, se crea una estructura de proyecto en el que los escenarios relevantes de negocio, procesos de negocio y pasos del proceso se organizan en una estructura jerárquica. También puede crear la documentación del proyecto y asignarlo a escenarios individuales, procesos o etapas del proceso. A continuación, asignar transacciones a cada paso del proceso, para especificar cómo sus procesos de negocio deben ejecutarse en los sistemas SAP. (Enrich, 2013)

2.1.3 Metodologías

2.1.3.1 Metodología RUP – Proceso Unificado de Rational

RUP es un proceso de desarrollo de Software, iterativo e incremental, conducido por casos de uso, centrado en una arquitectura, enfrenta riesgos, controla cambios, soporta varias

herramientas. Pero RUP no cubre por completo los procesos de gerenciamiento de proyectos, como por ejemplo la gestión de costos, recursos humanos y adquisiciones. Es porque RUP está orientado a Proyectos de Desarrollo de Software, no a la Gestión del Proyecto.

Se identificó las siguientes causas del problema con los proyectos TI de desarrollo de software: falta de involucramiento del usuario, falta de apoyo de la alta gerencia, objetivo de negocios difusos, gerente de proyecto inexperto, hitos de larga duración, requerimientos no administrativos, personal inexperto y planeamiento insuficiente.

Los dos grandes factores de éxito que pueden ayudar en la mejora de los proyectos de desarrollo de software a un 100% (en 10 años) son Proceso iterativo y gestión de proyectos profesional. (Delgado, 2008).

Las características principales del RUP serían:

Centrado en los modelos: Los diagramas son un vehículo de comunicación más expresivo que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema.

Guiado por los Casos de Uso: Los Casos de Uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.

Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.

Iterativo e incremental: Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

Entre los Beneficios que aporta RUP tenemos:

- Permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software mediante una gestión sistemática de riesgos.

- Permite la producción de software que cumpla con las necesidades de los usuarios, a través de la especificación de los requisitos, con una agenda y costo predecible.
- Enriquece la productividad en equipo y proporciona prácticas óptimas de software a todos sus miembros.
- Permite llevar a cabo el proceso de desarrollo práctico, brindando amplias guías, plantillas y ejemplos para todas las actividades críticas.
- Se integra estrechamente con herramientas Rational, permitiendo a los equipos de desarrollo aprovechar todas las ventajas de las características de los productos Rational, el Lenguaje de Modelado Unificado (UML) y otras prácticas óptimas de la industria.
- Optimiza la productividad de cada miembro del equipo al poner al alcance la experiencia derivada de miles de proyectos y muchos líderes de la industria.

No solo garantiza que los proyectos abordados serán ejecutados íntegramente, sino que además, evita desviaciones importantes respecto a los plazos. (Wong et al., 2010).

2.1.3.2 Microsoft Solutions Framework - MSF

Las aplicaciones de software son programas que tienen la función de agilitar los procesos que demandan una gran aplicación de tiempo o recursos a los usuarios, como la utilización de un programa para registro de entrada de empleados en una empresa, con el cual se permite determinar la hora de ingreso, horas de trabajo y faltas de cada empleado registrado, etc. El presente trabajo refleja un análisis de las aplicaciones de software, cuya documentación se realiza a través del uso de la metodología de desarrollo MSF (Microsoft Solutions Framework), lo que nos permite la evaluación de sus procesos a través de las fases de la metodología antes mencionada, dicha evaluación nos garantiza la eficacia y eficiencia durante el proceso de desarrollo de software. (Lomas, 2018).

2.1.3.3 Agile Modeling (AM)

"Agile Modeling es una metodología para el modelado de sistemas y documentación de software utilizando las mejores prácticas. Es una colección de valores, principios y buenas prácticas que se puede aplicar en un proyecto de desarrollo de software. Es decir, AM no es un proceso prescriptivo, no define los procedimientos detallados para la forma de crear un determinado tipo de modelo, sino que ofrece un conjunto de recomendaciones. No es una metodología que abarque todas las fases de un proyecto, sino que, como ya se ha dicho, se centra en el modelado y la documentación. No incluye gestionar el proyecto, la programación de actividades, testeo u otras fases.

Por lo tanto, AM es un complemento a otras metodologías ágiles como SCRUM y XP. Sin embargo, según estadísticas del 2011. AM se usa muy minoritariamente representando tan solo el 1% de todo el desarrollo ágil de software." (Ambler, 2002).

2.1.3.4 Adaptive Software Development (ASD)

La filosofía ASD que se basa en la colaboración humana y la auto organización del equipo, se adapta al cambio en lugar de estar luchando contra él, es decir, se basa en la adaptación continua a circunstancias cambiantes.

El ciclo del desarrollo ASD se dividen en 3 fases:

1. Especulación: Al comienzo del proyecto se establecen sus principales objetivos, limitaciones y riesgos, se hace una estimación del marco temporal del proyecto. Determina el número de iteraciones y su duración. Para cada iteración se definen sus objetivos y funcionalidades. En cada nueva iteración se volverá a analizar y recalcular en base a lo ya ejecutado.

2. Colaboración: Es la etapa donde se realiza concurrentemente el trabajo de desarrollo y gestión del producto. En esta fase son revisados en profundidad los requisitos y se define cómo se va a trabajar de acuerdo con las habilidades de cada miembro del equipo.

3. Aprendizaje: Como la filosofía es el aprendizaje continuo, esta fase es un elemento crítico para la eficacia de los equipos. En cada iteración se revisa: calidad del producto desde el punto de vista del cliente y por otro lado desde los desarrolladores, la gestión de rendimiento y la situación del proyecto (como paso previo a la planificación de la siguiente iteración del proyecto). (Highsmith , 2002).

2.1.3.5 Agile Unified Process (AUP)

Considera que es una versión simplificada de Rational Unified Process (RUP) desarrollado por IBM. Describe una manera simple de entender el desarrollo de aplicaciones de negocio usando técnicas ágiles y conceptos heredados de RUP. Usa técnicas ágiles tales como Test Driven Development (TDD), modelado ágil, gestión de cambios ágil y la refactorización del codigo. En 2012, AUP fue reemplazado por Disciplined Agile Delivery (DAD) dejándose de evolucionarse.

Se enumeran los 6 principios en los que se fundamenta AUP: los empleados saben lo que están haciendo, simplicidad, agilidad, centrarse en la actividad de alto valor, independencia de herramientas (cualquier conjunto que se adapte y optimice el trabajo) y adaptar AUP para cumplir con las necesidades propias del proyecto. (Ambler, 2002).

2.1.3.6 Crystal

Describe un conjunto de metodologías ligeras o agiles a las que llamo Crystal. Su nombre proviene de los minerales, que se caracterizan por 2 dimensiones: color y belleza. Análogamente los proyectos de software también pueden caracterizarse por 2 dimensiones: tamaño o dimensión (número de personas en el proyecto) y criticidad (consecuencia de los errores). La criticidad se puede clasificar de menos a más grave: pérdida de comfort o usabilidad, pérdidas económicas moderadas, pérdidas económicas graves y pérdida de vidas humanas. Según el número aproximado de personas se le llama con un color, algunos son: crystal clear, crystal yellow, crystal Orange, crystal Orange web, crystal red, crystal maroon, crystal diamond o crystal sapphire.

En sus conclusiones rompe con la idea que existen metodologías omnipotentes que se implantan directamente y por completo, así expone que no existe una metodología de desarrollo de software única, mejor y universal, sino que por cada tipo de proyecto hay una metodología optima, que permita la maniobrabilidad en el proyecto. Los parámetros fundamentales para seleccionar una metodología son el tamaño del equipo, su distribución, la criticidad del proyecto y las prioridades. (Cockburn, 2004)

2.1.3.7 Dynamic System Development Method (DSDM)

El método de Desarrollo de Sistemas Dinámicos, en ingles Dynamic Systems Development Method (DSDM), como un método que provee un framework para el desarrollo ágil de software, apoyado por su continua implicación del usuario en un desarrollo iterativo y creciente que sea sensible a los requerimientos cambiantes, para desarrollar un sistema que reúna las necesidades de la empresa en plazos y costes. Se podría decir que es la metodología ágil que más se aproxima a las tradicionales permitiendo alcanzar un nivel 2 de madurez según CMMI.

En algunas circunstancias hay posibilidades para integrar contenido de otros métodos, Por ejemplo, RUP, XP, PRINCE2 o PMBOK pueden complementar a DSDN en la realización de proyecto. (Cockburn, 2004)

2.1.3.8 Feature Drive Development (FDD)

Definen unos procesos adaptativos, agiles que se pueden aplicar a proyectos de tamaño medios y grandes.

Al igual que otras metodologías agiles, FDD adopta una filosofía que:

- Hace hincapié en la colaboración entre las personas del equipo.
- Gestiona los problemas y la complejidad del proyecto utilizando una descomposición basada en características (o funciones) mediante su integración en sucesivos incrementos del software.
- Comunica los detalles técnicos utilizando medios verbales, gráficos y escritos.

El ciclo de desarrollo se divide en 5 fases y es de tipo incremental, constando cada incremento (iteración) en 2 fases: diseño y construcción de una característica. (Palmer et al., 2002)

2.1.3.9 Lean Software Development (LSD)

Lean trata de aplicar los principios que revolucionaron la industria, provenientes de Toyota, y que se han trasladado al desarrollo de software.

Es una colección de seis principios que busca eliminar los trabajos que no generen valor para el cliente, que minimiza la deuda técnica, y que favorece una organización en busca de la mejora continua y facilite la labor de los equipos. No se trata de unas recetas a aplicar, si no de conceptos cuya implementación puede ser muy diferente según la casuística de las organizaciones. Es por eso por lo que son válidos universalmente. (Diaz , 2009)

El pensamiento Lean comparte conceptos simplificados, tales como: “centrarse en el valor”, “lotes de pequeño tamaño” y “eliminación de residuos”. Lean es un súper conjunto que comparte atributos con ágil y Kanban.

El Marco de referencia Ágil y Lean pueden utilizarse tal como están o ser combinados para adaptarse a lo que funciona mejor en un entorno o situación determinados. No es necesario utilizar ninguno de estos; se puede desarrollar un enfoque ágil desde cero, siempre y cuando se adhiera a la mentalidad, los valores y los principios del Manifiesto Ágil, Si se siguen los principios de Ágil para entregar valor a un ritmo sostenible, y el enfoque desarrollado promueve la colaboración con el cliente, no se requiere un enfoque especifico. (Institute, 2017).

2.1.3.10 Extreme Programming (XP)

Definen al XP como una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los programadores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes.

A continuación, se describen las características esenciales de XP organizadas en los apartados siguientes: historias de usuario, roles, proceso y prácticas.

Las Historias de Usuario, corresponden a la técnica utilizada para especificar los requisitos del software. Se trata de formatos en los cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

La propuesta original de Beck incluye los siguientes roles:

- Programador. El programador escribe las pruebas unitarias y produce el código del sistema.

- Cliente. Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.

- Encargado de pruebas (Tester). Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

- Encargado de seguimiento (Tracker). Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.

- Entrenador (Coach). Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

- Consultor. Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto.

- Gestor (Big boss). Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

Proceso XP

El ciclo de desarrollo consiste en los siguientes pasos:

1.El cliente define el valor de negocio a implementar.

2. El programador estima el esfuerzo necesario para su implementación.

3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.

4. El programador construye ese valor de negocio.

5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos.

De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste en seis fases:

Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

Prácticas XP

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas:

-El juego de la planificación. Hay una comunicación frecuente entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

-Entregas pequeñas. Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más de 3 meses.

-Metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida

que describe cómo debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).

-Diseño simple. Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.

-Pruebas. La producción de código está dirigida por las pruebas unitarias. Éstas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.

-Refactorización (Refactoring). Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.

- Programación en parejas. Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto con lleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores).

- Propiedad colectiva del código. Cualquier programador puede cambiar cualquier parte del código en cualquier momento.

- Integración continua. Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.

- 40 horas por semana. Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo.

- Cliente in situ. El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce

constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita. (Orjuela et al., 2018)

Definen al XP como la metodología ágil más conocida. Fue desarrollada por Kent Beck buscando guiar equipos de desarrollo de software pequeños o medianos, entre dos y diez desarrolladores, en ambientes de requerimientos imprecisos o cambiantes. XP tiene como base cinco valores:

Simplicidad, Comunicación, Retroalimentación, Respeto y Coraje.

Estos valores, a su vez, son la base para la definición de sus principios. De ellos, los fundamentales son: la retroalimentación rápida, asumir simplicidad, el cambio incremental, la aceptación del cambio y el trabajo de calidad.

Las prácticas de esta metodología se derivan de sus valores y principios y están enfocadas en darle solución a las actividades básicas de un proceso de desarrollo, esto es: escribir código, realizar pruebas, escuchar (planear) y diseñar.

Como se mencionó al inicio de la sección II, las metodologías ágiles no están limitadas a Scrum y XP, sino que incluyen varias más. A continuación, se presenta una breve descripción de las más relevantes. (Navarro et al., 2013)

2.1.3.11 Scrum

Mencionan que el Scrum fue desarrollado inicialmente por Ken Schwaber y Jeff Sutherland, con colaboraciones posteriores de Mike Beedle.

Scrum proporciona un marco de gestión de proyectos que enfoca el desarrollo en ciclos de Sprint de 30 días en los que se entrega un conjunto especifico de características de la cartera de pedidos. La práctica central en scrum es el uso de reuniones diarias de equipo de 15 minutos

para la coordinación e integración. Scrum ha estado en uso durante casi diez años y se ha utilizado para entregar con éxito una amplia gama de productos. (Highsmith, 2002).

Para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas Sprint, con una duración de 30 días. El resultado de cada Sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

En Scrum inicialmente planean el contexto y un estimado amplio de la entrega. Después desarrollan el estimado de la entrega basándose en desarrollo del ambiente del proyecto. El proceso de ciclo de vida de Scrum reconoce que el proceso de desarrollo fundamental está completamente indefinido y usa mecanismos de control para perfeccionar la flexibilidad, manipular lo impredecible y el control del riesgo (Orjuela et al., 2018)

La puerta de entrada a las metodologías ágiles de muchas empresas es Scrum.

Se trata de una colección de procesos pensada para la gestión de proyectos que permite centrarse en la entrega de valor al cliente y la potenciación del equipo para lograr su máxima eficiencia, dentro de un esquema de mejora continua. (Diaz, 2009).

La importancia del conflicto como oportunidad para el mejoramiento continuo dentro de los equipos que trabajan bajo la metodología de desarrollo de software, conocida como SCRUM. Consiste en un estudio investigativo de tipo cualitativo, que presenta los resultados de una investigación documental, en artículos académicos relacionados con el conflicto en equipos de trabajo, más específicamente en proyectos de desarrollo de software, además de una serie de entrevistas realizadas a trece personas que llevan a cabo el rol de Scrum Master dentro

de una de las empresas de desarrollo de software más importantes de Colombia. Esta empresa cuenta con más de 30 años de experiencia, presente en tres países más del continente, y ha sido reconocida constantemente por su liderazgo en las mejores prácticas de desarrollo de software. La conclusión más importante es la valoración que hacen los entrevistados del conflicto, como una dinámica que propicia la maduración y el mejoramiento continuo en los equipos de trabajo que usan metodologías de desarrollo ágil. (Villegas et al., 2017).

2.1.3.12 Kanban

Una manera de pensar acerca de la relación entre Lean, ágil y el Método Kanban es considerar a ágil y al Método Kanban como descendientes del pensamiento Lean.

Esta herencia compartida es muy similar y se centra en la entrega de valor, el respeto por las personas, la reducción del desperdicio, la transparencia, la adaptación al cambio y la mejora continua. Los equipos del proyecto a veces encuentran útil mezclar diversos métodos, lo que funcione para la organización o el equipo es lo que debe hacerse, independientemente de su origen. El objetivo es obtener el mejor resultado, independientemente del enfoque utilizado.

El Método Kanban es inspirado por el sistema de manufactura Lean original y se utiliza específicamente para trabajos relacionados con el conocimiento. Apareció a mediados de la década del 2000 como una alternativa a los métodos ágiles que prevalecían en ese momento.

El Método Kanban es menos prescriptivo que algunos enfoques ágiles y menos disruptivo, ya que es el enfoque original de “comenzar con lo que se hace ahora”. Los equipos de proyecto pueden comenzar a aplicar el Método Kanban con relativa facilidad y avanzar hacia otros enfoques ágiles, si eso es lo que consideran necesario o apropiado. (Institute, 2017).

2.1.3.13 Scrumban

Lo define como una metodología derivada de los enfoques Scrum y Kanban. Esta metodología, por cierto, híbrida, contempla componentes y conceptos de ambas que se complementan entre sí, para lograr una mejor optimización del proceso de desarrollo. El término "Scrumban" fue utilizado por primera vez por Ladas (2008), en su publicación "Scrumban-Essays on Kanban System for Lean Software Development". En la actualidad, muchas organizaciones definen a Scrumban como un enfoque avanzado y orientado a la mejora del proceso de desarrollo, ya que permite adoptar una combinación de reglas que ambas metodologías por separado no permiten. Existen dos (2) líneas de pensamiento en relación con Scrumban como enfoque híbrido: (a) se destacan algunos elementos de Scrum que son aplicados directamente al enfoque de Kanban, donde el proceso es mayormente inclinado hacia Kanban y (b) algunos elementos de Kanban que son aplicados al enfoque de Scrum, donde el proceso es mayormente inclinado hacia Scrum. Además, desde una perspectiva de implementación, Scrumban permite un grado de flexibilización mayor, para partir desde una base simple y de forma gradual, llegar a una base compleja. (Kniberg et al., 2010).

2.1.4 Desarrollo Ágil De Software

En las dos últimas décadas que las notaciones de modelado y posteriormente las herramientas pretendieron ser las "balas de plata" para el éxito en el desarrollo de software, sin embargo, las expectativas no fueron satisfechas. Esto se debe en gran parte a que otro importante elemento, la metodología de desarrollo, había sido postergado. De nada sirven buenas notaciones y herramientas si no se proveen directivas para su aplicación. Así, esta década ha comenzado con un creciente interés en metodologías de desarrollo. Hasta hace poco el proceso de desarrollo llevaba asociada un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y

documentación detallada. Este esquema "tradicional" para abordar el desarrollo de software ha demostrado ser efectivo y necesario en proyectos de gran tamaño (respecto a tiempo y recursos), donde por lo general se exige un alto grado de ceremonia en el proceso. Sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad. Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, muchos equipos de desarrollo se resignan a prescindir del buen hacer de la ingeniería del software, asumiendo el riesgo que ello conlleva. En este escenario, las metodologías ágiles emergen como una posible respuesta para llenar ese vacío metodológico. Por estar especialmente orientadas para proyectos pequeños, las metodologías ágiles constituyen una solución a medida para ese entorno, aportando una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto. (Canós, 2012).

2.1.5 Manifiesto Ágil

Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.

· Desarrollar software que funciona más que conseguir una buena documentación. La regla para seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental.

· La colaboración con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

· Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso de este. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta. (Calderón et al., 2007).

En la década de los noventa surgieron metodologías de desarrollo de software ligeras, más adelante nombradas como metodologías ágiles, que buscaban reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en los proyectos de desarrollo de software. Estas metodologías nacieron como reacción a las metodologías existentes con el propósito de disminuir la burocracia que implica la aplicación de las metodologías tradicionales en los proyectos de pequeña y mediana escala.

Las metodologías tradicionales buscan imponer disciplina al proceso de desarrollo de software y de esa forma volverlo predecible y eficiente. Para conseguirlo se soportan en un proceso detallado con énfasis en planeación propio de otras ingenierías. El principal problema de este enfoque es que hay muchas actividades que hacer para seguir la metodología y esto retrasa la etapa de desarrollo.

Las metodologías ágiles tienen dos diferencias fundamentales con las metodologías tradicionales; la primera es que los métodos ágiles son adaptativos –no predictivos-.

La segunda diferencia es que las metodologías ágiles son orientadas a las personas –no orientadas a los procesos.

Las metodologías ágiles son adaptativas. Este hecho es de gran importancia ya que contrasta con la predictibilidad buscada por las metodologías tradicionales. Con el enfoque de

las metodologías ágiles los cambios son eventos esperados que generan valor para el cliente. (Navarro, 2008).

2.1.5.1 Metodología tradicional:

Las metodologías tradicionales como Al inicio el desarrollo de software era artesanal en su totalidad, la fuerte necesidad de mejorar el proceso y llevar los proyectos a la meta deseada, tuvieron que importarse la concepción y fundamentos de metodologías existentes en otras áreas y adaptarlas al desarrollo de software. Esta nueva etapa de adaptación contenía el desarrollo dividido en etapas de manera secuencial que de algo mejoraba la necesidad latente en el campo del software. Entre las principales metodologías tradicionales tenemos los ya tan conocidos RUP y MSF entre otros, que centran su atención en llevar una documentación exhaustiva de todo el proyecto y centran su atención en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto.

Las metodologías tradicionales de desarrollo de software son orientadas por planeación. Inician el desarrollo de un proyecto con un riguroso proceso de elicitación de requerimientos, previo a etapas de análisis y diseño. Con esto tratan de asegurar resultados con alta calidad circunscritos a un calendario. En las metodologías tradicionales se concibe un solo proyecto, de grandes dimensiones y estructura definida; se sigue un proceso secuencial en una sola dirección y sin marcha atrás; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto, demandando grandes plazos de planeación previa y poca comunicación con el cliente una vez ha terminado ésta. (Figueroa, et al., 2015).

2.1.5.2 Metodologías Ágiles:

La metodología ágil contempla el desarrollo de software de manera integral, con un énfasis especial en la entrega de valor al cliente, en la generación de negocio y el retorno de la inversión (ROI). Sólo hay una manera efectiva de crear software que funcione, y es de manera

colaborativa. La colaboración entre cliente y desarrolladores es indispensable: se debe fomentar y apoyar. El software puede ser visto como un juego colaborativo.

Las metodologías ágiles son flexibles, pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto.

Los proyectos ágiles se subdividen en proyectos más pequeños mediante una lista ordenada de características. Cada proyecto es tratado de manera independiente y desarrolla un subconjunto de características durante un periodo de tiempo corto, de entre dos y seis semanas. La comunicación con el cliente es constante al punto de requerir un representante de él durante el desarrollo. Los proyectos son altamente colaborativos y se adaptan mejor a los cambios; de hecho, el cambio en los requerimientos es una característica esperada y deseada, al igual que las entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente.

Se permite a los desarrolladores expandir su aportación de valor a los proyectos, y se ofrece a los clientes transparencia sobre los mismos. (Diaz , 2009).

2.1.6 Metodología

Aunque los creadores e impulsores de las metodologías ágiles más populares han suscrito el manifiesto ágil y coinciden con los principios enunciados anteriormente, cada metodología tiene características propias y hace hincapié en algunos aspectos más especificados. a la metodología como al modelo aplicable que deben necesariamente seguir los métodos de investigación, aun cuando resulten cuestionables. Es la teoría normativa, descriptiva y comparativa acerca del método o conjunto de ellos, sumado al proceder del investigador.

La mayoría de ellas ya estaban siendo utilizadas con éxito en proyectos reales, pero les faltaba una mayor difusión y reconocimiento. (Canos, 2003)

2.1.7 Desarrollo De Software

La industria del software tiene una característica especial que la diferencia de las otras industrias. Cada año (o en menos tiempo), aparecen nuevas tecnologías que rápidamente llegan a ser populares (a modo de ejemplo, se pueden listar Java, Linux, XML, HTML, UML, .NET, JSP, ASP, PHP, flash, servicios de Web, etc.). Y muchas compañías necesitan aplicar estas nuevas tecnologías por buenas razones:

- Los clientes exigen esta nueva tecnología (por ejemplo, implementaciones para web).

- Son la solución para algunos problemas reales (por ejemplo, XML para el problema de intercambio de datos).

- La empresa que desarrolla la herramienta deja de dar soporte a las viejas tecnologías y se centra sólo en las nuevas (por ejemplo, el soporte para OMT fue reemplazado por soporte para UML).

Las nuevas tecnologías ofrecen beneficios concretos para las compañías y muchas de ellas no pueden quedarse atrás. Por lo tanto, tienen que pasar a utilizarlas cuanto antes. Como consecuencia del cambio, las inversiones en tecnologías anteriores pierden valor. Por consiguiente, el software existente debe migrar a una nueva tecnología, o a una versión nueva y diferente de la usada en su construcción. También puede darse otro caso en donde el software permanece utilizando la vieja tecnología, pero ahora necesita comunicarse con sistemas nuevos, los cuales sí han adoptado las nuevas tecnologías. Todas estas cuestiones nos plantean la existencia del problema de la flexibilidad tecnológica. (Pons et al., 2010)

2.1.8 Proyectos

Menciona que: “Es muy frecuente escuchar entre los conocedores del desarrollo de software (programas de computadoras), que un gran número de los proyectos de software

fracasan por no realizar una adecuada definición, especificación, y administración de los requerimientos.

Dentro de esa mala administración se pueden encontrar factores como la falta de participación del usuario, requerimientos incompletos y el mal manejo del cambio a los requerimientos.

Es necesario dar a conocer que alrededor del mundo existen estándares enfocados en el mejoramiento de los procesos de desarrollo de software, citando entre ellos a los estándares propuestos por la IEEE (Instituto de Ingenieros Eléctricos y Electrónicos), el SEI (que propone el modelo de capacidad de madurez, mejor conocido como CMM), el PMI (Project Management Institute, que ofrece certificaciones para el área de administración del proyectos) y las ya conocidas normas ISO, en cuyas normas también se involucran apartados referentes al desarrollo de software." (Arias, 2006)

2.1.9 Requerimientos

A través de los años se ha podido constatar que el levantamiento de requerimientos o requisitos son la pieza fundamental en un proyecto de desarrollo de software, ya que marcan el punto de partida para actividades como la planeación, principalmente en lo que se refiere a las estimaciones de tiempos y costos, así como la definición de recursos (mano de obra, infraestructura, etc.) necesarios y la elaboración de cronogramas que será uno de los mecanismos primordiales de control con los que se contará durante la etapa de desarrollo. Además, la especificación de requerimientos es la base que permite verificar si se alcanzaron o no los objetivos establecidos en el proyecto ya que estos son un reflejo detallado de las necesidades de los clientes o usuarios del sistema y es contra lo que se va a estar verificando si se están cumpliendo las metas trazadas.

1. “Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo”.

2. “Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal”. También, Ian Sommerville presenta una definición acerca de lo que es un “Requerimiento”:

3. “Un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste”.

Analizando las definiciones anteriores, un requerimiento es una descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso.

Es importante no perder de vista que un requerimiento debe ser: Especificado por escrito: Como todo contrato o acuerdo entre dos partes. Posible de probar o verificar. Si un requerimiento no se puede comprobar, entonces ¿cómo se sabe si se cumplió con él o no? Conciso: Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro. Completo: Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión. Consistente: Un requerimiento es consistente si no es contradictorio con otro requerimiento. No ambiguo: Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición no debe causar confusiones al lector. (Fiquitiva et al., 2015)

2.1.10 Cmmi.

En las estadísticas que menos del 20 % de los proyectos se completan en costes, plazos, alcance y nivel de calidad. Los factores que influyen son muchos y los puntos más importantes

del fracaso, son: Falta de experiencia generalizada; poco realismo de los planes de negocio; cultura de administración de los proveedores y exceso de neotecnicismo en los desarrolladores. Cuando se habla de procesos de desarrollo de software, no se refiere a temas puramente técnicos porque está demostrado que la mayoría de los problemas son organizativos. En objetivo consiste en mejorar los procesos de desarrollo de software de modo tal que los proyectos sean más predecibles (tiempo y costes), se reduzcan los riesgos en el desarrollo con el consiguiente ahorro de costes. En muchas organizaciones los principales técnicos han ido ocupando labores de responsabilidad sin haber sido correctamente preparados: Técnicamente pueden estar cualificados, pero presentan graves deficiencias en labores de gestión. El problema fundamental es que se han consolidado en las empresas procesos informales y poco estructurados, que propician un desarrollo poco predecible y repetible. Por ende, se requiere de elementos que brinden apoyo a la organización tales como CMMI (de las siglas en inglés Capability Maturity Model Integration).

El modelo CMMI fue adoptado por el área de defensa del gobierno americano, para asegurarse de que sus proveedores cumplieran los criterios mínimos de calidad, y exigir que estuvieran certificados en CMM. Dado el éxito del modelo, se extendió a otras disciplinas como la ingeniería de sistema, adquisición de material, etc., creándose posteriormente variaciones del CMM.

La metodología CMM se ha ampliado y modificado, ahora ha aparecido CMMI, que es una evolución de CMM que integra los distintos modelos de calidad que sirvieron de base para el modelo CMMI. (Huayta, 2006).

2.1.11 Metodología Asap

menciona que la metodología ASAP marca una pauta a seguir para abarcar todos los pasos para tener éxito en el proyecto.

Sin duda alguna vivimos en un mundo en donde la globalización provoca cambios constantes en la tecnología y la ciencia, por tanto, las organizaciones, así como los negocios deben de contar con un proceso formal para su administración, con el cual puedan poner como base y objetivo la innovación.

Como respuesta a estos múltiples cambios, gran parte de las empresas y organizaciones se están orientando a confiar sus sistemas de información a paquetes estándar pre-configurados como son los ERP (Enterprise Resource Planning). Estas soluciones se basan en módulos de software de aplicación que ayudan a gestionar las partes importantes del negocio, como lo son Ventas, Producción, Gestión de Materiales, Mantenimiento y recientemente se observa como incluyen, dentro de su estándar, las más novedosas tecnologías; Internet, Workflow, Gestión Documental, etc.

Dentro de los ERP se destaca el producto R/3 de SAP, que tanto por su tecnología como por su cuota de mercado está llamado a convertirse en el Standard empresarial. Uno de los grandes retos del R/3 es precisamente lograr una exitosa implementación dentro de las organizaciones, esto es de vital importancia ya que de esto depende que se eviten situaciones que generen insatisfacción en la empresa cliente lo cual puede poner en peligro un proyecto extenso tanto en tiempo como en recursos humanos y monetarios; por ello es necesaria la elaboración de un proyecto inicial de Diseño Conceptual de SAP R/3.

La implantación estándar de SAP consiste en implantar aquellos módulos que se necesiten para llevar a cabo los procesos de nuestra empresa según la metodología. Es posible configurarlos todos o parte de ellos, según nuestras necesidades.

El sistema es escalable, lo que quiere decir, que nosotros podemos implantar un módulo de ventas por ejemplo y más adelante implantar el de recursos humanos. Eso sí, hay algunos

que son obligados por así decirlos como el de finanzas que necesita tener creada una jerarquía y configurarlo para implantarlo en la empresa.

Para asegurar el éxito del cambio tecnológico, SAP propone su propia metodología de implementación llamada Accelerated SAP (ASAP), esto a razón de que una implementación de SAP está sujeta a múltiples factores, tanto tecnológicos como funcionales y organizativos (gestión de equipos, conocimiento del producto, reingeniería, dimensionamiento de máquinas, estructuras y análisis coste/beneficio).

A modo de resumen, se puede decir que SAP ya tiene hecho de antemano parte del trabajo de implantación, y ha empaquetado sus soluciones en las siguientes herramientas:

Accelerated SAP. Solución completa para la introducción de SAP R/3 en la compañía. ASAP y sus herramientas se pueden utilizar de forma independiente. Las principales herramientas son las siguientes:

• Project Estimator. Herramienta interna de gestión de proyectos.

• Ruta ASAP o Roadmap. La hoja de ruta que contiene las cinco fases de la implantación, con sus tareas y actividades, junto con el Plan de Proyecto.

• Question and Answer Database (Q&Adb). Base de datos de preguntas y respuestas. Un cuestionario para realizar al cliente en la fase de toma de requisitos, a fin de que el documento de análisis Bussines Blueprint sea lo más completo posible.

Fases del Proyecto:

Fase 1: Preparación del proyecto (Análisis)

En esta fase de la Hoja de ruta ASAP, los responsables de la toma de decisiones deben definir de forma clara los objetivos del proyecto. Se hace una primera planificación general del proyecto, se define y construye el equipo de proyecto y el entorno de trabajo.

El primer paso para los jefes de proyecto es establecer el proyecto de implantación. Se marcan una línea general, se junta el equipo de proyecto y se fija una reunión de kickoff. Este kickoff es crítico, porque es aquí donde el equipo de proyecto y los dueños de los procesos (cliente) visualizan juntos los objetivos que se pretenden conseguir y definen las responsabilidades de cada uno.

Fase 2: Business Blueprint

En esta fase, se define y documenta de forma detallada el alcance del proyecto de implantación, una vez terminadas las reuniones de trabajo. También se genera el Business Blueprint, que es un documento en formato Word con todos los requisitos de la compañía completamente detallados.

El equipo de consultores funcionales junto al equipo de negocio han de lograr un entendimiento común sobre cómo la empresa va a llevar a cabo sus procesos de negocio dentro del sistema R/3, a través de las reuniones de trabajo llamadas Business Blueprint Workshops. El siguiente esquema lo muestra perfectamente:

Fase 3: Realización

El objetivo de esta fase es que el sistema R/3 quede configurado y parametrizado, a fin de obtener una solución integrada y documentada que cumpla todos los requerimientos de negocio definidos previamente. La configuración del sistema se lleva a cabo en dos etapas, dentro de esta fase, configuración básica (Baseline) y configuración Final.

La configuración básica consiste en implementar alrededor del 80% de las transacciones de negocio diarias y completar la estructura organizativa y la carga de datos maestros. La configuración Final se realiza de forma cíclica, orientada a los procesos de negocio. El Business Blueprint se utiliza como guía para la configuración/parametrización del sistema, que se realiza a través de la Guía de Implementación IMG. Una vez terminada esta

tarea, se debe proceder a testear todos los desarrollos que se hayan realizado en el sistema, interfaces, programas de carga de datos, programas a medida.

Fase 4: Preparación Final

El objetivo de esta fase es completar la preparación final del sistema R/3 para salir a producción. Aquí se incluyen entre otras cosas las pruebas, la formación a usuarios, administración del sistema, preparación del corte (fechas, si se hace o no paralelo…) y prepararse para la puesta en producción.

En esta preparación final también se deben cerrar todos los puntos abiertos cruciales en el desarrollo del proyecto. Si esta fase se finaliza de forma completa y correcta, ya se está preparado para poner el sistema en producción. En esta fase, los usuarios finales recibirán la formación de la forma más completa posible. El último paso será la migración de datos al nuevo sistema.

Fase 5: Salida en vivo – Soporte

En este momento todo el mundo está ya preparado para comenzar a trabajar de forma real, con el sistema totalmente en producción. Posteriormente, el equipo de proyecto se centra en dar soporte al usuario final, ya que posiblemente la formación aún no haya finalizado, y aunque haya terminado ya la formación reglada, hasta que el usuario no se enfrente a problemas reales, debe tener el soporte de expertos.

También es necesario establecer procedimientos y medidas para revisar los beneficios de la inversión en R/3. (Enrich, 2013)

2.1.12 Norma Iso 31010

La norma de soporte proporciona directrices para la selección y aplicación de técnicas sistemáticas para la evaluación del riesgo.

Todas las actividades de una organización implican riesgos que se deberían gestionar. El proceso de gestión del riesgo ayuda a tomar decisiones teniendo en cuenta la incertidumbre y la posibilidad de futuros eventos o circunstancias (previstas o imprevistas) y sus efectos sobre los objetivos acordados.

La gestión del riesgo incluye la aplicación de métodos lógicos y sistemáticos para:

- comunicar y consultar a lo largo de este proceso.

- establecer el contexto para la identificación, análisis, evaluación, tratamiento del riesgo asociado con cualquier actividad, proceso, función o producto.

- realizar seguimiento y revisar los objetivos.

- informar y registrar los resultados de manera apropiada.

Esta norma es de carácter general, por lo que puede proporcionar directrices a seguir por numerosas industrias y diferentes tipos de sistemas. En estas industrias pueden existir normas más específicas que establezcan metodologías y niveles de evaluación preferentes para aplicaciones particulares. Si tales normas están en armonía con esta norma, las normas específicas generalmente serán suficientes.

Propósitos y beneficios:

Entre los principales beneficios de realizar la evaluación del riesgo, se incluyen:

- comprender el riesgo y su impacto potencial sobre los objetivos.

- proporcionar información a quienes toman decisiones.

- contribuir a comprender los riesgos, para ayudar en la selección de las opciones de tratamiento.

- identificar los factores principales que contribuyen a los riesgos, y los puntos débiles en los sistemas y organizaciones.

- comparar los riesgos en sistemas, tecnologías o enfoques alternativos; - comunicar los riesgos y las incertidumbres.

- ayudar a establecer prioridades.

- contribuir a la prevención de incidentes basados en investigaciones posteriores de incidentes; - seleccionar diferentes formas de tratamiento del riesgo.

- cumplir los requisitos reglamentarios.

- evaluar los riesgos unidos al final de la vida útil.

La evaluación del riesgo se puede realizar con diferentes grados de profundidad y de detalle, y utilizando uno o varios métodos que varían desde simples a complejos. La forma de la evaluación y de sus resultados debe ser consecuente con los criterios de riesgo desarrollados como parte del establecimiento del contexto. ISO/IEC 31010 (2013)

2.1.13 Ciclo De Demming

El Ciclo PDCA es la sistemática más usada para implantar un sistema de mejora continua. A continuación, vamos a explicar qué es lo que representa, cómo funciona y su estrecha relación con algunas normas ISO, concretamente con la ISO 9001 "Requisitos de los Sistemas de gestión de la calidad", donde aparece mencionado como un principio fundamental para la mejora continua de la calidad.

El nombre del Ciclo PDCA (o Ciclo PHVA) viene de las siglas Planificar, Hacer, Verificar y Actuar, en inglés "Plan, Do, Check, Act". También es conocido como Ciclo de mejora continua o Círculo de Deming, por ser Edwards Deming su autor. Esta metodología describe los cuatro pasos esenciales que se deben llevar a cabo de forma sistemática para lograr la mejora continua, entendiendo como tal al mejoramiento continuado de la calidad (disminución de fallos, aumento de la eficacia y eficiencia, solución de problemas, previsión y eliminación de riesgos potenciales…). El círculo de Deming lo componen 4 etapas cíclicas, de forma que una vez acabada la etapa final se debe volver a la primera y repetir el ciclo de nuevo, de forma que las actividades son reevaluadas periódicamente para incorporar nuevas mejoras. La aplicación de esta metodología está enfocada principalmente para para ser usada en empresas y organizaciones. (Bernal, 2013).

III. MÉTODO

3.1 Tipo De Investigación

El presente estudio cuenta con la intervención del investigador por ello es una investigación cuasi experimental, si habrá un cambio porque se realizará un experimento, se va a intervenir la variable independiente que es el nuevo Modelo Lacs con el objetivo de mejorar el proyecto de implementación del Sistema SAP, según el resultado es un estudio de intervención porque el investigador introduce variables de estudio, interviniendo en la realidad y desarrollo del mismo, se usara el estudio cuasiexperimental porque trabaja con grupos establecidos, por ese motivo se va a comparar el resultado del Pre-Test y el Post-Test.

Para ello se realizará en primer lugar, se realizará un Pre-Test para revisar como está la situación actual antes de aplicar el experimento, en segundo lugar, se aplicará el experimento y en tercer lugar se volverá a medir y se obtendrá el Post-Test, se va a medir el cambio ocasionado por el experimento, es lo que se llama Causa – Efecto.

La causa será aplicar el nuevo Modelo Lacs y el efecto será conseguir una mejora en la implementación del sistema SAP.

3.1.1 Diseño De La Investigación

Se utilizo la investigación cuasi-experimental de grupos definidos, con un diseño longitudinal que permite estudiar el efecto de cambio, se usó el Pretest y Postest, así mismo se utilizó un grupo experimental no aleatorio, la población de estudio comprendió a 10 integrantes entre analistas funcionales y desarrolladores. En la primera etapa del Pretest se entrevistó a 10 integrantes del equipo y en la segunda etapa Postest se entrevistó a los mismos 10 integrantes del equipo. Las variables utilizadas fueron la ejecución de proyecto del Sistema SAP y el nuevo modelo LACS.

Sobre el diseño cuasi-experimental podemos citar lo propuesto por Terry Hedrick, Leonard Bickman y Debra Rog, donde señalan que los diseños cuasi experimentales tienen el mismo propósito que los estudios experimentales: probar la existencia de una relación causal entre dos o más variables. Cuando la asignación aleatoria es imposible, los cuasi experimentos (semejantes a los experimentos) permiten estimar los impactos del tratamiento o programa. (Hedrick et al.,1993).

También podemos citar la fundamentación de la metodología cuasi experimental ofrecida por Pedhazur, E. y L. Schmelkin, donde indican: (Pedhazur et al., 1991) que un cuasi experimento es una investigación que posee todos los elementos de un experimento, excepto que los sujetos no se asignan aleatoriamente a los grupos. En ausencia de aleatorización, el investigador se enfrenta con la tarea de identificar y separar los efectos de los tratamientos del resto de factores que afectan a la variable dependiente.

ambos consideran a los cuasi experimentos una alternativa a los experimentos aleatorios, los cuasi experimentos son como experimentos de asignación aleatoria en todos los aspectos, excepto en que no se puede presumir que los diversos grupos de tratamiento sean inicialmente equivalentes dentro de los límites del error muestral. (Campbell et al.,1986).

Respecto al diseño longitudinal tiene como propósito, el estudio del cambio de un mismo grupo de sujetos entre dos ocasiones de observación (como consecuencia de algún hecho circunstancial, tratamiento, o por el simple paso del tiempo). (Bono Cabré R.), se utilizará el diseño longitudinal para medir la mejora la ejecución del proyecto de implementación utilizando el nuevo modelo LACS.

El diseño Longitudinal se muestra de la siguiente manera:

O1	X	O2

X: Es el nuevo método LACS para ser utilizado en la implementación del proyecto.

O1: Es el Pre-Test del grupo no aleatorio que realizara la ejecución del proyecto sin utilizar el nuevo modelo LACS.

O2: Es el Pos-Test del grupo no aleatorio que realizo la ejecución del proyecto utilizando el nuevo modelo LACS.

Como el diseño es cuasi experimental, manipulamos limitadamente la variable independiente X: El nuevo sistema LACS, para observar el efecto sobre la variable dependiente: Ejecución de proyectos de implementación de Sistema SAP.

3.2 Población y Muestra

Población:

Está formada por todos los profesionales que participaron de la implementación del Sistema SAP en la Universidad de San Martin de Porres en el año 2015, comprendida por 10 integrantes entre analistas funcionales y desarrolladores.

Muestra:

Para la muestra se consideró a 10 profesionales que participaron en la implementación del Sistema SAP que es el total de la población, por ser una población pequeña se realizó la encuesta a todos.

3.3 Operacionalización De Variables

Variable Independiente

- Nuevo Modelo LACS para la implementación del proyecto.

Variable Dependiente

- Ejecución del proyecto de implementación del sistema SAP.

Operacionalidad de las variables.

Tabla 1.

Operacionalización de Variables.

VARIABLES	DEFINICIÓN	DIMENSIONES	INDICADORES
Variable Dependiente: Ejecución de proyecto de implementación del Sistema SAP.	La ejecución del proyecto de implementación permitirá instalar el código y la configuración de los módulos de SAP.	Entrega oportuna del proyecto.	Cantidad de pruebas unitarias válidas.
		Productividad del Equipo del Proyecto.	Capacitación del personal actual, funcional y técnico.
		Entrega de la documentación completa.	Cantidad de documentación técnica y manuales.
Variable Independiente: Nuevo Modelo LACS.	Es el Nuevo Modelo que se ha creado llamado LACS	Costo del proyecto	Monto real que tiene el proyecto.
		Desempeño según cronograma.	Variación del cronograma.

		Comparación de los tiempos.	Índice del tiempo estimado comparado con el tiempo real.

3.4 Instrumentos

En la presente investigación se utilizará el cuestionario para recolectar los datos, se creara el formulario de Google para crear las encuestas por internet, el cual estará formulado según la escala de Likert, cualitativa ordinal de 5 alternativas, las preguntas serán de tipo cerradas, el cual nos permitirá medir actitudes y conocer el grado de conformidad del encuestado, resulta especialmente útil empleada en situaciones en las que queremos que la persona matice su opinión, en ese sentido, las categorías de las respuestas nos servirán para capturar la intensidad del encuestado hacia dicha afirmación.

A través de la técnica de juicio de expertos se realizará la validación de instrumentos, en esta prueba participaron 10 expertos que estuvieron en el proyecto de implementación de Sistema SAP en la empresa.

Se utilizo el Coeficiente de validez de Aiken, para verificar la relación que existe entre los indicadores y las preguntas del cuestionario.

El coeficiente de Aiken es el más adecuado para determinar el tipo de validez, ya que permite obtener valores factibles de ser contrastados estadísticamente según el tamaño de la muestra de jueces seleccionada. Este coeficiente combina la facilidad del cálculo y la evaluación de los resultados a nivel. (Escurra,1988), Una vez calculados los estadísticos de consistencia procedimos a revisar la adecuación de los ítems a los criterios de validez de contenido adoptados, mediante el coeficiente de validación V de Aiken. (Aiken,1985).

Coeficiente de Aiken

$$V = \frac{S}{(n(c\ \ 1))}$$

Donde:

S: sumatoria de si

Si: valor asignado por el juez i

N: número de jueces

c: número de valores en la escala de valoración.

3.4.1 Confiabilidad del Instrumento

Los datos que se obtienen con el instrumento que se utilizó para la recolección de datos en la evaluación de la "Nueva Metodología LACS", serán utilizados en el Alfa de Cronbach que representa la consistencia interna de la prueba, es decir el grado en que todas las preguntas de la prueba covarían entre sí.

Se utilizará el coeficiente Alfa de Cronbach para cuantificar el nivel de fiabilidad de la escala de medida.

Lo deseable para crear una escala fiable es que exista una fuerte correlación entre los items, el Alfa de Cronbach oscila entre el 0 y 1, cuanto más próximo este a 1, más consistentes serán los item entre sí, se debe tener presente que, a mayor longitud del test, será mayor el Alfa. (Martínez et al, 2014) menciona que cuanto más se aproxime a su valor máximo 1 mayor es la fiabilidad de la escala, se considera que valores de alfa superiores a 0,7 o 0,8 son suficientes para garantizarnos la fiabilidad de la escala que estamos utilizando. (Cohen et al., 2002), la confiabilidad del instrumento se muestra en el anexo 9.7.

3.5 Procedimientos

La aplicación Formularios de Google que se encuentra dentro de Google Drive. Con ella, se creará el formulario con las preguntas que se considere y después se enviará a los jurados expertos a sus cuentas de correo para que ingresen su información y puedan responder las preguntas, para luego agregarlo al SPSS.

Las respuestas a la encuesta se recopilan de forma automática y ordenada en Formularios, con gráficos y datos de las respuestas en tiempo real. También se puede analizar los datos en más profundidad si se exporta a una Hoja de cálculo Excel.

Se realizará una encuesta en la etapa Pre-Test y Post-Test, enviando el formulario a sus cuentas de correo del jurado experto, luego se realizará la prueba de hipótesis.

El estadístico Chi Cuadrado o ji Cuadrado, es una prueba estadística para evaluar hipótesis acerca de la relación entre dos variables categóricas, pertenecientes a un nivel de medición ordinal.

La prueba de Chi Cuadrado parte del supuesto de que las dos variables no están relacionadas, hay independencia.

En la hipótesis nula se establece, la independencia de las variables, en la hipótesis alternativa, se establece que las variables están relacionadas.

Ho: Independencia de las variables.

H1: Variables relacionadas.

El símbolo alfa hace referencia al nivel de significación y adopta uno de estos 2 valores, el 0,01 o el 0,05, es decir para el nivel de significación uno establece el nivel de significación el 0,01 o el 0,05.

El nivel de significación indica la probabilidad de cometer un error al rechazar la hipótesis nula, en el supuesto de que esta hipótesis sea cierta, cuando el estadístico Chi cuadrado, tiene una probabilidad menor o igual a 0,05 se rechaza la hipótesis de independencia, es decir la relación entre variables para tomar la hipótesis alternativa e indica que la relación entre las variables es estadísticamente significativa.

3.6 Análisis De Datos

Una vez terminadas la encuesta de los expertos se procederá a consolidar la información para ello se utilizará las Hojas de Cálculo de Excel y Software SPSS. Se realiza una verificación, clasificación y registro de datos.

Se utiliza el estadístico del chi cuadrado en las pruebas estadísticas, con un nivel de significación de 0.05 de probabilidad.

Se utilizo el mismo análisis para el estudio Pre-Test y Post-Test, según la investigación.

3.7 Consideraciones Éticas

La presente investigación encuentra su fundamento y sus antecedentes en documentos de ámbito internacional, entre ellos destaca el Código de Núremberg (1947) este código establece como principio básico el respeto por los derechos de las personas que de alguna manera forman parte de los experimentos.

En segundo lugar, se considera a la Declaración de Helsinki (2008), la cual menciona que los intereses de los participantes humanos que son parte de la investigación deben tener prioridad en todo momento, que los integrantes deben recibir el mejor trato posible y que en las instituciones se debe crear comités de ética para que regulen la producción científica en el aspecto ético.

Un tercer punto es el pacto internacional de Derechos civiles (1976), que fue adoptado por la Asamblea General de las Naciones Unidas, el pacto menciona que garantiza el respeto

irrestricto de los derechos humanos, que debe ser resguardado por parte de los investigadores científicos.

Un cuarto documento es el informe Belmont (1979) indica que los principios éticos que deben guiar a toda conducta de investigación con los seres humanos.

En tal sentido, la conducta del autor en la investigación es de responsabilidad sobre el nuevo modelo LACS basado en Lean IT y Desarrollo Ágil para mejorar la ejecución de proyectos de implementación de sistemas SAP.

IV. RESULTADOS

4.1 Análisis de resultados

¿Cuál es su opinión respecto al tiempo que se toma entre la corrección de una función del desarrollador implementador y el inicio de las pruebas unitarias que realiza el funcional del cliente?

Tabla 2.

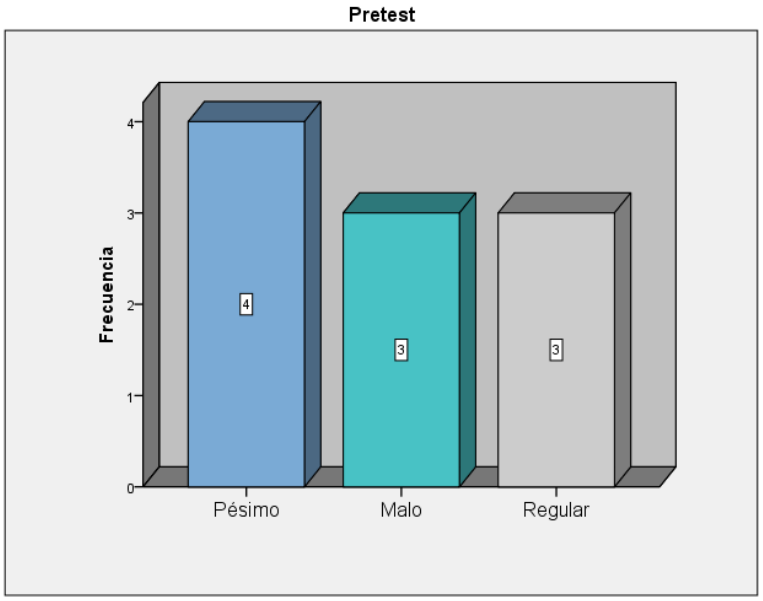
El tiempo que toma entre la corrección y el inicio de las pruebas - Pre-Test

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Pésimo	4	40,0	40,0	40,0
	Malo	3	30,0	30,0	70,0
	Regular	3	30,0	30,0	100,0
	Total	10	100,0	100,0	

Según los resultados que se muestran en la Tabla 2 y Figura 1 en la etapa Pre-Test se puede apreciar que el 40% de los entrevistados opinan como pésimo el tiempo que se toma entre el término de una corrección del programador y el inicio de las pruebas unitarias del funcional del cliente, 30% opina que es malo y un 30% que es regular el tiempo que demoran, es necesario indicar que los entrevistados participaron del proyecto de implementación del ERP y conocieron de cerca las causas que originaron muchos inconvenientes a lo largo de todo el proyecto.

Figura 1.

El tiempo que toma entre la corrección y el inicio de las pruebas - Pre-Test.



¿Cuál es su opinión respecto al tiempo que se toma entre la corrección de una función del desarrollador implementador y el inicio de las pruebas unitarias que realiza el funcional del cliente?

Tabla 3.

El tiempo que toma entre la corrección y el inicio de las pruebas – Pos-Test

Pos-Test

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Regular	3	30,0	30,0	30,0
	Bueno	4	40,0	40,0	70,0
	Excelente	3	30,0	30,0	100,0
	Total	10	100,0	100,0	

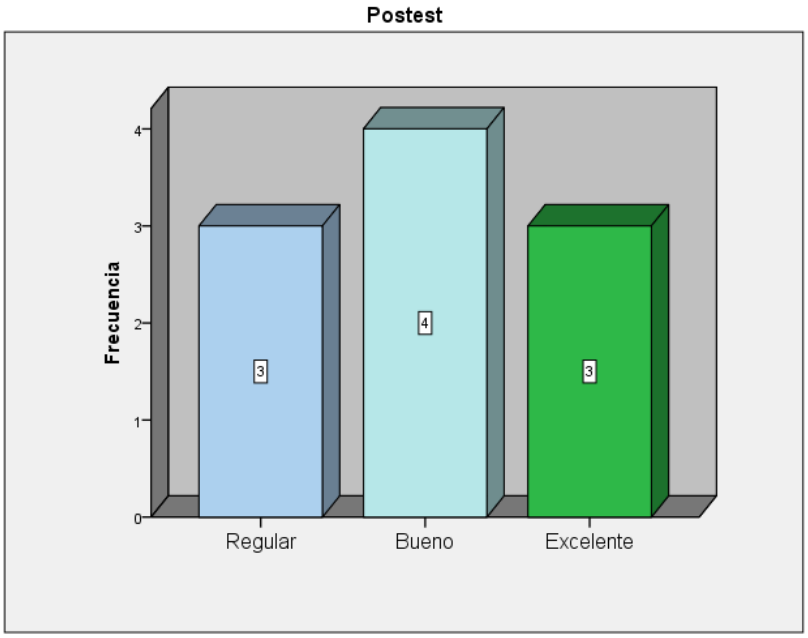
Según los resultados de la Tabla 3 y la Figura 2 se aprecia que en la etapa del Pos-Test aparece la alternativa Bueno con un 40%, seguido de Regular 30% y Excelente 30% por el

motivo que en la etapa Pos-Test se utiliza el nuevo modelo que acorta el tiempo entre el término del desarrollo de la función y el inicio de las pruebas unitarias del funcional, con la intención de realizar una labor sincronizada, sin perder demasiado tiempo para iniciar las pruebas e inmediatamente realizar la corrección si es necesaria.

Es necesario indicar que aplicar la nueva metodología LACS en este caso, es un gran apoyo para reducir el tiempo, continuar con el proyecto y cumplir con el cronograma establecido.

Figura 2.

El tiempo que toma entre la corrección y el inicio de las pruebas – Pos-Test.



¿Qué opinión tiene cuando el desarrollador implementador envía al funcional cliente todas las pruebas en bloque para que inicie con sus pruebas unitarias en el ambiente de Calidad?

Tabla 4.

Se envía todas las pruebas en bloque para iniciar las pruebas – Pre-Test

Pretest

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Pésimo	3	30,0	30,0	30,0
	Malo	4	40,0	40,0	70,0
	Regular	3	30,0	30,0	100,0
	Total	10	100,0	100,0	

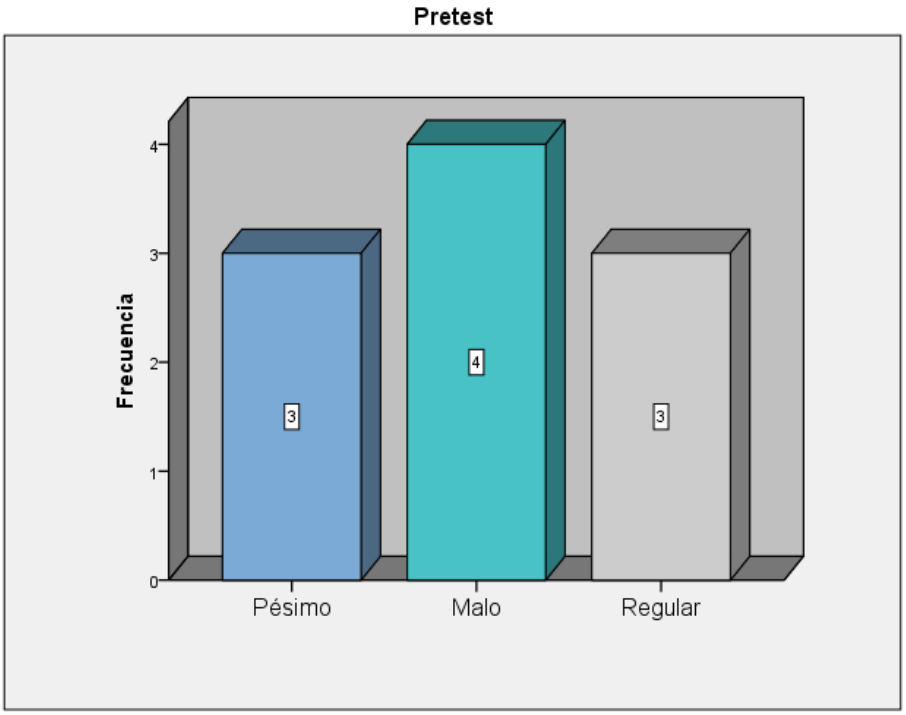
Según la Tabla 4 y en la Figura 3 en la etapa del Pre-Test se observa que el envío en bloque para el inicio de las pruebas unitarias, tiene un 40% que considera que es Malo esta decisión, también se observa un 30% que considera Pésimo y un 30% que es Regular, porque afecta el inicio de las pruebas y la correcta validación de los cambios porque el funcional del cliente tiene poco tiempo para empezar a realizar todas las pruebas unitarias y funcionales que necesita, si el funcional encuentra observaciones tendrá que nuevamente enviar los errores al desarrollador que está implementando los cambios y es allí donde el tiempo del proyecto se puede extender.

El funcional encargado de las pruebas unitarias y funcionales no puede realizar todas las pruebas que ha corregido el desarrollador y adicionar las pruebas que el funcional considera necesarias según su experiencia, muchas veces debe limitarse a realizar solo las pruebas que el desarrollador envío y por el corto tiempo no podrá realizar algunas pruebas particulares que el gustaría realizar.

Es importante mencionar que la decisión de enviar todo el set de pruebas en bloque para que el funcional inicie sus pruebas es un riesgo para la calidad del proyecto.

Figura 3.

Se envía todas las pruebas en bloque para iniciar las pruebas – Pre-Test.



¿Qué opinión tiene cuando el desarrollador implementador envía al funcional cliente todas las pruebas en bloque para que inicie con sus pruebas unitarias en el ambiente de Calidad?

Tabla 5.

Se envía todas las pruebas en bloque para iniciar las pruebas – Pos-Test

Pos-test

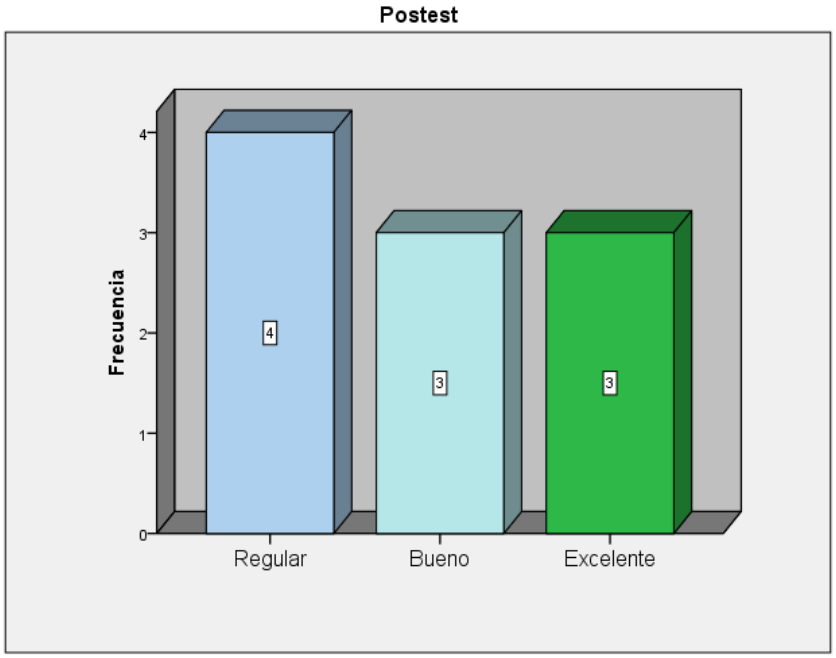
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Regular	4	40,0	40,0	40,0
	Bueno	3	30,0	30,0	70,0
	Excelente	3	30,0	30,0	100,0
	Total	10	100,0	100,0	

Según los resultados de la Tabla 5 y la Figura 4 en la etapa del Pos-Test se puede encontrar que usando el método LACS de una revisión en secuencia entre el desarrollador implementador y el funcional encargado de las pruebas unitarias que va realizando cada prueba conforme el desarrollador va terminando, así tiene tiempo para la prueba unitaria y agregar alguna otra prueba que considere necesaria hasta que la siguiente corrección del desarrollador este lista para probar, es así que tenemos un 40% que considera como Regular, un 30% como Bueno y un 30% Excelente el hecho de un trabajo sincronizado, coordinado y en línea entre el desarrollador implementador y el funcional del cliente.

Adicionalmente el desarrollador del cliente deberá revisar los cambios y realizar la documentación respectiva, indicando nombre de tablas, funciones, métodos y clases que se estén utilizando para la corrección de la implementación.

Figura 4.

Se envía todas las pruebas en bloque para iniciar las pruebas – Pos-Test.



¿Cómo considera realizar las pruebas unitarias de un sistema ERP en un tiempo muy corto, podría ocasionar errores en las pruebas funcionales?

Tabla 6.

Realizar todas las pruebas unitarias en un tiempo muy corto – Pre-Test.

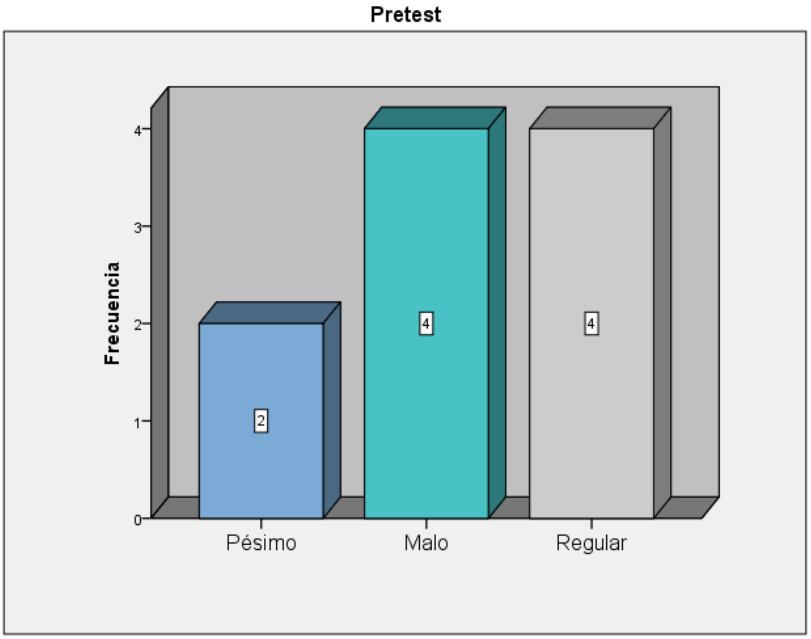
Pretest

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Pésimo	2	20,0	20,0	20,0
	Malo	4	40,0	40,0	60,0
	Regular	4	40,0	40,0	100,0
	Total	10	100,0	100,0	

Según la Tabla 6 y la Figura 5 en la etapa de Pre-Test se puede observar que se considera como Malo en un 40% el tiempo muy corto para realizar todas las pruebas unitarias necesarias y que puede impactar en más errores en las pruebas funcionales, porque un cambio en alguna función, clase o método en las pruebas unitarias ocasiona que se vuelvan a probar todas las pruebas funcionales, se observa que un 40% considera como Regular y un 20% como Pésimo el tiempo muy corto para realizar esta actividad que es crítica para asegurar la calidad del proyecto.

Figura 5.

Realizar todas las pruebas unitarias en un tiempo muy corto – Pre-Test.



¿Como considera realizar las pruebas unitarias de un sistema ERP en un tiempo muy corto, podría ocasionar errores en las pruebas funcionales?

Tabla 7.

Realizar todas las pruebas unitarias en un tiempo muy corto – Pos-Test.

Postest

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Regular	5	50,0	50,0	50,0
	Bueno	3	30,0	30,0	80,0
	Excelente	2	20,0	20,0	100,0
	Total	10	100,0	100,0	

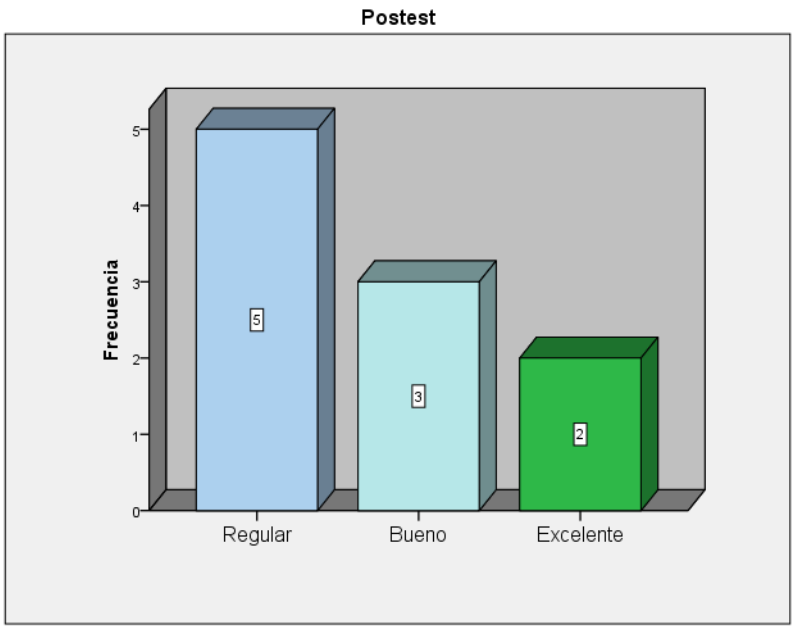
Según los resultados de la Tabla 7 y la Figura 6 en la etapa de Pos-Test se utiliza el método LACS para que el tiempo asignado para las pruebas unitarias y funcionales sea aprovechado al máximo y no tener tiempos muertos, cuando el funcional del cliente está esperando para iniciar con las pruebas unitarias y después las recibe todas juntas con el tiempo corto para realizar todas sus pruebas unitarias y después empezar con las pruebas funcionales,

es por ese motivo que el uso del método LACS va a servir para que las pruebas unitarias y funcionales se realicen sin perder tiempo, sincronizadas con el avance del desarrollador implementador.

De esta manera se plantea que el avance sea coordinado entre el desarrollador implementador, el funcional del cliente y el desarrollador del cliente, cada vez que el desarrollador termine de corregir una función, método o clase, inmediatamente será revisando por el funcional del cliente con una prueba unitaria y la respectiva prueba funcional en todo el sistema, de la misma forma el desarrollador del cliente ya estará documentando todos los cambios a nivel de código y a nivel de alguna configuración que se realizó en el módulo, de esta manera el aprendizaje para el personal del cliente también se beneficia para poder responder en los futuros cambios cuando la consultora termine con la implementación del producto en la empresa.

Figura 6.

Realizar todas las pruebas unitarias en un tiempo muy corto – Pos-Test.



¿Considera que las pruebas unitarias que realizara el funcional del cliente son de alta calidad, por recibir todas las pruebas juntas en bloque, cuando el desarrollador termino con todas las correcciones?

Tabla 8.

Recibir todas las pruebas en bloque garantiza una alta calidad – Pre-Test.

Pretest

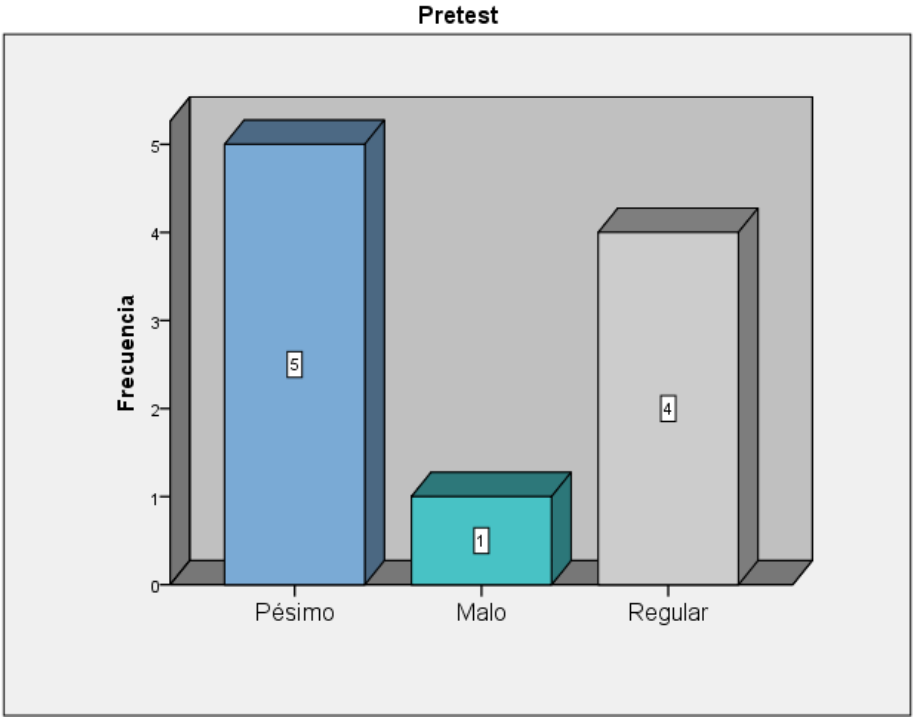
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Pésimo	5	50,0	50,0	50,0
	Malo	1	10,0	10,0	60,0
	Regular	4	40,0	40,0	100,0
	Total	10	100,0	100,0	

Según la Tabla 8 y la Figura 7 en la etapa del Pre-Test se observa que los encuestados consideran la calidad de las pruebas unitarias como Pésimo en un 50%, como Regular un 40% y como Malo en un 10%, esto es debido a que el funcional del cliente no tiene el tiempo y la tranquilidad necesaria para empezar a realizar todas sus pruebas unitarias y funcionales, que necesita como responsable de dar la aprobación a todos los cambios que está realizando la consultora que implementa el sistema en el cliente.

Es de mucha responsabilidad tener que realizar la prueba unitaria y funcional, tener que aprobar una modificación al sistema en un corto tiempo o tener que rechazar la modificación y tener que enviar a la consultora la observación con la evidencia de sus pruebas para que el desarrollador implementador nuevamente revise el error y realice la corrección del programa, luego de la corrección el funcional del cliente tendrá que realizar nuevamente la prueba unitaria y funcional para aprobar el cambio realizado, si vemos que todas las pruebas son recibidas en bloque por el funcional podemos suponer que la calidad de sus pruebas no serán las esperadas.

Figura 7.

Recibir todas las pruebas en bloque garantiza una alta calidad – Pre-Test.



¿Considera que las pruebas unitarias que realizara el funcional del cliente son de alta calidad, por recibir todas las pruebas juntas en bloque, cuando el desarrollador termino con todas las correcciones?

Tabla 1.

Recibir todas las pruebas en bloque garantiza una alta calidad – Pos-Test.

Postest

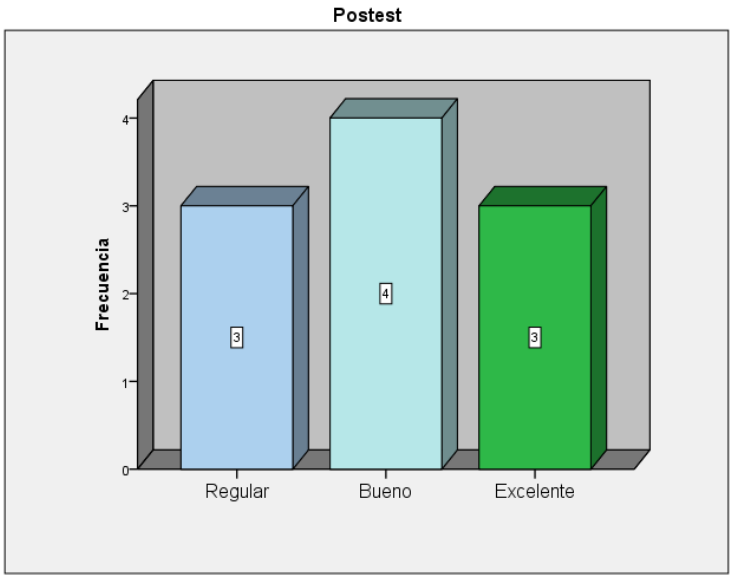
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Regular	3	30,0	30,0	30,0
	Bueno	4	40,0	40,0	70,0
	Excelente	3	30,0	30,0	100,0
	Total	10	100,0	100,0	

Según la Tabla 9 y la Figura 8 en la etapa de Pos-Test usando el método LACS se puede se observa que los encuestados consideran como Bueno a un 40%, se observa que consideran como Regular a un 30% y como Excelente a un 30%, se aprovecha más el tiempo entre el desarrollador implementador y el funcional del cliente, con el método LACS se plantea que el funcional estará iniciando sus pruebas unitarias apenas el desarrollador termine alguna corrección del programa, de esta manera el funcional podrá realizar con tranquilidad todas sus pruebas adicionales que considera necesarias, así podrá asegurar que la calidad de sus pruebas unitarias y funcionales sean las correctas y necesarias, para entregar un producto de mayor calidad y que no se encuentren errores en el pase a producción del nuevo sistema.

Sabemos que por una prueba unitaria mal realizada se tendrá muchos errores en las pruebas unitarias que si no son detectadas a tiempo, cuando se realice la salida en vivo del sistema, los usuarios del sistema, los clientes y el servicio que la empresa ofrece se verán afectados, incurriendo incluso en temas tributarios con entidades del estado, son esos motivos que la calidad del producto debe ser asegurada con una decisión a tiempo para que las personas encargadas de las pruebas empiecen y terminen en el tiempo adecuado con las facilidades correctas.

Figura 8.

Recibir todas las pruebas en bloque garantiza una alta calidad – Pos-Test.



¿Considera que alcanzara el tiempo al funcional cliente para realizar todas las pruebas unitarias y también crear otros escenarios de prueba considerando su realidad del día a día?

Tabla 20.

Alcanzara el tiempo para crear otros escenarios de prueba – Pre-Test.

Pretest

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Pésimo	4	40,0	40,0	40,0
	Malo	3	30,0	30,0	70,0
	Regular	3	30,0	30,0	100,0
	Total	10	100,0	100,0	

Según los datos presentados en la Tabla 10 y la Figura 9 en la etapa de Pre-Test se observa que los encuestados consideran como Pésimo 40%, Malo 30% y Regular 30% que el funcional del cliente no pueda crear escenarios adicionales para realizar todas sus pruebas unitarias y funcionales completas, porque el tiempo para sus pruebas no será suficiente.

Para los encargados de realizar las pruebas unitarias y funcionales es muy posible que también tengan un set de pruebas que les gustaría probar como parte de todas sus pruebas, para validar si el nuevo sistema soporta los escenarios o casos que se les presenta al funcional del cliente.

Se puede apreciar que los encuestados consideran que el funcional sino puede terminar con todas sus pruebas unitarias y funcionales que le solicita, entonces no podrán empezar a crear sus propios escenarios o casos que haya visto según su experiencia en su realidad en la empresa.

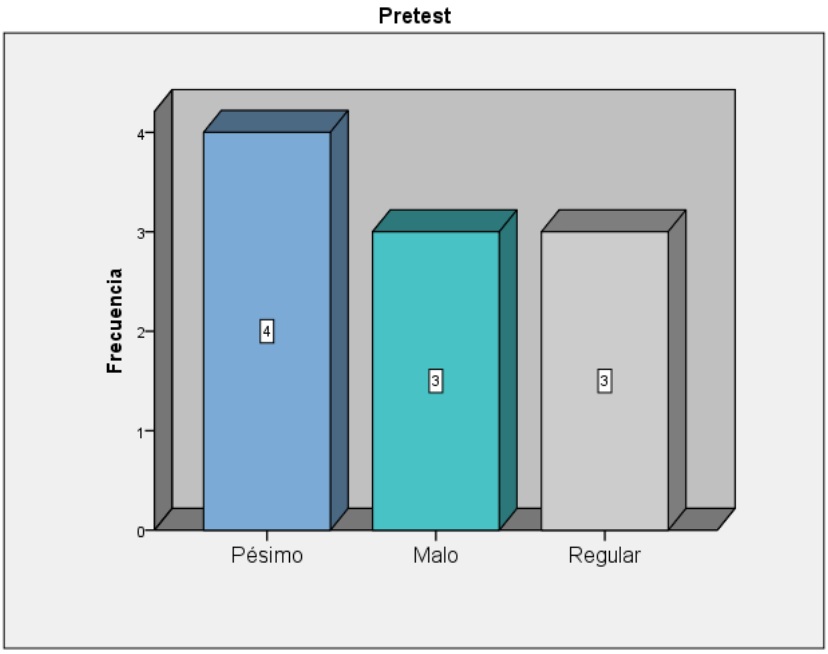


Figura 9.Alcanzara el tiempo para crear otros escenarios de prueba – Pre-Test.

¿Considera que alcanzara el tiempo al funcional cliente para realizar todas las pruebas unitarias y también crear otros escenarios de prueba considerando su realidad del día a día?

Tabla 31.

Alcanzara el tiempo para crear otros escenarios de prueba – Pre-Test.

Postest

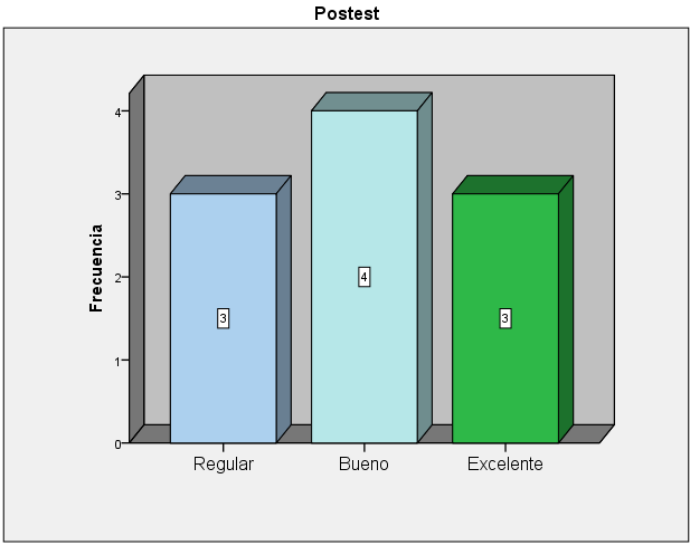
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Regular	3	30,0	30,0	30,0
	Bueno	4	40,0	40,0	70,0
	Excelente	3	30,0	30,0	100,0
	Total	10	100,0	100,0	

Según se observa en la Tabla 11 y la Figura 10 en la etapa de Pos-Test con el uso del método LACS opinan en un 40% como Bueno, Regular en un 30% y como Excelente en 30% porque el uso del nuevo método hará posible que el tiempo destinado para las pruebas unitarias, pruebas funcionales y la creación de otros escenarios que tenga el funcional se puedan realizar y el tiempo alcance para entregar un producto de calidad.

Al funcional del cliente encargado de realizar las pruebas unitarias y funcionales, le podrá alcanzar el tiempo para realizar otras pruebas adicionales, es decir crear nuevos escenarios y poner a prueba el nuevo sistema con otros casos que estime se le puedan presentar cuando el sistema este en línea.

Figura 10.

Alcanzara el tiempo para crear otros escenarios de prueba – Pre-Test.



¿Como considera la documentación técnica que entrega la consultora al término de la implementación, cubre todas las expectativas?

Tabla 42.

Considera que la documentación recibida cubre las expectativas - Pre-Test.

Pretest

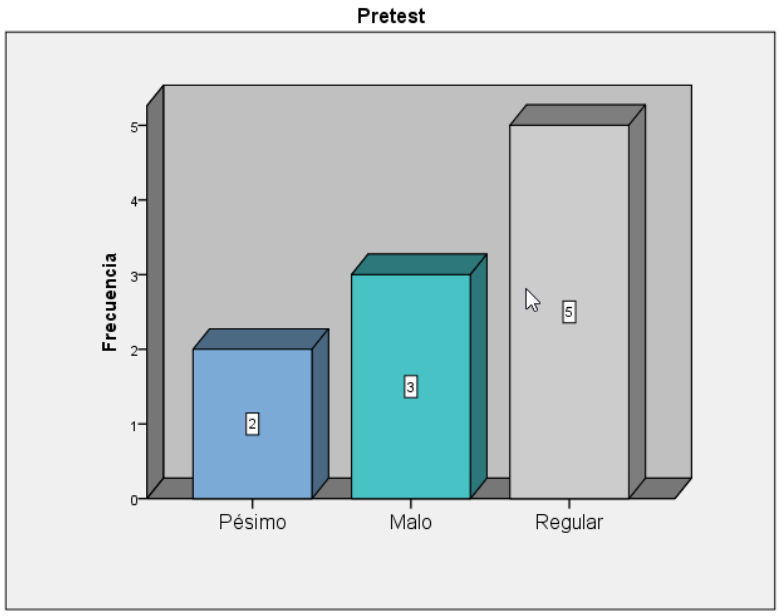
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Pésimo	2	20,0	20,0	20,0
	Malo	3	30,0	30,0	50,0
	Regular	5	50,0	50,0	100,0
	Total	10	100,0	100,0	

Según los resultados presentados en la Tabla 12 y la Figura 11 la documentación técnica que entrega la consultora al terminar la corrección del programa se considera Regular en un 50%, Malo en 30% y un Pésimo de 20%, la consultora muchas veces no entrega toda la documentación completa de todas las atenciones y si la entrega no se encuentra bien detallada, solo está a nivel general que dificulta el mantenimiento del sistema cuando la consultora ya no se encuentre en la empresa, talvez será para que los vuelvan a llamar para que realicen alguna ampliación al sistema, pero la documentación la dejan para ser entregada al final de todo el proyecto y como ha sucedido al final del proyecto ya no hay tiempo, hay que corregir errores por realizar pruebas unitarias y funcionales incompletas que ocasionan que se sacrifique las horas en realizar la entrega de toda la documentación por horas de corrección en el código, de esta manera la consultora en muchas ocasiones no entrega toda la documentación completa por dedicarle tiempo a rehacer y corregir las observaciones que encontró el funcional del cliente, como están previos a la salida en vivo del sistema tienen una prioridad máxima, así se comenta en las reuniones dejando la documentación en un segundo plano, que a las finales del

proyecto no se entrega con el todo el detalle necesario, perjudicando a la empresa quien asume la responsabilidad del mantenimiento desde ese momento.

Figura 11.

Considera que la documentación recibida cubre las expectativas - Pre-Test.



¿Como considera la documentación técnica que entrega la consultora al término de la implementación, cubre todas las expectativas?

Tabla 53.

Considera que la documentación recibida cubre las expectativas - Pos-Test.

Postest

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Regular	2	20,0	20,0	20,0
	Bueno	5	50,0	50,0	70,0
	Excelente	3	30,0	30,0	100,0
	Total	10	100,0	100,0	

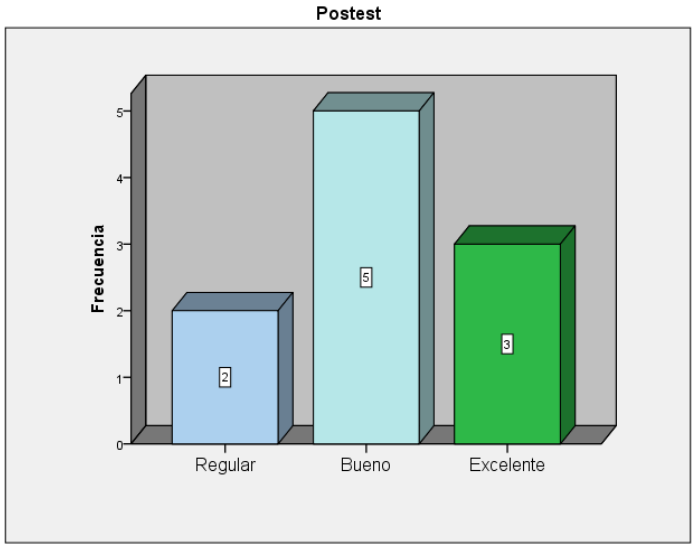
Según se observa en los resultados de la Tabla 13 y la Figura 12 en la etapa de Pos-Test considerando el utilizar el modelo LACS, los encuestados consideran como Bueno en 50%, Excelente 30% y Regular 20% el hecho de que utilizando el método LACS se logre desarrollar la documentación conforme se vaya avanzando, es decir se inicia con el desarrollador implementador corrigiendo el programa, luego el funcional del cliente realiza la prueba unitaria y la prueba funcional, si es correcto el desarrollador del cliente, ya empezaría a revisar el código nuevo, documentar las tablas, funciones, módulos y clases que utilizo el desarrollador implementador para crear su propio documento técnico, el cual le servirá para los futuros mantenimientos del sistema.

De esta manera se cumple el objetivo de tener toda la documentación completa de todos los cambios que se van realizando en el código del sistema, así como también registrar en el documento todas las configuraciones a nivel del módulo que realiza el implementador.

Adicionalmente el desarrollador del cliente está aumentando su conocimiento respecto a las tablas, funciones, métodos y clases que utiliza el desarrollador implementador, de esta manera aumenta la posibilidad de brindar un mayor soporte al sistema cuando la consultora que implementa el sistema allá terminado su trabajo de implementación y ya no se encuentre en la empresa del cliente.

Figura 12.

Considera que la documentación recibida cubre las expectativas - Pos-Test.



¿Considera que las capacitaciones de la consultora al personal desarrollador del cliente son las necesarias y suficientes, en cuanto al traspaso de conocimiento?

Tabla 64.

Considera que las capacitaciones recibidas son suficientes - Pre-Test.

Pretest

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Pésimo	3	30,0	30,0	30,0
	Malo	3	30,0	30,0	60,0
	Regular	4	40,0	40,0	100,0
	Total	10	100,0	100,0	

Según los resultados de la Tabla 14 y Figura 13 en la etapa de Pre-Test sin considerar el modelo LACS los encuestados indican como Regular en 40%, Malo 30% y Pésimo 30% a las capacitaciones que realiza la empresa consultora como parte del cronograma de trabajo y traspaso de conocimiento al desarrollador del cliente es insuficiente porque la empresa implementadora termina sin tiempo, para iniciar su etapa de capacitaciones al personal del cliente, por los motivos descritos como el reprocesos de errores, revisión de funciones, métodos y clases.

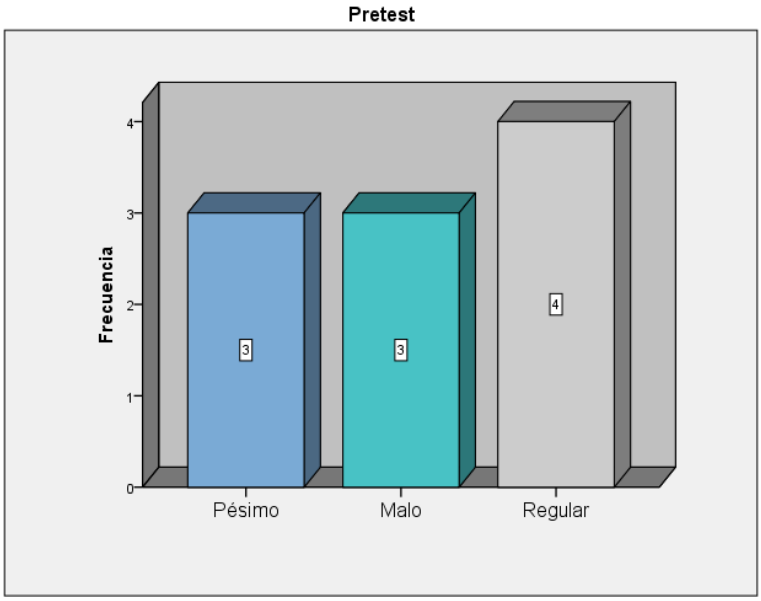
Al final del proyecto todos desean terminar sus pendientes y todos se encuentran realizando sus desarrollos, pruebas unitarias, pruebas funciones y documentación que cada uno tiene asignado, para tener todo listo para la salida en producción, por eso consideran de muy bajo nivel las capacitaciones de la consultora.

El personal del cliente muchas veces debe aprender por si sola, utilizando el mismo sistema ya en producción, encontrando la forma como solucionar las observaciones y atender las necesidades de la empresa.

En la implementación se debe considerar que las capacitaciones que son necesarias son a nivel de código del programa, capacitación a nivel de configuración del módulo, capacitación funcional del sistema que utilizara el usuario responsable del sistema todos los días en la empresa y la capacitación a nivel del mantenimiento de los servidores de aplicaciones y de base de datos.

Figura 13.

Considera que las capacitaciones recibidas son suficientes - Pre-Test.



¿Considera que las capacitaciones de la consultora al personal desarrollador del cliente son las necesarias y suficientes, en cuanto al traspaso de conocimiento?

Tabla 75.

Considera que las capacitaciones recibidas son suficientes - Pos-Test.

Postest

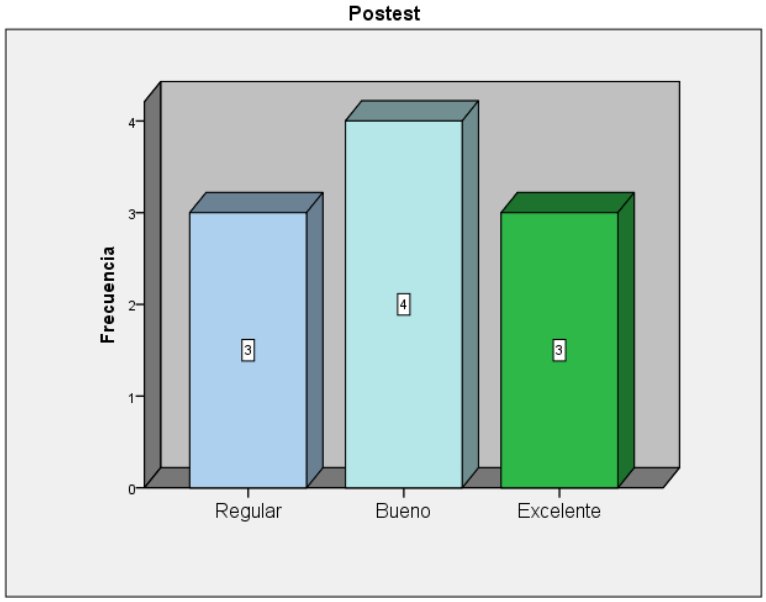
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Regular	3	30,0	30,0	30,0
	Bueno	4	40,0	40,0	70,0
	Excelente	3	30,0	30,0	100,0
	Total	10	100,0	100,0	

Según los resultados presentados en la Tabla 15 y la Figura 14 en la etapa de Pos-Test utilizando el método LACS el nivel de las capacitaciones aumentan a un nivel Bueno de 40%, Regular 30% y Excelente 30%, como se menciona en el método LACS se debe crear un equipo de trabajo entre el desarrollador implementador que realiza la corrección del programa, apenas termine el desarrollo debe liberar el código del programa al ambiente de Calidad para que el funcional del cliente pueda ingresar y realizar las pruebas unitarias y las pruebas funcionales aumentando su nivel de aprendizaje porque tendrá más tiempo para realizar las pruebas que sean necesarias, el desarrollador del cliente puede estar en paralelo revisando el avance del desarrollador implementador e ir documentando las tablas, funciones, métodos y clases que se han utilizado para resolver y atender el problema, debemos recordar que en el sistema SAP el código de los programas se guardan en ordenes de transporte el cual el código es visible para todos los desarrolladores que tienen el permiso de ingresar al ambiente de Desarrollo, es decir el código que va creando el desarrollador implementador siempre esta visible para que el desarrollador del cliente lo pueda ir revisando, de esta manera el desarrollador del también podrá aumentar el nivel de aprendizaje del código del sistema y de

la misma manera al crear la documentación técnica también estaría aprendiendo, el mismo documento se enviara a todo el personal técnico de la empresa cliente para que todos empiecen a revisar el programa y empiecen a aprender desde la etapa de realización y no al final de todo el proyecto cuando ya no hay el tiempo y la tranquilidad por salir en vivo con el sistema.

Figura 14.

Considera que las capacitaciones recibidas son suficientes - Pos-Test.



¿Qué opinión le merece, cuando el desarrollador implementador termina con las correcciones, no entregue la documentación completa para que el funcional del cliente inicie sus pruebas unitarias correctamente en el ambiente de Calidad?

Tabla 86.

Considera no recibir la documentación para iniciar las pruebas - Pre-Test.

Pretest

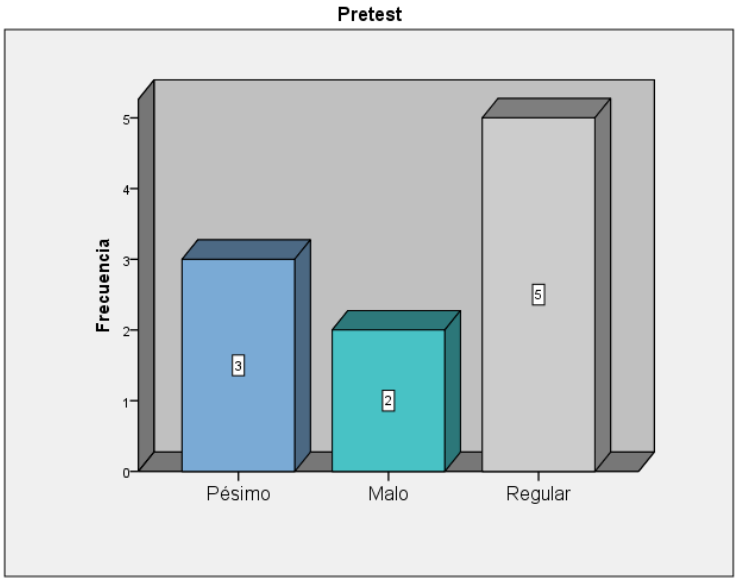
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Pésimo	3	30,0	30,0	30,0
	Malo	2	20,0	20,0	50,0
	Regular	5	50,0	50,0	100,0
	Total	10	100,0	100,0	

Según los resultados de la Tabla 16 y la Figura 15 en la etapa Pre-Test se considera como Regular en 50%, Pésimo en 30% y un nivel Malo de 20% cuando el desarrollador implementador no entrega la documentación necesaria a su funcional implementador para que desarrolle el manual de uso de la corrección, por ese motivo el funcional del cliente debe realizar las pruebas unitarias y funcionales sin el manual de uso del sistema, porque como se mencionó todos están ocupados para realizar la documentación y se deja este punto importante para el final del proyecto, que muchas veces no se llega a entregar porque siempre aparecen otras prioridades y como las horas asignadas a la implementación se terminan por la salida en vivo, ya no es posible entregar la documentación completa o si la entregan no tiene todo el detalle que es necesario para las futuras modificaciones en el sistema.

Como el funcional del cliente desea avanzar y realizar las pruebas unitarias y funcionales, muchas veces empieza las validaciones en el ambiente de Calidad, muchas veces sin recibir la documentación completa, como también recibe todas las correcciones en bloque no puede esperar que toda la documentación esté terminada para iniciar con sus pruebas, por ese motivo el funcional del cliente empieza a probar y documentar en ese momento lo que pueda revisar según su propia experiencia.

Figura 15.

Considera no recibir la documentación para iniciar las pruebas - Pre-Test.



¿Qué opinión le merece, cuando el desarrollador implementador termina con las correcciones, no entregue la documentación completa para que el funcional del cliente inicie sus pruebas unitarias correctamente en el ambiente de Calidad?

Tabla 97.

Considera no recibir la documentación para iniciar las pruebas - Pos-Test.

Postest

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Regular	2	20,0	20,0	20,0
	Bueno	4	40,0	40,0	60,0
	Excelente	4	40,0	40,0	100,0
	Total	10	100,0	100,0	

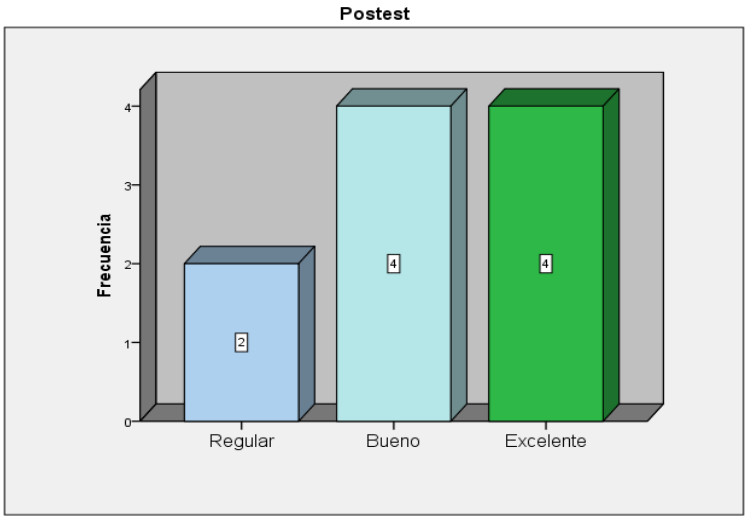
Según los resultados presentados en la Tabla 17 y la Figura 16 en la etapa de Pos-Test utilizando el método LACS se puede observar que los encuestados consideran como Bueno en 40%, Excelente 40% y un Regular de 20% a la propuesta que se cree un equipo coordinado entre el desarrollador implementador quien realiza la corrección del problema en el sistema, en ese momento el desarrollador del cliente ya está documentando el cambio y enviando al funcional implementador para que prepare el documento funcional y lo envíe al funcional del cliente para que empiece con sus pruebas unitarias y funcionales en el ambiente de Calidad.

Como se mencionó el código del programa del sistema SAP esta visible, porque el código que va creando el desarrollador implementador se guarda en ordenes de transporte que son visibles para todos los desarrolladores en el ambiente de Desarrollo.

De esta manera el funcional del cliente podrá empezar sus pruebas unitarias y funcionales con la documentación que le envíe el funcional implementador con el apoyo del desarrollador del cliente quien apoyo en enviar los cambios técnicos para que se complete la documentación necesaria.

Figura 16.

Considera no recibir la documentación para iniciar las pruebas - Pos-Test.



¿Como considera que el personal desarrollador del cliente no pueda revisar nada del código SAP hasta que el desarrollador implementador termine con todas las correcciones a las funciones y métodos que son necesarios para la implementación del sistema SAP?

Tabla 108.

El personal cliente no puede revisar el código hasta la última etapa - Pre-Test.

Pretest

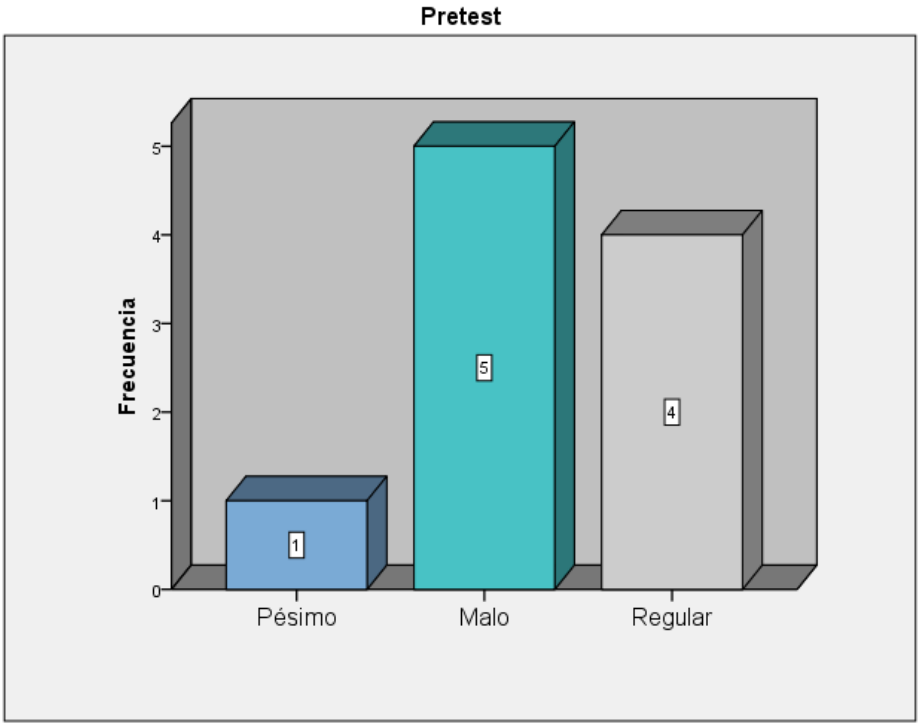
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Pésimo	1	10,0	10,0	10,0
	Malo	5	50,0	50,0	60,0
	Regular	4	40,0	40,0	100,0
	Total	10	100,0	100,0	

Según los resultados de la Tabla 18 y la Figura 17 en la etapa de Pre-Test se observa que los encuestados consideran como Malo en 50%, Regular 40% y Pésimo 10% al hecho que el desarrollador del cliente no pueda revisar el código del programa que está utilizando el desarrollador implementador, solo hasta que se termine con todo el desarrollo, esta parte es crítica porque el desarrollador del cliente nunca vera las tablas, funciones, métodos y clases que se están utilizando para corregir los problemas en el sistema y que en algún momento va a tener que realizar el mantenimiento del sistema.

Como se mencionó en el sistema SAP el código que está creando el desarrollador implementador esta siempre visible porque se guarda en una orden de transporte en el ambiente de Desarrollo, se puede configurar para que el código solo sea visible y no pueda modificar por ninguna persona y solo por el desarrollador implementador, de esta manera se asegura que no habrá ningún cambio y que el desarrollador implementador puede avanzar sin problemas.

Figura 17.

El personal cliente no puede revisar el código hasta la última etapa - Pre-Test.



¿Como considera que el personal desarrollador del cliente no pueda revisar nada del código SAP hasta que el desarrollador implementador termine con todas las correcciones a las funciones y métodos que son necesarios para la implementación del sistema SAP?

Tabla 19.

El personal cliente no puede revisar el código hasta la última etapa - Pos-Test.

Postest

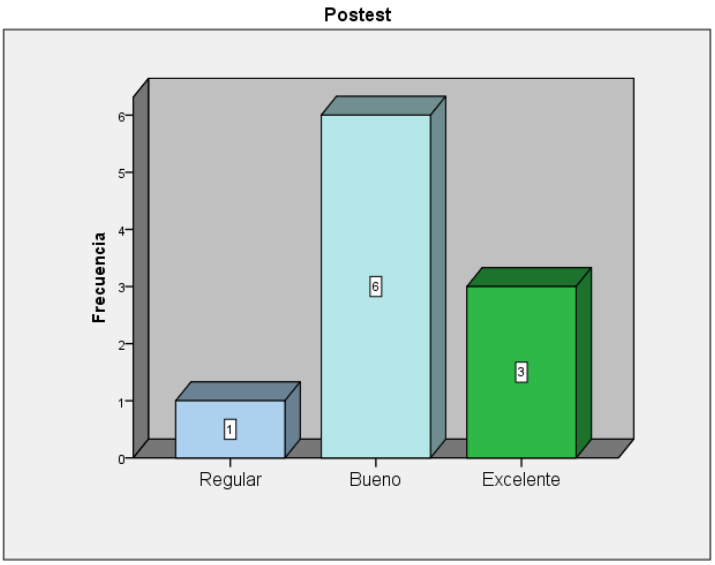
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Regular	1	10,0	10,0	10,0
	Bueno	6	60,0	60,0	70,0
	Excelente	3	30,0	30,0	100,0
	Total	10	100,0	100,0	

Según los resultados de la Tabla 19 y la Figura 18 en la etapa Pos-Test utilizando el método LACS los encuestados consideran como Bueno en 60%, Excelente en 30% y Regular 10%, consideran que el desarrollador del cliente debería conocer y revisar de manera paralela según el avance del desarrollador implementador todo el código, tablas que se utilizan, funciones, métodos y clases del sistema.

Como se indica en el sistema SAP el código que se desarrolla siempre esta visible para que todos los desarrolladores con el permiso respectivo puedan revisar, de esta manera el desarrollador implementador puede estar creando su código y el desarrollador del cliente puede observar sin tener los permisos de modificar, de esta manera se puede autocapacitar y tratar de elevar su nivel en el aprendizaje del lenguaje de programación Abap el cual se utiliza en todos los módulos del sistema SAP.

Figura 18.

El personal cliente no puede revisar el código hasta la última etapa - Pos-Test.



¿Como considera que el funcional responsable de las pruebas unitarias del cliente no pueda realizar ninguna prueba en el ambiente de Calidad hasta que el desarrollador implementador termine con todas las correcciones a las funciones y métodos que son necesarios?

Tabla 110.

El funcional cliente no puede iniciar ninguna prueba hasta que el desarrollador termine con todas las correcciones - Pre-Test.

Pretest

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Pésimo	2	20,0	20,0	20,0
	Malo	5	50,0	50,0	70,0
	Regular	3	30,0	30,0	100,0
	Total	10	100,0	100,0	

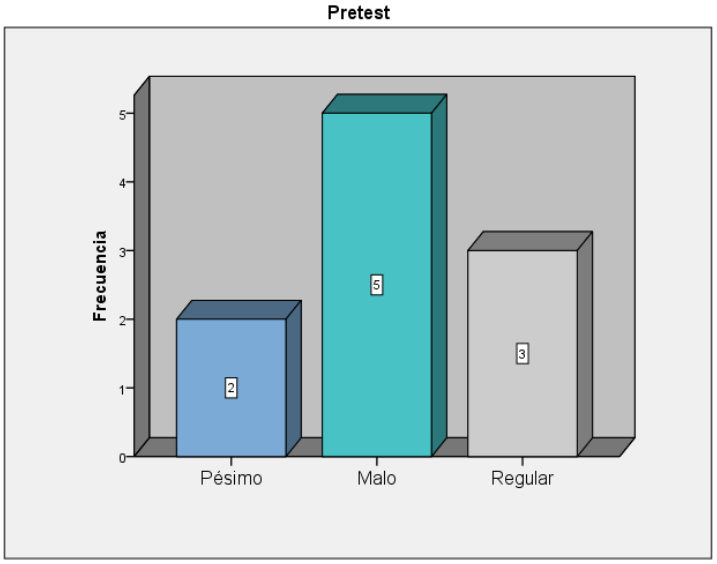
Según los resultados de la Tabla 20 y la Figura 19 en la etapa de Pre-Test sin considerar el método LACS los encuestados consideran como Malo al 50%, Regular 30% y Pésimo 20% a la propuesta que el funcional del cliente no pueda iniciar con ninguna prueba unitaria y prueba funcional hasta que el desarrollador implementador haya terminado en resolver todas las modificaciones que se le han solicitado.

De esta manera el funcional del cliente va a perder mucho tiempo esperando que el desarrollador implementador termine con todas las correcciones para que pueda empezar con sus pruebas unitarias y funcionales.

Por ese motivo aumenta el riesgo de realizar pruebas unitarias y funcionales incompletas porque el tiempo va a ser cada más corto y el funcional del cliente no podrá terminar a tiempo y realizar todas las pruebas que desee.

Figura 19.

El funcional cliente no puede iniciar ninguna prueba hasta que el desarrollador termine con todas las correcciones - Pre-Test.



¿Como considera que el funcional responsable de las pruebas unitarias del cliente no pueda realizar ninguna prueba en el ambiente de Calidad hasta que el desarrollador implementador termine con todas las correcciones a las funciones y métodos que son necesarios?

Tabla 121.

El funcional cliente no puede iniciar ninguna prueba hasta que el desarrollador termine con todas las correcciones – Pos-Test.

Pos-Test

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Regular	3	30,0	30,0	30,0
	Bueno	4	40,0	40,0	70,0
	Excelente	3	30,0	30,0	100,0
	Total	10	100,0	100,0	

Según los resultados presentados en la Tabla 21 y Figura 20 en la etapa de Pos-Test utilizando el modelo LACS según la respuesta de los encuetados se tiene como Bueno al 40%,

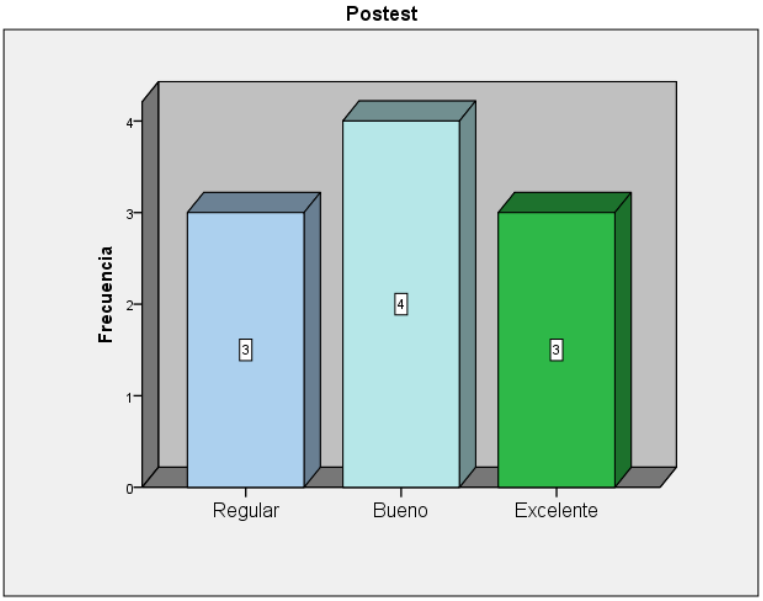
Regular al 30% y como Excelente el 30% a la propuesta que considera crear un equipo de trabajo que de manera coordinada inicie en el desarrollador implementador para corregir el programa, luego el desarrollador del cliente ya está revisando el código del programa que se modificó para atender el problema y en paralelo el funcional del cliente está ingresando al ambiente de Calidad para iniciar con sus pruebas unitarias y pruebas funcionales, inclusive le alcanzara el tiempo al funcional del cliente para crear sus propios escenarios de prueba entre cada programa que va corrigiendo el desarrollador implementador, de esta manera se consigue tener más pruebas aumentando el nivel y la calidad del sistema que se desea implementar.

Uno de los puntos que el desarrollador implementador y no le permite liberar una orden por cada atención, crea una orden de transporte para todas las correcciones por ese motivo no puede liberar la orden para realizar cada prueba independiente, por ese motivo debe crear una orden de transporte para cada problema que va a corregir.

De esta manera el equipo trabaja de manera coordinada, cada problema que se va resolviendo a nivel de código, se libera la orden de transporte en el ambiente de calidad y el mismo problema será verificado por el funcional del cliente, de esta manera cada corrección se va cerrando con su respectiva prueba unitaria, prueba funcional y documentación técnica, sería lo más importante para alcanzar el éxito del proyecto.

Figura 20.

El funcional cliente no puede iniciar ninguna prueba hasta que el desarrollador termine con todas las correcciones – Pos-Test.



¿Cuál es su opinión respecto al tiempo que se toma entre la corrección del desarrollador implementador y el inicio de las pruebas unitarias que realiza el funcional del cliente?

Tabla 132.

El tiempo que toma entre la corrección y el inicio de las pruebas unitarias.

	Pre-test Frecuencia	Porcentaje	Pos-test Frecuencia	Porcentaje
Pésimo	4	40%	0	0%
Malo	3	30%	0	0%
Regular	3	30%	3	30%
Bueno	0	0	4	40%
Excelente	0	0	3	30%
Total	10	100%	10	100%

Según los resultados presentados en la Tabla 22 y la Figura 21, respecto al tiempo que se toma entre la corrección del desarrollador implementador el inicio de las pruebas unitarias que realiza el funcional del cliente se observa que al principio en la etapa Pre-Test el resultado del cuestionario indicaba como Pésimo en 40%, Malo 30% y Regular 30%, no se tiene ninguna

respuesta que considere Bueno o Excelente este procedimiento, pero en la etapa del Pos-Test se observa que se mantiene la respuesta Regular en 30%, pero las respuestas de Bueno que se incrementa a 40% y la respuesta de Excelente se incrementa en 30%. En este sentido es importante mencionar que los encuestados consideran que el Nuevo Modelo LACS servirá para agilizar el tiempo en el inicio de las pruebas unitarias y funcionales en el ambiente de Calidad del Funcional del Cliente, de esta manera se podrá tener una respuesta más rápida, si la corrección se realizó con éxito o es necesario alguna verificación adicional, para que rápidamente sea revisado por el desarrollador implementador. Un hecho que es importante sobre las pruebas es lo mencionado por Gutierrez et al., (2010), en el indican que una de las misiones del cliente es escribir las pruebas, realizar las pruebas sobre el sistema y dar su aprobación al mismo, cada iteración concluye ejecutando un conjunto de pruebas de aceptación que permitan poder comprobar si el cliente está satisfecho con el resultado final.

Tabla 143.

Análisis de Fiabilidad de Alfa de Cronbach para el tiempo que toma entre la corrección y el inicio de las pruebas unitarias.

Estadísticos de fiabilidad

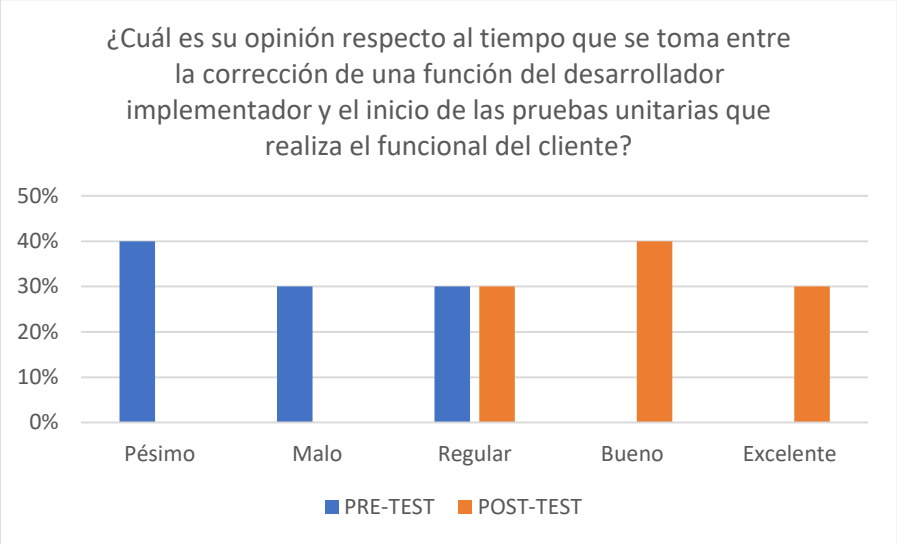
Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,964	,965	2

Según la Tabla 23, se puede observar la confiabilidad del instrumento de correlación de datos con el Alfa de Cronbach, para evaluar la consistencia interna de la escala de medición a través de las respuestas de los expertos, como consideran el tiempo entre el término de la

corrección del desarrollador implementador y el inicio de las pruebas unitarias del funcional del cliente, se mide las respuestas en la etapa del Pre-Test y Pos-Test se evaluó y se tiene un coeficiente de 0.964, indica que el instrumento es altamente confiable.

Figura 21.

El tiempo que toma entre la corrección y el inicio de las pruebas unitarias.



¿Qué opinión tiene cuando el desarrollador implementador envía al funcional cliente todas las pruebas en bloque para que inicie con sus pruebas unitarias en el ambiente de Calidad?

Tabla 154.

Cuando el funcional recibe todas las pruebas en bloque para iniciar con las pruebas unitarias.

	Pre-test		Pos-test	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Pésimo	3	30%	0	0%
Malo	4	40%	0	0%
Regular	3	30%	4	40%
Bueno	0	0%	3	30%
Excelente	0	0%	3	30%
Total	10	100%	10	100%

Según los resultados obtenidos en la Tabla 24 y la Figura 22, se puede observar que en la etapa del Pre-Test los encuestados consideran como Malo en 40%, Pésimo el 30% y Regular en 30% cuando el desarrollador implementador envía todas las pruebas en bloque para que el funcional del cliente pueda iniciar con sus pruebas unitarias y pruebas funcionales. Por otro lado en la etapa de Pos-Test se presentan que los encuestados opinan como Regular 40%, Bueno 30% y Excelente en 30% la propuesta del nuevo Modelo LACS que plantea un trabajo coordinado que inicia con la corrección del problema por parte del desarrollador implementador, en paralelo el desarrollador del cliente revisando la tablas, funciones, métodos y clases que se está utilizando para la corrección del programa, cuando el desarrollador implementador termine la corrección tendrá que liberar la Orden de transporte para importar el código y liberar la Orden de transporte al ambiente de Calidad, para que el funcional del cliente pueda ingresar y empezar con sus pruebas unitarias y funcionales, mientras se realiza esta actividad, el desarrollador implementador puede empezar analizando el código para la siguiente corrección.

Tabla 165.

Análisis de Fiabilidad de Alfa de Cronbach para cuando el funcional recibe todas las pruebas en bloque para iniciar con las pruebas unitarias.

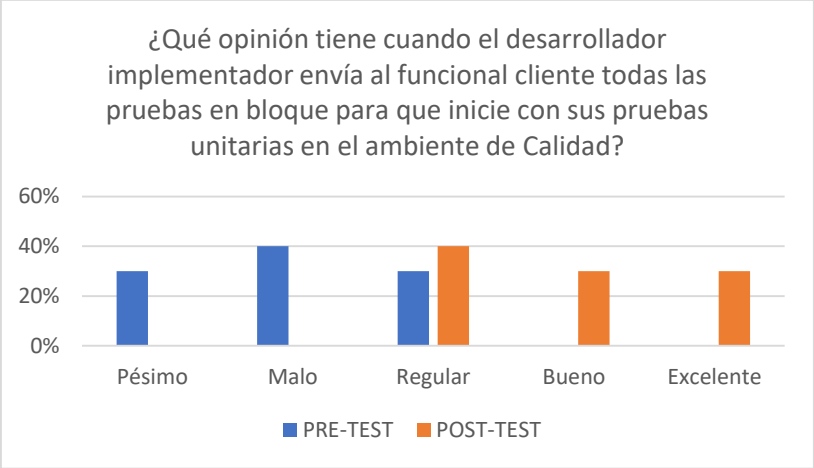
Estadísticos de fiabilidad

Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,873	,875	2

Según el resultado de la Tabla 25 se presenta el valor 0.873 de la confiabilidad del instrumento de correlación de datos del Alfa de Cronbach, para evaluar la consistencia interna de la escala de medición que se obtiene de la respuesta de los expertos, se mide la respuesta del Pre-Test y Pos-Test para la pregunta sobre el envío de todas las pruebas en bloque que realiza el desarrollador implementador al funcional del cliente para que inicie con sus pruebas unitarias y funcionales.

Figura 22.

Cuando el funcional recibe todas las pruebas en bloque para iniciar con las pruebas unitarias.



¿Como considera realizar las pruebas unitarias de un sistema ERP en un tiempo muy corto, podría ocasionar errores en las pruebas funcionales?

Tabla 176.

Como considera realizar pruebas unitarias en un tiempo muy corto.

	Pretest Frecuencia	Porcentaje	Postest Frecuencia	Porcentaje
Pésimo	2	20%	0	0%
Malo	4	40%	0	0%
Regular	4	40%	5	50%
Bueno	0	0%	3	30%
Excelente	0	0%	2	20%
Total	10	100%	10	100%

Según los resultados de la Tabla 27 y la Figura 23, respecto a realizar las pruebas unitarias en un tiempo muy corto, en la etapa de Pre-Test consideran en sus respuestas como Malo en 40%, Regular en 40% y Pésimo en 20%, pero en la etapa de Pos-Test se observa un incremento en la respuesta de Regular a 50%, Bueno en 30% y Excelente en 20%, en tal sentido es importante mencionar que los encuestados consideran que el Nuevo Modelo LACS ayudara a que el tiempo asignado para las pruebas unitarias se cumpla según el cronograma, utilizando el nuevo modelo LACS evitara tener tiempos del personal inactivos, porque se plantea que a

través del equipo desarrollador implementador, desarrollador del cliente y el funcional de cliente, estén de manera continua, iniciando una corrección, documento la corrección y realizando la prueba unitaria y prueba funcional, luego de ello se cierra el requerimiento y se continuar con la siguiente atención, de esta manera el tiempo asignado para las pruebas unitarias deberían ser suficientes para entregar un proyecto de calidad.

Tabla 187.

Análisis de Fiabilidad de Alfa de Cronbach para saber cómo considera realizar pruebas unitarias en un tiempo muy corto.

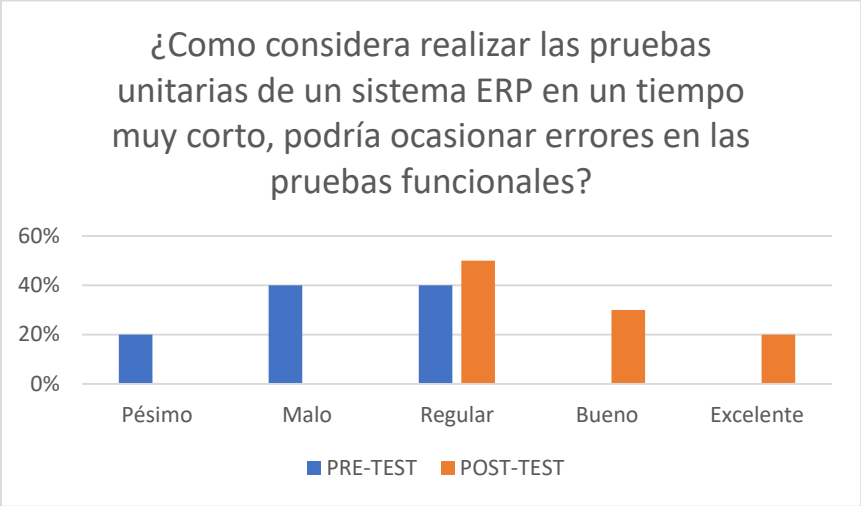
Estadísticos de fiabilidad

Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,880	,881	2

Según la Tabla 27 se observar la confiabilidad del instrumento de correlación del Alfa de Cronbach, que nos permite evaluar la consistencia interna de la escala de medición con la respuesta de las encuestas a los expertos, se mide las respuestas de la etapa del Pre-Test y la etapa del Pos-Test, para la consulta como considera realizar las pruebas unitarias en un tiempo muy corto de tiempo y que puede impactar en las pruebas funcionales, en este caso según las respuestas tenemos el valor del coeficiente de 0.880 y nos indica que el instrumento es altamente confiable.

Figura 23.

Como considera realizar pruebas unitarias en un tiempo muy corto.



¿Considera que las pruebas unitarias que realizara el funcional del cliente son de alta calidad, por recibir todas las pruebas juntas en bloque, cuando el desarrollador termino con todas las correcciones?

Tabla 198.

Considera que las pruebas unitarias serán de alta calidad por recibir todas las pruebas en bloque.

	Pretest Frecuencia	Porcentaje	Postest Frecuencia	Porcentaje
Pésimo	5	50%	0	0%
Malo	1	10%	0	0%
Regular	4	40%	3	30%
Bueno	0	0%	4	40%
Excelente	0	0%	3	30%
Total	10	100%	10	100%

Según los resultados presentados en la Tabla 28 y la Figura 24, en la etapa Pre-Test respecto a la calidad de las pruebas unitarias cuando el funcional del cliente recibe todas las correcciones en bloque por parte del desarrollador implementador, consideran como Pésimo

en 50%, Regular en 40% y Malo en 10%, pero en la etapa de Pos-Test considerando el uso del nuevo Modelo LACS consideran como Bueno en 40%, Regular en 30% y Excelente en 30%, en este sentido es muy importante indicar que los expertos encuestados consideran que el nuevo Modelo LACS servirá para mejorar la Calidad de las pruebas unitarias, porque se tendrá el tiempo necesario para ejecutar cada prueba, así como el tiempo para volver a revisar la observación según sea necesaria, hasta que se corrija adecuadamente, de esta manera se tendrá un sistema de mayor calidad implementado en el cliente.

Tabla 29.

Análisis de Fiabilidad de Alfa de Cronbach como considera que las pruebas unitarias serán de alta calidad por recibir todas las pruebas en bloque.

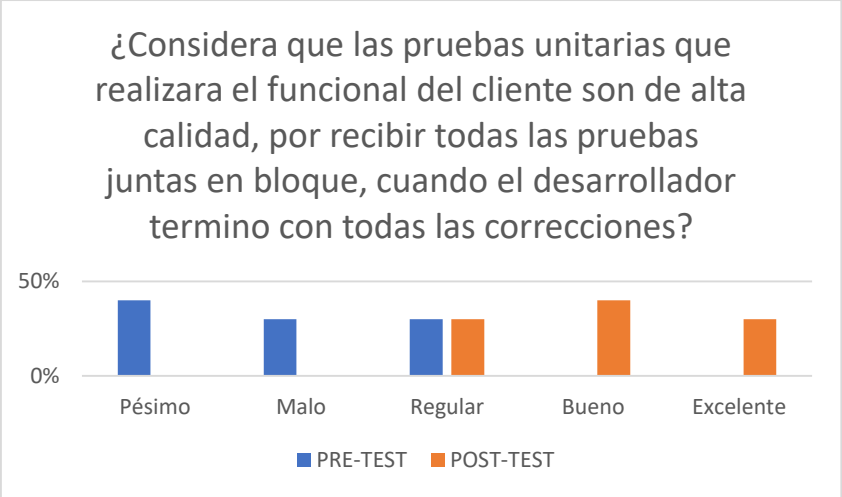
Estadísticos de fiabilidad

Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,892	,902	2

Según la Tabla 29, se observa que la confiabilidad del instrumento de correlación de datos de Alfa de Cronbach, con un coeficiente de 0.892, indica que el instrumento es altamente confiable para cuando se desee medir si las pruebas unitarias que realiza el funcional del cliente son de alta calidad cuando recibe todas las pruebas en bloque del desarrollador implementador.

Figura 24.

Considera que las pruebas unitarias serán de alta calidad por recibir todas las pruebas en bloque.



¿Considera que alcanzara el tiempo al funcional cliente para realizar todas las pruebas unitarias y también crear otros escenarios de prueba considerando su realidad del día a día?

Tabla 200.

Considera que alcanzara el tiempo para realizar todas las pruebas y agregar otros escenarios de prueba.

	Pretest Frecuencia	Porcentaje	Postest Frecuencia	Porcentaje
Pésimo	4	40%	0	0%
Malo	3	30%	0	0%
Regular	3	30%	3	30%
Bueno	0	0%	4	40%
Excelente	0	0%	3	30%
Total	10	100%	10	100%

Según los resultados presentados en la Tabla 30 y la Figura 25, en la etapa de Pre-Test respecto al tiempo para realizar las pruebas unitarias y crear otros escenarios de prueba que considere según su experiencia y el trabajo del día a día, el resultado del cuestionario nos

demuestra que consideran como Pésimo en 40%, Malo en 30% y Regular 30%, pero en la etapa de Pos-Test se considera el utilizar el nuevo Modelo LACS tenemos unas respuestas que muestran una calificación de Bueno en 40%, Regular en 30% y Excelente 30%, es importante mencionar que las personas encuestadas consideran que el uso del nuevo Modelo LACS propone el uso adecuado de los tiempos para las pruebas, desarrollo del código y la documentación necesaria, teniendo el tiempo suficiente para realizar las pruebas y poder crear otros escenarios de prueba que considere el funcional del cliente, cuanto más pruebas se realicen será mucho mejor para el proyecto, para tener un sistema de mayor calidad.

Tabla 211.

Análisis de Fiabilidad de Alfa de Cronbach como considera que alcanzara el tiempo para realizar todas las pruebas y agregar otros escenarios de prueba.

Estadísticos de fiabilidad

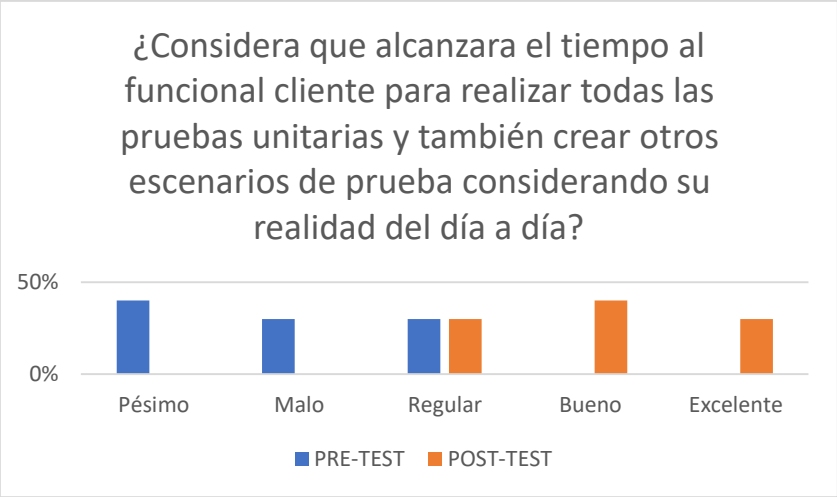
Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,964	,965	2

Según la Tabla 31, podemos observar un coeficiente de 0.964, indica que el instrumento es altamente confiable, se demuestra la confiabilidad del instrumento de correlación de datos con el Alfa de Cronbach, que evalúa la consistencia interna de la escala de medición de las respuestas de los expertos, sobre como considera el tiempo que tendrá el funcional del cliente si tendrá el tiempo suficiente para sus pruebas unitarias y crear otros escenarios de prueba según su experiencia del día a día en el puesto de trabajo, es muy importante este punto porque

el funcional del cliente, debe estar convencido que el sistema va a cumplir con todos los casos y requerimientos que va a tener cuando empiece a utilizar el sistema en la salida en vivo.

Figura 25.

Considera que alcanzara el tiempo para realizar todas las pruebas y agregar otros escenarios de prueba.



¿Como considera la documentación técnica que entrega la consultora al término de la implementación, cubre todas las expectativas?

Tabla 222.

Considera que la documentación que entrega la consulta cubre todas las expectativas.

	Pretest		Postest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Pésimo	2	20%	0	0%
Malo	3	30%	0	0%
Regular	5	50%	2	20%
Bueno	0	0%	5	50%
Excelente	0	0%	3	30%
Total	10	100%	10	100%

Según los resultados de la Tabla 32 y la Figura 26, respecto a cómo considera la documentación técnica que entrega la consultora al término de la implementación, podemos

observar que en la etapa de Pre-Test los encuestados expertos consideran como Regular en 50%, Malo en 30% y Pésimo en 20% el hecho de recibir la documentación incompleta y muchas veces una documentación sin tener todo el detalle necesario para las futuras correcciones en el código y en la configuración del sistema, pero en la etapa de Pos-Test se considera el nuevo Modelo LACS se observa que consideran como Bueno en 50%, Regular en 20% y Excelente en 30% con el nuevo método LACS se propone que el desarrollador del cliente este revisando constantemente el avance del desarrollador implementador, como se mencionó el código de lenguaje de programación es Abap, en SAP se puede revisar el código de programación de todos los programas, el código es compartido siempre que se tenga permiso a la transacción de programación, pero de manera visual si es posible, de esta manera el desarrollador del cliente podrá ir documentando cada tabla, función, método y clase que este utilizando el desarrollador implementador para aprender cada día y para tener un documento con todo el detalle necesario, para cuando los consultores ya no se encuentren en la empresa y se tenga que dar el soporte al sistema, tanto a nivel de desarrollo y a nivel de configuración.

Tabla 233.

Análisis de Fiabilidad de Alfa de Cronbach como considera que la documentación que entrega la consulta cubre todas las expectativas.

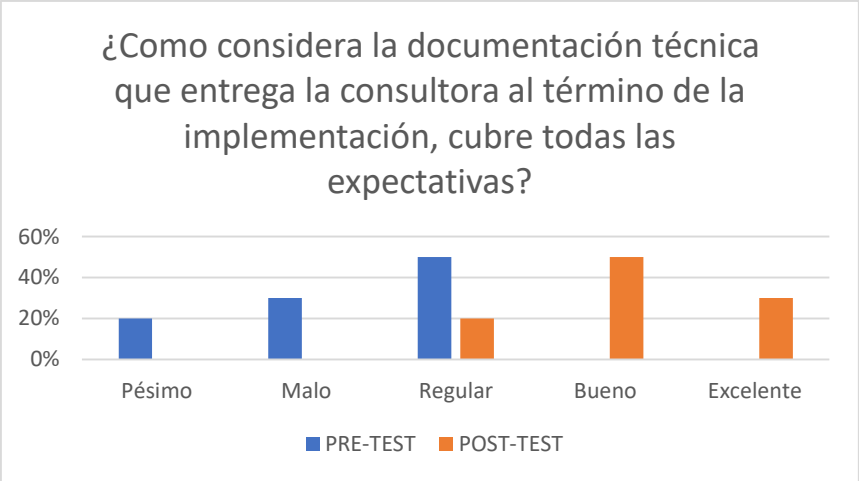
Estadísticos de fiabilidad

Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,922	,925	2

Según la Tabla 33, se observa que confiabilidad del instrumento de correlación de datos con el Alfa de Cronbach tiene un coeficiente de 0.922, nos indica que es un instrumento altamente confiable, el cual mide las respuestas en la etapa de Pre-Test y Pos-Test en la pregunta como considera la documentación técnica que entrega la consultora al término de la implementación.

Figura 26.

Considera que la documentación que entrega la consulta cubre todas las expectativas.



¿Considera que las capacitaciones de la consultora al personal desarrollador del cliente son las necesarias y suficientes, en cuanto al traspaso de conocimiento?

Tabla 244.

Considera que las capacitaciones realizadas por la consultora son las necesarias y suficientes.

	Pretest Frecuencia	Porcentaje	Postest Frecuencia	Porcentaje
Pésimo	3	30%	0	0%
Malo	3	30%	0	0%
Regular	4	40%	3	30%
Bueno	0	0%	4	40%
Excelente	0	0%	3	30%
Total	10	100%	10	100%

Según los resultados presentados en la Tabla 34 y la Figura 27, respecto como considera las capacitaciones de la consultora al personal desarrollador del cliente si son necesarias y suficientes, sobre el traspaso de conocimiento, en la etapa de Pre-Test el resultado de las encuestas indican como Regular en 40%, Malo en 30% y Pésimo en 30%, pero en la etapa de Pos-Test considerando el nuevo Modelo LACS las respuestas son diferentes, según las respuestas de los entrevistados consideran como Regular en 30%, Bueno en 40% y Excelente en 30%, en tal sentido es importante mencionar que los encuestados consideran que el nuevo Modelo LACS será necesario para que el desarrollador del cliente empiece a comprender las tablas, funciones, métodos y clases que va revisando el desarrollador implementador, desde el primer día en que se revise el primer programa a corregir, también con la documentación que va a tener que preparar podrá transmitir a todos los demás desarrolladores del cliente para que puedan empezar a revisar todos los objetos que se describen en la documentación técnica, de esta manera no se tendrá que esperar la última etapa si alcanza el tiempo recibir alguna capacitación muy rápida y con poco tiempo porque al final del proyecto todos están muy ocupados por los últimos ajustes de una salida en vivo, es mejor realizar el traspaso de la información desde la etapa de realización donde el desarrollador implementador recién está empezando con la primera corrección, de esta manera todos tendrán el tiempo suficiente para tener mayor conocimiento durante todo el tiempo que dure la implementación del proyecto y llegaran al final del proyecto con un nivel alto en el desarrollo del sistema.

Tabla 255.

Análisis de Fiabilidad de Alfa de Cronbach como considera que las capacitaciones realizadas por la consultora son las necesarias y suficientes.

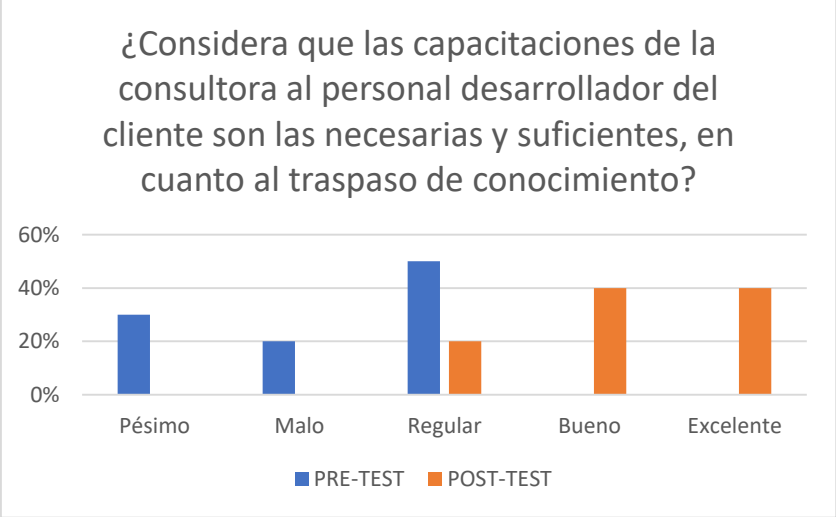
Estadísticos de fiabilidad

Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,964	,965	2

Según la Tabla 35, se puede observar la confiablidad del instrumento de correlación de datos del Alfa de Cronbach para evaluar la consistencia interna de la escala de medición a través de la respuesta de los expertos, tenemos un valor de 0.964, nos indica que el instrumento es altamente confiable, se mide las respuestas en la etapa del Pre-Test y Pos-Test, respecto a las capacitaciones que realiza la consultora responsable de la implementación al personal desarrollador de la empresa cliente, al final del proyecto de implementación.

Figura 27.

Considera que las capacitaciones realizadas por la consultora son las necesarias y suficientes.



¿Qué opinión le merece, cuando el desarrollador implementador termina con las correcciones, no entregue la documentación completa para que el funcional del cliente inicie sus pruebas unitarias correctamente en el ambiente de Calidad?

Tabla 266.

Entrega de la documentación completa para el inicio de las pruebas del funcional del cliente.

	Pretest		Postest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Pésimo	3	30%	0	0%
Malo	2	20%	0	0%
Regular	5	50%	2	20%
Bueno	0	0%	4	40%
Excelente	0	0%	4	40%
Total	10	100%	10	100%

Según los resultados de la Tabla 36 y la Figura 28, respecto a cuando el desarrollador implementador no entrega la documentación completa para que el funcional del cliente inicie sus pruebas en el ambiente de Calidad, en la etapa de Pre-Test los encuestados consideran como Regular en 50%, Malo en 20% y Pésimo en 30%, pero en la etapa de Pos-Test considerando el nuevo Modelo LACS los encuestados consideran en Regular 20%, Bueno en 40% y Excelente en 40%, en este sentido es importante mencionar que los encuestados consideran que el uso del nuevo modelo LACS va a ayudar a tener una documentación más completa, que se entregue a tiempo y con mayor grado de detalle, que considere las tablas, funciones, métodos y clases que realmente se están utilizando para corregir el programa.

Tabla 277.

Análisis de Fiabilidad de Alfa de Cronbach sobre la entrega de la documentación completa para el inicio de las pruebas del funcional del cliente.

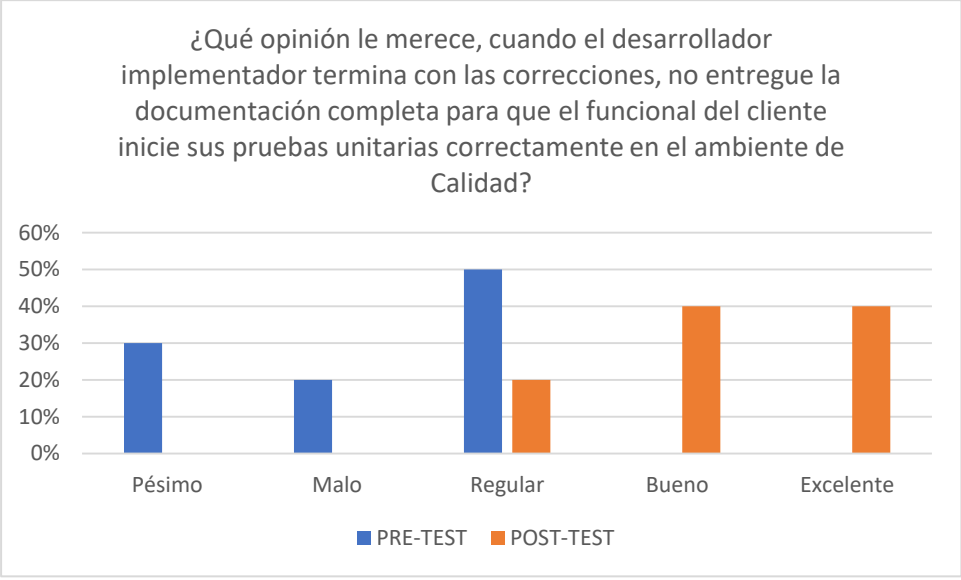
Estadísticos de fiabilidad

Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,918	,924	2

En la Tabla 37 se puede observar que la confiabilidad del instrumento de correlación de datos de Alfa de Cronbach nos muestra un coeficiente de 0.918, indica que el instrumento es altamente confiable, se utiliza para medir en la etapa de Pre-Test y Pos-Test la pregunta cuando el desarrollador implementador no entrega toda la documentación completa al funcional del cliente para que pueda iniciar las pruebas unitarias y funcionales.

Figura 28.

Entrega de la documentación completa para el inicio de las pruebas del funcional del cliente.



¿Como considera que el personal desarrollador del cliente no pueda revisar nada del código SAP hasta la última etapa del proyecto cuando el desarrollador implementador

termine con todas las correcciones a las funciones y métodos que son necesarios para la implementación del sistema SAP?

Tabla 288.

El desarrollador del cliente no puede revisar nada del código hasta la última etapa del proyecto.

	Pretest		Postest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Pésimo	3	30%	0	0%
Malo	2	20%	0	0%
Regular	5	50%	2	20%
Bueno	0	0%	4	40%
Excelente	0	0%	4	40%
Total	10	100%	10	100%

Según los resultados de la Tabla 38 y la Figura 29, respecto a que el personal desarrollador del cliente no pueda revisar nada del código que está desarrollando el desarrollador implementador, según las respuestas tenemos en la etapa del Pre-Test que opinan como Regular en 50%, Malo en 20% y Pésimo en 30%, pero en la etapa de Pos-Test se propone utilizar el nuevo Modelo LACS se puede observar que la respuesta de Regular disminuye a 20%, pero aumenta la respuesta de Bueno en 40% y Excelente en 40%, en este sentido es importante mencionar los encuestados consideran que el nuevo Modelo LACS ayudara a mejorar el hecho que los desarrolladores del cliente puedan conocer el código mucho antes, es decir se plantea que conozcan el código de programación desde la primera revisión que realice el desarrollador implementador, como se comentó el código está disponible desde la transacción de SAP, solo falta asignar los permisos de solo lectura para pueda ser revisado en paralelo conforme el desarrollador implementador va avanzando, el personal desarrollador del cliente, puede ir revisando las tablas, funciones, métodos y clases, de esta manera al terminar el proyecto tendrán un nivel mucho más alto de la programación en lenguaje Abap para brindar el soporte a cualquier módulo de SAP.

Tabla 39.

Análisis de Fiabilidad de Alfa de Cronbach para cuando el desarrollador del cliente no puede revisar nada del código hasta la última etapa del proyecto.

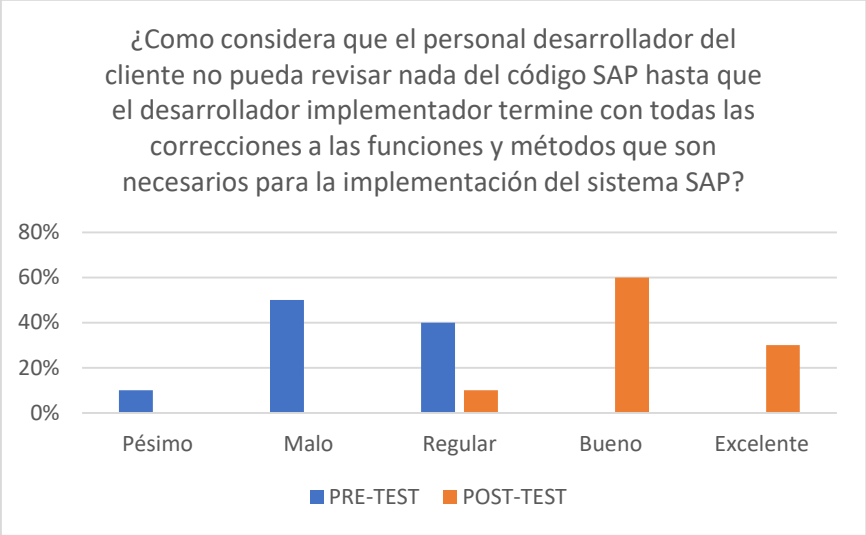
Estadísticos de fiabilidad

Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,938	,939	2

Según la Tabla 39, se presenta la confiabilidad del instrumento de correlación de datos de Alfa de Cronbach, que servirá para evaluar la consistencia interna de la escala de medición de las respuestas de los expertos en los cuestionarios, se muestra un coeficiente con un valor de 0.938 que nos indica un instrumento altamente confiable, se mide las respuestas en la etapa de Pre-Test y Pos-Test en la pregunta como considera que el personal desarrollador del cliente no pueda revisar nada del código creado por el desarrollador implementador hasta la última etapa del proyecto.

Figura 29.

El desarrollador del cliente no puede revisar nada del código hasta la última etapa del proyecto.



¿Como considera que el funcional responsable de las pruebas unitarias del cliente no pueda realizar ninguna prueba en el ambiente de Calidad hasta que el desarrollador implementador termine con todas las correcciones a las funciones y métodos que son necesarios para la implementación del sistema SAP?

Tabla 290.

El funcional del cliente no puede revisar nada del código hasta el término de todas las correcciones.

	Pretest		Postest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Pésimo	2	20%	0	0%
Malo	5	50%	0	0%
Regular	3	30%	3	30%
Bueno	0	0%	4	40%
Excelente	0	0%	3	30%
Total	10	100%	10	100%

Según los resultados presentados en la Tabla 40 y la Figura 30, respecto a que el funcional del cliente no pueda iniciar ninguna prueba unitaria hasta que el desarrollador implementador haya terminado con todas las correcciones de todo el grupo de correcciones, se observa en la etapa de Pre-Test que el resultado del cuestionario indica como Regular en 30%, Malo en 50% y Pésimo en 20%, pero en la etapa de Pos-Test considerando el nuevo Modelo LACS se observa que los encuestados consideran como Regular en 30%, Bueno en 40% y

Excelente en 30%, en este sentido es importante mencionar que los encuestados favorecen el nuevo Modelo LACS porque el nuevo modelo propone que no se cree solo una Orden de Transporte que contiene todas las correcciones que realizara el desarrollador implementador, sino que se crea una Orden de transporte que contenga solo una corrección, es decir se realiza la corrección de un programa, se libera la orden de transporte al ambiente de Calidad y está lista para que el funcional del cliente pueda ingresar y realizar la prueba unitaria y funcional, de tal modo que el desarrollador implementador tendrá que crear otra Orden de Transporte para la siguiente corrección, de esta manera se crea una Orden de transporte para cada atención del código, se realiza la prueba funcional y si es correcta, se va a cerrar y se continua con la siguiente corrección en el programa hasta terminar con todas las correcciones necesarias para una correcta implementación.

Tabla 301.

Análisis de Fiabilidad de Alfa de Cronbach como el funcional del cliente no puede revisar nada del código hasta el término de todas las correcciones.

Estadísticos de fiabilidad

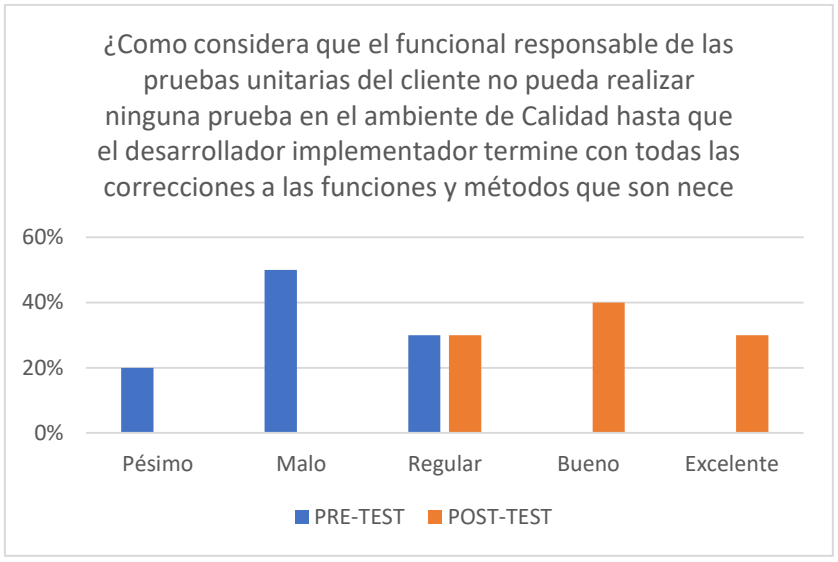
Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,957	,959	2

Según la Tabla 41 se puede observar que la confiabilidad del instrumento de correlación de datos con el Alfa de Cronbach necesaria para evaluar la consistencia interna de la escala de medición a través de la respuestas de los expertos, como se considera que el funcional del cliente no pueda iniciar ninguna prueba unitaria ni funcional hasta que el desarrollador implementador haya terminado con la atención de todas sus correcciones, se mide las

respuestas en la etapa de Pre-Test y Pos-Test, según se muestra tenemos un coeficiente de 0.957, indica que el instrumento es altamente confiable.

Figura 30.

El funcional del cliente no puede revisar nada del código hasta el término de todas las correcciones.



.

4.2 Contrastación de hipótesis

Para contrastar la hipótesis de estudio que se ha formulado en la sección 3.5, los cálculos que se van a presentar son para contrastar las hipótesis de estudio, empezaremos por la contrastación de las hipótesis específicas para luego llegar a la Hipótesis General.

4.2.1 Contrastación de la Primera Hipótesis Específica

En la Primera Hipótesis Especifica se indica lo siguiente:

Si se aplica el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software aumentará la cantidad de pruebas unitarias válidas en la ejecución de proyectos de implementación del sistema SAP.

En esta primera hipótesis especifica se utilizará el indicador de la dimensión asociada a la primera Hipótesis específica, de tal modo que podamos realizar la medida al indicador.

Ho: No existe diferencia significativa entre la cantidad de pruebas unitarias validas en la evaluación en el Pre-Test y Pos-Test para la implementación del nuevo Modelo LACS.

H1: Si existe diferencia significativa entre la cantidad de pruebas unitarias validas en la evaluación en el Pre-Test y Pos-Test para la implementación del nuevo Modelo LACS.

Nivel de Significación: 5%

Pruebas de chi-cuadrado			
	Valor	gl	Sig. asintótica (bilateral)
Chi-cuadrado de Pearson	15,625[a]	4	,004
Razón de verosimilitudes	17,279	4	,002
Asociación lineal por lineal	7,826	1	,005

Decisión: De la prueba de chi-cuadrado obtenemos 4 grados de libertad y un nivel de significancia de 0.004, por tal motivo el nivel de significancia es menor al valor: 0.05, entonces podemos rechazar la hipótesis nula Ho y podemos aceptar la hipótesis alterna H1.

Conclusión: Hay evidencia que indica que existe diferencia significativa en la cantidad de pruebas unitarias validas en la evaluación en el Pre-Test y Pos-Test para la implementación del nuevo modelo LACS.

4.2.2 Contrastación de la Segunda Hipótesis Específica

En la Segunda Hipótesis Especifica se indica lo siguiente:

Si se aplica el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará la capacitación del personal actual en la ejecución de proyectos de implementación del sistema SAP.

En esta segunda hipótesis especifica se utilizará el indicador de la dimensión asociada a la segunda Hipótesis específica, de tal modo que podamos realizar la medida al indicador.

Ho: No existe diferencia significativa entre incrementar las capacitaciones al personal actual en la evaluación en el Pre-Test y Pos-Test para la implementación del nuevo Modelo LACS.

H1: Si existe diferencia significativa entre incrementar las capacitaciones al personal actual en la evaluación en el Pre-Test y Pos-Test para la implementación del nuevo Modelo LACS.

Nivel de Significación: 5%

Pruebas de chi-cuadrado			
	Valor	gl	Sig. asintótica (bilateral)
Chi-cuadrado de Pearson	16,250[a]	4	,003
Razón de verosimilitudes	13,460	4	,009
Asociación lineal por lineal	7,049	1	,008

Decisión: De la prueba de chi-cuadrado obtenemos 4 grados de libertad y un nivel de significancia de 0.003, por tal motivo el nivel de significancia es menor al valor: 0.05, entonces podemos rechazar la hipótesis nula Ho y podemos aceptar la hipótesis alterna H1.

Conclusión: Hay evidencia que indica que existe diferencia significativa que se incrementara las capacitaciones al personal actual en la evaluación en el Pre-Test y Pos-Test para la implementación del nuevo modelo LACS.

4.2.3 Contrastación de la Tercera Hipótesis Específica

En la Tercera Hipótesis Especifica se indica lo siguiente:

Si se aplica el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará la cantidad de documentación técnica en la ejecución de proyectos de implementación del sistema SAP.

En esta tercera hipótesis especifica se utilizará el indicador de la dimensión asociada a la tercera Hipótesis específica, de tal modo que podamos realizar la medida al indicador.

Ho: No existe diferencia significativa entre incrementar la cantidad de documentación técnica valida en la evaluación en el Pre-Test y Pos-Test para la implementación del nuevo Modelo LACS.

H1: Si existe diferencia significativa entre incrementar la cantidad de documentación técnica valida en la evaluación en el Pre-Test y Pos-Test para la implementación del nuevo Modelo LACS.

Nivel de Significación: 5%

Pruebas de chi-cuadrado

	Valor	gl	Sig. asintótica (bilateral)
Chi-cuadrado de Pearson	13,600[a]	4	,009
Razón de verosimilitudes	13,863	4	,008
Asociación lineal por lineal	6,651	1	,010

Decisión: De la prueba de chi-cuadrado obtenemos 4 grados de libertad y un nivel de significancia de 0.009, por tal motivo el nivel de significancia es menor al valor: 0.05, entonces podemos rechazar la hipótesis nula Ho y podemos aceptar la hipótesis alterna H1.

Conclusión: Hay evidencia que indica que existe diferencia significativa que se incrementara la cantidad de documentación técnica valida en la evaluación en el Pre-Test y Pos-Test para la implementación del nuevo modelo LACS.

4.2.4 Contrastación de Hipótesis General

En la Hipótesis General se indica lo siguiente:

Si se aplica el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software entonces mejorará de manera sustantiva la ejecución de proyectos de implementación del sistema SAP.

Luego de contrastar las hipótesis específicas, las cuales indican que la implementación del nuevo modelo LACS aumentara la cantidad de pruebas unitarias validas, incrementara la capacitación del personal actual e incrementara la cantidad de documentación técnica valida, Por esos motivos se acepta la hipótesis general, de este modo se contrasta la hipótesis.

V. DISCUSIÓN DE RESULTADOS

Patricio & Carmen (2012), en su investigación: "Metodologías Ágiles en el Desarrollo de Software", mencionan que el desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte, tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software.

Por lo tanto, utilizar el nuevo modelo LACS propone aumentar la interacción entre los integrantes del equipo, una estrecha colaboración entre el desarrollador implementador de la consultora y el personal de cliente que serían el desarrollador y el encargado de realizar todas las pruebas unitarias y funcionales necesarias para asegurar un alto grado de calidad en el sistema que se desea implementar, sabemos que un sistema mal implementado puede ocasionar

perdidas para la empresa en diferentes aspectos como la perdida de una posición competitiva en el mercado, problemas con las compras de materiales, despacho de productos, las obligaciones tributarias, por esos motivos, el trabajo en equipo y colaborativo es la clave para el éxito del proyecto.

Según Jacobson et al. (2000), nos menciona que, para alcanzar el mayor grado de economía en el desarrollo, un equipo de proyecto intentará seleccionar sólo las iteraciones requeridas para lograr el objetivo del proyecto. Intentará secuenciar las iteraciones en un orden lógico. Un proyecto con éxito se ejecutará de una forma directa, sólo con pequeñas desviaciones del curso que los desarrolladores planifican inicialmente. Por supuesto, en la medida en que se añadan iteraciones o se altere el orden de estas por problemas inesperados, el proceso de desarrollo consumirá más esfuerzo y tiempo. Uno de los objetivos de la reducción del riesgo es minimizar los problemas inesperados. La iteración controlada acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan de manera más eficiente para obtener resultados claros a corto plazo, en lugar de tener un calendario largo, que se prolonga eternamente.

Por consiguiente, la propuesta del nuevo método LACS indica que se consumirá mayor esfuerzo y tiempo, sino se reduce los problemas, nos referimos a controlar la iteración, desarrollar el código necesario para realizar la implementación correcta de la función, método y clase necesario e inmediatamente el funcional del cliente inicie con su prueba unitaria y funcional, luego si esta correcto la atención se tendrá que realizar la importación de la orden de transporte al ambiente de producción, de esta manera se van cerrando cada una de las observaciones que tiene el desarrollador implementador por corregir, en ese sentido el desarrollador obtendrá resultados inmediatos en el menor tiempo posible, de esta manera

estimula el avance de los demás integrantes del equipo, porque los resultados se van viendo a corto plazo.

Según Cáceres et al. (2010) en su investigación menciona que las metodologías clásicas de ingeniería del software son en general pesadas y no facilitan el desarrollo rápido de aplicaciones, por esta razón se hace necesario una metodología que ayude al diseñador en el proceso de desarrollo de aplicaciones, la tendencia actual en el proceso de desarrollo apuesta por metodologías agiles que adaptan el proceso de desarrollo al desarrollo de software específico y que permiten fácilmente incorporar y contemplar cambios en los requisitos. Es por ello que parece adecuado para el tipo de desarrollo de software que sería más iterativo e incremental, es decir un proceso que se adapte al desarrollo de los diferentes tipos de aplicaciones, permitiendo la incorporación de nuevos requisitos y sin exigir una excesiva generación de documentación.

Por tal razón, el nuevo método LACS propone un desarrollo de software más iterativo e incremental, más dinámico entre el avance del desarrollador y reuniones diarias con funcional responsable de las pruebas unitarias y el desarrollador del cliente para avanzar con la documentación necesaria, crear los documentos de diseño que sean necesarios para la correcta identificación de las tablas, funciones, métodos y clases para un futuro mantenimiento, solo exigir la documentación necesaria pero detallada de los programas que están corrigiendo para una correcta implementación.

Según Jeffries et al. (2001), en su investigación menciona que depende de la complejidad del sistema, debe haber al menos una historia por cada característica importante, y propone realizar una o dos historias por programador por mes. Si se tienen menos, probablemente sea conveniente dividir las historias, si se tienen más lo mejor es disminuir el

detalle y agruparlas. Para efectos de planificación, las historias pueden ser de una a tres semanas de tiempo de programación (para no superar el tamaño de una iteración).

No hay que preocuparse si en un principio no se identifican todas las historias de usuario. Al comienzo de cada iteración estarán registrados los cambios en las historias de usuario y según eso se planificará la siguiente iteración. Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

En consecuencia, se menciona que las historias de usuario son una técnica utilizada en XP para especificar los requisitos del software, las historias de usuario son tarjetas de papel en el cual el cliente describe las características que el sistema debe tener, es muy dinámico y flexible, las historias de usuario pueden cambiar, romperse o reemplazarse por otras más específicas, agregar nuevas o ser modificadas, una historia de usuario es suficientemente comprensible para que el programador pueda realizar la implementación, es por ello que en la propuesta del nuevo modelo LACS se considera un desarrollo iterativo e incremental entre el desarrollador implementador encargado de realizar las modificaciones en el programa y el analista funcional que empieza a realizar las pruebas en el ambiente de calidad, según las pruebas en ese momento se podría ajustar alguna funcionalidad, si es necesario modificar el requerimiento, de esta manera se consigue un tratamiento con el usuario muy dinámico y flexible para lograr una implementación de alta calidad para el beneficio del proyecto.

Según Robert (2002), en su investigación señala que señala que el diseño del sistema de software es una cosa viviente. No se puede imponer todo en un inicio, pero en el transcurso del tiempo este diseño evoluciona conforme cambia la funcionalidad del sistema. Para mantener un diseño apropiado, es necesario realizar actividades de cuidado continuo durante el ciclo de vida del proyecto.

De hecho, este cuidado continuo sobre el diseño es incluso más importante que el diseño inicial. Un concepto pobre al inicio puede ser corregido con esta actividad continua, pero sin ella, un buen diseño inicial se degradará.

De manera que el nuevo método LACS propone reuniones constantes entre el desarrollador implementador, el desarrollador del cliente y el analista funciona encargado de las pruebas, para que constantemente se coordine porque como se indica el diseño del sistema de software es una cosa viviente, no se debe imponer desde un principio, que en el transcurso del tiempo puede cambiar, el requerimiento puede cambiar y se debe ajustar según la funcionalidad del sistema, para obtener un sistema adecuado a las necesidades del cliente.

Según Cockbun (2001), en su investigación en un estudio realizado para identificar los costos y beneficios de la programación en parejas, las principales ventajas de introducir este estilo de programación son: muchos errores son detectados conforme son introducidos en el código (inspecciones de código continuas), por consiguiente la tasa de errores del producto final es más baja, los diseños son mejores y el tamaño del código menor (continua discusión de ideas de los programadores), los problemas de programación se resuelven más rápido, se posibilita la transferencia de conocimientos de programación entre los miembros del equipo, varias personas entienden las diferentes partes sistema, los programadores conversan mejorando así el flujo de información y la dinámica del equipo, y finalmente, los programadores disfrutan más su trabajo. Dichos beneficios se consiguen después de varios meses de practicar la programación en parejas.

Por lo tanto, la propuesta del nuevo modelo LACS propone el trabajo coordinado en el desarrollo del programa entre el desarrollador implementador y el desarrollador del cliente, con ello se consigue que el desarrollador del cliente conozca todas las tablas, funciones, métodos y clases que está utilizando el desarrollador implementador para atender el problema, sino

también se está capacitando y revisando el código ABAP, de la misma manera trabajando en parejas pueden encontrar más rápido la solución de algún problema entre 2 desarrolladores, al final se beneficia la consultora porque su desarrollador tiene el apoyo del programador del cliente, se beneficia el desarrollador implementador porque tiene un apoyo para revisar, documentar y probar, se beneficia la empresa cliente porque tiene 2 personas revisando las correcciones, para asegurar el avance y cumplir con el cronograma, a la vez también se beneficia porque que su desarrollador a la vez que apoya en el código, también se está capacitando, se puede apreciar que ganan todos y se beneficia el proyecto de implementación.

Según Schwaber et al. (2001) en su investigación Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente.

La segunda característica importante son las reuniones a lo largo proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Por esta razón, el nuevo modelo LACS propone una interacción constante con el cliente, mostrar el avance de cada iteración que realiza el desarrollador implementador, cada programa que se termina de corregir se debe liberar la orden de transporte y pasar el código al ambiente de Calidad para que sea revisado por el funcional responsable de las pruebas unitarias y funcionales, también agregar que las reuniones cortas de 15 minutos para llenar algún cuadro de Excel donde se agregue el porcentaje diario de avance, sino es un avance agregar el motivo del problema y revisar las alternativas para solucionar el problema, cada día debe haber un

avance en el porcentaje, ya sea de coordinación para el desarrollo o empezando la integración con el sistema, las reuniones diarias de 15 minutos sirven para monitorear el avance de las pruebas unitarias, el avance en el desarrollo de la documentación técnica y el avance en las capacitaciones para seguir conociendo los objetos del nuevo sistema.

Según Baruah (2015), en su investigación comenta que la comprensión y el cumplimiento de los requisitos individuales de cada cliente se ha reconocido como un desafío urgente para las industrias del software. Producir productos de software de alta calidad y cumplir con los requisitos de las partes interesadas es un desafío importante en los requisitos de software. Los requisitos deficientes y los cambios en los requisitos son una de las causas del exceso de proyectos y los problemas de calidad en el software entregado. El documento analiza cómo las diferentes metodologías ágiles siguen los pasos de gestión de requisitos en un proyecto. Intenta dar una idea a aquellas organizaciones que se encuentran en proyectos con cambios frecuentes en los requisitos para que puedan producir productos de calidad y sobrevivir en la estrategia de mercado.

En definitiva, el método LACS propone realizar un producto de software de alta calidad, el seguimiento diario y la iteración continua con el cliente, cualquier cambio o ajuste detectado a tiempo, a causa de algún requisito no detectado inicialmente, asegura evitar problemas de calidad en la implementación final del proyecto.

VI. CONCLUSIONES

Conclusión general:

- El estudio realizado para la implementación del nuevo modelo LACS mejora significativamente la implementación del Sistema SAP.
- Luego de contrastar las hipótesis específicas y general, las cuales indican que la implementación del nuevo modelo LACS aumentara la cantidad de pruebas unitarias validas en 40%, también se incrementara la capacitación del personal actual en 30% y finalmente se incrementara la cantidad de documentación técnica valida en 40%, siendo estos puntos muy significativos para el éxito del proyecto de implementación del Sistema SAP, Por esos motivos se acepta la hipótesis general, de este modo se contrasta la hipótesis.

Conclusiones especificas:

- La implementación del nuevo modelo LACS mejora la cantidad de pruebas unitarias para la ejecución del proyecto SAP, se observa una mejora y un incremento del 40% en la cantidad de pruebas unitarias realizadas por el personal encargado de realizar las pruebas unitarias y funcionales por parte del cliente, también se observa una disminución del 30% en cuanto al tiempo que le tomaba iniciar las pruebas unitarias, por tal motivo se considera que la mejora es significativa para el beneficio de la implementación del Sistemas SAP.
- La implementación del nuevo modelo LACS mejora las capacitaciones al personal para la ejecución del proyecto SAP, se observa una mejora y un incremento del 30% en las capacitaciones y el traspaso de conocimiento de la consultora al personal del cliente, también se observa una disminución del 30% a las capacitaciones que no se

realizaban, por tal motivo se considera que la mejora es significativa para el beneficio de la implementación del Sistemas SAP.

- La implementación del nuevo modelo LACS mejora la cantidad de documentación técnica válida para la ejecución del proyecto SAP, se observa una mejora y un incremento del 40% en la cantidad de documentación técnica valida, completa y con el detalle necesario para realizar en un futuro cualquier tipo de mantenimiento por parte del cliente, también se observa una disminución del 30% en cuanto a la documentación que no se entrega completa ni con el detalle necesario, por tal motivo se considera que la mejora es significativa para el beneficio de la implementación del Sistemas SAP.

VII. RECOMENDACIONES

- Según los resultados obtenidos el inicio de las pruebas unitarias y funcionales podría mejorar aún más, si el desarrollador implementador libera el código creado al ambiente de Calidad por cada programa que va resolviendo, de esta manera junto a las reuniones diarias para revisar si hubo algún cambio en el requerimiento, de esta manera se podrá avanzar de manera segura para el éxito del proyecto.
- Se observa que se incrementara la capacitación al personal de la empresa porque van a desarrollar junto al personal desarrollador de la consultora, para que vayan revisando el código, es necesario que también se les involucre en las reuniones para revisar el avance de la corrección del código, de esa manera habrá mayor compromiso en aprender y empezar a participar en reportes menores que ayuden a su aprendizaje.
- Se puede verificar que la documentación que se va a preparar y va a quedar en el cliente después de la implementación, será una documentación completa con el detalle necesario para el futuro mantenimiento, pero es necesario complementar con documentación estándar, así como se debe agregar la documentación de configuración y las respectivas actualizaciones necesarias para el buen funcionamiento del sistema SAP.

VIII. REFERENCIAS

Aiken, L. (1985). Una vez calculados los estadísticos de consistencia inter-jueces procedimos a revisar la adecuación de los ítems a los criterios de validez. *Educational and Psychological Measurement*.

Amaro Calderón, S., & Valverde Rebaza, J. (2007). *Metodologías Ágiles.* Trujillo.

Ambler. (2002). *Agile Modeling.* New York: John Wiley & Sons, Inc.

Arias, M. (2006). La Ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes*.

Baruah, N. (2015). *Gestión de requisitos en un entorno de software ágil* . ELSEVIER.

Benkler, Y. (2012). *La Riqueza de las Redes.* Estados Unidos: Yale.

Bernal, J. (23 de Agosto de 2013). Obtenido de https://www.pdcahome.com/5202/ciclo-pdca/.

Bono Cabré, R. (s.f.). *Diseños cuasi-experimentales y longitudinales (Universidad de Barcelona)*. Obtenido de http://diposit.ub.edu/dspace/bitstream/2445/30783/1/D.%20cuasi%20y%20longitudinales.pdf

Brooks, F. (1995). *Libro de Administración de proyectos de software, The Mythical Man Month.* Massachusetts: Addison-Wesley.

Cáceres, P., & Marcos, E. (2010). *Procesos Ágiles para el Desarrollo de Aplicaciones web.* Madrid: Grupo Kybele.

Canos, J. (2003). *Metodologías Ágiles en el Desarrollo de Software. Taller de Metodologías Ágiles en el Desarrollo de Software.* Alicante: Universidad Politecnica de Valencia.

Castillo Asencio, P. (2016). *Desarrollo e implementación de un sistema web para generar valor en una pyme aplicando una metodología ágil.* Lima: Universidad Nacional Mayor de San Marcos.

Cockbun, A. (2001). *Agile Software Development.* Boston: Addison Wesley.

Cockburn, A. (2004). *Crystal Clear: A Human-Powered Methodology for Small Teams.* Londres, Reino Unido: Pearson Education.

Cohen, & Manion, L. (2002). *Métodos de investigación cuantitativa.* Madrid: La Muralla.

Cook, T., & Campbell, D. (1986). *The causal assumptions of quasiexperimental practice.* Synthese.

Delgado Exposito, E. (2008). Metodologías de desarrollo de software. *Revista de Arquitectura e Ingeniería*.

Diaz , J. (2009). Las metodologías ágiles como garantía de calidad del software. *REICIS. Revista Española de Innovación*, 40-43.

Enrich, R. (2013). *Implantación de un sistema ERP SAP en una empresa.* Barcelona: Universitat Politécnica de Catalunya.

Erazo Pérez, D. (2017). *Metodologías agiles para el desarrollo de proyectos en la gestión de organizaciones publicas en Colombia.* Colombia: Universidad Nacional UNAD.

Escurra M., L. (1988). Cuantificación de la validez de contenido por criterio de jueces. *Revista de Psicologia*.

Figueroa G., R., & J. Solis, C. (2015). *Metologías tradicionales.* Ecuador: Universidad Técnica de Loja.

Fiquitiva, N., & Lopez, M. (2015). *Prototipo de aplicativo para especificar requerimientos de software.* Bogota: Universidad Catolica de Colombia.

Forero, F. (2018). *Implementación de la Metodología SCRUM en un Ambiente Bancario.* Colombia: Universidad Nacional de Colombia.

García-Bustelo, P., & Begoña, C. (2007). *Desarrollo ágil de software con arquitecturas dirigidas por modelos.* Oviedo - España: Universidad de Oviedo.

Gutierrez, J., & Escalona, M. (2010). *PRUEBAS DEL SISTEMA EN PROGRAMACIÓN.* Sevilla: University of Sevilla.

Hedrick, T., Bickman, L., & Rog, D. (1993). *Applied Research Design.* Florida, USA: SAGE Publications, Inc.

Highsmith , J. (2002). *Agile Software Development Ecosystems.* San Francisco: Addison Wesley.

Huaylinos Gonzales, E. (2017). *Metodologías ágiles en la implementación de una aplicación móvil para la gestión de citas en la clínica dental "Perio Dent".* Huancayo: Universidad Nacional del Centro del Perú.

Huayta Garcia, M. (2006). *Aseguramiento de la Calidad.* Cuba: Instituto Superior Politécnico Jose Echevarría.

Institute, P. M. (2017). *Guida al Project Management Body of Knowledge.* Estados Unidos: Project Management Institute.

Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El proceso unificado de desarrollo del software.* Addison Wesley.

Jeffries, R., Anderson, A., & Hendrickson, C. (2001). *Extreme Programming Installed.* Boston: Addison Wesley.

Kniberg, H., & Skarin, M. (2010). *Kanban and Scrum - making the most of both.* United States of America: c4media.

Lomas, B. (2018). *Analisis y diseño para evaluar el proceso de software basados en msf.* Ecuador: Machala.

Martínez, & Hernández, M. (2014). *Psicometría.* Madrid: Alianza.

Molina Montero, B., Vite Cevallos, H., & Dávila Cuesta, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Revista Multidisciplinaria de investigación científica*, 2(17).

Navarro Cadavid, A., Fernandez Martinez, J., & Morales Vélez, J. (2013). *Revisión de metodologías ágiles para el desarrollo de software.* Colombia: Universidad Icesi.

Navarro, M., Moreno, M., & Aranda, J. (2017). *Integración de arquitectura de software en el ciclo de vida de las metodologías ágiles.* Buenos Aires: SEDICI.

Navarro, M., Moreno, M., & Rueda, J. (2017). *Selección de metodologías ágiles e integración de arquitecturas de software en el desarrollo de sistemas de información.* Buenos Aires: SEDICI.

Orjuela, A., & Rojas, M. (2018). Las metodologias de desarrollo ágil como una oportunidad para la ingeniería del software educativo. *Revista UNAL - Colombia, 5*(2).

Palmer, S., & Felsing, J. (2002). *Practical Guide to Feature-Driven Development.* United States of America: Prentice Hall.

Patricio, L., & Carmen, P. (2012). *Métodologías ágiles para el desarrollo de software.* Cuba: Ingenieria de Software.

Pedhazur, E., & Schmelkin, L. (1991). *Measurement, design, and analysis. An integrated approach.* New Jersey: Hilladale, Lawrence Erlbaum Associates.

Pons, C., & Giandini, R. (2010). *Desarrollo de software dirigido por modelos.* Buenos Aires: McGraw-Hill Educación.

R., M. (2002). *Continuos Care vs. Initial Design*. Obtenido de www.objectmentor.com: www.objectmentor.com/resources/articles/Continuous_Care.pdf

Raymond, E. (2001). *La Catedral y el bazar.* Buenos Aires.

Robert, M. (2002). *Continuos Care vs. Initial Design.* Obtenido de www.objectmentor.com/resources/articles/Continuous_Care.pdf

Samamé, J., & Humberto, J. (2013). *Aplicación de una metodología ágil en el desarrollo de un sistema de información.* Lima : Pontificia Universidad Católica del Perú.

Saravia, G. (2012). *Metodologías ágiles y desarrollo basado en conocimiento.* Buenos Aires - Argentina: Universidad Nacional de la plata.

Schwaber, K., Beedle, M., & Martin, R. (2001). *Agile Software Development with SCRUM.* Nueva Jersey: Prentice Hall.

Villegas, E., & Ruiz, J. (2017). *El conflicto en el agilismo: Una perspectiva desde el scrum.* Medellín: Universidad EAFIT.

Wong Portillo, L., & Torres Sánchez, F. (2010). Mejorando las debilidades de RUP para la gestión de proyectos. *Revista de Investigación de Sistemas e Informática*, 50-58.

IX. ANEXOS

A. Matriz De Consistencia

NUEVO MODELO LACS BASADO EN LEAN IT Y DESARROLLO AGIL PARA MEJORAR EL PROYECTO DE IMPLEMENTACION DE SAP

Definición del Problema	Objetivos	Formulación de Hipótesis	Clasificación de variables	Indicadores	Metodología
Problema General ¿De qué manera el nuevo Modelo LACS basado en LEAN IT y el Desarrollo Ágil de software mejorará la ejecución de proyectos de implementació	**Objetivo General** Determinar en qué medida el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software mejorará la	**Hipótesis General** Si se aplica el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software entonces mejorará de manera sustantiva la ejecución de	**Variable Dependiente** La ejecución del proyecto de implementació n del sistema SAP.	-Cantidad de pruebas unitarias válidas. - Capacitación del personal actual. - Cantidad de documentaci ón técnica	**Método de Investigació n** Es un diseño experimental, se interviene la variable independiente que es el nuevo Modelo LACS, se va a comparar el Pre-Test y el

n del sistema SAP? **Problemas Específicos** 1.- ¿De qué manera un nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software aumentará la cantidad de pruebas unitarias válidas en la ejecución de proyectos de implementació n del sistema SAP?	ejecución de proyectos de implementac ión del sistema SAP. **Objetivos Específicos** 1.- Determinar en qué medida el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software aumentará la cantidad de	proyectos de implementaci ón del sistema SAP. **Hipótesis Especificas** 1.- Si se aplica el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software aumentará la cantidad de pruebas unitarias válidas en la ejecución de proyectos de implementaci	**Variable Independiente** Es el Nuevo Modelo que se ha creado llamado LACS	- Monto real que tiene el proyecto - Variación del cronograma - Índice del tiempo que se estima con el tiempo real.	Post-Test, se medirá el cambio ocasionado en el experimento, es lo que se llama Causa-Efecto.

2.- ¿De qué manera un nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará la capacitación del personal actual en la ejecución de proyectos de implementación del sistema SAP? 3.- ¿De qué manera un nuevo modelo LACS basado	pruebas unitarias válidas en la ejecución de proyectos de implementación del sistema SAP. 2.- Determinar en qué medida el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará la	ón del sistema SAP. 2.- Si se aplica el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará la capacitación del personal actual en la ejecución de proyectos de implementación del sistema SAP.			

en LEAN IT y el Desarrollo Ágil de software incrementará la cantidad de documentación técnica en la ejecución de proyectos de implementació n del sistema SAP?	capacitación del personal actual en la ejecución de proyectos de implementac ión del sistema SAP. 3.- Determinar en qué medida el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará	3.- Si se aplica el nuevo modelo LACS basado en LEAN IT y el Desarrollo Ágil de software incrementará la cantidad de documentació n técnica valida en la ejecución de proyectos de implementaci ón del sistema SAP.			

	la cantidad de documentación técnica en la ejecución de proyectos de implementación del sistema SAP.				

B. Validación Y Confiabilidad De Instrumentos

La Observación Sistemática

Mediante esta técnica, nos permitirá observar la coordinación entre el consultor desarrollador de la implementación junto al desarrollador de la empresa y el usuario funcional encargado de las pruebas unitarias, para atender a cada modificación en el código y pruebas de los métodos y funciones, de esta manera se podrá avanzar y cerrar cada prueba.

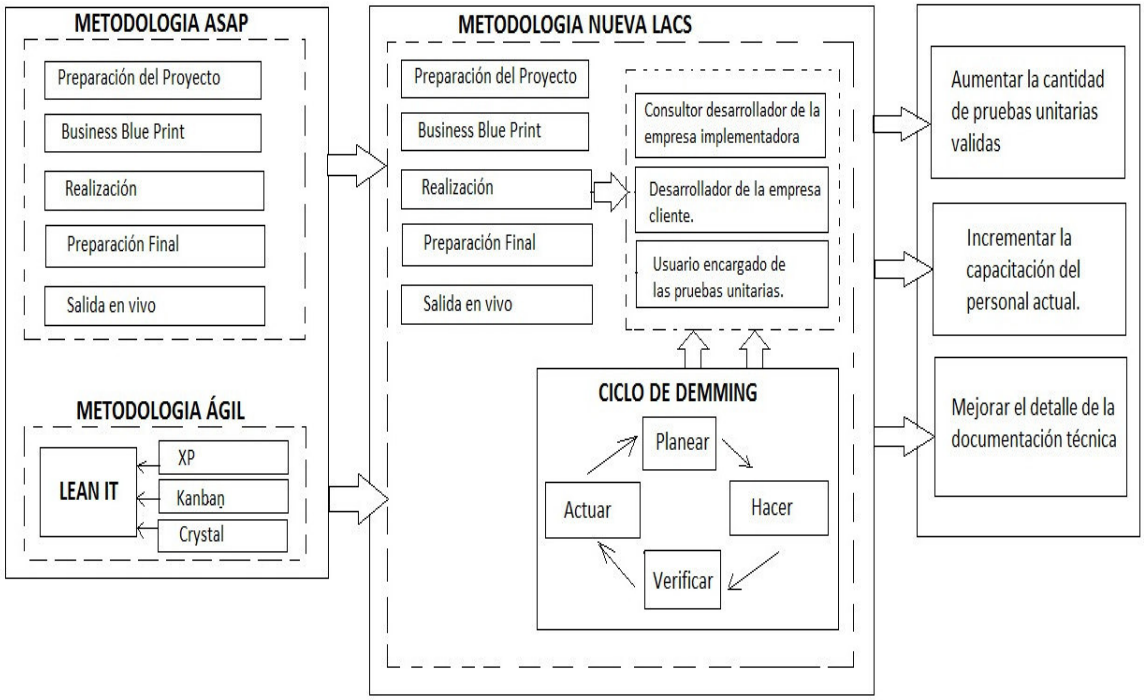
La Encuesta

Se utilizará la técnica de la encuesta para identificar la correcta ejecución del proyecto de implementación del sistema SAP y los índices en relación con el nuevo método LACS.

El Cuestionario

Para la determinación de los factores que influyen en la ejecución del proyecto de implementación del sistema SAP en base al nuevo método LACS, se elaborará y se aplicará un cuestionario.

C. Metodología Propuesta



Fuente Propia

D. Documento Para Invitar Al Juez Experto:

Señor(a)

…….

Presente. –

Tengo el agrado de dirigirme a Ud., para saludarlo (a) cordialmente y a la vez manifestarle que, conocedores de su trayectoria académica y profesional, molestamos su atención al elegirlo JUEZ EXPERTO para revisar el contenido del instrumento que pretendemos utilizar en la Tesis para optar el grado de Doctor en Ingeniería de Sistemas, por la Escuela de PosGrado de la Universidad Federico Villarreal.

El instrumento tiene como objetivo medir la variable independiente: Nuevo Modelo LACS, por lo que, con la finalidad de determinar la validez de su contenido, solicitamos marcar con una X el grado de evaluación a los indicadores para los ítems del instrumento, de acuerdo con su amplia experiencia y conocimientos. Se adjunta el instrumento y la matriz de operacionalización de la variable considerando dimensiones, indicadores, categorías y escala de medición.

Agradecemos anticipadamente su colaboración y estamos seguros de que su opinión y criterio de experto servirán para los fines propuestos.

Atentamente,

Luis Angel Contreras Sagástegui

E. Informe De Opinión De Experto.

Informe De Opinión De Expertos Del Instrumento De Investigación

I. Datos Generales:

1.1. Apellidos y nombres del informante (Experto): ……

1.2. Grado Académico: ……

1.3. Profesión: ……..

1.4. Institución donde labora: ……..

1.5. Cargo que desempeña: ……..……

1.6. Autor del instrumento: ……..

1.7. Programa de Postgrado: …….....

II. Validación:

INDICADORES DE EVALUACIÓN DEL INSTRUMENTO	CRITERIOS	Muy malo	Malo	Regular	Bueno	Muy Bueno
		1	2	3	4	5
1. CLARIDAD	Están formulados con lenguaje apropiado que facilita su comprensión.					
2. OBJETIVIDAD	Están expresados en conductas observables, medibles.					
3. CONSISTENCIA	Existe una organización lógica en los contenidos y relación con la teoría.					

4. COHERENCIA	Existe relación de los contenidos con los indicadores de la variable					
5. PERTINENCIA	Las categorías de respuestas y sus valores son apropiados.					
6. SUFICIENCIA	Son suficientes la cantidad y calidad de ítems presentados en el instrumento.					
	SUMATORIA PARCIAL					
	SUMATORIA TOTAL					

III. Resultados De La Validación

3.1 Valoración total cuantitativa: _____

3.2 Opinión: FAVORABLE: _____ DEBE MEJORAR: _____

NO FAVORABLE: _____

3.3 Observaciones: _____

_____

Firma

F. Cuestionario.

Cuestionario De Tesis

El presente instrumento forma parte del trabajo de investigación titulada:

"Nuevo Modelo LACS basado en Lean IT y Desarrollo Ágil para mejorar la ejecución de proyectos de implementación de sistemas SAP."

La información es de carácter confidencial y reservado; ya que los resultados serán manejados solo para la investigación.

Agradezco anticipadamente su valiosa colaboración.

Instrucciones:

A continuación, se les presenta 10 preguntas que deberá responder:

- Marcando con un aspa (x) en la letra donde indique la respuesta que más se acerca a su modo de pensar.

Pregunta 1. – Cuál es su opinión respecto al tiempo que se toma entre la corrección de una función del desarrollador implementador y el inicio de las pruebas unitarias que realiza el funcional del cliente.

a. Excelente
b. Bueno
c. Regular
d. Malo
e. Pésimo

Pregunta 2.- Qué opinión tiene cuando el desarrollador implementador envía al funcional cliente todas las pruebas en bloque para que inicie con sus pruebas unitarias en el ambiente de Calidad.

a. Excelente

b. Bueno

c. Regular

d. Malo

e. Pésimo

Pregunta 3. – Como considera realizar las pruebas unitarias de un sistema ERP en un tiempo muy corto, podría ocasionar errores en las pruebas funcionales.

a. Excelente

b. Bueno

c. Regular

d. Malo

e. Pésimo

Pregunta 4. – Considera que las pruebas unitarias que realizara el funcional del cliente son de alta calidad, por recibir todas las pruebas juntas en bloque, cuando el desarrollador termino con todas las correcciones.

a. Excelente

b. Bueno

c. Regular

d. Malo

e. Pésimo

Pregunta 5. – Considera que alcanzara el tiempo al funcional cliente para realizar todas las pruebas unitarias y también crear otros escenarios de prueba considerando su realidad del día a día.

a. Excelente

b. Bueno

c. Regular

d. Malo

e. Pésimo

Pregunta 6. – Como considera la documentación técnica que entrega la consultora al término de la implementación, cubre todas las expectativas.

a. Excelente

b. Bueno

c. Regular

d. Malo

e. Pésimo

Pregunta 7. – Considera que las capacitaciones de la consultora al personal desarrollador del cliente son las necesarias y suficientes, en cuanto al traspaso de conocimiento.

a. Excelente

b. Bueno

c. Regular

d. Malo

e. Pésimo

Pregunta 8. - Qué opinión le merece, cuando el desarrollador implementador termina con las correcciones, no entregue la documentación completa para que el funcional del cliente inicie sus pruebas unitarias correctamente en el ambiente de Calidad.

a. Excelente

b. Bueno

c. Regular

d. Malo

e. Pésimo

Pregunta 9. – Como considera que el personal desarrollador del cliente no pueda revisar nada del código SAP hasta que el desarrollador implementador termine con todas las correcciones a las funciones y métodos que son necesarios para la implementación del sistema SAP.

a. Excelente

b. Bueno

c. Regular

d. Malo

e. Pésimo

Pregunta 10. - Como considera que el funcional responsable de las pruebas unitarias del cliente no pueda realizar ninguna prueba en el ambiente de Calidad hasta que el desarrollador implementador termine con todas las correcciones a las funciones y métodos que son necesarios para la implementación del sistema SAP.

a. Excelente

b. Bueno

c. Regular

d. Malo

e. Pésimo

G. Confiabilidad De Instrumentos

El cuestionario integrado por 10 preguntas que exploran el conocimiento de los entrevistados se realizó a 10 personas expertas y que participaron durante todo el proyecto de implementación del sistema, que duró aproximadamente 2 años para la salida en vivo.

Con las respuestas obtenidas se creó una base de datos en SPSS para Windows, se calculó el coeficiente Alfa de Cronbach para los ítems más relevantes para los efectos de determinar la confiabilidad del cuestionario.

Se obtuvo un cuestionario de confiabilidad aceptable, se concluye que el análisis de fiabilidad es un procedimiento efectivo, los resultados que se obtuvieron se presentan en la siguiente tabla.

Indice de Confiabilidad de Alfa de Cronbach para la etapa del Pre-Test respecto al cuestionario de 10 preguntas.

Estadísticos de fiabilidad	
Alfa de Cronbach	N de elementos
,959	10

Índice de Confiabilidad de Alfa de Cronbach para la etapa del Pos-Test respecto al cuestionario de 10 preguntas.

Estadísticos de fiabilidad	
Alfa de Cronbach	N de elementos
,946	10