



UNIVERSIDAD CÉSAR VALLEJO

**ESCUELA DE POSGRADO
PROGRAMA ACADÉMICO DE MAESTRÍA EN INGENIERÍA
DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA
INFORMACIÓN**

**Marco de trabajo Scrum para la gestión de proyectos de software en una
Empresa Consultora, Lima 2023**

**TESIS PARA OBTENER EL GRADO ACADÉMICO DE:
MAESTRA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN
TECNOLOGÍAS DE LA INFORMACIÓN**

AUTOR:

Gutierrez Estrella, John Williams (orcid.org/0000-0002-8366-6988)

ASESOR:

Dr. Acuña Benites, Marlon Frank (orcid.org/0000-0001-5207-9353)

CO-ASESOR:

Dr. Flores Zafra, David (orcid.org/0000-0001-5846-325X)

LÍNEA DE INVESTIGACIÓN:

Sistemas de Información y Comunicaciones

LÍNEA DE RESPONSABILIDAD SOCIAL UNIVERSITARIA:

Desarrollo económico, empleo y emprendimiento

LIMA – PERÚ
2023

Dedicatoria

A mis padres por ser el soporte y guía permanente, incentivando mis deseos de superación y desarrollo profesional, gracias por enseñarme a enfrentar los desafíos y por ayudarme a salir adelante en los momentos más difíciles. A mi esposa y a mis hijos Alonso y Mateo quienes son la fuente de mi motivación e inspiración. A mis profesores, gracias por su tiempo y apoyo, gracias por los conocimientos transmitidos durante el desarrollo de mi formación profesional.

Agradecimiento

Especialmente a mi esposa e hijos por su comprensión y paciencia. A mis padres por enseñarme desde muy pequeño a perseverar y luchar por nuestros sueños. Agradezco profundamente a todos los docentes que me acompañaron y guiaron a lo largo de este proceso. Este logro alcanzado más que el mío es de todos aquellos que lo hicieron posible.

Índice de contenidos

	Pág.
Caratula	i
Dedicatoria	ii
Agradecimiento	iii
Índice de contenidos	iv
Índice de tablas	v
Índice de gráficos y figuras.....	vi
Resumen.....	vii
Abstract.....	viii
I. INTRODUCCIÓN.....	1
II. MARCO TEÓRICO.....	6
III. METODOLOGÍA.....	20
3.1. Tipo y diseño de investigación.....	20
3.2. Variables y operacionalización.....	22
3.3. Población, muestra, muestreo, unidad de análisis.....	24
3.4. Técnicas e instrumentos de recolección de datos.....	26
3.5. Procedimientos.....	27
3.6. Método de análisis de datos.....	27
3.7. Aspectos éticos.....	28
IV. RESULTADOS.....	30
V. DISCUSIÓN.....	48
VI. CONCLUSIONES.....	55
VII. RECOMENDACIONES.....	57
REFERENCIAS.....	58
ANEXOS.....	66

Índice de tablas

	Pág.
Tabla 1 Definición de variables.....	22
Tabla 2 Lista de verificación.....	25
Tabla 3 Opciones para la lista de verificación.....	25
Tabla 4 Validación del instrumento	26
Tabla 5 Análisis descriptivo para cantidad de fallas del sistema.....	30
Tabla 6 Análisis descriptivo para el tiempo de entrega del producto	33
Tabla 7 Análisis descriptiva para cantidad de solicitudes de cambio.....	36
Tabla 8 Prueba de normalidad para el indicador 1	39
Tabla 9 Prueba de normalidad para el indicador 2	40
Tabla 10 Prueba de normalidad para el indicador 3	40
Tabla 11 Resultados obtenidos de las pruebas de normalidad.....	41
Tabla 12 Resultados de la prueba de U-Mann Whitney para el indicador cantidad de fallas del sistema.....	42
Tabla 13 Resumen de la prueba de U-Mann Whitney para el indicador cantidad de fallas del sistema.....	42
Tabla 14 Prueba de homogeneidad de varianzas para el indicador tiempo de entrega del producto	44
Tabla 15 Resultados de la prueba T-Student Welch para el indicador tiempo de entrega del producto	44
Tabla 16 Resultados de la prueba de U-Mann Whitney para el indicador cantidad de solicitudes de cambio	46
Tabla 17 Resumen de la prueba de U-Mann Whitney para el indicador cantidad de solicitudes de cambio	46

Índice de gráficos y figuras

	Pág.
Figura 1 Décimo Reporte de Agilidad.....	14
Figura 2 Estudio cuasiexperimental.....	21
Figura 3 Comparación de las medias para cantidad de fallas del sistema	31
Figura 4 Gráfico de cajas para cantidad de fallas del sistema.....	32
Figura 5 Comparación de las medias para el tiempo de entrega del producto .	34
Figura 6 Gráfico de cajas para el tiempo de entrega del producto	35
Figura 7 Comparación de las medias cantidad de solicitudes de cambio	37
Figura 8 Gráfico de Cajas cantidad de solicitudes de cambio	38
Figura 9 Reducción de la cantidad de fallas del sistema.....	43
Figura 10 Reducción del tiempo de entrega del producto	45
Figura 11 Reducción de la cantidad de solicitudes de cambio	47

Resumen

La investigación tuvo como objetivo Determinar la influencia del marco de trabajo Scrum en la gestión de los Proyectos de Software en una Empresa Consultora, midiendo los resultados de la gestión de proyectos de manera tradicional y utilizando del marco Scrum en los indicadores cantidad de fallas del sistema, tiempo de entrega del producto y solicitudes de cambio al producto.

La investigación es aplicada con enfoque cuantitativo y diseño cuasiexperimental, la muestra lo conformaron 50 proyectos, divididos en dos muestras independientes, los datos se obtuvieron utilizando fichas de registro. Para validar las hipótesis se determinará la normalidad de datos utilizando la Z de Kolmogórov-Smirnov para cada indicador, con los resultados obtenidos se aplicó el test paramétrico de T-Student y el test no paramétrico de U-Mann Whitney.

Los resultados arrojaron que se logró disminuir la cantidad de fallas del sistema en 15.58%, igualmente se logró reducir el tiempo de entrega de los productos en 9.09% y por último se redujo la cantidad de solicitudes de cambio en 28.36%. Concluyendo que la implementación del Marco de Trabajo Scrum influye positivamente en la Gestión de los Proyectos de Software en una Empresa Consultora de Software.

Palabras clave: Scrum, Enfoque Predictivo y Enfoque Ágil.

Abstract

The objective of the research was to determine the influence of the Scrum framework on the management of Software Projects in a Consulting Company, measuring the results of project management in a traditional way and using the Scrum framework in the indicators number of system failures, product delivery time and product change requests.

The research is applied with a quantitative approach and quasi-experimental design, the sample was made up of 50 projects, divided into two independent samples, the data was obtained using registration sheets. To validate the hypotheses, the normality of the data will be determined using the Kolmogórov-Smirnov Z for each indicator, with the results obtained the parametric T-Student test and the non-parametric U-Mann Withney test were applied.

The results showed that it was possible to reduce the number of system failures by 15.58%, it was also possible to reduce the delivery time of the products by 9.09% and finally the number of change requests was reduced by 28.36%. Concluding that the implementation of the Scrum Framework positively influences the Management of Software Projects in a Software Consulting Company.

Keywords: Scrum, Predictive Approach and Agile Approach.

I. INTRODUCCIÓN

El desarrollo de proyectos en general ha estado gobernado por una gestión basado en metodologías de trabajo tradicional (framework o guía de buenas prácticas), la cual se caracteriza por una planificación detallada en etapas iniciales, dado que los objetivos que se esperan alcanzar eran claros y el producto siempre se entregaba al final. Hoy en día la globalización y el crecimiento tecnológico requiere que las organizaciones se adapten a estos cambios constantes, poniendo énfasis en adaptar sus procesos a estos cambios, es aquí donde aparecen los marcos de trabajo ágiles, cuya principal característica es que son flexibles a los cambios que se puedan presentar y a la colaboración total entre todo el equipo de trabajo.

Hoy en día el desarrollo de software es una de las industrias más grandes de la economía mundial, industria que se encuentra en constante crecimiento y cambio, por lo que es fundamental la entrega rápida y adecuada de los productos de software los cuales deben estar soportados sobre un proceso de calidad deseado, por lo tanto, los proyectos complejos pueden demorar años bajo un metodologías de trabajo tradicional debido a los constantes cambios a los que estaría sujeto el producto, teniendo en cuenta que el producto se entrega al final, mientras que bajo una metodología de trabajo ágil se entrega pequeños incrementos de trabajo (software) a través de iteraciones o espacios de tiempo. Entonces las metodologías de trabajo tradicionales son adecuadas para un entorno predecible, donde se tiene un control de todo el proyecto, mientras que las metodologías de trabajo ágil se adaptan mejor en entornos en constante cambio y escenarios de alta incertidumbre (Fagarasan et al., 2021).

En el ámbito internacional mencionamos el caso de una farmacéutica de Brasil, que utilizó el marco Scrum para la fabricación de un módulo para el control de inventario ERP, el cual se planificó desarrollarse en cinco iteraciones, el objetivo fue obtener productos funcionales al término de cada iteración y así sucedió, con lo cual se demostró como el marco Scrum ayuda en la entrega temprana de productos funcionales, en comparación con los marcos tradicionales (Azanha et al., 2017).

Las organizaciones en general impulsan sus estrategias a través de la implementación de tecnologías, siendo los proyectos de software uno de sus principales impulsores, siendo así, es sumamente importante adoptar marcos de trabajo ágiles y diseñar procesos estandarizados que nos permitan gestionar los proyectos eficazmente, a fin de reducir costos, tiempos y mejorar la calidad. Según una investigación en Latinoamérica, actualmente el 97% de las organizaciones practican marcos de trabajo ágiles, de los cuales el 83% se encuentra en proceso de maduración en la adaptación de las prácticas de los marcos ágiles y un 78% sostiene que no todo el equipo de trabajo lo ha adoptado (Flores et al., 2021).

En el ámbito nacional también se han adoptado e implementado muchos marcos de trabajo ágiles, como hoja de ruta para la gestión de los proyectos. En el año 2019 International Data Corporation (2019), realizó un estudio en América Latina sobre cómo las metodologías ágiles han sido adoptadas y aceptadas, arrojando que un 40% de las organizaciones en el Perú, adoptaron y aplicaron alguna de las metodologías ágiles para gestionar sus proyectos, este porcentaje significó el auge y cuán importante fue su adopción en el Perú. El mismo estudio manifiesta que la adopción y sobre todo la aplicación de algunas de las metodologías ágiles en organizaciones, han originado cambios significativos y comprobados en el desarrollo de sus proyectos, específicamente señala mejoras del 80% en la productividad y en entregar de manera rápida los productos y servicios.

De igual manera, la guía del SBOK en su tercera edición, señala a Scrum como una de las prácticas ágiles más empleadas en el desarrollo de proyectos, debido a que los principios y herramientas que propone nos permiten abordar los proyectos en situaciones complejas de manera flexible y en periodos cortos de tiempo llamados iteraciones (Satpathy, 2017).

En el Perú, Scrum es una de las metodologías más utilizadas por las organizaciones, pero esta se ha adoptado de manera paulatina y las organizaciones han adaptado sus procesos y el manejo de sus proyectos, de acuerdo a lo que establece Scrum. Asimismo, han puesto radical interés en inculcar las prácticas de

Scrum en sus equipos de trabajo, con la finalidad de atender de manera rápida sus proyectos. Pero aún se presentan dificultades en el proceso de adopción.

En las empresas privadas la realidad es similar a la mencionada anteriormente, la aplicación de Scrum como metodología para un correcto manejo de los proyectos de software está en etapa de adopción, por lo tanto, no hay un proceso definido que oriente y guíe el desarrollo de los proyectos, esta situación es la que se presenta en la empresa consultora de proyectos, en la cual se aplicó la presente investigación. la adopción del marco Scrum se dio a inicios del año 2021, antes de este año los proyectos eran gestionados bajo el enfoque predictivo, teniendo en cuenta que muchos de esos proyectos tenían un alto grado de incertidumbre en cuanto a lo que el cliente quería como producto al final, se presentaron muchos problemas en la gestión de los proyectos bajo este enfoque, con un mercado cambiante se solicitaba celeridad en la entrega de los productos, para lo cual en muchas ocasiones se dejaba de lado la etapa de pruebas, el resultado es un producto con muchos errores o fallas, evidenciando una baja calidad, algunas veces esto involucra que se corrija el producto, ocasionando, entregas tardías del mismo, asimismo al no tener muy claro lo que el cliente solicitaba, producto de la incertidumbre en plena etapa de desarrollo del producto se originaban cambios al producto, debido a que se detectaban acciones a corregir y muchos problemas que se presentaban, estos problemas fueron identificados mediante el desarrollo de un árbol de problemas el cual se muestra en el anexo 7.

La problemática mencionada, obliga a tomar acción y propone la implementación del Marco Scrum en la gestión de los proyectos de desarrollo de software, observando cómo influye en los resultados del proyecto, enfocándose específicamente en los problemas mencionados en el párrafo anterior. De acuerdo a lo mencionado planteamos la problemática general: ¿Cómo influye el marco de trabajo Scrum en la gestión de los proyectos de software en una empresa consultora, Lima 2022? Del mismo modo planteamos los problemas específicos como sigue: (i) ¿Cómo influye el marco de trabajo Scrum en la mejora de la calidad del producto de los proyectos de software en una empresa consultora, Lima 2022?; ¿Cómo influye el marco de trabajo Scrum en la disminución del tiempo de entrega

del producto de los proyectos de software en una empresa consultora, Lima 2022?; ¿Cómo influye el marco de trabajo Scrum en la disminución de las solicitudes de cambio al producto de los proyectos de software en una empresa consultora, Lima 2022?

En cuanto a la metodología la investigación se justifica en su contribución en la utilización de un método centrado en los fundamentos de Scrum que sirvan como guía de referencia para una gestión adecuada de los proyectos en una consultora de tecnologías. La justificación práctica se enmarca en que la presente investigación permitió elaborar un procedimiento basado en los fundamentos de Scrum, que permita una gestión adecuada de los proyectos en todas sus etapas, que son ejecutados en la consultora. La justificación temática de la investigación radica en cómo el marco de trabajo Scrum ha tomado mayor notoriedad en estos últimos años, especialmente en el entorno sanitario que aún seguimos atravesando, ya que hoy en día los marcos de trabajo ágiles en general han sido implantados en sectores distintos al de software (como marketing, educación, venta, servicios, y otros) y han aprovechado los beneficios que estos marcos de trabajo ofrecen. La justificación teórica se enfoca en el marco Scrum y cuánto ayuda en la gestión de los proyectos, además de cómo las variables se relacionan sistémicamente en el proceso de gestión de los proyectos, enmarcándose en la teoría de sistemas. En cuanto a la teoría de restricciones radica en que, utilizando el marco Scrum se busca mejorar la productividad, minimizando el tiempo de entrega, los cambios y acrecentando la calidad del entregable. En general las teorías sustentan la mejora de la productividad de la gestión de proyectos al aplicar el marco Scrum.

El objetivo general de esta investigación es: Determinar la influencia del marco de trabajo Scrum en la gestión de los proyectos de software en una empresa consultora, Lima 2022. Como objetivos específicos se plantea: (i) Determinar la influencia del marco de trabajo Scrum en la mejora de la calidad del producto de los proyectos de software en una empresa consultora, Lima 2022; (ii) Determinar la influencia del marco de trabajo Scrum en la disminución del tiempo de entrega del producto de los proyectos de software en una empresa consultora, Lima 2022; (iii) Determinar la influencia del marco de trabajo Scrum en la disminución de las

solicitudes de cambio al producto de los proyectos de software en una empresa consultora, Lima 2002.

De lo mencionado anteriormente planteamos la siguiente hipótesis general: El marco de trabajo Scrum influye positivamente en la gestión de los proyectos de software en una empresa consultora, Lima 2002. Como hipótesis específicas tenemos: (i) El marco de trabajo Scrum mejora significativamente la calidad del producto de los proyectos de software en una empresa consultora, Lima 2022; (ii) El marco de trabajo Scrum disminuye significativamente el tiempo de entrega del producto en los proyectos de software en una empresa consultora, Lima 2022; (iii) El marco de trabajo Scrum disminuye significativamente la cantidad de solicitudes de cambio al producto de los proyectos de software en una empresa consultora, Lima 2022.

II. MARCO TEÓRICO

En los antecedentes nacionales se ha considerado a Reyna (2018), quien en su investigación trazó como objetivo definir la eficacia de la aplicación de Scrum y CMMI en el proceso de elaboración de software en una consultora, para lo cual se analizó los principios de los marcos de trabajo Scrum y CMMI y como aportarían en la gestión de proyectos. La población estuvo definida por los requerimientos y requisitos de los diferentes proyectos de software de la consultora entre los años 2016 a 2017, para la muestra se eligieron tres proyectos y se utilizó la herramienta T-Students para determinar la diferencia entre las muestras de los marcos de trabajo Scrum y CMMI. Se concluye que para la recopilación de los requisitos las desviaciones presentadas sobre lo planificado se encuentran por debajo del 50% de la media, por lo que la aplicación los marcos de trabajo Scrum y CMMI mejoran la tarea de recopilación de los requisitos del producto a fabricar.

Por su parte, Falen (2020), señala como objetivo principal de su investigación emplear la práctica ágil Scrum en la gestión de proyectos de software en el sector privado, al respecto el investigador sostiene que la participación y colaboración recurrente de parte de todo el equipo de trabajo sumado al involucramiento y comunicación con los interesados permitirá optimizar el producto. La investigación es aplicada con un enfoque cualitativo, como instrumentos de medición se utilizaron la observación y entrevista, la población la conformaron los trabajadores de la empresa. La conclusión de la investigación afirma que emplear la práctica ágil Scrum en los proyectos de fabricación de software requiere de documentación, procedimientos y elección de las herramientas idóneas y adaptarlas a la realidad situacional de la organización.

De igual manera, Carbajal (2020), indica la necesidad de seguir un modelo que permita gestión de manera óptima las nuevas propuestas tecnológicas en los hospitales de Lambayeque, debido a los continuos fracasos que presentan los proyectos, ante la falta de buenas prácticas de gestión, la investigación plantea como objetivo proporcionar una guía para gestionar los proyectos tecnológicos, teniendo en cuenta el análisis de los diferentes marcos de trabajo existentes y el entorno situacional donde se aplica la propuesta. En cuanto a la metodología la

investigación tiene un diseño cuasiexperimental, los datos se recogieron empleando el cuestionario y el juicio de expertos para su validez, la muestra la conformaron 4 hospitales. En cuanto a los resultados se obtuvo que el 100% de encuestados no efectúan el monitoreo de los proyectos de manera adecuada, de los cuales el 75% no informa sobre el avance o estado del proyecto, luego de aplicado el modelo de gestión, este obtuvo un nivel alto de confiabilidad de 97%, se concluyó con la entrega de un modelo de gestión efectivo y adecuado al entorno de los hospitales.

En su investigación Guzmán (2019), propuso la implementación de una práctica Ágil, que perfeccione el proceso de dirección para la elaboración de software en un retail de telecomunicaciones, el autor sostiene que producto del entorno cambiante debido a la globalización, las organizaciones tienen la necesidad de adecuarse a estos cambios con la finalidad de entregar sus productos de manera rápida, muchas veces arriesgando la calidad de sus productos, lo cual origina la insatisfacción de los usuarios, así como costos de no conformidad y mala reputación en la organización. De lo mencionado la investigación plantea aplicar el marco de trabajo Scrum a los proyectos y validar cuáles son los resultados de aplicar este marco, la población estuvo conformado por 20 proyectos, la investigación presentó enfoque cuantitativo explicativo, con diseño cuasiexperimental, utilizando la entrevista y cuestionario para la medición. Concluyendo que al implementar Scrum en los proyectos se minimizaron las fallas en un 90%, aumentando la sensación de calidad en los productos, por parte de los usuarios.

Carrasco (2022), plantea como objetivo perfeccionar el proceso de gestión de proyectos de software en una entidad de justicia, para lo cual propone la implementación de un modelo enfocado Scrum. Con respecto a la metodología el estudio es aplicado, cuantitativo y experimental con característica pre-experimental, Para la muestra se consideró a 25 profesionales de la entidad. Los resultados que se obtuvieron indicaron que al utilizar un modelo basado en enfoques tradicionales el promedio del tiempo de entrega de producto es equivalente a 13.83%, mientras que al utilizar el modelo planteado se obtuvo 8.83%, evidenciándose una

disminución del 5%. Concluyendo que al implementar un modelo de gestión planteado se mejora el desarrollo de los proyectos de software.

Por último, Flores (2020), propuso realizar un análisis del marco ágil Scrum y la guía del PMBOK, para determinar cómo estas se relacionan con la eficiencia en el desarrollo de software para las nuevas iniciativas de negocio en empresas del sector. Para lo cual el autor aplica un enfoque cuantitativo no experimental-transversal, el método utilizado es hipotético deductivo, los datos fueron recabados utilizando la encuesta, la población estuvo conformada por 44 empresas del rubro, el modelo de escala utilizado es el de Likert. Como resultados el valor de Spearman obtenido fue de 0,859, indicando una alta correlación, con lo cual se concluye, que la aplicación de ambos marcos de referencia impacta positivamente en la eficiencia en el desarrollo de software.

En los antecedentes internacionales se ha considerado a Hernández y Bravo (2020), quienes indican en su artículo la gran importancia que los marcos de trabajo ágiles han adquirido en estos tiempos, si bien el ámbito de adopción se da en los proyectos de elaboración de software, también se ha dado en otras disciplinas. El artículo propone la integración de los frameworks o marcos de trabajo, entre los cuales utilizó las guías del PMBOK, ágil y Scrum, para la gestión ágil de los proyectos de software. El estudio planteó como objetivo elaborar un procedimiento que sirva de guía para una buena administración de los proyectos de elaboración de software que permita el cumplimiento de sus objetivos. Para lograr los objetivos los autores analizaron la relación entre los postulados de las guías del PMBOK, ágil y relacionarlos con el marco Scrum. Con los resultados del análisis realizado se identifican y establecen los procedimientos para un adecuado desarrollo de proyectos. El artículo concluye indicando que para un desarrollo de proyectos adecuado es necesario que los procesos de gestión sean flexibles y se adapten a los cambios que puedan presentarse, además manifiestan que se requiere de equipos de trabajo multifuncionales y autoorganizados. Asimismo, recomiendan que, en investigaciones futuras se deberá validar y adecuar el procedimiento propuesto de acuerdo al contexto, y realidad específica de la organización.

De igual manera, Díaz (2019), manifiesta en su investigación la importancia de implementar un método ágil para optimizar el procedimiento de solicitud de requerimientos de una empresa del rubro financiero. En cuanto a la metodología, la investigación se presenta como aplicada, utilizando la entrevista como técnica y los cuestionarios para la recopilación de los datos. Con los datos obtenidos se implementan los procesos y procedimientos adecuados en torno a la metodología ágil, enfocándose en el equipo de desarrollo y de sensibilizar a todos los interesados sobre la metodología garantizando entregas anticipadas de productos usables de calidad, generando valor a los clientes o usuarios.

Villavicencio (2017), plantea implementar un modelo enfocado en ITIL y Scrum para controlar los cambios en la ejecución de servicios, debido al elevado número de cambios que se presentan durante el desarrollo de los proyectos. La investigación presenta un enfoque cuantitativo, conformándose la muestra por 19 especialistas de la entidad, utilizando el cuestionario para el levantamiento de datos. Los resultados que se obtuvieron indicaron que, al no utilizar ninguna metodología o procedimiento para controlar los cambios, se presenta en promedio un 78.95% de solicitudes, luego de aplicar el modelo planteado, el promedio de las solicitudes se redujo a 36.84%, evidenciándose que la utilización de un modelo de basado en ITIL y Scrum disminuye significativamente los requerimientos de cambios durante el desarrollo de los proyectos de software.

En su investigación, Merizalde (2018), se enfoca en identificar aquellos factores que perjudican al proceso de fabricación de software y determinar cuál es la influencia del marco Scrum. La variedad de los proyectos analizados, establecerá una guía para las demás empresas. En cuanto a la metodología, el estudio es aplicado, de enfoque cuantitativo; se utilizó como herramienta de análisis el SPSS. La muestra estuvo conformada por 8 empresas dedicadas al rubro del desarrollo de proyectos bajo los enfoques ágiles, los resultados arrojaron porcentajes de 89.50% de éxito en la etapa de planificación del proyecto. Se concluye que el éxito de la aplicación de Scrum, se centra en la planificación, ya que en esta etapa se realizan todas las estimaciones de acuerdo a las historias de usuario identificadas.

Por su parte Forero (2018), quien realiza una investigación sobre la implementación de Scrum en el sector bancario, la cual tuvo como objetivo principal implementar y analizar cómo impacta el marco de trabajo en el equipo de desarrollo de la entidad bancaria, ya que se presentaba serias dificultades en el desempeño del equipo de trabajo durante el desarrollo de los proyectos, lo que originó lentitud al entregar los productos. El alcance de la investigación consistió en implementar el marco Scrum en un proyecto real, previa inducción sobre el marco de trabajo al equipo de trabajo. Asimismo, se desarrolló una hoja de ruta para la implementación analizando variables que permitan identificar las necesidades, así como las herramientas necesarias para una implementación exitosa. La investigación concluye con lo importante que es identificar todos los requisitos y plasmarlos en un Product Backlog bien definido, e igual manera se resaltó la importancia de hacer transparente toda la información sobre el estado del proyecto, permitiendo que los interesados tengan una participación activa, lo que garantiza el éxito del proyecto.

Por su parte la investigación de Rivera (2017), propuso elaborar un método para la administración de proyectos, el objetivo de la investigación fue la mejora del desarrollo de los productos a entregar y optimizar la calidad de los mismos, para lo cual el autor realizó un análisis para determinar por qué los productos desarrollados son de baja calidad. La investigación es aplicada y descriptiva el enfoque utilizado fue de carácter cualitativo, utilizando entrevistas guiadas y encuestas como instrumentos de medición, con lo cual se diseñaron procesos y procedimientos que se implementaron de manera gradual y progresiva. Se concluye que la metodología diseñada y aplicada sirve como guía estandarizada para gestionar la calidad en los proyectos de software.

De acuerdo con la investigación realizada por Sánchez (2019), cuyo objetivo fue implementar marcos ágiles durante el desarrollo de proyectos en el rubro retail, a fin de permitir la resolución de problemas que se presentan cuando se aplican marcos de trabajo tradicionales en entornos de alta incertidumbre. La Investigación finaliza concluyendo que la gestión de proyectos en el sector retail adopta un modelo de gestión de tipo evolutivo, esta conclusión se fundamenta en el estudio empírico de los datos conformado por más de 14.000 solicitudes de información a

más de 100 proyectos, con el fin de verificar si el trabajo en general realizado por la organización parten de definiciones evolutivas. Por lo tanto, se pueden aplicar herramientas de los enfoques ágiles.

De la misma manera en la investigación realizada por Romero (2017), cuyo objetivo fue implementar un modelo que mejore los procesos de desarrollo de productos basado en CRM y Scrum en el sector outsourcing en México, con el propósito de mejorar la calidad del entregable. La investigación es de tipo aplicativo, ya que el autor utiliza el marco de trabajo Scrum y el modelo de gestión de clientes CRM, para crear un modelo que desarrolle la calidad del producto. El autor concluye que luego de aplicar su modelo, la calidad de los productos desarrollados en los outsourcing se incrementa satisfactoriamente en 92%, considerando características internas y externas como la confiabilidad, usabilidad y mantenibilidad.

Las entidades públicas no son ajenas a la adopción de los marcos de trabajo ágiles. De acuerdo a la investigación de Rivera (2021) que, tuvo por objetivo analizar la adopción de los marcos de trabajo ágiles en el sector público en Argentina, la investigación señala cómo se han adoptado y adaptado los enfoques, así como cuáles han sido las herramientas más utilizadas, partiendo de la recopilación de casos de estudio internacionales donde el enfoque ágil ha sido aplicado en diferentes entornos, así como de la elaboración y realización de entrevistas a los funcionarios, los cuales permitieron conocer la percepción de los marcos de trabajo ágil, el nivel de su aplicación y en qué grado aportan valor. La investigación finaliza proporcionando propuestas que promoverán la utilización de las herramientas ágiles. Se concluye que los principios y prácticas de los marcos de trabajo ágiles se convertirán en el eje potenciador de las capacidades a nivel institucional, permitiendo agilizar los procesos.

Respecto a los sustentos teóricos de la investigación, Cruz et al. (2020) manifestaron que, todas las organizaciones para lograr cumplir con sus objetivos y metas requieren hacer uso de procesos eficientes que permitan ejecutar todas las actividades de manera ordenada, con los recursos necesarios que permitan obtener la calidad establecida en los procesos, es así que se establecen modelos de gestión

aplicables a estas actividades realizadas por un grupo de recursos, lo que se conoce como proyectos, la gestión tradicional de los proyectos establece procesos definidos que se ejecutan de manera secuencial, en la cual en un primer proceso de inicio se establece el alcance de lo que se quiere de manera detallada, así como los plazos para la entrega de lo que se quiere y los costos involucrados, a esto se le conoce como enfoque tradicional o predictivo o cascada.

El enfoque tradicional funcionó por mucho tiempo, pero las condiciones que hoy se presentan en cuanto al mercado y la tecnología obligan a la aparición de nuevos modelos de gestión, de los cuales ya se hablaba tiempo atrás. De acuerdo a Takeuchi y Nonaka (1986), desde hace mucho ya se hablaba del crecimiento de la competencia de las organizaciones en los mercados, haciéndose hincapié en que ya no era suficiente que las organizaciones entreguen productos de calidad y a bajo costo, sino que el diferenciador era ser flexibles y veloces en el mercado, pero que esto no sería posible si se seguía con la adopción de metodologías tradicionales (predictivos o secuenciales), que hasta ese entonces gobernaban los proyectos. Se requería de una metodología holística, que englobe a toda la organización como una unidad de trabajo altamente competitiva, es así que ya se daban indicios sobre las metodologías ágiles, comenzando a utilizarse terminología como equipos auto organizados y entrega de productos de manera incremental.

Pero recién en el año 2001, se realizó una cumbre formada por 17 profesionales dedicados al desarrollo de software, la finalidad de la cumbre fue identificar métodos en común utilizados hasta aquel entonces y crear una guía que permitiera agilizar el proceso del desarrollo de software de acuerdo a las dificultades que se presentaba en el entorno. Al culminar la reunión se llegó a un consenso que se plasmó en 12 principios, al cual denominaron Manifiesto Ágil, los cuales sientan las bases de las metodologías ágiles (Agile Alliance, 2001).

Entre algunas definiciones de los métodos ágiles, podemos citar a Amaro y Valverde (2007), quienes conceptualizaron a las metodologías ágiles como aquellas caracterizadas por su sencillez, permitiendo a pequeños equipos de

desarrollo mentalizarse en fabricar software con prácticas que permitan una rápido y fácil entendimiento, en un entorno controlado que garantice el éxito del proyecto.

Mientras que Moran (2015), definió a los métodos ágiles como una doctrina flexible y adaptable a los cambios, de manera rápida, haciendo uso del aprendizaje y retroalimentación constante, creando y entregando valor de modo incremental. Complementado lo mencionado por el autor, es cierto que los métodos ágiles tienen como uno de sus principios la adaptación a los cambios debido a no tener claro el producto, pero los métodos ágiles van mucho más allá de los principios y herramientas que establece, la esencia radica en la manera correcta de hacer las cosas, en tener claro una actitud y vocación de trabajo, así como colaborar con los demás.

Por último, Castaño y Cadavid (2017) indican que, los métodos ágiles son una serie de recomendaciones, que de ser seguidas o practicadas permiten crear un producto de calidad. Asimismo, definieron una serie de metodologías ágiles, las cuales presentaban diferentes características, haciendo énfasis en elegir y adoptar la que mejor se adapte a las necesidades y entorno del negocio.

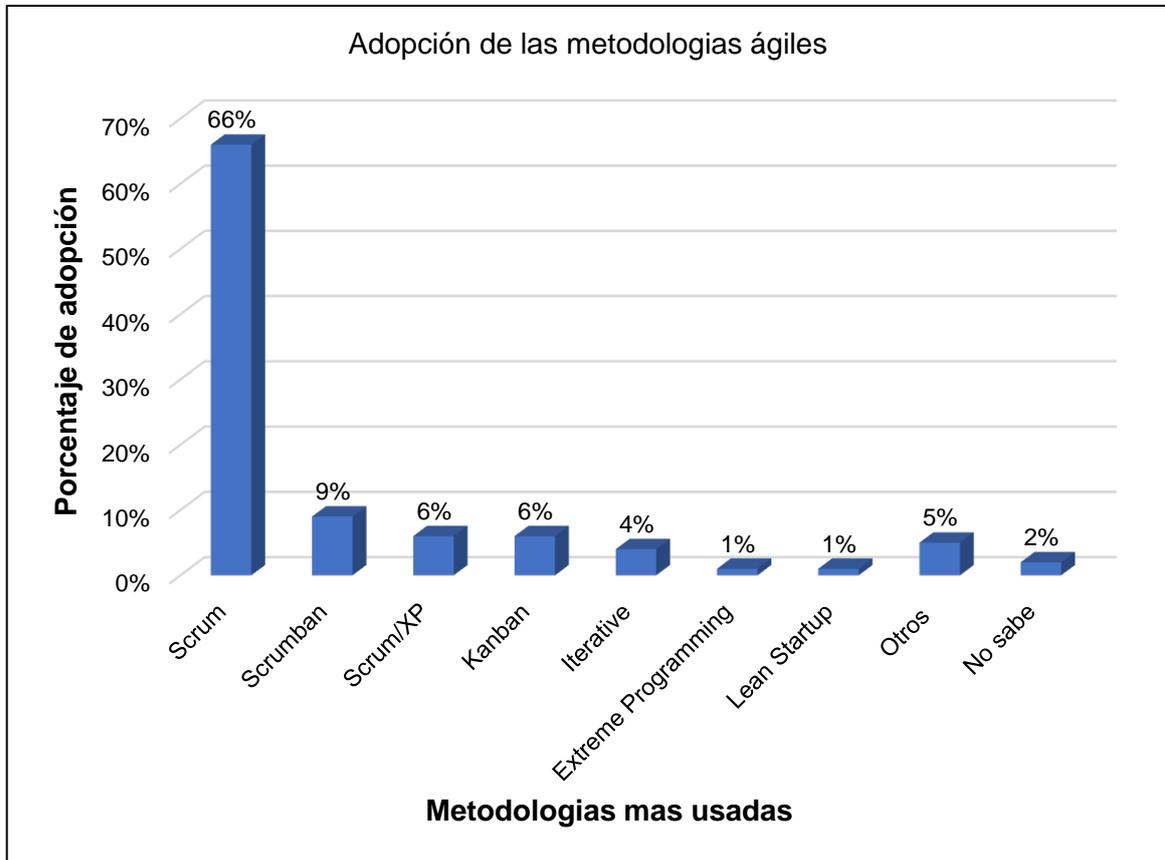
Hoy en día la globalización y el cambio tecnológico acelerado, ha hecho que las organizaciones adopten metodologías ágiles, algunas empresas han nacido ágiles. Según Cappelli y Tavis (2018), las metodologías ágiles, no es usado únicamente para el desarrollo de software, sino que ha sido adoptado por otros mercados como el marketing, retail, y otros. Efectivamente en un inicio los métodos ágiles nacieron por la necesidad de buscar nuevas maneras de desarrollar software, pero esto ha cambiado producto de cómo están cambiando los mercados y de cómo las empresas deben adaptarse a estos cambios, para lo cual ya no debe de planificarse por adelantado, sino planificar mientras se define el proyecto en cualquier rubro.

Dentro del grupo de metodologías ágiles, el último reporte de agilidad “15th Annual State of Agile” publicado por Digital.ai (2021) indica que, el enfoque ágil más utilizado es Scrum, seguido por Scrumban, como se observa en la figura 1, se debe

indicar que el reporte se basó en una encuesta realizada a la comunidad de desarrollo de software.

Figura 1

Décimo Reporte de Agilidad



Nota: Digital.ai

Satpathy (2017) señala que, Scrum es una práctica ágil, que realiza entregas de manera incremental y periódica, fomentando buenas prácticas para incentivar un trabajo colaborativo de los equipos, lo que contribuye a las soluciones de los posibles inconvenientes que puedan presentarse durante la etapa de desarrollo del proyecto. Asimismo, el autor manifiesta que Scrum es el framework más adoptado en la actualidad, debido a su adaptabilidad, flexibilidad y eficacia, siendo creado para entregar valor de manera rápida a lo largo del desarrollo del proyecto.

Respecto a Scrum, establece ciclos de trabajo cortos llamados Sprints, en cada uno de estos se realizan entregas pequeñas del producto o parte del producto

que, con la adecuada retroalimentación por parte de los usuarios se van agregando funcionalidades. En cada Sprint el equipo de desarrollo elige cómo se desarrollarán las actividades, de tal manera que pueda entregar valor al término de cada Sprint (Marcal et al., 2007).

Scrum cuenta con tres roles: el Scrum Master, Product Owner y el Team Developer o equipo encargado del desarrollo. El Scrum Master es el responsable de orientar, guiar e inculcar principios, valores y prácticas de Scrum en toda la organización, es más un rol de coach y debe asegurar que el equipo disponga de lo necesario para el éxito del proyecto. El Product Owner, es el responsable de reunir los requerimientos del cliente, definir cuáles son prioritarios, así como definir las condiciones finales del producto. Por último, el Team Developer, que es el equipo de trabajo autoorganizado y multifuncional, encargado del desarrollo del producto (Woodward et al., 2010).

Srivastava et al. (2017) manifiestan que, el proceso Scrum radica en el desarrollo del equipo, para lo cual el Scrum master y el Product Owner, deben de involucrarse y trabajar con el equipo, para elevar su moral y productividad durante todas las iteraciones que involucra el desarrollo del producto. Esto se logra manteniendo una comunicación constante y fluida tomando en cuenta las opiniones para el logro de acuerdos.

Por su parte, James y Walter (2017) señalan que, los artefactos de Scrum son elementos de información transparente a todos los interesados, es decir todos deben de conocer el avance del desarrollo del producto, los tres artefactos o herramientas de Scrum son: Sprint Backlog o pila de producto, que es una relación de los requerimientos, tareas y son la primera fuente de información del producto a entregar. El Sprint Backlog, es una lista extraída del Product Backlog y que se tiene que fabricar en la próxima iteración. El incremento, que es lo que se obtiene al final de cada sprint.

Morandini (2021), define al Sprint Backlog como a la lista de requerimientos provenientes del Product Backlog, esta lista de requerimientos será el trabajo a

realizar durante el Sprint, el Team Developer realizará las actividades necesarias para convertir los requerimientos del Product Backlog en aumentos. Al final de cada sprint, se genera un incremento, que es la suma de todos los resultados de las actividades desarrolladas en el sprint. Buffardi (2020) indica que, las historias de usuario son descripciones generales de cada requerimiento del software a desarrollar y son narradas desde la perspectiva del usuario.

Beedle et al. (1998) sostienen que, Scrum prescribe cuatro eventos: sprint planning, que es un evento que se da al inicio del sprint, aquí se decide que se va a hacer en el sprint y como se hará. El daily Scrum, que es un evento que se efectúa de manera diaria, aquí solo está involucrado el development team, en esta reunión se da a conocer lo que se realizó el día anterior y lo que se va a realizar, así como los impedimentos que retrasan el trabajo. El sprint review, es un evento que se lleva a cabo al finalizar un sprint, donde muestra el incremento por parte del equipo. El último evento es el sprint retrospective cuyo objetivo es la autoevaluación del equipo e identificar situaciones para su mejora y aplicarlas en el siguiente sprint.

Respecto a la gestión de proyectos Moura et al. (2018) manifiestan que es una disciplina que propone el uso de habilidades, herramientas y conocimiento, en todas las actividades ejecutadas en el proyecto, necesarias para el logro del éxito. Complementando lo indicado por el autor, la gestión de proyectos es un componente estratégico dentro de la organización y que como disciplina también involucra liderazgo, capacidades resolutivas, conocimientos, así como tecnologías.

Mientras que Gómez et al. (2012), definen a la dirección de proyectos como el monitoreo y seguimiento a la ejecución de las actividades, las cuales son realizadas por los recursos necesarios en el tiempo planificado. Todo esto sujeto a limitaciones que el entorno sobre el cual se desarrolla el proyecto cambie.

De igual manera Ramos et al. (2017) señalan que, el desarrollo de software engloba a todas las actividades necesarias para la fabricación del producto, las actividades se distribuyen en etapas, las cuales de manera general son: identificación y especificación de requerimientos, desarrollo o construcción del

producto, y pruebas o validación. Estas etapas deben ser transparentes y flexibles para una adecuada gestión del proyecto.

Según Saeedi y Visvizi (2021), el modelo para la construcción de software más utilizado es el “Ciclo de vida del desarrollo de software”, que es una serie de pasos estructurados para construir el software deseado. Además, indican que este ciclo de vida tiene 5 etapas: la planificación, aquí se analizan los riesgos y se determina si el proyecto es viable, se estiman los costos y recursos; análisis, aquí se identifican y recopilan todos los requisitos que el producto debe tener; diseño, aquí se estudian las maneras posibles de fabricación del producto; Fabricación, etapa de construcción del software de acuerdo al diseño; prueba, se verifica lo fabricado con la finalidad de encontrar errores y que se realicen todos los requisitos cumplan los requisitos planteados en la etapa de análisis.

Los proyectos de software con el tiempo se han caracterizado por tener una valoración alta de fracaso. La firma Internacional Standish Group que es reconocida por entregar reportes sobre la implementación de los proyectos de sistemas de información ha realizado encuestas referentes a los logros de los proyectos tecnológicos en general. Las valoraciones iniciales respecto al éxito de los proyectos de TI indicaron que menos del 20% de los proyectos tienen éxito, pero con el transcurrir del tiempo, esto ha cambiado. En su informe del 2018, la valoración del éxito de los proyectos es del 31%, y según indica el informe, este aumento del éxito de los proyectos se debe a que muchos de los proyectos han adoptado métodos ágiles para su desarrollo. Además, se tomaron en cuenta indicadores como la entrega de valor al cliente, la satisfacción del mismo y la alineación del proyecto con los objetivos estratégicos de la organización. Asimismo, el informe indica que hay muchos factores que causan el fracaso de los proyectos de software como: Requerimientos nada claros o no identificados, Poco involucramiento del usuario originado por una pobre comunicación entre todos los involucrados. Asimismo, otros factores como la deficiencia de recursos, expectativas de los involucrados poco realistas, cambios en los requerimientos y especificaciones, entre los más importantes. Estos factores de fracaso en los

proyectos afectan a la Calidad de producto, al tiempo de entrega del producto ocasionando la insatisfacción de cliente o usuario.

En lo que respecta a la calidad, hay diferentes puntos de vista, ya que la calidad puede tener diferente apreciación. La ISO/IEC (1999b) delimita a la calidad como sigue: "el grado en el cual un conjunto de características inherentes cumple los requisitos". Según Highsmith (2002), la calidad se aprecia desde el lado del cliente y de cómo este recibe valor. Otra definición de calidad es "la destreza de terminar los productos cumpliendo con las condiciones iniciales, logrando la entrega del valor que el cliente espera" (Satpathy 2017).

Según Azeem et al. (2017), sostienen en su artículo que la calidad es lo más importante en el proceso de desarrollar software y esto puede derivar al fracaso o éxito del proyecto. El desarrollo de software a través de modelos de ciclo de vida permite mejorar los tiempos de desarrollo y entrega, mejorar la calidad del software. En general mejorar el proceso.

Por otro lado, Akingbehin (2005), propone una definición alterna al concepto de calidad de software, basándose en Taguchi, definiendo a la calidad como la pérdida entregada y percibida por la sociedad luego de recibir el producto. Al respecto podemos mencionar que en la gestión de proyectos la calidad puede tener diferentes percepciones, por ejemplo, la calidad para el equipo que desarrolla el producto es distinta a la calidad percibida desde el lado del cliente. Asimismo, la calidad también se puede definir de acuerdo al valor que este genera.

En lo que respecta al tiempo de entrega, este es importante en todo proyecto de desarrollo de software, determinar o estimar el tiempo que vamos a tardar en entregar todo el producto, lo que se ha desarrollado hasta cierta fecha o lo que falta por desarrollar son parámetros que intervienen en la medición de la calidad de un producto (Martín, 1991).

Pressman (2010), manifiesta la importancia de una entrega temprana de software al cliente, el tiempo que demora en entregarse el producto es relevante

para ellos, por lo que, al demorar la entrega, la necesidad que tiene el cliente por el producto podría cambiar. Además, una entrega temprana y continua al cliente permitirá interactuar con el cliente y conocer su apreciación del software entregado e identificar puntos de mejora, se requiere su participación activa con los equipos motivados (Beck et al., 2001).

El PMI (2017) manifiesta que, una solicitud de cambio es un requerimiento que formaliza la modificación de los entregables o de la documentación del proyecto. Cuando se ejecuta todo el trabajo necesario para lograr los entregables, se suelen presentar inconvenientes que requieren ser corregidos, asimismo por eventos ajenos al proyecto es necesario modificar algún procedimiento en cuanto al alcance presupuesto y cronograma, para ambos casos es necesario solicitar de manera formal los cambios. Estas solicitudes suelen involucrar la identificación de nuevos requerimientos, los cuales afectan lo planificado en las etapas iniciales, así como al entregable mismo del proyecto.

III. METODOLOGÍA

3.1. Tipo y diseño de investigación

Tipo de investigación: Gabriel (2017) señala que, las investigaciones aplicadas, también denominadas prácticas o empíricas se enfocan en la búsqueda del conocimiento y lo aplica para resolver un problema o un planteamiento. Asimismo, Ñaupas et al. (2014) afirma que, las investigaciones aplicadas se fundamentan en una teoría básica y de acuerdo a esta teoría se plantean objetivos e hipótesis para solucionar una problemática, concluyendo que este tipo de investigación nace ante la necesidad de perfeccionar un proceso, procedimiento o el funcionamiento de una solución en particular. Por lo expuesto esta investigación es aplicada, debido a que las prácticas del marco de trabajo Scrum son aplicados al desarrollo de los proyectos de software, en busca de mejorar la calidad del producto desarrollado, así como optimizar el tiempo de entrega y reducir las solicitudes de cambio.

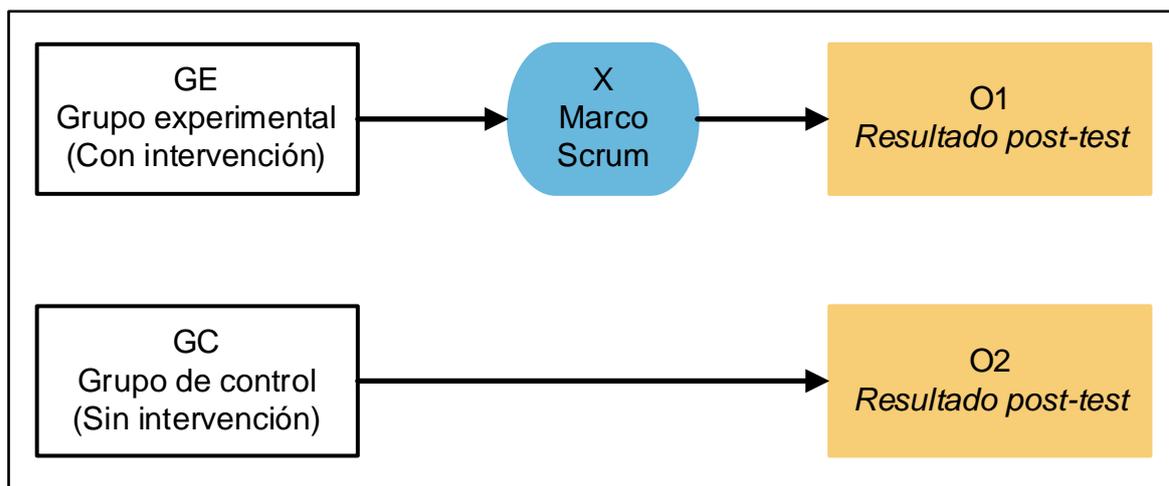
Enfoque de la investigación: Galeano (2020) afirma que, una investigación con enfoque cuantitativo se sostiene sobre una recopilación de datos estrictos y confiables, además las investigaciones cuantitativas se basan en procedimientos estadísticos que permiten demostrar su validez de los mismos. Por su parte Atmowardoyo (2018) indica que, las investigaciones con enfoque cuantitativo ponderan los datos numéricos mediante análisis estadístico ya que son fáciles de medir y cuantificar. De lo mencionado la presente investigación es de enfoque cuantitativo, pues se recopilan datos cuantificables mediante instrumentos como fichas de observación, luego son analizados, para las pruebas de hipótesis.

Diseño de la investigación: Ñaupas (2014) afirma que, el diseño experimental tiene como propósito guiar de manera correcta al investigador a realizar el experimento. Asimismo, el autor señala que los métodos experimentales se dividen en preexperimentales, cuasiexperimentales y experimentales puros, de acuerdo al grado o acercamiento al experimento puro. Para White y Saburral (2014), los diseños cuasiexperimentales comprueban el efecto de una variable sobre la otra. Asimismo, el diseño trabaja con un grupo de control o comparación y otro de tratamiento al cual se le aplica el experimento, para luego observar y comparar los resultados para la validación de las hipótesis. Asimismo, Hernández et al. (2018)

manifiesta que, un tipo de diseño cuasiexperimental es aquel con un solo post-test y grupos intactos, en el cual participan dos grupos, uno el que recibe el experimento y el otro no, los cuales son comparados al realizar el post-test y analizar qué efecto tuvo el experimento en la variable independiente. Por lo mencionado en el párrafo anterior esta investigación es cuasiexperimental solo con post-test y grupos intactos, pues la muestra es no probabilística (no aleatorio), usando dos grupos uno de control y un grupo de tratamiento, a los cuales se les aplicarán las fichas de observaciones, al grupo de control sin la intervención del experimento y al grupo de tratamiento con la intervención del experimento, con los resultados que se obtienen se validan las hipótesis planteadas (figura 2).

Figura 2

Estudio cuasiexperimental



Dónde:

GC = Grupo de control (proyectos ejecutados con enfoque tradicional)

GE = Grupo de experimental (proyectos ejecutados con enfoque Scrum)

X = Experimento (marco de trabajo Scrum)

O₁ y O₂ = Post-test

Para Andia (2017), las investigaciones transversales son de tipo observacional, porque los datos son recopilados en un instante de tiempo y luego son analizados sobre la población o muestra definida. Mientras que Arbaiza (2019)

indica que, las investigaciones transversales o también denominadas transeccionales se basan en la obtención de datos de un determinado suceso en un momento dado. Esta investigación es de corte transversal, fundamentándose en que los datos recogidos mediante los instrumentos se realizan en un periodo de tiempo determinado.

3.2. Variables y operacionalización

Según Palomino et al. (2015), la variable es la propiedad o cualidad de un objeto de estudio y esta propiedad puede cambiar entre objetos. Este cambio puede tomar múltiples valores, los cuales pueden ser observados y medidos. Para García (2016), una variable es una propiedad que sirve de apoyo para que el investigador pueda probar su hipótesis. Para complementar lo mencionado, Webster (2000) menciona que, una variable es todo aquello que puede ser medido y que forma parte de la muestra. En la tabla 1 se muestran las variables definidas.

Tabla 1

Definición de variables

Variable	Tipo	Perspectiva
Marco de trabajo Scrum	Independiente	Cuantitativo
Gestión de proyectos de software	Dependiente	Cuantitativo

De acuerdo a Gamarra et al. (2015), clasifica a las variables de acuerdo al papel que desempeñan en una investigación, encontrando variables dependientes, variables independientes, variables intervinientes y variables extrañas.

Para definir las variables, citamos a Arias y Covinos (2021), quienes definen a una variable independiente como a la variable que se aplica tal cual o manipulada, y observar las consecuencias o cambios que se presentan en la variable dependiente. Por su parte la variable dependiente está sujeta o es susceptible a cambio al intervenir la variable independiente.

De acuerdo con Icart y Pulpón (2012), la definición conceptual de una variable implica describir sus funciones, particularidades y comportamientos. Mientras que para Córdova (2018), la definición conceptual es determinar el significado de la variable utilizando libros, como diccionarios, por lo que una variable puede tener múltiples conceptos.

De igual manera para Icart y Pulpón (2012), la definición operativa de una variable comprende dedicar procedimientos y actividades necesarias para poder medirlas. Mientras que para Moran y Alvarado (2010), primero se debe identificar aquella variable más precisa y amplia dentro de su contexto, a fin de que pueda proporcionarnos mayor información.

La operacionalización de la variable según Ríos (2017), nos ayuda a identificar a las variables involucradas dentro de una investigación de manera precisa y comprensible. Asimismo, las conceptualiza, dimensiona y establece los indicadores. Por su parte Cabezas et al. (2018) señala que, operacionalizar una variable consiste en establecer la forma en que será observada y medida en el ámbito del estudio de la investigación, este procedimiento es netamente práctico, por lo que no solo identifica a las variables y a sus dimensiones, sino que debe de definir las al detalle.

Variable independiente: Marco de Trabajo Scrum.

Definición conceptual: Scrum es un proceso iterativo e incremental que contiene un conjunto de buenas prácticas o recomendaciones, para trabajar de manera colaborativa y lograr el mejor resultado de los proyectos, Scrum nos orienta mas no define las actividades que se tiene que hacer para lograr el éxito (Gaborov et al., 2021).

Con respecto al marco Scrum adoptado por la empresa consultora, se desarrolló una metodología basada en el marco, a partir del cual se desarrollaron los proyectos, la metodología se detalla en el anexo 8.

Variable dependiente: Gestión de Proyectos de Software.

Definición conceptual: Es el proceso que permite desarrollar software, mediante una serie de actividades, modelos, normas y políticas que deben de ser cumplidos por un equipo de trabajo y generar un producto (Ruvalcaba, 2020).

Definición Operacional: con respecto a la gestión de proyectos de software será medida a través de fichas de registro. Las dimensiones consideradas para la variable dependiente son: Calidad del producto; tiempo de entrega del producto, y solicitudes de cambio al producto.

3.3. Población, muestra, muestreo, unidad de análisis

Población.

De acuerdo a Vara (2015), la población está formada por un conjunto de elementos (personas, objetos y otros), que tienen cualidades similares entre sí y que son observables. Por otro lado, Spiegel y Stephens (2009) sostienen que, hay dos tipos de poblaciones las finitas e infinitas. Las infinitas son elementos demasiado grandes, en cuyo caso no se accederá a toda la información, mientras que en las finitas el número de objetos es mucho menor, siendo muy accesible. La población para esta investigación es finita y se han considerado 50 proyectos de software que se han ejecutado en una empresa consultora.

Criterio de inclusión: Para la elección de los proyectos se han incluido a aquellos cuyas características son idóneas para ser gestionados con el marco de trabajo Scrum, para evaluar esta condición se utilizó una lista de verificación que consta de 8 preguntas, las cuales se muestran en la tabla 2, la lista fue aplicada a los profesionales que gestionaron los proyectos en la empresa, esta lista de verificación es de elaboración propia y se aplica como complemento del análisis estadístico, la escala de calificación de la lista de verificación de muestra en la tabla 3. Los proyectos seleccionados fueron aquellos que obtuvieron una puntuación mayor o igual a 6 puntos.

Tabla 2*Lista de verificación*

Nro.	Pregunta	Cumple
1	¿El producto presento una incertidumbre al inicio?	SI/NO
2	¿El producto requiere entregables en periodos cortos?	SI/NO
3	¿El producto puede ser dividido en pequeños entregables?	SI/NO
4	¿El producto puede ser desarrollado por un equipo dedicado?	SI/NO
5	¿El equipo que desarrolla el producto es experimentado?	SI/NO
6	¿El producto es innovador?	SI/NO
7	¿El producto se desarrolla bajo un contrato del tipo tiempo y materiales?	SI/NO
8	¿El desarrollo del producto es complejo?	SI/NO

Tabla 3*Opciones para la lista de verificación*

Nro.	Cumple	Peso
1	SI	1
2	NO	0

Muestra

Según Amiel (2014), la muestra es una parte del total de la población, los cuales tienen particularidades similares, de los cuales se recolectan datos. Asimismo, para Hayes (1999), existen tres formas de muestreo, uno de ellos es el muestreo censal, en el cual la muestra es toda la población, este tipo de muestra se usa cuando se tiene facilidad para acceder a la base de datos.

La investigación utiliza una muestra de tipo censal no probabilística (fueron seleccionados mediante el criterio de inclusión) conformada por 50 proyectos ejecutados en la empresa consultora, la muestra está conformada por dos muestras independientes un primer grupo conformado por 25 proyectos ejecutados con enfoque predictivo los cuales formarán parte del grupo de control y otro grupo formado también por 25 proyectos ejecutados con el enfoque Scrum, los cuales formarán parte del grupo de tratamiento y que estarán sujetos al experimento.

Muestreo

Toda la población participará del estudio de la investigación, por lo que el muestreo no se aplicará.

3.4. Técnicas e instrumentos de recolección de datos

Técnicas e instrumentos.

Hernández et al. (2017), define a una técnica como una cadena de actividades utilizadas para recopilar datos y lograr obtener información de cierto fenómeno dado. En nuestra investigación utilizamos la observación como técnica y la ficha de registro (anexo 2), para recolectar los datos, luego ser analizados y permitir validar las tres hipótesis planteadas, las fichas de registro se aplicarán a toda la muestra conformada por los proyectos de software ejecutados en la empresa consultora, cuya documentación se encuentra almacenada en un repositorio local, en la cual se encuentran toda la información de cada proceso ejecutado en los proyectos, de los cuales se obtendrán los datos de cantidad de fallas de los productos, tiempo de entrega de los productos y cantidad de solicitudes de cambio de los proyectos. Esta base de datos documental de proyectos es propiedad de la empresa consultora.

Validez

Según Useche et al. (2019), define la validez como el nivel de medición que tiene el instrumento con respecto al objetivo planteado en la investigación. En nuestra investigación para la validez del instrumento (anexo 2) se toma en cuenta el juicio de expertos quienes validan el instrumento y la información estructurada del mismo, obteniendo la aplicabilidad del instrumento (véase tabla 4).

Tabla 4

Validación del instrumento

Instrumento	Apellidos y nombres	Institución	Validez
Ficha de registro 1			
Ficha de registro 2	Dr. Acuña Benites Marlon Frank	UCV	Aplica
Ficha de registro 3			

3.5. Procedimientos

La recolección de los datos se realizó aplicando los instrumentos (Fichas de registro) a la muestra censal. Las fichas de registro contienen los indicadores diseñados y validados mediante el juicio de expertos, la aplicación de las fichas se realizó en un único post-test, al grupo de control (muestra formada por los proyectos ejecutados con enfoque predictivo) sin intervención del experimento y al grupo de tratamiento (muestra formada por los proyectos ejecutados con enfoque Scrum) sujeto al experimento. Los datos se obtuvieron accediendo a la base de información de los proyectos, propiedad de la empresa consultora. Para ambas muestras se utilizó como criterio de inclusión aquellos proyectos cuya característica es aplicable el marco Scrum para su gestión, para lo cual se aplicó una lista de verificación a cada proyecto. Con los datos obtenidos de cada grupo en el post-test se realizará el análisis estadístico, para luego evaluar los resultados que el grupo de control obtuvo y validar la mejora del proceso.

Los procedimientos descritos se aplicaron a un conjunto de proyectos ejecutados en la empresa consultora, quien autorizó la aplicación de los instrumentos a sus proyectos, de acuerdo a la carta de aceptación (anexo 6).

3.6. Método de análisis de datos

Como se indicó la obtención de los datos se llevará a cabo a través de las Fichas de Registro, estas se procesarán en hojas de cálculo, creando nuestra base de datos para su análisis descriptivo e inferencial utilizando el software IBM SPSS v27.

El análisis descriptivo se realizó calculando los parámetros estadísticos (media, mediana, varianza, entre otros.) para cada uno de los indicadores, para luego realizar la comparación de los promedios (medias), para ambos grupos (control y experimental) y determinar la influencia de una variable sobre la otra para cada uno de los indicadores propuestos.

En el análisis inferencial primero se determinó la normalidad de los datos recopilados, la normalidad se obtiene aplicando los estadísticos de Shapiro Wilk o Kolmogórov-Smirnov a los datos recopilados a través las fichas de registro aplicadas al objeto de estudio, en un único post-test, al grupo de control sin estar sujeto al experimento y al grupo de tratamiento sujeto al experimento, para la investigación se utilizó el estadístico de Kolmogórov-Smirnov ya que, esta se aplica a muestras mayores o iguales a 30 objetos (muestra conformada por 50 proyectos). El parámetro a determinar es la significancia (Sig.) del estadístico de Kolmogórov-Smirnov, para lo cual hacemos uso del software SPSS. La significancia nos indicó si hay normalidad o no normalidad en los datos obtenidos para cada indicador, esto nos permitió determinar qué tipo de prueba estadística se utilizará para validar las hipótesis, si se presenta normalidad utilizaremos la distribución paramétrica de T-Student de muestras independientes y si se presenta no normalidad se utilizará la prueba no paramétrica de U-Mann Whitney de muestras independientes, comprobando el rechazo de las hipótesis nula y la aceptación de las hipótesis de investigación.

3.7. Aspectos éticos

La presente investigación es de mi autoría y cumple la normativa plasmada en la Resolución del Consejo Universitario N° 00126-2017-UCV, que en su Art. 15 aborda las políticas anti plagio de toda investigación, en este sentido la investigación será sometido al análisis mediante el software Turniting, para probar la originalidad del mismo, en cuanto al contenido como el marco teórico, la metodología y las referencias bibliográficas se desarrollaron en base a lo indicado en la “Guía de Elaboración de Productos de Investigación de Fin de Programa” emitida con resolución del Vicerrectorado de Investigación N° 020-2020-VI/UCV, Asimismo se han tomado en consideración los códigos éticos, referentes a responsabilidad, integridad, honestidad, así como el respeto de la propiedad intelectual y a las leyes. El presente estudio no implica daños físicos ni psicológicos a ningún involucrado. Asimismo, la información se obtuvo con la autorización verbal de los involucrados, por lo tanto, serán tratados con absoluta confidencialidad. Las actividades de

recolección de datos se realizaron previa solicitud y aprobación del coordinador del área de tecnologías de información.

IV. RESULTADOS

Análisis descriptivo

Indicador 1: Cantidad de fallas de sistema

Tabla 5

Análisis descriptivo para cantidad de fallas del sistema

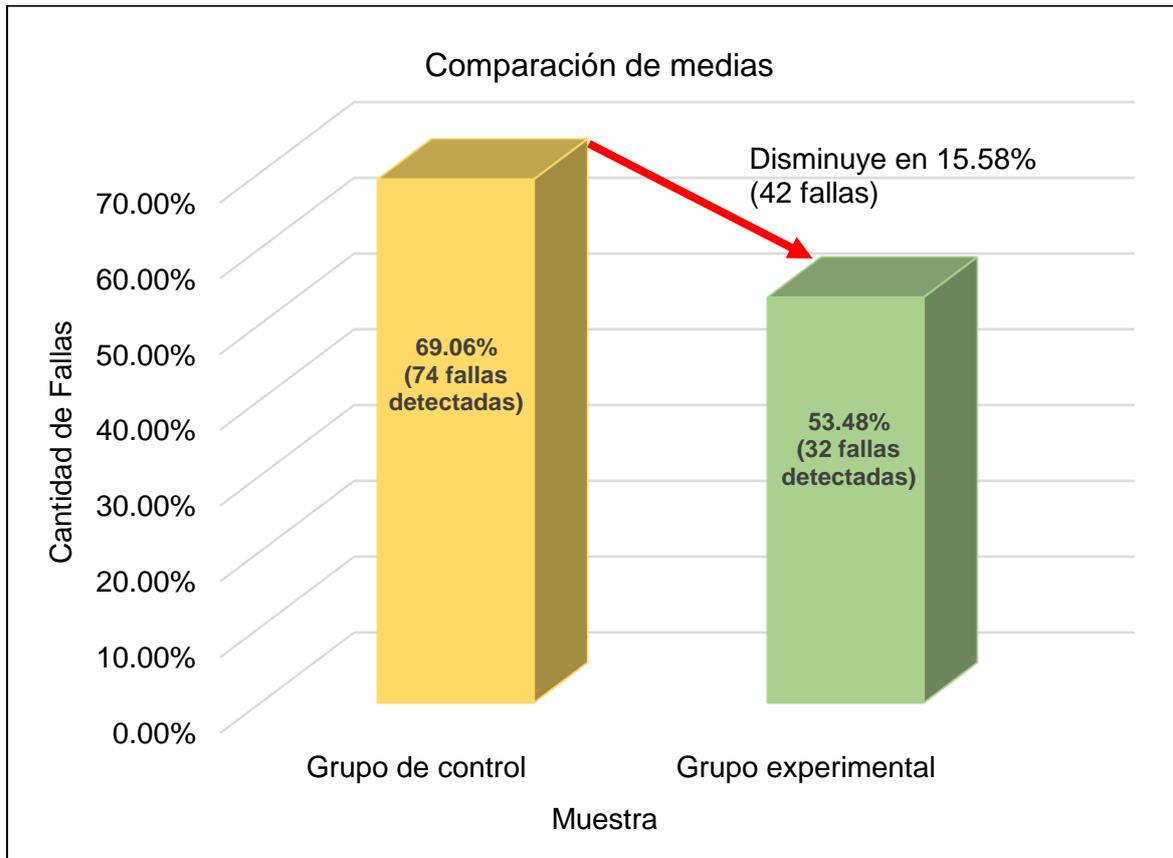
	Muestra	Estadístico	
Cantidad_de_fallas	Control	Media	69,0592
		Mediana	68,9200
		Varianza	13,321
		Desviación estándar	3,64984
		Mínimo	62,50
		Máximo	75,38
	Experimento	Media	53,4756
		Mediana	54,5500
		Varianza	10,021
		Desviación estándar	3,16561
		Mínimo	50,00
		Máximo	58,33

Nota: Se obtuvo con el software SPSS v27.

Los datos obtenidos en la tabla 5, para el indicador cantidad de fallas del sistema, nos indicó que el promedio para la muestra conformada por el grupo de control (proyectos ejecutados bajo un enfoque predictivo) es de 69.05%, mientras que el promedio para la muestra conformada por el grupo experimental (proyectos ejecutados bajo el enfoque Scrum) fue de 53.47%, demostrándose una reducción del indicador cantidad de fallas del sistema en el grupo sometido al experimento, con lo cual se evidenció que la adopción del marco Scrum en los proyectos de software disminuye la cantidad de fallas en los sistemas.

Figura 3

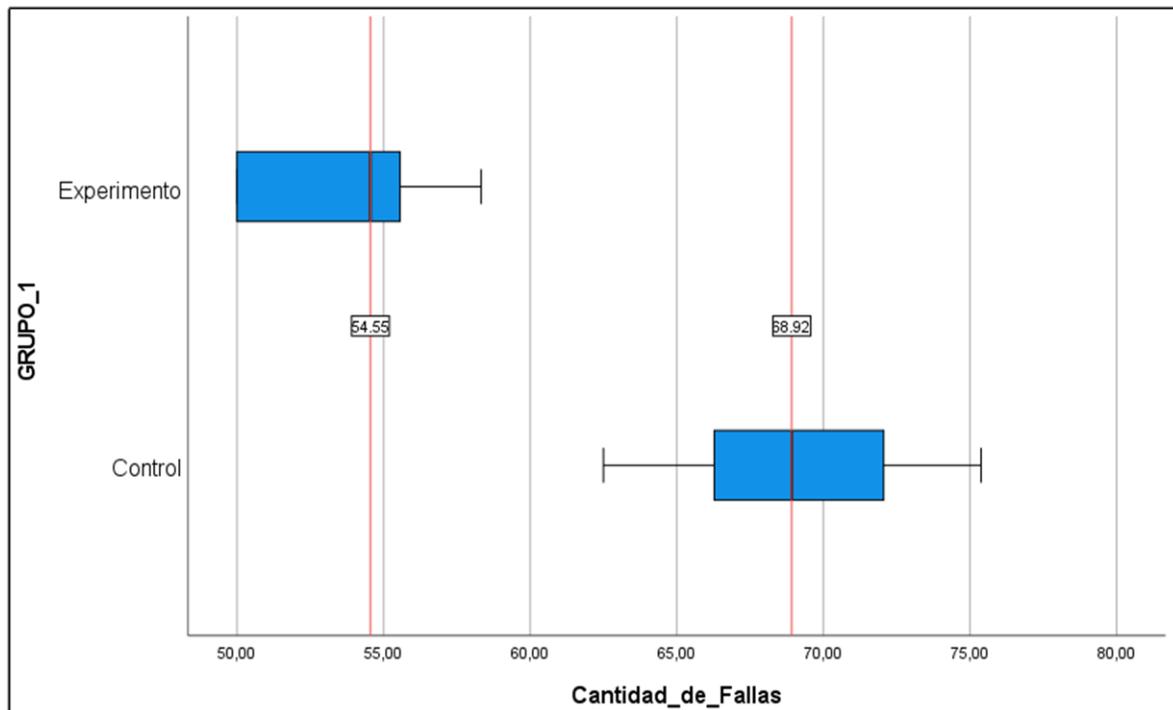
Comparación de las medias para cantidad de fallas del sistema



De igual manera, la figura 3 nos muestra como varía la media para el indicador cantidad de fallas de sistema para ambos grupos de muestra (control y experimental), observándose que los proyectos implementados con el enfoque tradicional presentaron fallas en los sistemas parte del producto, el promedio de estas fallas fue de 74 reportados por proyecto, las fallas más comunes eran errores de código entre otros. Mientras que, los proyectos implementados bajo el enfoque ágil Scrum el promedio de fallas presentados en los sistemas fue de 32 reportados por proyecto. Evidenciándose una disminución de 42 fallas detectadas en los sistemas, lo cual representa una reducción del 15.58%, observándose la deficiencia en el proceso de implementación de los proyectos cuando se desarrollan con los enfoques tradicionales o predictivos. Concluyendo que la adopción del marco Scrum reduce la cantidad de fallas del sistema.

Figura 4

Gráfico de cajas para cantidad de fallas del sistema



Nota: Se obtuvo con el software SPSS v27.

En el gráfico de cajas mostrado en la figura 4, para el indicador cantidad de fallas del sistema, se representa a la mediana para ambos grupos con una línea roja, este gráfico de cajas (Box-Plot) nos permitió tener una idea intuitiva de la normalidad de los datos. Para el grupo de control (proyectos ejecutados bajo un enfoque predictivo) el gráfico indicó que la mediana y la media están prácticamente superpuestas ($68,9200 \approx 69,0592$), como se puede observar en la tabla 5, encontrándose prácticamente igual cantidad de datos en ambas partes de la caja o también conocido como simetría respecto a la media, por lo que se presumió normalidad de los datos. Para el grupo experimental (proyectos ejecutados bajo el enfoque Scrum) el gráfico indicó que de igual manera la mediana es mayor que la media ($54,5500 > 53,4756$), como se observó en la tabla 5, encontrándose mayor concentración de datos en la parte izquierda de la media, por lo tanto, los datos presentan asimetría con respecto a la media, por lo que se presumió normalidad de los datos.

Indicador 2: Tiempo de entrega del producto

Tabla 6

Análisis descriptivo para el tiempo de entrega del producto

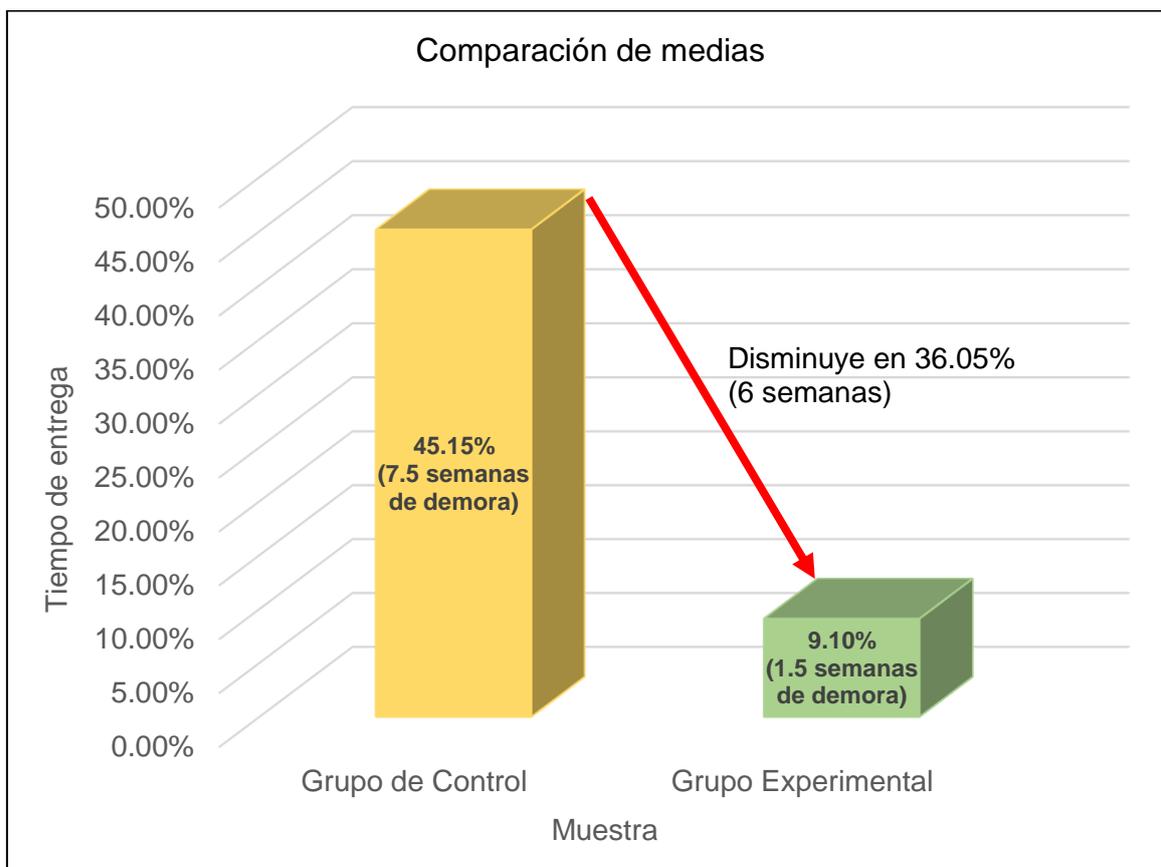
	Muestra	Estadístico	
Tiempo_de_entrega	Control	Media	45,1524
		Mediana	45,0000
		Varianza	96,024
		Desviación estándar	9,79917
		Mínimo	27,27
		Máximo	66,67
	Experimento	Media	9,0984
		Mediana	8,3300
		Varianza	10,216
		Desviación estándar	3,19620
		Mínimo	5,00
		Máximo	16,67

Nota: Se obtuvo con el software SPSS v27.

Los datos obtenidos en la tabla 6, para el indicador tiempo de entrega del producto, nos indicó que el promedio para la muestra conformado por el grupo de control (proyectos ejecutados bajo un enfoque tradicional o predictivo) es de 45.15%, mientras que el promedio para el grupo experimental (proyectos ejecutados bajo el enfoque Scrum) fue de 9.09%, demostrándose una reducción en el indicador tiempo de entrega del producto cuando la muestra conformada por el grupo de control es sometido al experimento, con lo cual se evidencio que la adopción del marco Scrum en los proyectos de software disminuye significativamente el tiempo de entrega del producto.

Figura 5

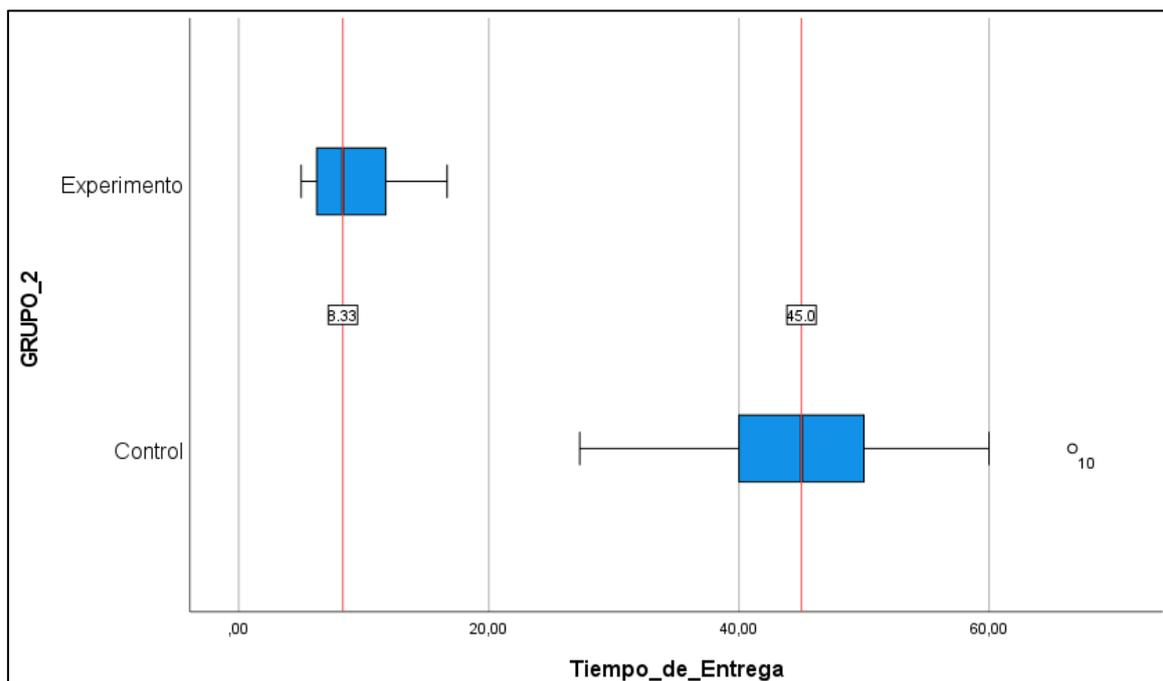
Comparación de las medias para el tiempo de entrega del producto



De igual manera, la figura 5 nos muestra como varía la media para el indicador tiempo de entrega del producto para ambas muestras (grupos de control y grupo experimental), observándose que, en el grupo de control (proyectos ejecutados bajo un enfoque tradicional), los tiempos de entrega de los productos presentaban en promedio una demora de 7 semanas y media por encima de los tiempos establecidos en la planificación. Mientras que, en el grupo experimental (proyectos ejecutados bajo el enfoque Scrum), los tiempos de entrega de los productos presentaban en promedio una demora de 1 semana y media, por encima de lo planificado, evidenciándose una disminución de 6 semanas en el tiempo de entrega del producto, lo cual representa una reducción de 36.05% en el post. Concluyendo que la adopción del marco Scrum reduce el tiempo de entrega del software.

Figura 6

Gráfico de cajas para el tiempo de entrega del producto



Nota: Se obtuvo con el software SPSS v27.

En el gráfico de cajas mostrado en la figura 6, para el indicador tiempo de entrega de software, se representa a la mediana para ambos grupos con una línea roja, este gráfico de cajas (Box-Plot) nos permitió tener una idea intuitiva de la normalidad de los datos, además en la figura se observa que se presentó un dato atípico. Para el grupo de control (sin experimento) el gráfico indicó que la mediana y la media están prácticamente superpuestas ($45,0000 \approx 45,1524$), como se puede observar en la tabla 6, encontrándose prácticamente igual cantidad de datos en ambas partes de la caja, siendo los datos simétricos respecto a la media, por lo que se presumió normalidad de los datos. Para el grupo experimental el gráfico indicó que de igual manera la mediana se acerca a la media ($8,3300 \approx 9,0984$), pero al encontrarse la caja a la izquierda del dato mínimo habrá mayor cantidad de datos a la derecha de la media, siendo los datos asimétricos respecto a la media, por lo que se presumió no normalidad de los datos.

Indicador 3: Cantidad de solicitudes de cambio

Tabla 7

Análisis descriptiva para cantidad de solicitudes de cambio

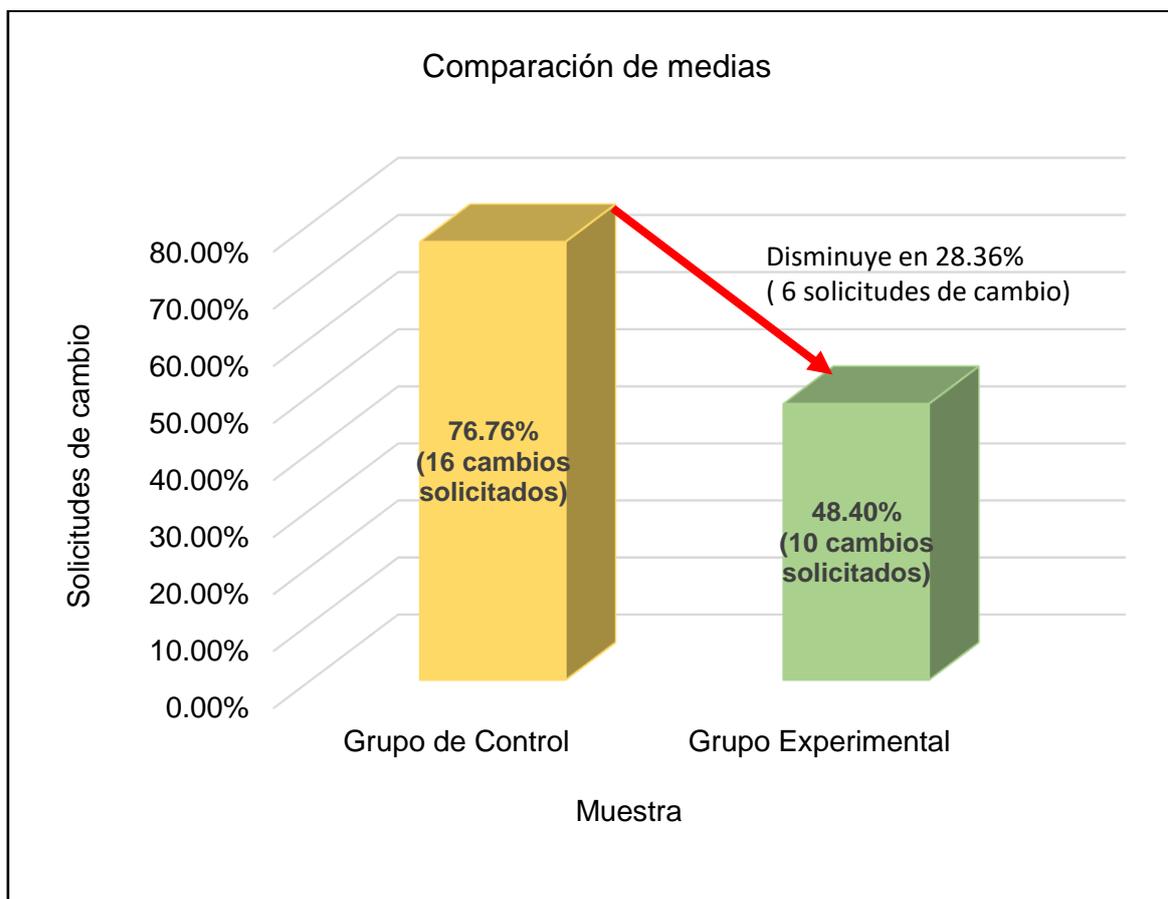
	Muestra	Estadístico	
Cantidad_de_cambios	Control	Media	76,7628
		Mediana	76,9200
		Varianza	81,509
		Desviación estándar	9,02822
		Mínimo	56,25
		Máximo	92,31
	Experimento	Media	48,3996
		Mediana	50,0000
		Varianza	128,864
		Desviación estándar	11,35185
		Mínimo	33,33
		Máximo	66,67

Nota: Se obtuvo con el software SPSS v27.

Los datos obtenidos en la tabla 7, para el indicador cantidad de solicitudes de cambio, nos indicó que el promedio para la muestra conformada por el grupo de control (proyectos ejecutados bajo un enfoque tradicional o predictivo) es de 76.76%, mientras que el promedio para el grupo experimental (proyectos ejecutados bajo el enfoque Scrum) fue de 48.40%, demostrándose una reducción del indicador cantidad de solicitudes de cambio, cuando la muestra es sometido al experimento, con lo cual se evidencio que la adopción del marco Scrum en los proyectos de software disminuye la cantidad de solicitudes de cambio en el producto.

Figura 7

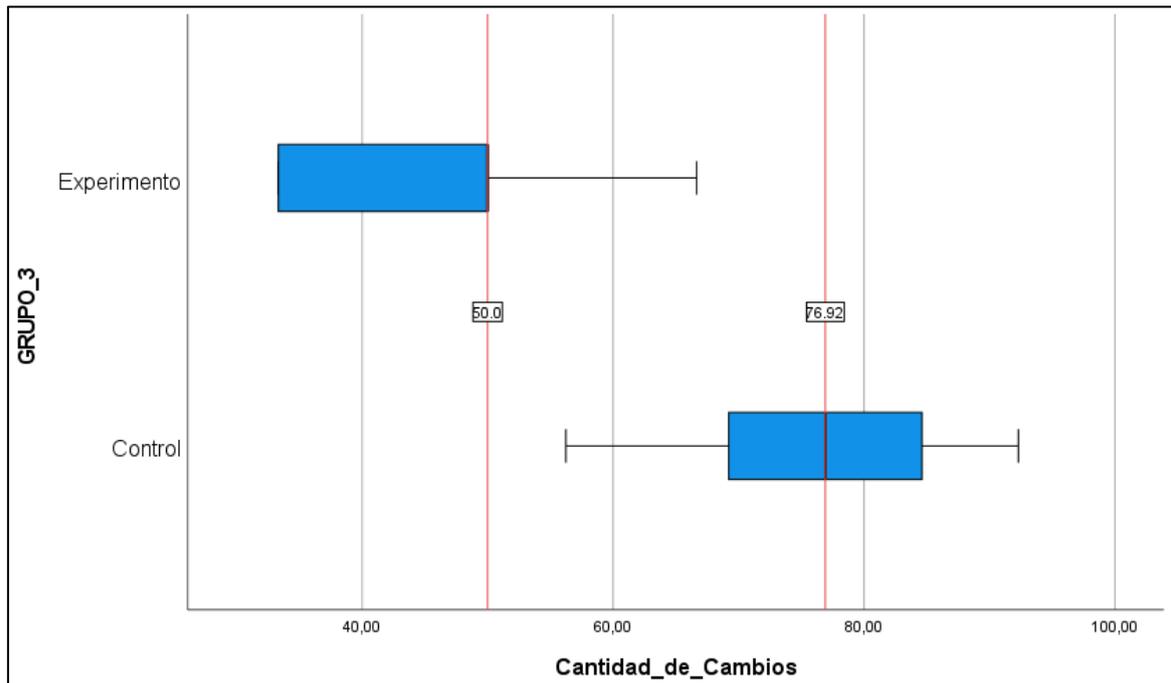
Comparación de las medias cantidad de solicitudes de cambio



De igual manera, la figura 7 nos muestra como varía la media para el indicador cantidad de solicitudes de cambio del producto para ambas muestras (grupos de control y grupo experimental), observándose que, en el grupo de control (proyectos ejecutados bajo un enfoque tradicional), la cantidad de solicitudes de cambio que sufre el producto es de 16 por proyecto en promedio. Mientras que, en el grupo experimental (proyectos ejecutados bajo el enfoque Scrum), la cantidad de solicitudes de cambio que sufre el producto es de 10 por proyecto en promedio, evidenciándose una disminución significativa de 6 solicitudes de cambio, lo cual representa una reducción de 28.36% en el post. Concluyendo que la adopción del marco Scrum reduce la cantidad de solicitudes de cambio que sufre el producto.

Figura 8

Gráfico de Cajas cantidad de solicitudes de cambio



Nota: Se obtuvo con el software SPSS v27.

En el gráfico de cajas mostrado en la figura 8, para el indicador cantidad de solicitudes de cambio, se representa a la mediana para ambos grupos con una línea roja, este gráfico de cajas (Box-Plot) nos permitió tener una idea intuitiva de la normalidad de los datos. Para el grupo de control (proyectos ejecutados bajo un enfoque tradicional o predictivo) el gráfico indicó que la mediana y la media están prácticamente superpuestas ($76,9200 \approx 76,7628$), como se puede observar en la tabla 7, encontrándose prácticamente igual cantidad de datos en ambas partes de la caja, siendo los datos simétricos respecto a la media, por lo que se presumió normalidad de los datos. Para el grupo experimental (proyectos ejecutados bajo un enfoque Scrum) el gráfico indicó que de igual manera la mediana es mayor que la media ($50,0000 > 48,3996$), como se observó en la tabla 7, encontrándose mayor distribución de datos a la izquierda de la media, siendo los datos asimétricos respecto a la media, por lo que se presumió no normalidad de los datos.

Pruebas de normalidad

Para determinar si se presenta normalidad en los datos, se utilizó la prueba de Kolmogórov-Smirnov y se aplicó a cada uno de los indicadores, para evidenciar la normalidad o no normalidad de los datos, se evaluó la significancia obtenida (p-valor) y se contrastó con las hipótesis estadísticas propuestas:

H₀: Distribución de datos normal en la muestra.

H₁: Distribución no normal de datos en la muestra.

En la investigación se trabajó con una confianza del 95%, por consiguiente, se asumió un porcentaje de error del 5% ($\alpha = 0.05$). Por lo tanto, el p-valor se evaluó de la siguiente manera:

p-valor: Nivel crítico de contraste.

Donde sí:

$p < 0.05$; Rechazo de la hipótesis nula - datos con distribución no normal.

$p \geq 0.05$; Acepta la hipótesis nula - datos con distribución normal.

Indicador 1: Cantidad de fallas del sistema

Tabla 8

Prueba de normalidad para el indicador 1

	Muestra	Kolmogorov-Smirnov		
		Estadístico	gl	Sig.
Cantidad_de_fallas	Control	,107	25	,200
	Experimento	,264	25	,000

Nota: Se obtuvo con el software SPSS v27.

Los resultados de la prueba (tabla 8), evidenciaron que la significancia del grupo de control (sin experimento) es mayor a 0.05 ($p = 0.200$), esto significó que los datos presentan normalidad, de igual manera la significancia del grupo experimental es menor que 0.05 ($p = 0.000$), esto significó que los datos presentan no normalidad.

De los datos obtenidos se cumple que basta con que uno de ellos sea menor a 0.05 ($p < 0.05$), para descartar la hipótesis nula, concluyendo que los datos tienen distribución no normal.

Indicador 2: Tiempo de entrega de software

Tabla 9

Prueba de normalidad para el indicador 2

	Muestra	Kolmogorov-Smirnov		
		Estadístico	gl	Sig.
Tiempo_de_entrega	Control	,110	25	,200
	Experimento	,170	25	,061

Nota: Se obtuvo con el software SPSS v27.

Los resultados de la prueba (tabla 9), evidenciaron que la significancia del grupo de control (sin experimento) es mayor a 0.05 ($p = 0.200$), esto significó que los datos presentan normalidad, de igual manera la significancia del grupo experimental es mayor que 0.05 ($p = 0.061$), esto significó que los datos presentan normalidad. Ambos obtuvieron un p mayor a 0.05 ($p > 0.05$) aceptando la hipótesis nula, concluyendo que los datos tienen distribución normal.

Indicador 3: Cantidad de solicitudes de cambio

Tabla 10

Prueba de normalidad para el indicador 3

	Muestra	Shapiro-Wilk		
		Estadístico	gl	Sig.
Cantidad_de_cambios	Control	,967	25	,567
	Experimento	,819	25	,000

Nota: Se obtuvo con el software SPSS v27.

Los resultados de la prueba (tabla 10), evidenciaron que la significancia del grupo de control (sin experimento) es mayor a 0.05 ($p = 0.567$), esto significó que los datos presentan normalidad, de igual manera la significancia del grupo experimental es menor que 0.05 ($p = 0.000$), esto significó que los datos presentan no normalidad. De los datos obtenidos se cumple que basta con que uno de ellos sea menor a 0.05 ($p < 0.05$), para descartar la hipótesis nula, concluyendo que los datos tienen distribución no normal.

Prueba de hipótesis

Para validar las hipótesis se utilizaron las pruebas estadísticas de acuerdo a la tabla 11 obtenida de las pruebas de normalidad.

Tabla 11

Resultados obtenidos de las pruebas de normalidad

Indicador	Normalidad	Pruebas
Cantidad de fallas	No normal	U-Mann Whitney
Tiempo de entrega del producto	Normal	T-Student Welch
Cantidad de solicitudes de cambio	No normal	U-Mann Whitney

Hipótesis específica 1: Cantidad de fallas

De acuerdo a los resultados de la tabla 11, para el indicador cantidad de fallas, se aplicó el estadístico no paramétrico de U-Mann Whitney, para validar lo siguiente:

H_0 : El marco de trabajo Scrum NO mejora significativamente la calidad del producto de los proyectos de software en una empresa consultora, Lima 2023.

H_1 : El marco de trabajo Scrum mejora significativamente la calidad del producto de los proyectos de software en una empresa consultora, Lima 2023.

Tabla 12

Resultados de la prueba de U-Mann Whitney para el indicador cantidad de fallas del sistema

Estadístico	Cantidad_de_fallas
U de Mann-Whitney	,000
W de Wilcoxon	325,000
Sig. asintótica(bilateral)	,000

Nota: Se obtuvo con el software SPSS v27.

Tabla 13

Resumen de la prueba de U-Mann Whitney para el indicador cantidad de fallas del sistema

Hipótesis nula	Prueba	Sig.	Decisión
La distribución de Cantidad_de_fallas es la misma entre categorías de Muestra	Prueba U de Mann-Whitney para muestras independientes	,000	Rechace la hipótesis nula.

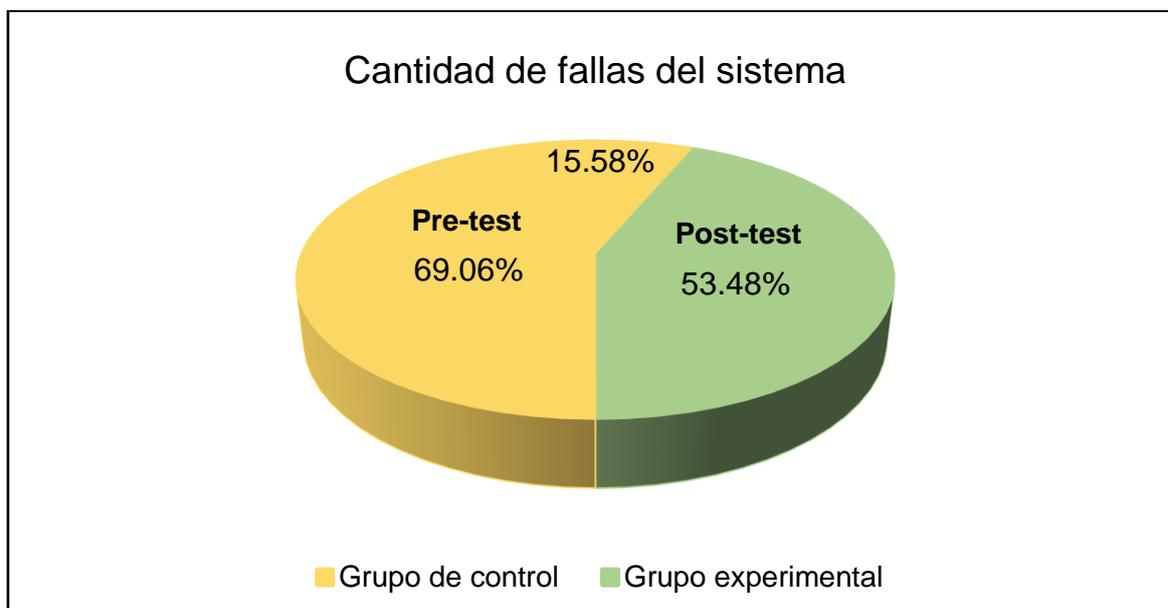
Nota: Se obtuvo con el software SPSS v27.

La tabla 12 evidencia los resultados de aplicar la prueba no paramétrica, donde la significancia (Sig. asintótica) bilateral tuvo un valor de $p = 0.000$, lo que significa que se descartó la hipótesis nula (tabla 13) y se tomó como válida la hipótesis de investigación. Por lo tanto, se sustentó que el marco de trabajo Scrum mejora significativamente la calidad del producto en los proyectos de software.

En resumen, en la figura 9 se presenta la disminución del 15.58% en la cantidad de fallas de los sistemas, traduciéndose en 15 fallas menos cometidas en los sistemas de los productos a entregar, cuando estos proyectos son ejecutados utilizando el marco Scrum. Evidenciándose en una mejora en la calidad del producto.

Figura 9

Reducción de la cantidad de fallas del sistema



Hipótesis específica 2: Tiempo de entrega del producto

De acuerdo a los resultados de la tabla 11, para el indicador tiempo de entrega del producto, se utilizó el estadístico paramétrico de T-STUDENT para validar lo siguiente:

H₀: El marco de trabajo Scrum NO disminuye significativamente el tiempo de entrega del producto en los proyectos de software en una empresa consultora, Lima 2023.

H₁: El marco de trabajo Scrum disminuye significativamente el tiempo de entrega del producto en los proyectos de software en una empresa consultora, Lima 2023.

Se debe tomar en consideración que la prueba T-Student compara la diferencia entre las medias y se aplica para datos normales y con varianza homogénea. Entonces el siguiente paso fue determinar si hay homogeneidad de varianza de las muestras utilizando la prueba de Levene o prueba de homocedasticidad. Esta prueba plantea como hipótesis nula la existencia de homogeneidad de varianzas la cual se acepta si el p-valor es mayor o igual a 0.05

($p \geq 0.05$) y como hipótesis no nula la no existencia de homogeneidad de varianzas la cual se acepta si el p-valor es menor a 0.05 ($p < 0.05$).

Tabla 14

Prueba de homogeneidad de varianzas para el indicador tiempo de entrega del producto

		Estadístico			
		de Levene	gl1	gl2	Sig.
Tiempo_de	Se basa en la media	15,859	1	48	,000
_entrega	Se basa en la mediana	15,648	1	48	,000

Nota: Se obtuvo con el software SPSS v27.

La tabla 14 evidencia los resultados de aplicar la prueba de homogeneidad, donde la significancia basada en la media tuvo un valor de $p = 0.000$, lo cual desestima la hipótesis nula y tomó como válida la hipótesis no nula, comprobándose que no existe homogeneidad en las varianzas de las muestras independientes, con este resultado se utilizó la prueba paramétrica T-Students mejorada también conocida como T-Students Welch.

Tabla 15

Resultados de la prueba T-Student Welch para el indicador tiempo de entrega del producto

Tiempo_de_entrega	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	95% de intervalo de confianza de la diferencia	
				Inferior	Superior
Se asumen varianzas iguales	,000	36,05400	2,06145	31,90918	40,19882
No se asumen varianzas iguales	,000	36,05400	2,06145	31,83817	40,26983

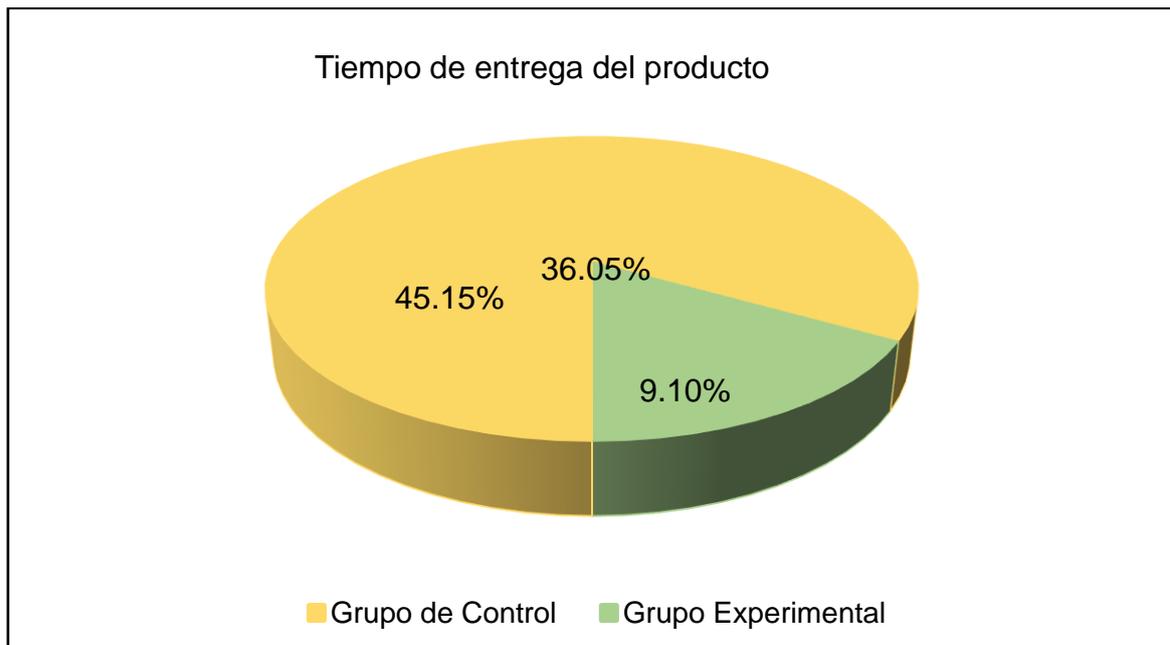
Nota: Se obtuvo con el software SPSS v27.

La tabla 15 evidencia los resultados de aplicar la distribución paramétrica T-Student, se prestó atención para el caso en la cual las varianzas no son iguales (no homogeneidad), observando que la significancia (Sig.) bilateral tuvo un valor de $p = 0.000$, dejando de lado la hipótesis nula y tomando como válida la hipótesis de investigación, es decir el marco de trabajo Scrum disminuyó significativamente el tiempo de entrega del producto en los proyectos de software en una Empresa Consultora, otra manera de validar la hipótesis es visualizar gráficamente si los intervalos de confianza inferior y superior incluyen al cero, observándose en la tabla 16 que los valores inferior y superior (31.83 y 40.27) graficados en una línea recta no incluyen el cero, por lo que el p es menor a 0.05 ($p < 0.05$), tomando como válida la hipótesis de investigación. Por lo tanto, se sustentó que el marco de trabajo Scrum disminuye significativamente el tiempo de entrega de los productos de los proyectos de software.

En resumen, en la figura 10 se presenta la disminución del 36.05% en el tiempo de entrega del producto, traduciéndose en 6 semanas menos en el tiempo de entrega, cuando los proyectos son ejecutados utilizando el marco Scrum. Evidenciándose una reducción en el tiempo de entrega del producto.

Figura 10

Reducción del tiempo de entrega del producto



Hipótesis específica 3: Cantidad de solicitudes de cambio

De acuerdo a los resultados de la tabla 11, para el indicador cantidad de solicitudes de cambio, se utilizó el estadístico no paramétrico de U-Mann Whitney para validar lo siguiente

H₀: El marco de trabajo Scrum NO disminuye significativamente la cantidad de solicitudes de cambio al producto de los proyectos de software en una empresa consultora, Lima 2023.

H₁: El marco de trabajo Scrum disminuye significativamente la cantidad de solicitudes de cambio al producto de los proyectos de software en una empresa consultora, Lima 2023.

Tabla 16

Resultados de la prueba de U-Mann Whitney para el indicador cantidad de solicitudes de cambio

Estadístico	Cantidad_de_cambios
U de Mann-Whitney	11,000
W de Wilcoxon	336,000
Sig. asintótica(bilateral)	,000

Nota: Se obtuvo con el software SPSS v27.

Tabla 17

Resumen de la prueba de U-Mann Whitney para el indicador cantidad de solicitudes de cambio

Hipótesis nula	Prueba	Sig.	Decisión
La distribución de Cantidad_de_cambios es la misma entre categorías de Muestra.	Prueba U de Mann-Whitney para muestras independientes	,000	Rechace la hipótesis nula.

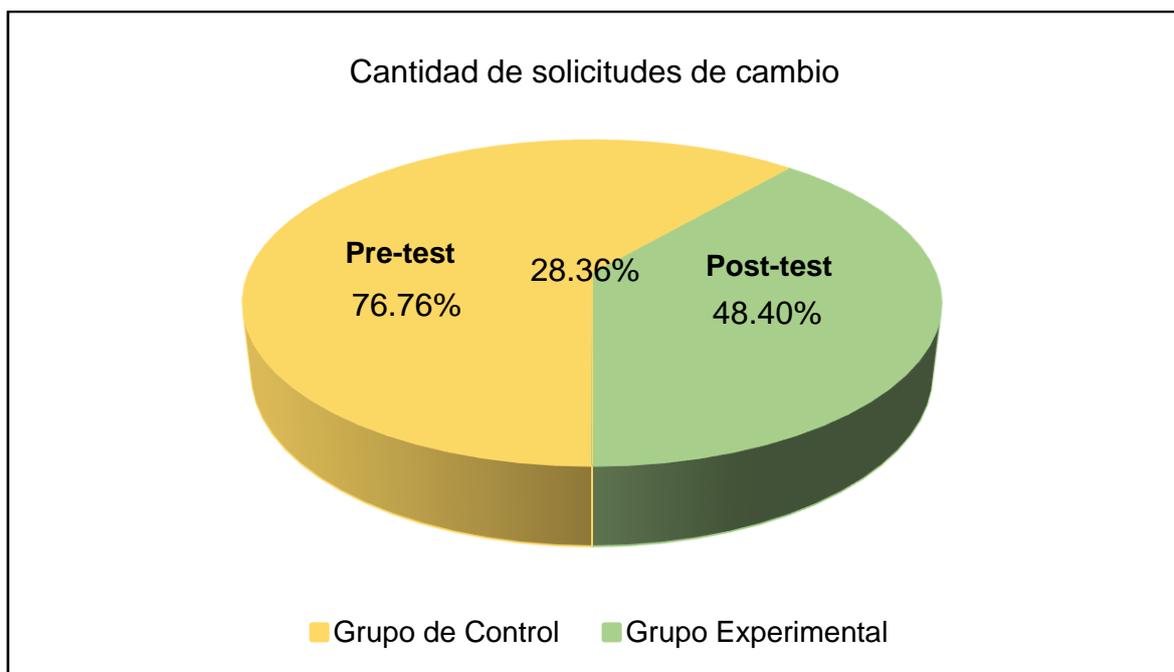
Nota: Se obtuvo con el software SPSS v27.

La tabla 16 evidencia los resultados de aplicar la prueba no paramétrica, donde la significancia (Sig. Asintótica bilateral) tuvo un valor de $p = 0.000$, lo que significa que se dejó de lado la hipótesis nula (tabla 17) y se tomó como válida la hipótesis de investigación. Por lo tanto, se sustentó que el marco de trabajo Scrum disminuye la cantidad de solicitudes de cambio en el producto.

En resumen, en la figura 11 se presenta la disminución del 28.36% en la cantidad de solicitudes de cambio, traduciéndose en 6 solicitudes menos, cuando los proyectos son ejecutados utilizando el marco Scrum. Evidenciándose una reducción en la cantidad de solicitudes de cambio.

Figura 11

Reducción de la cantidad de solicitudes de cambio



V. DISCUSIÓN

Con los resultados obtenidos a partir de las fichas de registro utilizados en la presente investigación, se puede evidenciar de manera general que la adopción de Scrum si influye positivamente en la gestión de los proyectos de software en la empresa de consultoría en Lima 2022, específicamente influye en los indicadores cantidad de fallas del sistema, tiempo de entrega del producto y cantidad de solicitudes de cambio.

Indicador 1: Cantidad de fallas del sistema.

Con respecto a la dimensión calidad medida a través del indicador Cantidad de Fallas del Sistema, se observó que después del tratamiento al grupo experimental se redujo la cantidad de fallas que se presentó en los sistemas, esto se demostró con el análisis descriptivo realizado a las muestras independientes de 25 objetos consideradas en la investigación, en donde se recopilaron los datos de ambas muestras (control y experimental), las cuales se procesaron estadísticamente arrojando las medias de 69.05% para el grupo de control y de 53.47% para el grupo experimental (véase tabla 5), encontrándose una disminución del 15.58% en el índice de fallas en los sistemas (figura 3) en la muestra sometida al experimento. Concluyendo que la adopción del marco de trabajo Scrum disminuye la cantidad de fallas del sistema, pero debemos mencionar que comparando las medias no será suficiente para asegurar que hay una diferencia y por lo tanto haya una mejora, para asegurarlo fue necesario aplicar las pruebas de hipótesis, en este caso para 2 muestras independientes. Se dispone de 2 pruebas de hipótesis, la aplicación de una u otra dependió de determinar si los datos se distribuyen de manera normal, una manera de predecir la normalidad de los datos es a través del diagrama de cajas (Box-Plot), del cual podemos obtener mucha información como la normalidad de los datos, esta se determinó visualmente atendiendo a la posición de la línea mediana dentro del bloque azul (caja). Para el grupo sin experimentó el gráfico (véase figura 4) indicó que la mediana divide simétricamente a la caja encontrándose igual cantidad de datos en ambas partes de la caja, en esta situación se intuye que los datos están distribuidos normalmente. Para el grupo experimental el gráfico indicó que la mediana divide asimétricamente a la caja, hay

mayor cantidad de datos al lado izquierdo de la caja, en esta situación se intuye que los datos no están distribuidos normalmente. Pero como se mencionó este procedimiento nos brinda una idea de cómo están distribuidos los datos, se debe de comprobar estos resultados, aplicando las pruebas de normalidad respectiva.

El análisis inferencial involucro realizar las pruebas de normalidad a los datos para cada indicador y realizar el contraste de hipótesis, para el indicador cantidad de fallas del sistema se determinó con el estadístico de Kolmogórov-Smirnov, los resultados arrojaron que para el grupo de control (proyectos gestionados con enfoque predictivo, llámese PMBOK) se obtuvo una significancia superior a 0.05, lo cual indicó que los datos presentan normalidad, por su parte para el grupo experimental (proyectos gestionados con enfoque Scrum) la significancia obtenida estuvo por debajo del valor de 0.05, lo cual indicó que los datos no presentan normalidad, con lo cual se comprobó el resultado del análisis del gráfico de cajas (Box-Plot).

Concluyendo que los datos para el indicador cantidad de fallas del sistema presentan normalidad. La validez de la hipótesis para el indicador se realizó utilizando la prueba no paramétrica de U-Mann Withney, ya que se tuvo 2 muestras independientes, no teniendo distribución normal de datos, la prueba nos permitió comparar dos muestras independientes pertenecientes a una misma población y validar si las medias son iguales o no, los resultados (tabla 13) arrojaron una significancia (Sig. Asintótica) menor a 0.05 ($p = 0.000$), no superando el margen de error del 5% planteado, adicionalmente el software estadístico nos permitió ejecutar la prueba de otra manera, mostrando un resumen de contraste con la decisión de rechazo (véase tabla 14), comprobando que se descartó la hipótesis nula y se aceptó la hipótesis de investigación. Concluyendo que la adopción del marco de trabajo Scrum mejora significativamente la calidad del producto de los proyectos de software.

Los resultados mostrados fueron similares a los obtenidos por Romero (2017), quien aplicó su modelo basado en modelos ágiles CRM y Scrum que permiten detectar fallas en etapas tempranas, con la finalidad de desarrollar

productos de calidad en los outsourcing. Al aplicar el modelo se elaboraron procesos que permitieron una adecuada evaluación de la calidad de producto considerando factores internos y externos del software, los procesos permitieron mejorar la calidad tanto interna como externa en un 92%. concluyendo que la utilización de un modelo basado en CRM y Scrum aumenta la calidad del software. Asimismo, Merizalde (2018) obtuvo resultados similares al evaluar el uso de la metodología Scrum en la gestión de proyectos de software en empresas de guayaquil, debido a la necesidad de muchas empresas de buscar nuevos métodos de gestión para la entrega de productos de calidad, para los cuales los enfoques predictivos ya no estaban dando resultados, la autora centró su investigación en los defectos detectados, los cuales mostraron valores bajos, al ejecutar los proyectos bajo el enfoque ágil Scrum que al ejecutarlos bajo el enfoque tradicional, la autora señala que ambos valores están en relación de 2.5/5, lo que evidencia una disminución de los defectos detectados en un 50% mejorando la calidad del producto.

Indicador 2: Tiempo de entrega del producto.

Con respecto a la dimensión tiempo de entrega medida a través del indicador tiempo de entrega del producto, se observó que luego de procesar estadísticamente los datos de ambas muestras (control y experimental) se obtuvo una media de 45.15% en el grupo de control y 9.09% en el grupo experimental (véase tabla 6), observándose una disminución del 36.06% (figura 5) en el índice de fallas en los sistemas en la muestra sometida al experimento. Concluyendo que la adopción del marco de trabajo Scrum disminuye el tiempo de entrega del producto, pero al igual que el caso anterior debemos aplicar las pruebas de hipótesis respectiva. Para elegir la prueba de hipótesis a aplicar primero se determinó la normalidad de los datos, utilizando el gráfico de cajas (Box-Plot), del cual se observó para el grupo de control (sin experimentó) que la mediana divide simétricamente a la caja encontrándose igual cantidad de datos en ambas partes de la caja (véase figura 6), en esta situación se intuye que los datos están distribuidos normalmente. Para el grupo experimental la mediana podría considerarse que se encuentra al centro de la caja, pero la caja está orientada a la izquierda (véase figura 6), intuyendo que los

datos están distribuidos normalmente, estos resultados nos dan solo una idea, se deben comprobar aplicando las pruebas de normalidad respectiva.

En el análisis inferencial se aplicó el estadístico de Kolmogórov-Smirnov, arrojando como resultado un valor de significancia superior a 0.05 en el grupo de control, lo cual indicó que los datos presentan normalidad, por su parte para el grupo experimental la significancia obtenida fue superior a 0.05, lo cual indicó que los datos presentan normalidad. Estos resultados permitieron validar que lo observado en el gráfico de cajas es solo una idea, debiendo comprobarse con la prueba de normalidad.

Concluyendo que los datos para el indicador tiempo de entrega del producto se distribuyen normalmente. La validez de la hipótesis para el indicador se realizó aplicando la distribución paramétrica de T-Student, ya que las muestras tienen distribución normal de datos, en este caso de normalidad de datos de ambas muestras se debe de comprobar que la variabilidad (distribución) de los datos de los grupos sea no distintas, esto se puede visualizar en la figura 6, en la cual la distribución de los datos dentro de las cajas deben de ser similares y esto se cumple cuando las cajas son de igual tamaño o hay homogeneidad de varianzas, de la figura podemos observar que no hay igualdad de tamaños en las cajas por lo que no hay homogeneidad de varianzas, pero como mencionamos el gráfico de cajas nos brinda una idea del comportamiento de nuestros datos, debiendo validarse con la prueba de Levene, los resultado de la prueba (véase tabla 15), arrojaron una significancia menor al 0.05, comprobándose que no hay homogeneidad de varianzas. Para varianzas no homogéneas utilizaremos igual la distribución de T-Student cuyo resultado (véase tabla 16) arrojó significancias para homogeneidad y para no homogeneidad, se utilizó la última a la cual se le denomina distribución de T-Student Welch, del resultado obtenido la significancia bilateral es inferior a 0.05, comprobándose que se descartó la hipótesis nula y se aceptó la hipótesis de investigación, concluyendo que la adopción del marco de trabajo Scrum disminuyó significativamente el tiempo de entrega del producto en los proyectos de software en una Empresa Consultora.

Los resultados son similares a los obtenidos por Reyna (2018) quien en su investigación evaluó el impacto de aplicar un marco de trabajo basado en Scrum y CMMI en el tiempo planificado de los proyectos de desarrollo de software, en el análisis la media para un grupo de proyectos que no utilizaron un marco de trabajo basado en Scrum y CMMI formado por 84 proyectos fue de 53.75%, mientras que para un grupo de proyectos que utilizaron Scrum y CMMI formado por 99 proyectos fue de 22.19%. El autor concluye y estima que, si un proyecto tiene un tiempo de ejecución planificado de 100 horas, utilizando el marco de trabajo basado en Scrum y CMMI el proyecto podría extenderse hasta 23 horas adicionales en promedio, y de no utilizar el marco de trabajo Scrum y CMMI el proyecto podría extenderse 54 horas en promedio. Asimismo, Carrasco (2022), obtuvo resultados similares al implementar un modelo de gestión basado en Scrum en una entidad de justicia, debido a la necesidad de buscar nuevo método de gestión para la entrega de productos en menor tiempo, para los cuales los enfoques predictivos ya no estaban dando resultados, el autor manifestó que en la etapa de pre-test (gestionado proyectos con enfoque tradicionales), el promedio del tiempo de entrega de producto es 13.83%, mientras que en el post-test (gestionando proyectos con el modelo planteado), el promedio del tiempo de entrega de producto es 8.83% del 5%, evidenciándose una mejora en el tiempo de entrega de los productos.

Indicador 3: Cantidad de solicitudes de cambio.

Con respecto a la dimensión solicitudes de cambio medida a través del indicador cantidad de solicitudes de cambio, se observó que luego de procesar estadísticamente los datos de ambas muestras (control y experimental) se obtuvo una media de 76.76% en el grupo de control y de 48.39% en el grupo experimental (véase tabla 7), observándose una disminución del 28.36% (véase figura 7) en la cantidad de solicitudes de cambio, este resultado debe de ser validado con la prueba de hipótesis. En un primer paso se recurrió al gráfico de cajas (Box-Plot), del cual se observó que para el grupo de control (sin experimentó) que la mediana divide simétricamente a la caja encontrándose igual cantidad de datos en ambas partes de la caja, en esta situación se intuye que los datos están distribuidos normalmente (véase figura 8). Para el grupo experimental el gráfico indico que la mediana divide asimétricamente a la caja, hay mayor cantidad de datos al lado

izquierdo de la caja, en esta situación se intuye que los datos no están distribuidos normalmente. Pero como se mencionó este procedimiento nos brinda una idea de cómo están distribuidos los datos, se debe de comprobar estos resultados, aplicando las pruebas de normalidad respectiva.

En el análisis inferencial se aplicó el estadístico de Kolmogórov-Smirnov, obteniendo como resultado una significancia superior a 0.05 en el grupo de control, lo cual indicó que los datos presentan normalidad, por su parte para el grupo experimental la significancia obtenida fue inferior a 0.05, lo cual indicó que los datos no presentan normalidad, con lo cual se comprobó el resultado del análisis del gráfico de cajas (Box-Plot).

Concluyendo que los datos para el indicador cantidad de solicitudes de cambio no presentan normalidad. La validez de la hipótesis para el indicador se realizó aplicando la prueba no paramétrica de U-Mann Withney, debido a la no normalidad de los datos, se obtuvo como resultado (véase tabla 17) una significancia bilateral inferior a 0.05. Asimismo, se pudo ejecutar la prueba de otra manera, permitiéndonos obtener el resumen de contraste con la decisión de rechazo (véase tabla 18), comprobando que se descartó la hipótesis nula y se aceptó la hipótesis de investigación. Concluyendo que la adopción del marco de trabajo Scrum disminuye la cantidad de solicitudes de cambio en los proyectos de software en la empresa consultora.

Los resultados fueron similares a los obtenidos por Guzmán (2019), quien planteó implementar la práctica ágil Scrum en la gestión de los proyectos de software en un retail, para disminuir la cantidad de solicitudes de cambio en el producto en proyectos gestionados con un enfoque predictivo, los cuales se caracterizaban por una planificación inicial y un seguimiento riguroso del plan, los cambios presentados ocasionaron correcciones al producto ocasionando gastos adicionales y cambios en el plan. La implementación de la práctica ágil Scrum involucró una serie de procesos en cada etapa del proyecto con roles definidos, introduciendo las iteraciones y estando abierto a los cambios que solo generen valor, los resultados que se obtuvieron arrojaron que para los 10 proyectos objetos

del estudio se redujeron 40 solicitudes de cambio, siendo equivalente a un 71% de reducción. Igualmente, Villavicencio (2020), obtuvo resultados similares al implementar un modelo para controlar los cambios de los proyectos de una empresa, el modelo propuesto se base en ITIL y Scrum, previamente el autor evaluó el estado actual del área de desarrollo de la empresa, encontrando un déficit en la organización con respecto a los cambios que se dan en relación al requerimiento inicial del producto, y que estos cambios se presentan durante la etapa de desarrollo del producto, la evaluación arrojó que las solicitudes de cambio son efectuadas por un 78.95% de los involucrados, mientras que luego de aplicar el modelo se elaboró una matriz con los responsables de realizar las solicitudes de cambio, siendo un 36.84% de los involucrados, evidenciándose una disminución del 42.11% en las solicitudes de cambio. Concluyendo que la implementación de un modelo basado en ITIL y Scrum mejora significativamente las solicitudes de cambio en los proyectos de software.

VI. CONCLUSIONES

Primera: Los hallazgos presentados, producto del análisis realizado en la presente investigación, comprobó la influencia del marco Scrum en la gestión de proyectos de software en una empresa consultora, sustentándose en la mejora de la calidad del producto entregado. Asimismo, se redujo el tiempo de entrega del producto, por último, se logró disminuir los requerimientos de cambio en el producto.

Segunda: Se comprobó la influencia del Marco de trabajo Scrum en la mejora de la calidad del producto, el cual fue medido a través del indicador cantidad de fallas del sistema, mostrándose una disminución del 15.58% luego de implementar el Marco de Trabajo Scrum, en comparación con respecto al grupo gestionado con enfoque tradicional. Con lo cual se mejoró la productividad de las actividades durante el desarrollo de los sistemas en la empresa, ya que se redujeron las incidencias durante las pruebas antes de colocar a producción los sistemas. Asimismo, se redujo el retrabajo y los costos adicionales en los proyectos.

Tercera: Se comprobó la influencia del Marco de trabajo Scrum en la reducción del tiempo de entrega del producto, el cual fue medido a través del indicador tiempo de entrega, mostrándose una disminución del 36.05% luego de implementar el Marco de Trabajo Scrum, en comparación con respecto al grupo gestionado con enfoque tradicional. Por tal motivo la empresa mejoró su reputación comercial en su nicho de mercado, entregando sus productos en sus plazos establecidos generando la satisfacción de sus clientes.

Cuarta: Se comprobó la influencia del Marco de Trabajo Scrum en la disminución de las solicitudes de cambio al producto, El cual fue medido a través del indicador cantidad de solicitudes de cambio al producto, mostrándose una disminución del 28.36% luego de

implementar el Marco de Trabajo Scrum, en comparación con respecto al grupo gestionado con enfoque tradicional. De igual manera se incrementó la productividad de los proyectos en general al implementar procesos menos burocráticos que no afectaban a la línea base del proyecto, reduciendo los costos del proyecto.

VII. RECOMENDACIONES

Primera: Luego de validar las hipótesis de los indicadores planteados en la presente investigación después de implementar el marco Scrum en la gestión de proyectos de software en una empresa consultora, se recomienda al gerente de proyectos de la consultora enfocarse en los equipos de trabajo, inculcándoles los principios y valores del marco Scrum y guiarlos hasta lograra equipos colaborativos y autogestionados, lo cual va a permitir mejorar significativamente el proceso de gestión de proyectos de software.

Segunda: Es muy importante plantear al gerente de proyectos que se elabore talleres de sensibilización para todo el personal en la adopción del marco de trabajo y que tengan muy en claro que, Scrum no es aplicable a cualquier proyecto, si se tiene muy claro y definido el alcance el proyecto debe de desarrollarse bajo un enfoque predictivo, por el contrario, si no se tiene definido el alcance, los enfoque ágiles como Scrum es el indicado.

Tercera: Se recomienda al gerente de proyectos medir el desempeño del proyecto como parte de la mejora continua, haciendo uso de los indicadores clave, como los gráficos de avance de proyecto y medir periódicamente los resultados obtenidos, permitiendo identificar las mejoras de todo el proceso.

Cuarta: El presente trabajo de investigación pretende servir de modelo para los gerentes de proyectos de otras empresas que aún no tengan muy en claro los beneficios de aplicar el Marco Scrum a los proyectos cuyas características sean adecuadas para su aplicación, teniendo muy en cuenta su adaptación de acuerdo al entorno de la organización.

REFERENCIAS

- Akingbehin, K. (2005). *A quantitative supplement to the definition of software quality*.
Published in: Third ACIS Int'l Conference on Software Engineering Research,
Management and Applications (SERA'05)
<https://ieeexplore.ieee.org/abstract/document/1563182>
- Amaro, S. y Valverde, J. (2007). Metodologías ágiles. Universidad Nacional de Trujillo. Facultad de Ciencias Físicas y Matemáticas. Escuela de Informática.
https://www.academia.edu/23325522/Universidad_Nacional_de_Trujillo
- Amiel, J. (2014). *Metodología y diseño de la investigación Científica*. [En línea]. (1ª ed.). Miraflores, Perú: Editorial de la Universidad Científica del Sur. ISBN 978-992-967-707-4.
[https://books.google.es/books?hl=es&lr=&id=jwejBAAQBAJ&oi=fnd&pg=PP6&dq=Metodolog%C3%ADa+y+dise%C3%B1o+de+la+investigaci%C3%B3n+Cient%C3%ADfica.+\(1%C2%B0ed.\).&ots=WPbdsE9dRx&sig=f2JbPpIZ0uY8l2W5NOrpFTDpiXo#v=onepage&q&f=false](https://books.google.es/books?hl=es&lr=&id=jwejBAAQBAJ&oi=fnd&pg=PP6&dq=Metodolog%C3%ADa+y+dise%C3%B1o+de+la+investigaci%C3%B3n+Cient%C3%ADfica.+(1%C2%B0ed.).&ots=WPbdsE9dRx&sig=f2JbPpIZ0uY8l2W5NOrpFTDpiXo#v=onepage&q&f=false)
- Andia, W. (2017). *Manual de investigación universitaria, Pautas para la planificación de una tesis*. 1era edición. Ediciones arte y pluma. Lima Perú.
<https://n9.cl/l3ev>
- Arbaiza, L. (2019). *Cómo elaborar una tesis de grado*. Colombia: Esan Ediciones.
- Arias J. y Covinos, M. (2021). *Diseño y metodología de la investigación*. Enfoques Consulting EIRL.
<http://hdl.handle.net/20.500.12390/2260>
- Atmowardoyo H. (2018). *Research Methods in TEFL Studies: Descriptive Research, Case Study, Error Analysis, and R & D*. Journal of Language Teaching and Research, 9(1), 197-204.
<http://dx.doi.org/10.17507/jltr.0901.25>
- Azanha, A., Terra, A., Camargo, J. y Antonioli, P. (2017), "Agile project management with Scrum: A case study of a Brazilian pharmaceutical company IT project", International Journal of Managing Projects in Business, Vol. 10 No. 1, pp. 121-142.
<https://doi.org/10.1108/IJMPB-06-2016-0054>

- Azeem, M., Sang, J., Ali, Arif., Fazal-E-Amin, N. y Shafi, M. (2017). *Improving the Quality of Software Development Process by Introducing a New Methodology–AZ-Model*. <https://doi.org/10.1109/ACCESS.2017.2787981>.
<https://ieeexplore.ieee.org/abstract/document/8241771/authors#authors>
- Beedle, M., Devos, M., Schwaber, K. y Sutherland, J. (1998). *SCRUM: An extension pattern language for hyperproductive software development*. ResearchGate. Recuperado de <https://bit.ly/2Oda7ks>
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. y Thomas, D. (2001). *Manifiesto for Software Agile Development* [Internet]. Disponible en <http://agilemanifesto.org/>
- Buffardi, K. (2020). *Assessing Individual Contributions to Software Engineering Projects with Git Logs and User Stories*. Proceedings of the 51st ACEM Technical Symposium on Computer Science Education.
Doi: 10.1145/3328778.3366948.
- Cabezas, E., Andrade, D. y Torres, J. (2018). *Introducción a la metodología de la investigación científica*. 1era edición electrónica. Lima: Universidad de las Fuerzas Armadas ESPE.
<http://repositorio.espe.edu.ec/jspui/bitstream/21000/15424/1/Introduccion%20a%20la%20Metodologia%20de%20la%20investigacion%20cientifica.pdf>
- Cappelli, P. y Tavis, A. (2018). *HR goes agile*. Harvard Business Review. Recuperado de: <https://hbr.org/2018/03/the-new-rules-of-talent-management>
- Carbajal, D. (2020). *Modelo de gestión de proyectos para mejorar la efectividad de las iniciativas tecnológicas en hospitales de nivel II y III de la región Lambayeque*.
https://tesis.usat.edu.pe/bitstream/20.500.12423/3105/1/TM_CarbajalVasquezDiegoAlonso.pdf
- Carrasco, J. (2022). *Modelo de Gestión de Proyectos para mejorar el Proceso de Desarrollo de Software de la Corte Superior de Justicia de Lambayeque, 2021*.
https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/85518/Carrasco_ZJK-SD.pdf?sequence=1&isAllowed=y

- Castaño, C. y Cadavid, J. (2017). *La transformación de las metodologías de desarrollo y la tendencia ágil*. Fundación Universitaria Católica Lumen Gentium SNIES 2731. Recuperado de: [https://repository.unicatolica.edu.co/bitstream/handle/20.500.12237/990/ T_RANSFORMACIÓN_METODOLOGÍAS_DESARROLLO_TENDENCIAS_Á_GIL.pdf?sequence=1&isAllowed=y](https://repository.unicatolica.edu.co/bitstream/handle/20.500.12237/990/T_RANSFORMACIÓN_METODOLOGÍAS_DESARROLLO_TENDENCIAS_Á_GIL.pdf?sequence=1&isAllowed=y)
- Córdova, I. (2018). *Instrumentos de Investigación*. 1era edición. Lima, Perú: Editorial San Marcos E.I.R.L., editor.
- Cruz, J., Guevara, H. y Flores, J. (2020). *Knowledge areas and key phases in project management: Considerations theoretical*. Article. Universidad del Zulia. <https://www.scopus.com/record/display.uri?eid=2-s2.0-85086823584&doi=10.37960%2frvg.v25i90.32409&origin=inward&txGid=f47706a07f8e3c08c1d36029d141365a>
- Díaz, P. (2019). *Caso de negocio: Implementación de metodología ágil en el proceso de mejora de requerimientos e incidencias sistémicas en una empresa de retail financiero*. Recuperado de: <https://bit.ly/2YqWR4G>
- Digital.ai (2021). 15th Annual State of Agile Report. Recuperado de: <https://digital.ai/resource-center/analyst-reports/state-of-agile-report>
- Falen, R. (2020). *Scrum en la gestión de proyectos de desarrollo de software en la empresa Innovatec, Magdalena del Mar*. <https://repositorio.ucv.edu.pe/handle/20.500.12692/49080>
- Fagarasan, C., Popa, O., Písla, A. y Cristea, C., (2021). *Waterfall and iterative approach in information technology projects*. IOP Conference Series. Materials Science and Engineering; Bristol Tomo 1169, N° 1. DOI:10.1088/1757-899X/1169/1/012025 <https://www.proquest.com/results/825D8E9F71844361PQ/1?accountid=37408>
- Flores, F., Sanhueza, V., Valdés, H. y Reyes, L. (2021). *Metodologías ágiles: un análisis de los desafíos organizacionales para su implementación*. Revista Científica, 43(1), pág. 38-49. <https://doi.org/10.14483/23448350.183321>

- Flores, M., (2020). *El marco de trabajo Scrum junto con la guía de PMBOK y su relación con la eficiencia en la gestión de proyectos de desarrollo de software para nuevas líneas de negocio MVP*.
<https://repositorio.unfv.edu.pe/bitstream/handle/20.500.13084/4336/FLORES%20TORRES%20MARCO%20ANTONIO%20-%20MAESTRIA.pdf?sequence=1&isAllowed=y>
- Forero, A. (2018). *Implementación de la Metodología SCRUM en un Ambiente Bancario*. Recuperado de <https://bit.ly/3cowZs0>
- Gaborov, M., Karuović, D., Kavalić, M., Radosav, D., Milosavljević, D., Stanisavljević, S y Bushati, J. (2021). *Comparative analysis of agile and traditional methodologies in IT project management*. Journal of Applied Technical and Educational Sciences; Budapest Tomo 11, N.º 4, (2021): 1-24. DOI:10.24368/jates.v11i4.279
<https://www.proquest.com/docview/2621872224/BADAC614949D41B8PQ/1?accountid=37408>
- Gabriel, J. (2017). *Cómo se genera una investigación científica que luego sea motivo de publicación*. Journal of the Selva Andina Research Society vol.8 no.2. La Paz.
http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S2072-92942017000200008
- Galeano, M. (2020). *Diseño de proyectos en la investigación cualitativa*. Editorial Universidad Eafit (13 – 25), 84 páginas.
<https://books.google.com.co/books?id=Xkb78OSRMI8C&printsec=frontcover#v=onepage&q&f=false>
- Gamarra, G., Wong, F., Rivera, T. y Pujay, O. (2015). *Estadística e investigación con aplicación de SPSS*. Lima: San Marcos.
<http://librodigital.sangregorio.edu.ec/librosusqp/B0031.pdf>
- García, J. (2016). *Metodología de la investigación para Administradores*. Bogotá: Ediciones de la U.
- Gómez, M., Cervantes, J. y González, P. (2012). *Administración de proyectos: notas de curso*. Segunda edición. Editada por: Universidad Autónoma Metropolitana, pág. 7 -14, 169 páginas.
<http://ilitia.cua.uam.mx:8080/jspui/bitstream/123456789/169/1/620%20->

[%20G%c3%b3mez%20Fuentes%20Mar%c3%ada%20del%20Carmen%2c%20Cervantes%20Jorge%20y%20Gonz%c3%a1lez%20Pedro%20Pablo.pdf](#)

- Guzmán, J. (2019). *Implementación del enfoque ágil y la mejora en los proyectos de desarrollo de software en una empresa de Telecomunicaciones*. <https://repositorio.urp.edu.pe/handle/20.500.14138/2425>
- Hayes, B. (1999). *Cómo medir la satisfacción del cliente: desarrollo y utilización de cuestionarios*. Segunda edición. España
- Hernández, L. y Bravo, B. (2020). *Diseño de un procedimiento para la gestión ágil de proyectos de desarrollo de software alineados a la guía del PMBOK*. Revista Ibérica de Sistemas y Tecnologías de Información; Lousada N.º E32. <https://www.proquest.com/docview/2452331778?parentSessionId=VDEVpb4rvWYy8anS0bHxeHOpqais7CloPZIITPkIpbQ%3D>
- Hernández, R., Fernández, C., Baptista, L., Méndez, S. y Mendoza, C. (2014). *Metodología de la investigación*. sexta edición. Editorial Mc Graw Hill / Interamericana. (p. 173 - 240). 634 páginas. Recuperada de <https://www.uca.ac.cr/wp-content/uploads/2017/10/Investigacion.pdf>
- Highsmith, J. (2002). *Agile Software Development Ecosystems, Problems, Practices, and Principles*. Primera edición, Editorial: Addison-Wesley Professional.
- Icart, T. y Pulpón, A. (2012). *Cómo elaborar y presentar un proyecto de investigación, una tesina y una tesis*. Barcelona: Universidad de Barcelona. <http://www.publicacions.ub.edu/refs/indices/07632.pdf>
- ISO/IEC. (1999). *ISO/IEC 9000:2000 Quality management systems -- Fundamentals and vocabulary*. Geneva, Switzerland: International Organization for Standardization. <https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-4:v1:en>
- International Data Corporation (2019). *Everis ágil: estudio de la agilidad en América Latina por IDC*. Everis an NTT Data Company. Recuperado de <https://www.everisnewhumanera.com/estudioagileesp>
- James, M. y Walter, L. (2017). *Scrum Reference Card*. CollabNet, Inc. Recuperado de: <https://scrumreferencecard.com/ScrumReferenceCard.pdf>

- Marcal, A., Furtado, F. y Belchior, A. (2007). *Mapping CMMI Project Management Process Areas to Scrum Practices*. IEEE, 31st IEEE Software Engineering Workshop.
<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.965.3329&rep=rep1&type=pdf>
- Martin, J. (1991). "Rapid Application Development". Editorial Macmillan Inc., New York.
- Merizalde, M. (2018). *Aplicación de la Metodología Scrum en la Gestión y Desarrollo de Proyectos. Caso de Estudio: Empresas Consultoras de Software de Guayaquil*. Recuperado de: <https://bit.ly/3IKCB6j>
- Morán, A. (2015). *Managing Agile: Strategy, implementation, organization and people*. Springer. International. Editorial: Publishing Switzerland. (p. 1 - 33). 266 páginas.
- Morán, G. y Alvarado, D. (2010). *Métodos de investigación*. (3ra. ed.) México: Pearson Education. (Pag. 42)
- Morandini M., Coleti T., Oliveira E. y Corrêa P. (2021). *Considerations about the efficiency and sufficiency of the utilization of the Scrum methodology: A survey for analyzing results for development teams*. Recuperado de: <https://bit.ly/3GPI30T>
- Moura, R., Carneiro, T. y Diniz, B. (2018). *Influence of the project manager's personal characteristics on project performance*. Gestao e Producao [en línea], vol. 25, no. 4, pp. 751-763. DOI 10.1590/0104-530X3595-16. Disponible en: <https://doi.org/10.1590/0104-530X3595-16>
- Ñaupas, H., Mejía, E., Novoa, E. y Villagómez, A. (2014). *Metodología de la investigación*. Cuarta edición. Ediciones de la U.
<http://librodigital.sangregorio.edu.ec/librosusgp/B0028.pdf>
- Palomino, J., Peña, J., Zevallos, G. y Orizano, L. (2015). *Metodología de la investigación*. Lima: Editorial San Marcos.
- Pressman, R. (2010). *Ingeniería del Software, un enfoque práctico*. Séptima edición. D.F.: McGraw-Hill Interamericana Editores, S.A.
- Ramos, D., Noriega, R., Laínez, J. y Durango, A. (2017). *Curso de Ingeniería de Software*. Segunda edición. Recuperado de:

<https://books.google.com.pe/books?id=G2Q4DgAAQBAJ&printsec=frontcover#v=onepage&q&f=false>

- Reyna, B. y Benjamín, D. (2018). *Modelo Integrado de Scrum y CMMI para la eficacia de la desviación del tiempo en el Ciclo de Vida de los Proyectos de Ingeniería de Software de la Consultora Assembly Solution SAC*.
https://repositorio.upeu.edu.pe/bitstream/handle/20.500.12840/1467/Benjamin_Tesis_Maestro_2018.pdf?sequence=5&isAllowed=y
- Ríos, R. (2017). *Metodología para la Investigación y Redacción*. Servicios Académicos Intercontinentales S.L.
- Rivera, N. (2021). *Metodologías Ágiles aplicables a la administración pública: Análisis de aplicabilidad de las metodologías ágiles en la administración pública*. (Proyecto del consejo federal de inversiones, Argentina).
<http://biblioteca.cfi.org.ar/wp-content/uploads/sites/2/2021/06/2-metodologias-agiles-en-el-sector-publico-informe-final.pdf>
- Rivera, O. (2017). *Propuesta de una metodología de gestión de la calidad que apoye a la administración de proyectos de software en la empresa Avantica Technologies*. <https://n9.cl/3ut79>
- Romero, D. (2017). *Modelo de mejora de procesos basado en la implementación de la estrategia CRM y la metodología SCRUM (MOPICS). Caso de estudio: Sector Consultoría en Tecnologías de la Información como Outsourcing*. Instituto Politécnico Nacional.
<https://tesis.ipn.mx/bitstream/handle/123456789/26481/T1912.pdf?sequence=1&isAllowed=y>
- Ruvalcaba, M. (2020). *Procesos de Software*. [en línea], Disponible en: <https://sg.com.mx/revista/1/procesos-software>
- Saeedi, K. y Visvizi, A. (2021). *Software Development Methodologies, HEIs, and the Digital Economy*. Education Sciences [en línea], vol. 11, no. 73. Disponible en: <https://files.eric.ed.gov/fulltext/EJ1288479.pdf>
- Sánchez, H., (2019). *Aplicación de las metodologías ágiles en la gestión de la información y comunicación en proyectos de reforma en el sector retail*. (Tesis Doctoral, Universidad Politécnica de Madrid).
https://oa.upm.es/54778/1/HUGO_SANCHEZ_VICENTE_01.pdf

- Satpathy, T. (2017). *Una guía para el cuerpo de conocimiento de Scrum*. (Guía SBOK™) 3ra Edición. Avondale, Arizona 85392 USA: SCRUMstudy™, una marca de VMEdU, Inc.
- Spiegel, M. y Stephens, L. (2009). *Estadística de Shawm*. Cuarta edición. México: Mc Graw Hill.
- Srivastava, A., Bhardwaj, S. y Saraswat, S. (2017). *SCRUM model for agile methodology*. International Conference on Computing, Communication and Automation (ICCCA), 2017. doi:10.1109/CCAA.2017.8229928.
- Takeuchi, H. y Nonaka, I. (1986). *The New Product Development Game*. Harvard Business Review. <https://hbsp.harvard.edu/product/86116-PDF-ENG>
- Useche, M., Artigas, W., Queipo, B. y Perozo, E. (2019). *Técnicas e Instrumentos de Recolección de Datos Cual-cuantitativos*. Primera edición. Universidad de la Guajira. Colombia.
- Vara, A. (2015). *7 pasos para elaborar una tesis. Cómo elaborar y asesorar una tesis para Ciencias Administrativas, Finanzas, Ciencias Sociales y Humanidades*. Lima: Universidad San Martín de Porres. <https://www.administracion.usmp.edu.pe/investigacion/files/7-PASOS-PARA-UNA-TESIS-EXITOSA-Desde-la-idea-inicial-hasta-la-sustentaci%C3%B3n.pdf>
- Villavicencio, J. (2017). *Modelo de gestión para el control de cambios y la dirección de proyectos basados en ITIL y metodologías ágiles dentro de la provisión de servicios de un departamento de desarrollo de software*. Tesis Universidad de Cuenca. Ecuador. <http://dspace.ucuenca.edu.ec/bitstream/123456789/28422/1/Trabajo%20de%20titulaci%C3%B3n.pdf>
- Webster, A. (2000). *Estadística aplicada a los negocios y la economía*. (3ra ed.) Colombia: Mc Graw Hill. (Pag. 9)
- White, H. y Sabarwal, S. (2014). *Diseño y métodos cuasiexperimentales*, Síntesis metodológica: evaluación de impacto n.º 8, Centro de Investigaciones de UNICEF, Florencia. <https://www.unicef-irc.org/publications/pdf/MB8ES.pdf>
- Woodward, E., Surdek, S. y Ganis, M. (2010). *A Practical Guide to Distributed Scrum*. Editorial: IBM Press.

ANEXOS

Anexo 1: Matriz de consistencia

Título: Marco de Trabajo Scrum para la Gestión de Proyectos de Software en una Empresa Consultora, 2023						
Problema	Objetivo	Hipótesis	Variable 1: Marco de Trabajo Scrum			
			Variable 2: Proyectos de software			
			Dimensiones	Indicadores	Instrumentó	Escala
<p>Problema General: ¿Cómo influye el marco de trabajo Scrum en la gestión de los proyectos de software en una empresa consultora?</p>	<p>Objetivo General: Determinar la influencia del marco de trabajo Scrum en la gestión de proyectos de software en una empresa consultora.</p>	<p>Hipótesis General: El marco de trabajo Scrum influye positivamente en la gestión de los proyectos de software en una empresa consultora.</p>	Calidad	Cantidad de Fallas	Ficha de Registro	de Razón
<p>Problema Específico 1: ¿Cómo influye el marco de trabajo Scrum en la mejora de la calidad del producto de los proyectos de software en una empresa consultora?.</p>	<p>Objetivo Específico 1: Determinar la influencia del Marco de trabajo Scrum en la mejora de la calidad del producto de los Proyectos de Software en una empresa consultora.</p>	<p>Hipótesis Especifica 1: El Marco de trabajo Scrum influye positivamente en la calidad del producto de los proyectos de software en una empresa consultora.</p>	Tiempo de entrega	Tiempo de entrega del producto	Ficha de Registro	de Razón
<p>Problema Específico 2: ¿Cómo influye el marco de trabajo Scrum en la disminución del tiempo de entrega del producto de los proyectos de software en una empresa consultora?.</p>	<p>Objetivo Específico 2: Determinar la influencia del Marco de trabajo Scrum en la disminución del tiempo de entrega del producto de los Proyectos de Software en una empresa consultora.</p>	<p>Hipótesis Especifica 2: El Marco de trabajo Scrum influye positivamente en el tiempo de entrega del producto en los proyectos de software en una empresa consultora.</p>	Solicitudes de Cambio	Cantidad de Solicitudes de Cambio	Ficha de Registro	de Razón
<p>Problema Específico 3: ¿Cómo influye el marco de trabajo Scrum en la disminución de las solicitudes de cambio al producto de los proyectos de software en una empresa consultora?.</p>	<p>Objetivo Específico 3: Determinar la influencia del Marco de Trabajo Scrum en la disminución de las solicitudes de cambio al producto de los Proyectos de Software en una empresa consultora.</p>	<p>Hipótesis Especifica 3: El Marco de trabajo Scrum influye positivamente en la cantidad de solicitudes de cambio al producto de los proyectos de software en una empresa consultora.</p>				

Método y Diseño	Población y muestra	Técnicas e instrumentos	Método de Análisis de datos
<p>Enfoque: Cuantitativo</p> <p>Tipo de Investigación: Aplicada</p> <p>Diseño de investigación: Cuasi Experimental</p> <p>De Tipo: Cuasi Experimental con Único Post-test con Grupo de Control</p>	<p>Población: Los Proyectos Ejecutados en la empresa Consultora.</p> <p>Tipo de Muestreo: No Aplica.</p> <p>Tamaño de la Muestra: 50 Proyectos.</p>	<p>Variable 2: Gestión de Proyectos de Software</p> <p>Técnica: Observación</p> <p>Instrumento: Ficha de Registro</p> <p>Ámbito de aplicación: Área de Proyectos de una Empresa Consultora.</p> <p>Forma de Recopilación de Datos: Individual</p>	<p>DESCRIPTIVA:</p> <p>Se aplicaron los instrumentos de recolección de datos se recopiló los datos, luego fueron procesados para su análisis descriptivo y observar la comparación de medias y el gráfico de cajas utilizando la herramienta SPSS v27.</p> <p>INFERENCIAL:</p> <p>Para determinar la normalización de los datos se utilizó el estadístico de Kolmogórov-Smirnov validando la interpretación del gráfico de cajas, para la pruebas y validación de las hipótesis se aplicó la prueba de T-Student y la de U-Mann Whitney.</p>

Anexo 2: Matriz de operacionalización de las variables

Variables de Estudio	Definición Conceptual	Definición Operacional	Dimensión	Indicadores	Formula	Escala
Variable Dependiente: Gestión de Proyectos de Software.	Es el proceso que permite desarrollar software, mediante una serie de actividades, modelos, normas y políticas que deben de ser cumplidos por un equipo de trabajo y generar un producto (Ruvalcaba, 2020).	Con respecto a la gestión de proyectos de software será medida a través de cuestionarios	Calidad	Cantidad de Fallas del Producto	$CP_{FALL} = \frac{CF_{INT}}{CF_{TO}} \times 100$ <p> CP_{FALL} = Cantidad promedio de fallas del producto. CF_{INT} = Cantidad de fallas internas. CF_{TO} = Cantidad total de fallas. </p>	De Razón
			Tiempo de Entrega	Tiempo de Entrega del Productos	$TP_{ENT} = \frac{T_{DEM}}{T_{PLAN}} \times 100$ <p> TP_{ENT} = Tiempo promedio de entrega del producto. T_{DEM} = Tiempo de demora para la entrega del producto T_{PLAN} = Tiempo planificado para la entrega del producto </p>	De Razón
			Solicitudes de Cambio	Cantidad de Solicitudes de Cambio	$CS_{CAM} = \frac{CS_{EJE}}{CS_{TO}} \times 100$ <p> CS_{CAM} = Cantidad promedio de solicitudes de cambio. CS_{EJE} = Cantidad de solicitudes de cambio ejecutadas. CS_{TO} = Cantidad de solicitudes de cambio totales. </p>	De Razón

Anexo 3: Instrumentos para la recolección de los datos

Ficha de registro para determinar la calidad.

FICHA DE REGISTRO PARA DETERMINAR LA CALIDAD DEL PRODUCTO DE LOS PROYECTOS DE DESARROLLO DE SOFTWARE				
Investigador				
Fecha de inicio		Fecha de Fin		
PRE-TEST				
Ítem	Cantidad de Defectos Internos	Cantidad de Defectos Externos	Cantidad de Defectos Total	% Cantidad de Defectos
Entregable 1				
Entregable 2				
Entregable 3				
Entregable 4				
Entregable 5				
Entrega del Producto				
POST-TEST				
Ítem	Cantidad de Defectos Internos	Cantidad de Defectos Externos	Cantidad de Defectos Total	% Cantidad de Defectos
Entregable 1				
Entregable 2				
Entregable 3				
Entregable 4				
Entregable 5				
Entrega del Producto				

Ficha de registro para el tiempo de entrega.

FICHA DE REGISTRO PARA DETERMINAR EL TIEMPO DE ENTREGA DE LOS PRODUCTOS DE LOS PROYECTOS DE DESARROLLO DE SOFTWARE				
Investigador				
Fecha de inicio		Fecha de Fin		
PRE-TEST				
Ítem	Tiempo Planificado	Tiempo Real	Tiempo de Demora	% Tiempo de Entrega
Entregable 1				
Entregable 2				
Entregable 3				
Entregable 4				
Entregable 5				
Entrega del Producto				
POST-TEST				
Ítem	Tiempo Planificado	Tiempo Real	Tiempo de Demora	% Tiempo de Entrega
Entregable 1				
Entregable 2				
Entregable 3				
Entregable 4				
Entregable 5				
Entrega del Producto				

Ficha de registro para determinar la cantidad de cambios.

FICHA DE REGISTRO PARA DETERMINAR LA CANTIDAD DE CAMBIOS EN LOS PROYECTOS DE DESARROLLO DE SOFTWARE				
Investigador				
Fecha de inicio		Fecha de Fin		
PRE-TEST				
Ítem	Tiempo Planificado	Tiempo Real	Tiempo de Demora	% Tiempo de Entrega
Entregable 1				
Entregable 2				
Entregable 3				
Entregable 4				
Entregable 5				
Entrega del Producto				
POST-TEST				
Ítem	Tiempo Planificado	Tiempo Real	Tiempo de Demora	% Tiempo de Entrega
Entregable 1				
Entregable 2				
Entregable 3				
Entregable 4				
Entregable 5				
Entrega del Producto				

Anexo 4: Validación del instrumento de recolección de datos



Ficha de Registro para Determinar la Calidad.

FICHA DE REGISTRO PARA DETERMINAR LA CALIDAD DEL PRODUCTO DE LOS PROYECTOS DE DESARROLLO DE SOFTWARE				
Investigador				
Fecha de inicio		Fecha de Fin		
PRE TEST				
ITEM	Cantidad de Defectos Internos	Cantidad de Defectos Externos	Cantidad de Defectos Total	% Cantidad de Defectos
Entregable 1				
Entregable 2				
Entregable 3				
Entregable 4				
Entregable 5				
Entrega del Producto				
POST TEST				
ITEM	Cantidad de Defectos Internos	Cantidad de Defectos Externos	Cantidad de Defectos Total	% Cantidad de Defectos
Entregable 1				
Entregable 2				
Entregable 3				
Entregable 4				
Entregable 5				
Entrega del Producto				

Ficha de Registro para el Tiempo de Entrega.

FICHA DE REGISTRO PARA DETERMINAR EL TIEMPO DE ENTREGA DE LOS PRODUCTOS DE LOS PROYECTOS DE DESARROLLO DE SOFTWARE				
Investigador				
Fecha de inicio		Fecha de Fin		
PRE TEST				
ITEM	Tiempo Planificado	Tiempo Real	Tiempo de Demora	% Tiempo de Entrega
Entregable 1				
Entregable 2				
Entregable 3				
Entregable 4				
Entregable 5				
Entrega del Producto				
POST TEST				
ITEM	Tiempo Planificado	Tiempo Real	Tiempo de Demora	% Tiempo de Entrega
Entregable 1				
Entregable 2				
Entregable 3				
Entregable 4				
Entregable 5				
Entrega del Producto				

Ficha de Registro para Determinar la Cantidad de Cambios.

FICHA DE REGISTRO PARA DETERMINAR LA CANTIDAD DE CAMBIOS EN LOS PROYECTOS DE DESARROLLO DE SOFTWARE				
Investigador				
Fecha de inicio		Fecha de Fin		
PRE TEST				
ITEM	Tiempo Planificado	Tiempo Real	Tiempo de Demora	% Tiempo de Entrega
Entregable 1				
Entregable 2				
Entregable 3				
Entregable 4				
Entregable 5				
Entrega del Producto				
POST TEST				
ITEM	Tiempo Planificado	Tiempo Real	Tiempo de Demora	% Tiempo de Entrega
Entregable 1				
Entregable 2				
Entregable 3				
Entregable 4				
Entregable 5				
Entrega del Producto				

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO:

VARIABLE: Gestión de Proyectos de Software.

Nº	INDICADORES	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Sí	No	Sí	No	Sí	No	
1	Cantidad de Fallas del Producto							
2	Tiempo de Entrega del Producto							
3	Cantidad de Solicitudes de Cambio							

Observaciones (precisar si hay suficiencia): _____

Opinión de aplicabilidad: Aplicable [] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador. D^o/ Mg: ACUÑA BENITES MARLON FRANK DNI: 42097456

Especialidad del validador:

Lima, 17 de octubre del 2022

¹Pertinencia: El ítem corresponde al concepto teórico formulado.
²Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo
³Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo
 Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

 Firma del Experto Informante.

Anexo 5: Base de datos

Muestra de control			
Ítem	% Cantidad de defectos	% Tiempo de entrega	% Solicitudes de cambio
Proyecto P1	64	42.86	66.67
Proyecto P2	68.92	50	66.67
Proyecto P3	75	57.14	85.71
Proyecto P4	66.28	30	84.62
Proyecto P5	69.23	31.25	90
Proyecto P6	69.41	50	80
Proyecto P7	65	46.67	69.23
Proyecto P8	71.19	27.27	86.67
Proyecto P9	73.33	38.89	66.67
Proyecto P10	67.95	66.67	68.42
Proyecto P11	73.91	50	92.31
Proyecto P12	74.58	40	87.5
Proyecto P13	72.73	45	80
Proyecto P14	62.5	53.33	75
Proyecto P15	66.67	36.36	83.33
Proyecto P16	69.74	44.44	68.42
Proyecto P17	64.71	50	78.95
Proyecto P18	72.06	50	70
Proyecto P19	69.23	30	73.68
Proyecto P20	66.22	47.06	71.43
Proyecto P21	68.42	56.25	56.25
Proyecto P22	66.67	44.44	86.67
Proyecto P23	67.07	60	75
Proyecto P24	66.28	41.18	76.92
Proyecto P25	75.38	40	78.95

Muestra experimental			
Item	% Cantidad de defectos	% Tiempo de entrega	% Solicitudes de cambio
Proyecto P26	54.55	12.5	33.33
Proyecto P27	58.33	6.25	33.33
Proyecto P28	55.56	8.33	50
Proyecto P29	50	10	50
Proyecto P30	53.33	5.56	50
Proyecto P31	50	7.14	33.33
Proyecto P32	58.33	16.67	50
Proyecto P33	50	12.5	50
Proyecto P34	57.14	10.53	66.67
Proyecto P35	54.55	13.64	60
Proyecto P36	50	7.14	50
Proyecto P37	54.55	11.76	50
Proyecto P38	50	5.88	33.33
Proyecto P39	55.56	10	66.67
Proyecto P40	54.55	10.53	50
Proyecto P41	58.33	6.25	33.33
Proyecto P42	50	9.09	33.33
Proyecto P43	55.56	8.33	66.67
Proyecto P44	50	7.14	50
Proyecto P45	50	6.25	50
Proyecto P46	50	13.33	50
Proyecto P47	53.85	12.5	66.67
Proyecto P48	50	5	50
Proyecto P49	57.14	5.88	33.33
Proyecto P50	55.56	5.26	50

Anexo 6: Carta de presentación



UNIVERSIDAD CÉSAR VALLEJO



"Año del Fortalecimiento de la Soberanía Nacional"

Lima, 18 de octubre de 2022
Carta P. 1036-2022-UCV-VA-EPG-F01/J

Ingeniero
Pedro Abelardo Galindo Castro
Jefe de Proyectos
Domain Consulting S.A.C

De mi mayor consideración:

Es grato dirigirme a usted, para presentar a Gutierrez Estrella, John Williams; identificado con DNI N° 10146269 y con código de matrícula N° 7002517263; estudiante del programa de MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN quien, en el marco de su tesis conducente a la obtención de su grado de MAESTRO, se encuentra desarrollando el trabajo de investigación titulado:

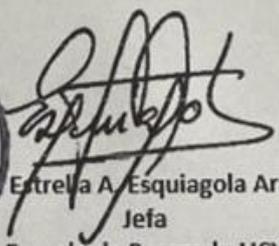
Marco de Trabajo Scrum para la Gestión de Proyectos de Software en una Empresa Consultora, Lima 2023.

Con fines de investigación académica, solicito a su digna persona otorgar el permiso a nuestro estudiante, a fin de que pueda obtener información, en la institución que usted representa, que le permita desarrollar su trabajo de investigación. Nuestro estudiante investigador Gutierrez Estrella, John Williams asume el compromiso de alcanzar a su despacho los resultados de este estudio, luego de haber finalizado el mismo con la asesoría de nuestros docentes.

Agradeciendo la gentileza de su atención al presente, hago propicia la oportunidad para expresarle los sentimientos de mi mayor consideración.

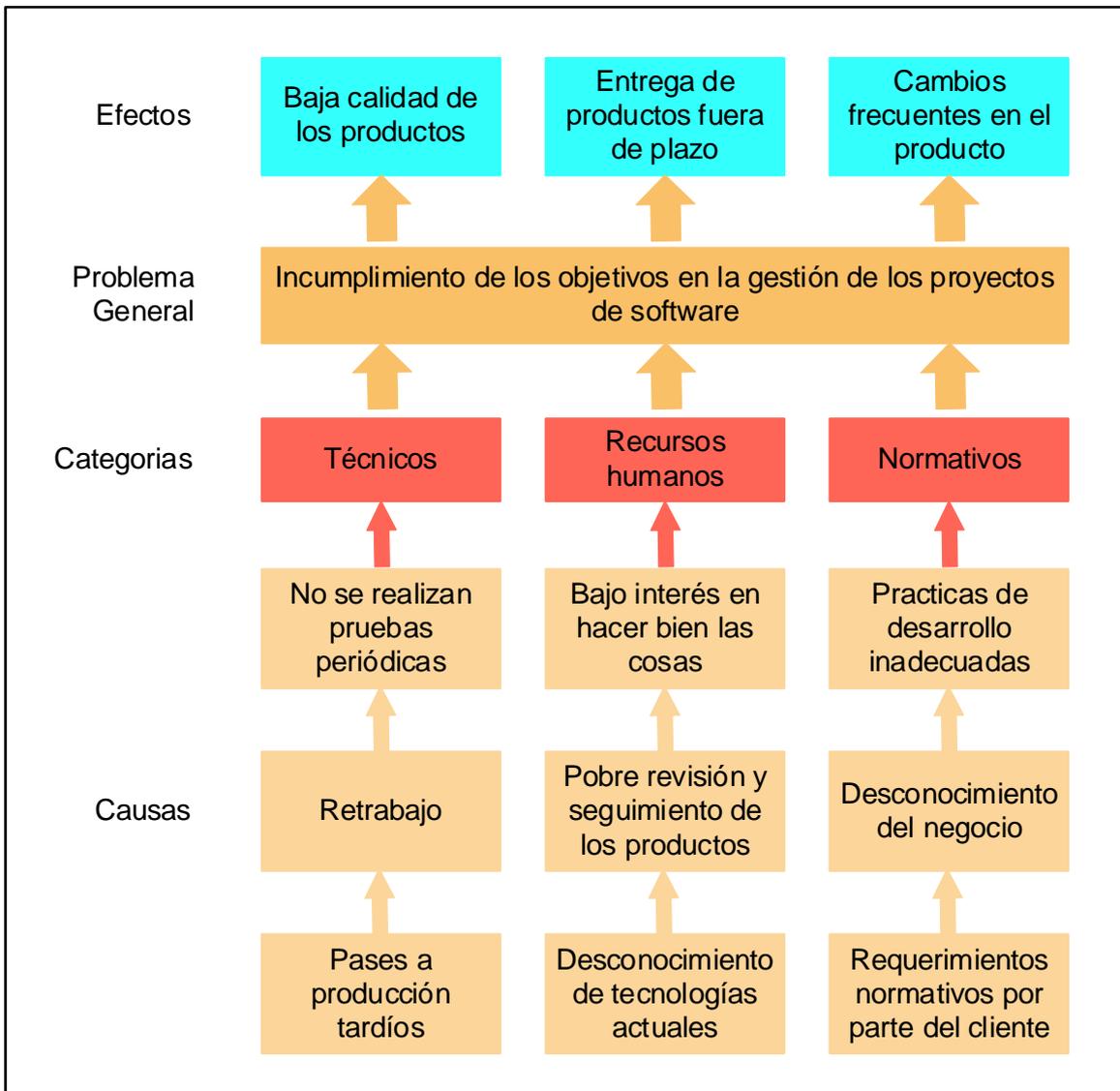
Atentamente,




Dra. Estrella A. Esquiagola Aranda
Jefa
Escuela de Posgrado UCV
Filial Lima Campus Los Olivos


Pedro Galindo
18/10/2022

Anexo 7: Árbol de problemas



Anexo 8: Metodología de gestión utilizada

Proceso de Gestión de Proyectos de Software Utilizando el Marco de trabajo Scrum

I. Introducción

En la actualidad en la consultora, existen procesos que no están sistematizados y que en muchos casos carecen de documentación, por lo que fue necesario iniciar la elaboración de un modelo de gestión para la ejecución de los proyectos de software, a fin de ser utilizado en la implementación de los proyectos de software.

El presente modelo propuesto pretende integrar el marco de trabajo Scrum en el modelo utilizado actualmente, creando un modelo formal y ágil, para el desarrollo de software, que permita al equipo desempeñarse de manera colaborativa y elaborar entregables de valor.

II. Objetivo

Proponer un modelo basado en el marco de trabajo Scrum, para el desarrollo de proyectos de software, con la finalidad de entregar productos y servicios de software de valor.

III. Metodología

El proceso se fundamenta en los principios y prácticas del marco de trabajo Scrum, proponiendo un flujo continuo de actividades, que permita una gestión ágil de los proyectos de desarrollo de software en la empresa consultora. Previo a la descripción de las etapas se definen los roles del proceso.

3.1. Roles

Cliente/Usuario	<p>Especifica los requerimientos.</p> <ul style="list-style-type: none">• Solicita y financia el desarrollo de software.• Cuando el desarrollo se libera, toma el rol de usuario.• Puede ser externo a la organización o pertenecer a ella.
Product Owner	<p>Es el responsable del producto.</p> <ul style="list-style-type: none">• Debe poseer una visión clara de las necesidades del negocio.• Encargado de priorizar los requerimientos y definir las funcionalidades.• Inspecciona los entregables, garantizando que se cumplan los criterios de aceptación.• Debe identificar los riesgos que puedan presentarse y apoyar a que estos sean tratados, con la finalidad de que no impacten en el entregable final.• Es la voz del cliente.
Scrum Master	<p>Es el facilitador.</p> <ul style="list-style-type: none">• Es el responsable de orientar al equipo de desarrollo en el uso de los principios y prácticas de Scrum, ejerciendo las funciones de facilitador, ayudando al equipo de desarrollo a eliminar los impedimentos que se puedan presentar y eliminando todo aquello que no genere valor.• Asimismo, es el responsable de garantizar que el equipo de desarrollo tenga todos los recursos necesarios para la realización de sus actividades y de evitar distracciones al equipo de desarrollo.• Coordina las reuniones con el cliente y con el Product Owner.• Buena relación y comunicación con el cliente.

Scrum Team	<p>Equipo de desarrollo.</p> <ul style="list-style-type: none"> • Equipo formado por especialistas con conocimientos y habilidades que permitan llevar a cabo la visión del responsable del producto. • Equipo multidisciplinario y autoorganizado. • El equipo debe estar formado por un mínimo de 3 personas y un máximo de 9.
------------	---

3.2. Reuniones

Reunión de inicio	<p>Realizado entre el cliente, Scrum Master y Product Owner.</p> <ul style="list-style-type: none"> • Se recogen los requerimientos del cliente y se detallan los mismos. • Se crea el Product Backlog • Time-box de 2 a 4 horas.
Reléase Planning	<p>Realizado entre el Product Owner, Scrum Master y el equipo de trabajo.</p> <ul style="list-style-type: none"> • Se crean las historias de usuario. • Se definen los criterios de aceptación. • Time-box de 2 horas.
Sprint Planning	<p>Realizado el Scrum Master y el equipo de trabajo.</p> <ul style="list-style-type: none"> • Se identifican y definan las actividades específicas. • Se estiman los tiempos y recursos de cada actividad. • Se crea el Sprint Backlog. • Time-box de 8 horas (Sprint de un mes).
Daily Scrum (Reunión diaria)	<p>Realizado entre el Scrum Master y equipo de desarrollo.</p> <ul style="list-style-type: none"> • Se indica que actividades se realizó, que actividades se realizaran y cuáles han sido los obstáculos que se han presentado y han impedido la realización de las actividades. • La reunión debe ser realizada todos los días y a la misma hora.

	<ul style="list-style-type: none"> • Time-box de 15 minutos.
Sprint Review	<p>Realizado entre el equipo de desarrollo y el cliente.</p> <ul style="list-style-type: none"> • Se muestra, revisa y valida el producto culminado en el Sprint. • Time-box de 4 horas (Sprint de un mes).
Sprint Retrospective	<p>Realizado entre todo el equipo Scrum.</p> <ul style="list-style-type: none"> • Se reflexiona sobre lo bueno y malo realizado en el Sprint, para la mejora continua. • Time-box de 4 horas (Sprint de un mes).
Reunión de cierre	<p>Realizado con el cliente.</p> <ul style="list-style-type: none"> • Se presenta al cliente el proyecto en su totalidad. • Se firma el acta de cierre del proyecto. • Time-box de 2 a 4 horas.
Retroalimentación.	<p>Realizado entre todo el equipo Scrum.</p> <ul style="list-style-type: none"> • Se revisa el cumplimiento del proyecto con los objetivos establecidos inicialmente. • Se establecen las mejoras para aplicarlos en proyectos futuros. • Se documentan las lecciones aprendidas. • Time-box de 2 a 4 horas.

3.3. Entregables

El modelo propuesto genera los siguientes entregables:

- Visión inicial del proyecto
- Project Charter
- Product Backlog
- Sprint Backlog
- Plan del proyecto
- Registro de impedimentos
- Acta de mejoras aceptadas

- Acta de aceptación del entregable
- Acta de Cierre
- Lecciones aprendidas

3.4. Etapas

Este modelo propuesto presenta cinco etapas:

Inicio	<p>En esta etapa participan el cliente, el Product Owner, Scrum Master y los stakeholders.</p> <ul style="list-style-type: none"> • En esta etapa se toma la visión del proyecto y en una reunión de inicio en conjunto con el cliente se identifican y establecen los requerimientos que el producto debe tener. • El Scrum Master guía la reunión y deberá estar en comunicación constante con el cliente. • El Scrum Master deberá elaborar el Project Chárter del proyecto, documento que formaliza la iniciación del proyecto. • Se recomienda que el Product Owner y Scrum Master firmen el Project Charter. • El Product Owner define el Product Backlog.
Planificación	<p>Priorizar la lista de requerimientos o backlog (Product Backlog priorizado)</p> <ul style="list-style-type: none"> • La lista de requerimientos o backlog del producto, es una lista de todas las peticiones de los usuarios, que permita plasmar la visión del producto. La lista es dinámica, se actualiza constantemente de acuerdo a los nuevos requerimientos que se presentan. • La lista de requerimientos o producto backlog es única y representa todo el trabajo que el equipo de desarrollo deberá de realizar, el orden de estos requerimientos es

priorizado por el responsable del producto (Product Owner).

Estimar realmente la lista de requerimientos que están pendientes

- Es relevante que el equipo de desarrollo estime el esfuerzo real y necesario que les demandara cada requerimiento.
- Previa a la estimación se debe tener muy en cuenta toda la información existente de cada requerimiento que se va a estimar. Además, el requerimiento debe de ser lo suficientemente pequeño para poder realizar una estimación real y que permita realizar una definición de terminado (DoD) y que todos los involucrados estén de acuerdo en lo que se tiene que cumplir para considerar que el requerimiento se encuentra “terminado”.
- Es muy importante que la técnica de estimación es criterio de cada equipo Scrum.

Planificación adecuada de los Sprints (Sprint Planning)

- La reunión de planificación es para todo el equipo Scrum (el responsable del producto, el scrum master y el equipo scrum).
- Los sprints tienen una duración máxima de 4 semanas, de preferencia debe durar entre 1 a 2 semanas para una mejor flujo y retroalimentación (la velocidad del equipo es relevante para la duración del Sprint).
- Se debe de seleccionar que elementos del Product Backlog se van a desarrollar en cada Sprint, esta priorización debe ser definida por el responsable del producto.
- Los involucrados deben estar de acuerdo con lo que se va a conseguir al final de cada Sprint, de acuerdo al objetivo trazado, evitando falsas expectativas.

	<ul style="list-style-type: none"> • Es muy importante tener muy claro que, al finalizar la planificación del sprint, este ya no puede ser modificado. <p>Estimar las fechas de entrega del producto (Release Planning).</p> <ul style="list-style-type: none"> • Elaborar una propuesta de lanzamiento a nivel gerencial, para que se tenga una hoja de ruta y que el responsable del producto tenga presente las fechas de implementación y entrega. • La propuesta de lanzamiento es variable en el tiempo, de acuerdo a como se va desarrollando el Sprint.
Implementación y Seguimiento	<p>El trabajo realizado por el equipo de desarrollo, debe hacerse visible.</p> <ul style="list-style-type: none"> • La información del proyecto debe ser compartida de forma pública a todos los interesados del proyecto. • Para compartir la información, se debe hacer uso de radiadores de información, como tableros Kanban o simplemente en pizarras o en paredes. • El tablero Kanban debe estar conformado como mínimo por 3 columnas. La lista de requerimientos con sus 3 estados (pendiente, en proceso y hecho) y la los impedimentos a la derecha. Asimismo, se puede considerar alguna información adicional a criterio del Scrum Máster, Product Owner y Equipo Scrum. • El tablero Kanban se deberá actualizar de forma diaria, de acuerdo a la información obtenida de las reuniones diarias (Daily Scrum) o cuando algún miembro del equipo tenga una actualización. <p>Reuniones Diarias (Daily Scrum).</p> <ul style="list-style-type: none"> • Previo a la realización de las reuniones, se deben de establecer las reglas para el desarrollo de la reunión.

	<ul style="list-style-type: none">• Asimismo, se define que las reuniones no pueden superar los 15 minutos de duración y deben de iniciar a la misma hora todos los días, debiendo asistir todos los miembros de manera puntual.• En la reunión diaria el equipo responde a 3 preguntas: ¿Qué hiciste ayer para ayudar al equipo? ¿Qué vas a hacer mañana para ayudar al equipo? ¿Qué impedimentos se han presenta que dificulten tu avance?• Es responsabilidad del Scrum Master garantizar que el equipo de desarrollo avance de manera fluida, eliminando cualquier impedimento. <p>Demostración del entregable culminado en el Sprint (Sprint Review).</p> <ul style="list-style-type: none">• Se presenta el entregable culminado en el Sprint al responsable del producto y a los interesados. También podrían participar el Scrum Master y otros directivos de la empresa.• En la reunión el equipo de desarrollo explica todos los requerimientos que se culminaron en el Sprint y establecer si está terminado. El equipo de desarrollo solo explicara lo que se alcance en la definición de Terminado (DoD).• El principal alcance de la reunión es la retroalimentación de lo realizado en el sprint, para la mejora del producto y entregar valor al cliente o usuario.
Cierre	<p>Reuniones de retrospectiva (Sprint Retrospective).</p> <ul style="list-style-type: none">• Luego de la demostración de los resultados en el Sprint Review y al culminar el proyecto, todo el equipo se reúne con la finalidad de reflexionar sobre lo realizado en el último Sprint e identificar lo que no resulto como

	<p>se esperaba y que se podría hacer para mejorar en corto plazo (la mejora se aplica en el siguiente Sprint).</p> <ul style="list-style-type: none"> • En esta reunión es primordial que los miembros del equipo asuman su responsabilidad de sus actividades y resultados, tratando de encontrar soluciones y mejorar en equipo. <p>Las mejoras se implementan en el siguiente sprint.</p> <ul style="list-style-type: none"> • Cerrado el Sprint, se inician con la planificación del sprint siguiente, tomando en cuenta las mejoras identificadas en el Sprint anterior.
--	---

3.5. Proceso

Crear Project Charter. Se revisa el caso de negocio del proyecto, con la finalidad de crear los objetivos, alcances e identificar los riesgos. Asimismo, se identifica al Product Owner, Scrum Master, el equipo de desarrollo y los stakeholders. El Product Owner debe definir el Project Charter en conjunto con el Scrum Master. En el ítem formatos, se muestra la plantilla propuesta.

Crear Product Backlog. Se identifica y lista los requerimientos y/o funcionalidades generales, los cuales son priorizados para crear el Product Backlog priorizado. Es importante considerar que el Product Backlog siempre está actualizándose a medida que se va desarrollando el producto. En el ítem formatos, se muestra la plantilla propuesta.

Crear historias de usuario. Se crean las historias de usuario, que son requisitos puntuales de los usuarios, también se definen los criterios de aceptación por parte del Product Owner. La historia de usuario está conformada por tres elementos sobre el requisito: ¿Quién? ¿Qué? y ¿Por qué? Los requisitos expresados en las historias de usuario definiciones breves, y sencillas de entender. Los criterios de aceptación deben estar en concordancia a las historias de usuario y deben ser entendibles para todos,

esto permitirá una entrega eficaz y oportuna de las funcionalidades definidas. En el ítem formatos, se muestra la plantilla propuesta.

Especificar tareas. Las historias de usuario son desagregadas en tareas específicas por parte del equipo de desarrollo, considerando que el tiempo máximo para ejecutar cada tarea o debe ser mayor a 4 horas. Caso contrario deberá subdividirse la tarea. En el ítem formatos, se muestra la plantilla propuesta.

Estimación y asignación de tareas. El equipo de desarrollo estimara el esfuerzo necesario para culminar una tarea, para luego asignarse entre todos las tareas, comprometiendo a culminarlas en el Sprint. Como recomendación utilizar el formato horas como unidad de tiempo.

Crear Sprint Backlog. El equipo de desarrollo establece el número de tareas que se realizarán y que deberán ser completadas en los diferentes Sprints. En el ítem formatos, se muestra la plantilla propuesta.

Plan del proyecto. Se determina el cronograma de actividades, a partir de las historias de usuario priorizadas en el Product Backlog, así como las tareas correspondientes.

Desarrollar tareas. El equipo de desarrollo realiza las tareas planificadas para el Sprint correspondiente. El seguimiento a las tareas, se realiza utilizando un Scrumboard. En el ítem formatos, se muestra la plantilla propuesta.

Realizar Daily Scrum. El Scrum Master inicia la reunión diaria, de preferencia de pie y a la misma hora todos los días, para que los integrantes del equipo se sincronicen con respecto a sus actividades realizadas, así como de los impedimentos presentados. En caso de presentarse impedimentos, el Scrum Master debe tomar nota del mismo en un registro de impedimentos. En el ítem formatos, se muestra la plantilla propuesta.

Validar Sprint. El equipo de desarrollo el producto entregado en el Sprint al Product Owner y al cliente, esta entrega debe de ser supervisado por el Scrum Master. Se debe de validar que se cumple con los criterios de aceptación y obtener la aprobación y aceptación del Product Owner. Asimismo, se podrían revisar las prioridades de las actividades, así como cualquier cambio o actualización al Product Backlog. En el ítem formatos, se muestra la plantilla propuesta.

Retrospectiva del Sprint. El Scrum Master y el equipo de desarrollo reflexionan sobre lo aprendido durante el Sprint concluido. El principal objetivo es identificar lo que se ha hecho mal, para mejorar y aplicarlo en los siguientes Sprint. En el ítem formatos, se muestra la plantilla propuesta.

Cierre. Se entrega el producto aceptado dándose por terminado el proyecto, procediendo a elaborar el acta de cierre.

IV. Formatos

Formato visión del proyecto
<ol style="list-style-type: none">1. Fecha de elaboración del documento2. Título del proyecto:3. Siglas del proyecto:4. Visión del Proyecto La declaración de la visión del proyecto (Project Vision Statement) no debe ser demasiado específica y debe dejar espacio a la flexibilidad. Es posible que el conocimiento actual sobre el proyecto esté basado en suposiciones cambien conforme avanza el proyecto, por lo que es importante que la visión del proyecto sea lo suficientemente flexible como para adaptarse a estos cambios. La visión del proyecto debe centrarse en el problema y no en la solución.
<p><i>Ejemplo:</i></p> <p>VMFoods, una cadena de supermercados, quiere expandirse con un portal de comercio electrónico en línea y se ha comunicado con su empresa para crear el producto.</p> <p>Visión del proyecto: Desarrollar una herramienta fácil de usar y un canal de ventas en línea para VMFoods que sea estéticamente agradable.</p>

Formato de acta de constitución del proyecto

1. Fecha de elaboración del documento

2. Título del proyecto:

3. Siglas del proyecto:

4. Gerente del proyecto

Nombre completo
¿Quién es su jefe inmediato?
¿A qué organización pertenece?

5. Nivel de autoridad del Gerente de Proyecto

Capacidad de decisión sobre el personal del proyecto
Gestión y variación del presupuesto
Intervención sobre decisiones técnicas del proyecto
Resolución de conflictos

6. Iniciador o Patrocinador del proyecto

Nombre completo
Nivel de autoridad
¿Quién es su jefe inmediato?
¿A qué organización pertenece?

7. Cliente del proyecto

Nombre completo
Nivel de autoridad
¿Quién es su jefe inmediato?
¿A qué organización pertenece?

8. Descripción del proyecto

¿De qué se trata el proyecto?
¿Cuál es el producto o resultado de su proyecto?
¿Qué necesidad logrará solucionar su proyecto?

9. Alineación del proyecto con los objetivos estratégicos de la organización cliente

Objetivos estratégicos de la organización	Propósito del proyecto
Objetivo estratégico 1	
Objetivo estratégico 2	
Objetivo estratégico n	

10. Propósito o justificación del proyecto

(Línea base de costos + Reserva de Gestión)

16. Requerimientos de aprobación del proyecto

¿Quién firma formalmente la aprobación del término del proyecto?

¿Cuáles son los criterios de aprobación que utilizará?

17. Supuestos del proyecto

18. Restricciones del proyecto

Restricciones	Impuesto por
Restricción 1	
Restricción 2	
Restricción n	

19. Relación de los interesados principales del proyecto

Interesados	Rol
Interesado 1	
Interesado 2	
Interesado n	

20. Nombre, fecha y firma

Nombre, fecha y firma del patrocinador del proyecto

Nombre, fecha y firma del Gerente de Proyecto

Dependiendo de la organización se podrán incluir a otros interesados del proyecto

Formato de Product Backlog

1. Fecha de elaboración del documento

2. Título del proyecto:

3. Siglas del proyecto:

4. Definición de la metodología de priorización de las historias de usuario

< El Product Owner desarrolla un Backlog Priorizado del Producto que contiene una lista priorizada de los requerimientos del negocio y de los proyectos escritos en forma de épica(s), que son historias de usuario de alto nivel.>

El Backlog Priorizado del Producto se basa en tres factores principales: valor, riesgo o incertidumbre y dependencias.

N° Req. Negocio	Requerimiento de negocio	Riesgo	Nivel de impacto del riesgo (Alto – Medio - Bajo)	Estimación (alto nivel) en días
RN1				
RN1				
RN2				

RN2				
RN2				
RN3				

Formato de historias de usuario

1. Fecha de elaboración del documento

2. Título del proyecto:

3. Siglas del proyecto:

4. Funcionalidades y criterios de aceptación

<< La historia de usuario debe ser creada en lenguaje usuario >>

Una historia de usuario debe estar conformada por las 3C:

- Card (tarjeta): Descripción escrita de lo que necesita el usuario.
- Conversación: El PO y el DT aclaran los detalles.
- Confirmación: Sirve para determinar lo que se espera.

Formato de historia de usuario:

Como <rol/prototipo de cliente>, yo debería <requerimiento> a fin de <beneficio>.

Ejemplo de historia de usuario:

Como administrador de una base de datos, yo debería contar con la capacidad de revertir una cantidad selecta de actualización de la base de datos a fin de que se restablezca a la versión deseada.

Nº Historia Usuario	Descripción de la Historia de Usuario	¿Quién lo necesita?	¿Cuál es el beneficio?	Prioridad	Estimación en horas	Criterio de aceptación
UH1						
UH2						
UH3						
UH4						
UH5						
UH6						

Formato de tareas

1. Fecha de elaboración del documento

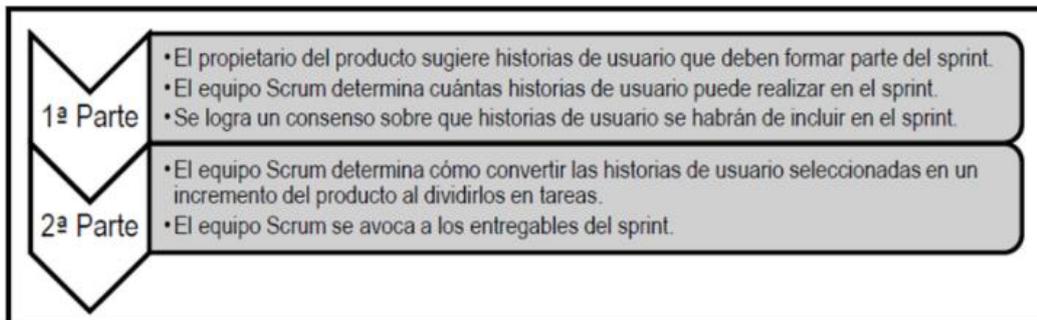
2. Título del proyecto:

3. Siglas del proyecto:

4. Funcionalidades y criterios de aceptación

La tarea debe ser redactada en un lenguaje técnico a fin de que el Equipo de Desarrollo lo entienda.

Reunión de planificación de las tareas:



N° Historia Usuario	Descripción de la Historia de Usuario	N° Tarea	Responsable Tarea	Descripción Tarea	Estimación Tarea
UH1		T1			
UH2		T2			
UH3		T3			
UH4		T4			
UH5		T5			
UH6		T6			

Formato de Sprint Backlog

1. Fecha de elaboración del documento

2. Título del proyecto:

3. Siglas del proyecto:

4. Lista de Pendientes del SPRINT

<< La Lista completa que contiene todas las tareas a las que se ha abocado el equipo Scrum en el actual sprint. Contiene descripciones de cada tarea, así como las estimaciones derivadas durante el proceso de creación de tareas >>

SPRINT	N° Historia Usuario	Descripción de la Historia de Usuario	N° Tarea	Descripción Tarea
SPRINT 1	UH1		T1	
SPRINT 1	UH2		T2	
SPRINT 1	UH3		T3	
SPRINT 1	UH4		T4	
SPRINT 1	UH5		T5	
SPRINT 1	UH6		T6	

Formato Scrumboard

1. Fecha de elaboración del documento

2. Título del proyecto:

3. Siglas del proyecto:

4. SCRUMBOARD

El tablero contiene cuatro columnas para indicar el progreso de las tareas estimadas para el sprint: una columna "por hacer" para las tareas que aún no se inician; una columna "en progreso" para las tareas que ya iniciaron, pero no se han concluido; una columna "en prueba" para las tareas concluidas pero que están en proceso de evaluación y una columna de "terminado" para las tareas que se han concluido y evaluado satisfactoriamente.

HISTORIA DE USUARIO	POR HACER	EN PROGRESO	EN PRUEBA	TERMINADO
Historia de usuario 1	Tarea 5	Tarea 4 Tarea 3	Tarea 2	Tarea 1
Historia de usuario 2				
Historia de usuario 3				

Formato registro de impedimentos

1. Fecha de elaboración del documento
2. Título del proyecto:
3. Siglas del proyecto:
4. Registro de impedimentos

Id Impedimento	Interno/Externo	Nº Sprint	Descripción del impedimento	Impacto en los objetivos	Urgencia	Acciones	Estado

Formato de aceptación de entregables

1. Fecha de elaboración del documento
2. Título del proyecto:



UNIVERSIDAD CÉSAR VALLEJO

ESCUELA DE POSGRADO

MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

Declaratoria de Autenticidad del Asesor

Yo, ACUÑA BENITES MARLON FRANK, docente de la ESCUELA DE POSGRADO MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN de la UNIVERSIDAD CÉSAR VALLEJO SAC - LIMA NORTE, asesor de Tesis titulada: "Marco de Trabajo Scrum para la Gestión de Proyectos de Software en una Empresa Consultora, Lima 2023", cuyo autor es GUTIERREZ ESTRELLA JOHN WILLIAMS, constato que la investigación tiene un índice de similitud de 9.00%, verificable en el reporte de originalidad del programa Turnitin, el cual ha sido realizado sin filtros, ni exclusiones.

He revisado dicho reporte y concluyo que cada una de las coincidencias detectadas no constituyen plagio. A mi leal saber y entender la Tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad César Vallejo.

En tal sentido, asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

LIMA, 06 de Enero del 2023

Apellidos y Nombres del Asesor:	Firma
ACUÑA BENITES MARLON FRANK DNI: 42097456 ORCID: 0000-0001-5207-9353	Firmado electrónicamente por: MACUNABE el 06- 01-2023 14:59:52

Código documento Trilce: TRI - 0511376