



UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

“Aplicación de Scrum en el desarrollo de software en TeamSoft S.A.C.”

TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS

AUTOR:

Villalva Castañeda, Luis Aarón

ASESOR:

Dra. Yesenia Vásquez Valencia

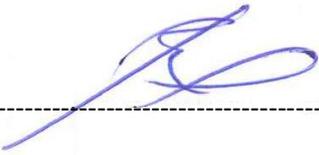
LÍNEA DE INVESTIGACIÓN:

Sistemas de Información estratégicos y toma de decisiones

LIMA – PERÚ

2017

PÁGINA DEL JURADO



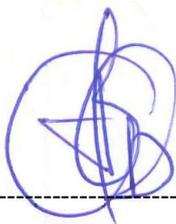
PRESIDENTE

MG. RENE RIVERA CRISÓSTOMO



SECRETARIO

DRA. YESENIA VÁSQUEZ VALENCIA



VOCAL

ING. IVÁN CRISPÍN SANCHEZ

Dedicatoria

En primer lugar a Dios todopoderoso porque de Él, por Él y para Él, son todas las cosas, sin su ayuda no hubiera sido posible terminar mi carrera profesional, es por ello que en primer lugar agradezco al Dios de mi salvación. Segundo, a mi madre Amanda que me enseñó con su ejemplo a ser esforzado y valiente y que todo se puede en el nombre de Jesús, a mi padre Abraham que se esforzó siempre por sacar a sus hijos adelante, por su valentía y amor por mi madre, son y seguirán siendo un ejemplo para mí, es por ello que dedico y honro con estas palabras el esfuerzo de mis padres y familia que en todo momento me dieron el apoyo necesario para poder culminar de manera satisfactoria este objetivo profesional.

Agradecimiento

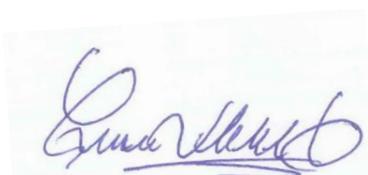
Agradezco a Dios por mi familia porque siempre se esforzaron en ayudarme a lograr mis objetivos profesionales, he tenido la oportunidad de recibir el apoyo de mis hermanos que siempre me ayudaron a seguir adelante, en mis luchas estuvieron presente y me levantaron cuando lo necesitaba, a todos mis seres queridos que guardo en el corazón y que me alentaron a llegar a lograr este objetivo, a cada uno de ellos les tengo presente siempre y les agradezco inmensamente su gran ayuda.

DECLARATORIA DE AUTENTICIDAD

Yo Villalva Castañeda, Luis Aarón con DNI N° 43804281, a efecto de cumplir con las disposiciones vigentes consideradas en el Reglamento de Grados y Títulos de la Universidad César Vallejo, Facultad de Ciencias Empresariales, Escuela de Negocios Internacionales, declaro bajo juramento que toda la documentación que acompaño es veraz y auténtica.

Así mismo, declaro también bajo juramento que todos los datos e información que se presenta en la presente tesis son auténticos y veraces. 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 de la Universidad César Vallejo.

Lima, 12 de Octubre del 2017

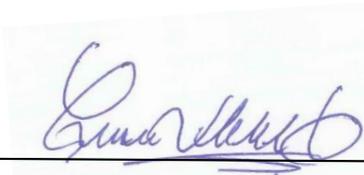


Villalva Castañeda, Luis Aarón

DNI: 43804281

PRESENTACIÓN

Señores Miembros del Jurado, presento ante ustedes la tesis titulada “Aplicación de Scrum en el Desarrollo de Software de TeamSoft S.A.C”, la misma que someto a vuestra consideración y espero que cumpla con los requisitos de aprobación para obtener el título Profesional de Ingeniero de Sistemas. El presente trabajo de investigación consta de seis capítulos: El primero capítulo lleva por título Introducción, en él se describe la realidad problemática, trabajos previos, teorías relacionadas al tema los cuales son el sustento base de esta tesis, además de manifestarse las justificaciones, los objetivos e hipótesis generales y específicas que persigue la investigación. En el segundo capítulo se detalla la metodología aplicada describiendo el tipo de investigación y diseño aplicado, además se determinan la población y muestra sobre la cual se realizaron las pruebas de pre-test y post-test y se plantearon los métodos de análisis de datos y desarrollaron las técnicas e instrumentos de recolección de datos. En el tercer capítulo, se dan a conocer los resultados obtenidos por cada indicador planteado al realizar las pruebas respectivas tanto antes como después de la aplicación de Scrum, las cuales fueron descritas en el anterior capítulo con sus respectivos gráficos y tablas para hacer la explicación más entendible. En el cuatro capítulo se hicieron las comparaciones de los resultados del trabajo con los resultados obtenidos en otras investigaciones con la intención de respaldar estos trabajos o discrepar de ellos en el caso de no concertar con la solución planteada. En el quinto capítulo, fueron expuestas las conclusiones finales del proyecto de investigación por cada indicador basados en los resultados obtenidos en el capítulo anterior. En el sexto capítulo están las recomendaciones a futuras investigaciones tomando como base la experiencia de la investigación y las observaciones que surgieron en la aplicación de su desarrollo.



Villalva Castañeda, Luis Aarón

RESUMEN

El estudio comprende la aplicación del marco de trabajo Scrum sobre los procesos de desarrollo de software en la empresa TeamSoft SAC, actualmente la empresa presenta algunas problemáticas en ámbitos como la productividad, cobertura de requisitos y eficiencia de los costos, estos como parte de los procesos de desarrollo de software. El objetivo principal fue determinar los efectos de la aplicación de un método ágil para los procesos de desarrollo de software y poder ver el efecto de su aplicación sobre el porcentaje de la productividad, porcentaje de los requerimientos cubiertos y el porcentaje de la eficiencia del costo. La muestra estuvo conformada por los proyectos de software de la cartera de proyectos del área en un periodo de 150 días. El diseño del estudio es Experimental de tipo pre-experimental, y el tipo de estudio es aplicado.

Como resultado se obtuvo que el porcentaje que representó la productividad antes de la aplicación de Scrum tuvo un 39%, luego de la aplicación alcanzó un porcentaje de 79%, lo que significa un aumento del 40% en productividad. Para el indicador de cobertura de requisitos, antes de la aplicación se obtuvo un 34%, luego de la aplicación fue de 95%, esto significa un aumento porcentual de 61% en cobertura de requisitos, para el indicador de eficiencia del costo, antes de la aplicación se obtuvo un resultado de 83%, luego de la aplicación el porcentaje fue de 94%, esto representa un aumento del 11% en la eficiencia del costo. Finalmente, se concluyó fue que la aplicación de un marco de trabajo ágil tuvo un efecto positivo en los procesos de desarrollo de software para la empresa TeamSoft S.A.C.

Palabras claves: Marco de Trabajo, Scrum, Metodología de desarrollo de Software, metodología ágil.

ABSTRACT

This study includes the application of the SCRUM framework on software development processes in the company TeamSoft SAC, which currently presents certain problems in productivity, coverage of requirements and inefficiencies in software development costs. The main purpose was to determine the effects of the application of an agile method for software development processes and to see its effect on the percentage of productivity, the percentage of covered requirements and the percentage of cost efficiency. The sample consisted of the software projects of the project portfolio of the area in a period of 150 days. The design of the study was Experimental of pre-experimental type. The type of study was applied.

As a result, it was obtained that the percentage that represents the productivity before the application of Scrum ended with 39% and after the application was 79%, which means a 40% increase in productivity. For the requirement coverage indicator, 34% was obtained before application, after application it was 95%, this means a percentage increase of 61% in requirements coverage, for the cost efficiency indicator, before the application was obtained a result of 83%, after the application the percentage was 94%, and this represents an increase of 11% in the cost efficiency. Finally, the conclusion was that the application of an agile framework had a positive effect on the software development processes for the company TeamSoft S.A.C.

Keywords: Framework, Scrum, Software Development Methodology, agile methodology.

ÍNDICE GENERAL

I. INTRODUCCIÓN.....	1
1.2 Trabajos Previos.....	7
2.2 Teorías Relacionadas al Tema	10
2.2.1 Marco de trabajo Scrum.....	10
2.2.2 Principios de Scrum.....	12
2.2.3 Los Roles que maneja el equipo Scrum.....	13
2.2.4 Los Artefactos de Scrum.....	15
2.2.5 Reuniones de Scrum	18
2.2.6 Ingeniería del Software.....	22
2.6.7 Medidas en la Ingeniería de Software.....	28
2.6.8 Análisis comparativo Metodologías Ágiles vs. Tradicionales	38
2.6.9 Elección del marco de trabajo Scrum.....	41
2.3 Formulación del Problema	43
2.5 Hipótesis.....	45
2.6 Objetivos.....	46
II. MÉTODO.....	47
2.1 Diseño de la Investigación	48
2.2 Variables – Operacionalización.....	49
2.3 Población y Muestra50	
2.4 Técnicas e Instrumentos de recolección De Datos	50
2.5 Validez y Confiabilidad del Instrumento	50
2.6 Métodos de Análisis de Datos.....	51
2.7 Aspectos Éticos.....	53
III. RESULTADOS	54
3.1 Resultados de la estadística descriptiva	55
3.2 Prueba de Hipótesis..	59
3.2.1 Resultados del indicador de Productividad	59
3.2.2 Resultados del indicador de Cobertura de Requisitos	61
3.2.3 Resultado del indicador de la eficiencia del Costo	62
3.2.4 Resultado general de las dimensiones	64
IV. DISCUSIÓN.....	66
V. CONCLUSIONES	69
VI. RECOMENDACIONES.....	72
VII. REFERENCIAS	74
VIII.ANEXOS.....	80

ÍNDICE DE TABLAS

Tabla 1: Efecto de la productividad de Scrum	55
Tabla 2: Efecto de la cobertura de requisitos de Scrum	56
Tabla 3: Efecto de la eficiencia del costo de Scrum	57
Tabla 4: Resultado de las dimensiones generales y promedios estadísticos	58
Tabla 5: Pruebas de normalidad - Productividad.....	59
Tabla 6: Prueba de muestras emparejadas - Productividad	60
Tabla 7: Resultado de la hipótesis - Productividad	60
Tabla 8: Prueba de normalidad – Cobertura de Requisitos	61
Tabla 9: Estadísticos de prueba – Cobertura de Requisitos	61
Tabla 10: Resultado de la hipótesis – Cobertura de Requisitos	62
Tabla 11: Prueba de normalidad – Eficiencia del costo	62
Tabla 12: Prueba de muestras emparejadas – Eficiencia del costo.....	63
Tabla 13: Resultado de hipótesis – Eficiencia del costo	63
Tabla 14: Prueba de normalidad – dimensiones de la investigación	64
Tabla 15: Estadísticos de prueba – dimensiones de la investigación	64
Tabla 16: Resultado general de la hipótesis – dimensiones de la investigación ...	65
Tabla 17: Plantilla del Product Backlog	87
Tabla 18: Desarrollo del Product Backlog	87
Tabla 19: Peso relativo	90
Tabla 20: Desarrollo del Peso Relativo	91
Tabla 21: Priorización del Product Backlog	91
Tabla 22: Plantilla del Sprint Backlog	92
Tabla 23: Desarrollo del Sprint Backlog – Sprint 1	93
Tabla 24: Desarrollo del Sprint Backlog - Sprint 2	94
Tabla 25: Desarrollo del Sprint Backlog - Sprint 3	95
Tabla 26: Desarrollo del Sprint Backlog – Sprint 5	96
Tabla 27: Desarrollo del Sprint Backlog – Sprint 6	96
Tabla 28: Desarrollo del Sprint Backlog – Sprint 7	98
Tabla 29: Desarrollo del Sprint Backlog – Sprint 8	98
Tabla 30: Tabla de retrospectiva del proyecto.....	104
Tabla 31: Tabla de respuesta a la retrospectiva del proyecto.....	105

ÍNDICE DE FIGURAS

Figura 1: Gráfico de productividad de recursos	4
Figura 2: Gráfico de cobertura de requisitos	5
Figura 3: Gráfico de sobre costos en los proyectos	6
Figura 4: Scrum Team – Kenneth S, Rubín and Innolution.....	14
Figura 5: Product Backlog - Kenneth S, Rubín and Innolution.....	17
Figura 6: Spring Planning - Kenneth S, Rubín and Innolution.....	18
Figura 7: Scrum – Ciclo de Aplicación (Álvarez, et al., 2012 p. 86)	21
Figura 8: Modelo en cascada – Sánchez (2012, p.40)	23
Figura 9: Modelos en “V” – Sánchez (2012, p.41)	24
Figura 10: Modelo espiral – Sánchez (2012, p.45)	25
Figura 11: Ágiles vs cascada – Lynch (2015, párr. 14).....	38
Figura 12: Pequeños - ágiles vs cascada - Lynch (2015, párr. 22)	39
Figura 13: Grandes - ágiles vs cascada - Lynch (2015, párr. 23)	39
Figura 14: Metodologías ágiles versus Cascada - Lynch, 2015 Chaos Report.	40
Figura 15: Comparativa Ágiles vs Tradicionales. Maida et al. (2015, p. 19)	41
Figura 16: Análisis Metodologías ágiles (Tinoco Gómez, et al., 2010 p. 73).....	42
Figura 17: Causa – Efecto (Hernández, et al., 2010 p. 121).....	48
Figura 18: Cuadro de Operacionalización de las variables.....	49
Figura 19: Resultados de la productividad pre y post test	55
Figura 20: Resultados de la productividad pre y post test	56
Figura 21: Resultados de la eficiencia del costo post y pre-test	57
Figura 22: Resultados pre y post – Tradicional Vs. Scrum	58
Figura 23: Indicadores finales de las dimensiones del proceso.....	59
Figura 24: Matriz de consistencia de los procesos de ingeniería de software	81
Figura 25: Ficha de Registro Nro. 1	82
Figura 26: Ficha de Registro Nro. 2	83
Figura 27: Desarrollo del Product Backlog	89
Figura 28: Formulario de afiliación de clientes de fija	93
Figura 29: Formulario de afiliación al débito	94
Figura 30: Formulario de afiliación al débito	96
Figura 31: Formulario de afiliación al débito	97
Figura 32: Formulario de recepción de la afiliación	99
Figura 33: Desarrollo del Sprint Backlog	99
Figura 34: Desarrollo del Sprint Backlog en la pared	100
Figura 35: Seguimiento del Sprint Backlog en la pizarra	101
Figura 36: Desarrollo del Burdown Chart	101
Figura 37: Objetivo del desarrollo del Daily Meeting.....	102
Figura 38: Review del sprint de desarrollo – Fase 1	103
Figura 39: Acta de constitución del proyecto	106
Figura 40: Propuesta de solución del proyecto.....	106
Figura 41: Requerimientos de Usuario – Nivel 0	107

Figura 42: Autenticidad de Tesis	108
----------------------------------------	-----

ÍNDICE DE ANEXOS

Anexo 1: Matriz de Consistencia	81
Anexo 2: Ficha de Registro	82
Anexo 3: Desarrollo del proyecto de Investigación	84
Anexo 4: Resultados de Autenticidad.....	108

I. INTRODUCCIÓN

1.1 Realidad Problemática

La creciente competencia de las organizaciones por alcanzar sus objetivos estratégicos introduce la necesidad del uso de las Tecnologías de la Información en sus procesos de negocio, dentro del campo de desarrollo, los productos software obtenidos ayudan a las compañías y sirven como herramientas de soporte, operativo y estratégico para la toma de las decisiones, en este sentido es de gran importancia llevar un control adecuado sobre las formas de cómo desarrollar software más eficiente, esto conlleva a hacer uso de metodologías comprobadas que coadyuven a esta estrategia, respecto de estas metodologías de software (MDS) Rivas, Corona y Hernández (2015, p. 982) dijeron que “[...] son indispensables para crear, o modificar software de calidad que cumpla con los requisitos de los usuarios, ya que si no se utiliza la metodología apropiada, seguramente no se alcanzará el objetivo”.

En general, Rivas et al. (2015, p. 982) explicaron que “el problema actual es que de las diversas MDS que existen no se selecciona la adecuada, y en el peor de los casos no se emplea ninguna”. Las metodologías tradicionales de desarrollo de software actualmente están siendo absorbidas por los métodos ágiles, para Landry y Mc Daniel (2015, p. 1) indicaron que “el uso de prácticas ágiles es cada vez más y más prevalente [...] aproximadamente el 88% de los encuestados son practicantes ágiles en su lugar de trabajo”.

El especialista Zuker (2016, párr. 15) explicó las diferencias entre métodos tradicionales y ágiles, el indica que: “En general, los proyectos ágiles tienen tres veces más probabilidades de tener éxito que los proyectos Waterfall (39% versus 11%). Los proyectos grandes y medianos muestran una mejora espectacular con Agile. Los proyectos grandes aumentan sus tasas de éxito del 3% al 18% y los proyectos medianos del 7% al 27%. Los proyectos pequeños demuestran ganancias menores (44% versus 58%)”.

A nivel regional, la problemática en base a los resultado de las metodologías tradicionales versus las ágiles mantiene sus diferencias, así lo explicaron Maida y Pacienza (2015, p. 14) donde dijeron que las metodologías tradicionales tienen desventajas porque su planificación y costos son imprecisas, además de tener en su mayoría falta de productividad, insatisfacción con el cliente, vaga indicación de los requisitos del cliente, baja calidad del producto final, finalmente en muchas ocasiones es muy costoso el mantenimiento del software.

En el ámbito nacional, la problemática no es ajena a lo descrito anteriormente, el Ingeniero especialista Oscar Camacho (2015, párr. 4) sostuvo que las problemáticas en la gestión de proyectos para el desarrollo de software se presentan cuando “el proceso de cada proyecto incluye la gestión, el ciclo de vida, la metodología, los estándares, la ingeniería, la arquitectura, el componente técnico – administrativo, entre otros”. Oscar Camacho (2015, párr. 4), bajo esta afirmación indicó que son cuatro pilares los que causan la problemática, estas se basan en la no adecuada definición del alcance del proyecto, la no aplicación de metodologías de desarrollo de software, la selección de tecnologías de maduración y la selección de personas para el equipo de TI. Asimismo, Singh indicó que los proyectos bajo el modelo en cascada presentan algunas debilidades que incluyen:

[...] su falta de flexibilidad, lentitud, costos relativamente altos [...] que restringe las oportunidades de iterar o responder al cambio en requisitos o ideas de diseño. También da pocas oportunidades para que los clientes participar en el proceso hasta el final del proyecto (2012, p. 59).

Problemática en TeamSoft SAC.

La problemática en TeamSoft se deriva por la aplicación de metodología tradicional que se usa para llevar a cabo la ejecución de los proyectos de software, el modelo que se maneja actualmente es el ciclo de vida en cascada o

modelo predictivo el cual no permite la flexibilidad adecuada sobre los proyectos, ya sean por los cambios que hay en el proceso del negocio u otro tipo de evento que se suscite e implique un cambio en la funcionalidad del producto, en tal sentido los entregables se dan en tiempos bastante largos y con alto riesgo de fallos en la entrega del producto final.

1. Problemática con la productividad de los Recursos

Según Maldonado (2017, p. 4) como citó Melo et al. (2011) “[...] se ha mostrado que los métodos ágiles incrementan la productividad respecto a métodos como cascada, para esto se han realizado estudios de variables como: la composición del equipo y asignación, las dependencias externas y la rotación de personal”. En TeamSoft según la prueba de Pre-Test realizada, el modelo actual del ciclo de vida en cascada solo permite hacer la entrega del producto de software al final del proyecto, los proyectos en promedio tienen 5 meses de desarrollo y la cantidad de entregables que se dan al Usuario son dos como máximo, los recursos de un equipo de software en promedio son de 5 personas por proyecto, en base a este análisis se vio que la productividad estaba en un nivel bajo debido al uso de muchos recursos y pocos entregables funcionales, sumado a esto, se observó también que a través de los años se ha venido trabajando continuamente con un único modelo de desarrollo lo cual ha provocado un estancamiento y monotonía en las formas de producir software en los equipos de proyectos.

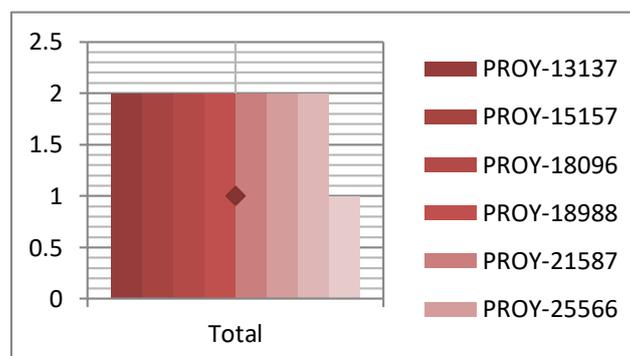


Figura 1: Gráfico de productividad de recursos

2. Problemática con la cobertura de requisitos del proyecto

El cumplimiento de los requisitos del software siempre ha sido un paradigma en el proceso de producción del software, la definición y comprensión de los mismos es de vital importancia para el cumplimiento del objetivo buscado, Lipkin (2011, p. 55) indicó que la impresión de los requisitos podría afectar el rendimiento del proyecto, y además son parte de los problemas relacionados a los proyectos. Panizzi, respecto a los constantes problemas en los proyectos de software que están relacionados a los requerimientos de Usuario sobre las funcionalidades de los sistemas, dijo lo siguiente:

En 1995, el Grupo de Standish realizó un estudio (el informe CHAOS), de las encuestas realizadas a los directores de los proyectos que participaron en el estudio, indicaron que, en su opinión, que los tres principales factores de fracaso eran: Falta de información por parte de los usuarios, especificaciones y requisitos incompletos, especificaciones y requisitos cambiantes. (2015, p. 4)

En TeamSoft, según el Pre-Test desarrollado, los requisitos de Usuario que se añaden como cambios aprobados para el producto de software no son realizados en su totalidad. Es decir, del 100% de cambios aprobados solo el 33.8% fueron efectuados por el equipo de desarrollo, los tiempos adicionales que se programan para efectuar cambios en el producto software representan un 21.3% del tiempo inicial planificado, esto representa un excedente significativo para poder dar un entregable al Usuario final y además con requerimientos incompletos que por normativa de la empresa serán atendidos en otro proyecto.

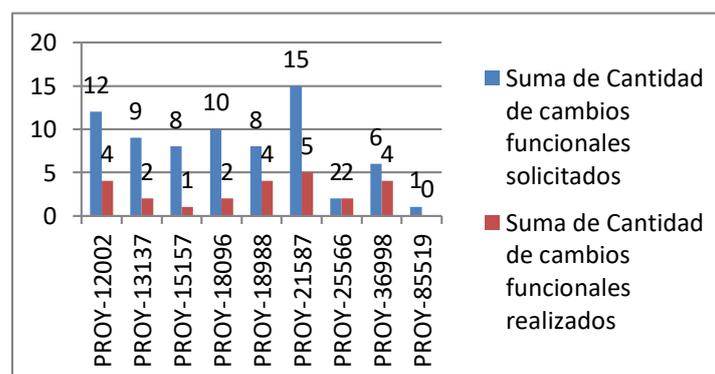


Figura 2: Gráfico de cobertura de requisitos

3. Problemática con los sobrecostos en los proyectos

La competitividad es muy común hoy en día, en el ámbito de las tecnologías de la información las empresas se ven con la necesidad de adquirir productos o servicios que les ayuden a poder competir, por ello, es importante que tomen una buena decisión frente a una nueva aplicación de este tipo, Hanadi (2015, p. 183) explicó que “Los líderes empresariales y tecnológicos están dirigiendo a los equipos para que se centren en el retorno de la inversión y el negocio cuantificado”, referente a este problema Alfaro enunció lo siguiente:

Es criticable que se inicien procesos de adquisición e implementación de tecnologías de información sin el correspondiente análisis de generación de valor financiero, sobre todo, cuando los montos de inversión o gasto son considerables para la organización, según su tamaño. Cuando no se dan, vemos las consecuencias en los aumentos de los costos o una dependencia tecnológica con el proveedor de la tecnología de información adquirida” (2012, p. 17).

La problemática en TeamSoft según el Pre-Test realizado indicó que la metodología tradicional utilizada para el desarrollo no permite el retorno óptimo de la inversión debido a que no es flexible para entregar productos de software en tiempos cortos, esto provoca que el Usuario deba hacer las validaciones al final del proyecto, en la mayoría de veces existen incongruencias en el producto de software, esto provoca que los costos exceden el límite planificado por nuevas horas hombre asignadas a la resolución de los problemas, estos sobre costos alcanzan la suma de S/. 126,267.00 que representan el 19% del presupuesto inicial planificado.

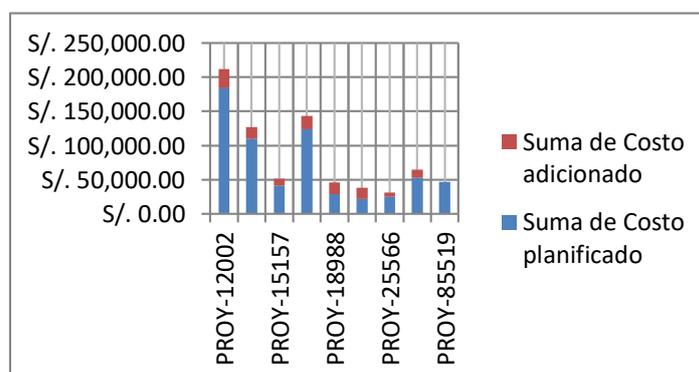


Figura 3: Gráfico de sobre costos en los proyectos

1.2 Trabajos Previos

1.2.1 Trabajos previos – Internacionales

En la Tesis de Rey (2017) con el título “Estudio de la efectividad de la aplicación de la metodología ágil de desarrollo Scrum” con motivo por optar por el título de Ingeniero de sistemas de la Universidad tecnológica de Israel, para lo cual el objetivo fue comparar la eficiencia y efectividad de los proyectos de desarrollo de software haciendo uso de metodologías ágiles. Se concluyó en este trabajo de investigación de manera general que las metodologías ágiles posibilitaron que el trabajo de proyectos pueda ser más eficiente y mucho mayor organizado alcanzando un mayor involucramiento con los requerimientos de software, asimismo la metodología SCRUM versus las metodología tradicionales alcanzaron un resultado del 98% de efectividad, por ello tuvo un margen positivo de 4%, frente a la metodología tradicional eso es por su flexibilidad en la administración de los proyectos asumiendo los cambios que se suscitaron en el transcurso del proceso de desarrollo de software.

En la Tesis de Flores (2016) con el título “Framework de trabajo Para Proyectos de Tesis aplicando la metodología Scrum en la Ingeniería de Software” con motivo por optar el título de Ingeniero de Sistemas Computacionales de la Universidad de Guayaquil, para lo cual el objetivo fue desarrollar un producto de software que permita realizar las gestiones académicas de una entidad educativa aplicando la metodología SCRUM enfocado al estudio de factibilidad de la ingeniería de software. Se concluyó del siguiente trabajo que se completó satisfactoriamente el trabajo planteado inicialmente haciendo entrega de los 6 Sprint considerados en la etapa inicial y cumpliendo el objetivo de lograr un sistema educativo usado por los usuarios.

En la Tesis de Pérez (2015) con *título* “Una Metodología Ágil para el desarrollo de software en una compañía financiera” con motivo por optar el título de Ingeniero de Sistemas de la Universidad Militar de Nueva Granada, para lo cual el objetivo fue plantear una alternativa para la mejora del proceso de desarrollo de software a través de la metodología Scrum que permita mejorar la calidad del software así como los retrasos en la entrega de software y la sobrecarga laboral y la flexibilidad para el cumplimiento de los requerimientos de Usuario aprovechando al máximo los recursos de la empresa. Se concluyó que se tienen las siguientes ventajas por uso de la metodología ágil Scrum:

- Scrum genera dinamismo en los procesos de desarrollo de software
- Scrum reduce los costos y ayuda a mejorar los servicios que se les ofrecen a los clientes.
- Mejora el nivel de comunicación.
- La división del proyecto en pequeñas tareas hace que el proceso de desarrollo sea mejor controlado.

2.1.1 Trabajos previos – Nacionales

En esta sección de la investigación se suscriben trabajos que se han realizado anteriormente por otros investigadores, se ha recopilado información nacional e internacional del tema de estudio, los informes revisados demuestran que desde hace mucho tiempo la tendencia de la aplicación de Scrum ha sido positiva. Ticona en su libro de metodologías para el desarrollo indicó lo siguiente:

En el mundo existen aproximadamente 300,000 empresas utilizando esta metodología ágil [...] en la actualidad la mayoría de los proyectos informáticos poseen una planificación al inicio de los mismos que poco tienen que ver con la realidad del día a día (2014, p. 89).

En la Tesis de Malpica (2014) con el título “Aplicación de la metodología Scrum para incrementar la productividad del proceso de desarrollo de software en la empresa CCJ S.A.C”, con motivo por optar el título de Ingeniero de Sistemas de la Universidad Nacional del Centro del Perú, para lo cual el objetivo fue aplicar la metodología Scrum a efectos de poder determinar la influencia de la metodología Scrum en el incremento de la productividad en el proceso de desarrollo de la empresa. Se concluyó de este trabajo que la aplicación de la metodología Scrum influye positivamente sobre el incremento de la productividad logrando que se cumplan los plazos estimados teniendo un desfase de 0 días y una pérdida neta de S/. 0.00, en términos generales la productividad de incrementó en un 30%.

En la Tesis de Vargas (2013) con el título “Mejora de Procesos de Desarrollo de Software mediante metodologías ágiles” con motivo por optar el título de Ingeniero de Sistemas de la Universidad Nacional de Ingeniería, para lo cual el objetivo era mejorar la entrega efectiva de los productos de software evitando retrasos en las fechas planificadas, así mismo mejorar el margen de ganancia de los proyectos de software, esto generaba presión al equipo de desarrollo y por ende sobre costo. Se concluyó del siguiente trabajo que se llegó a mejorar el margen operativo de los proyectos de software desarrollados en un periodo de 5 meses, el promedio mejorado fue de 5% siendo el margen inicial 39.1% y el final 44.1%.

2.2 Teorías Relacionadas al Tema

2.2.1 Marco de trabajo Scrum

Para Álvarez, De las Heras y Laza (2012, p. 39) Scrum es “Una de las más populares metodologías o métodos ágiles. Se trata de un marco de trabajo iterativo e incremental, de propósito general, aunque muy utilizado en el desarrollo de software”. Mariño y Alfonso muestran también explicaron que Scrum es “[...] una colección de procesos para la gestión de proyectos, que permite centrarse en la entrega de valor para el cliente y la potenciación del equipo para lograr su máxima eficiencia, dentro de un esquema de mejora continua”. (Mariño y Alfonso, 2014 p. 414). Finalmente, según lo descrito por Schwaber y Sutherland respecto a Scrum:

Es un marco dentro del cual puede emplear varios procesos y técnicas. Scrum deja en claro la eficacia relativa de las técnicas de gestión y trabajo de su producto para que pueda mejorar continuamente el producto, el equipo y el entorno de trabajo” (2017 p. 3).

Definición de un Sprint

El Sprint para Álvarez et al. (2012, p. 71) indicó que es una etapa que dura 1 a 4 semanas, en muchos casos 2 o 3, también nos dijo que 1 o varios Sprint pueden convertirse en un entregable final al Cliente, el conjunto de Sprint se le denomina Reléase. Según Schwaber et al. (2017) dijeron acerca del Sprint:

El corazón de Scrum es un Sprint, un cuadro de tiempo de un mes o menos durante el cual un "Hecho", utilizable, y se crea un Incremento de producto potencialmente liberable. Los sprints tienen duraciones consistentes a lo largo de un esfuerzo de desarrollo. Un nuevo Sprint comienza inmediatamente después de la conclusión de la Sprint anterior (2017, p. 9).

Efectos de la Aplicación de Scrum en el desarrollo de Software

▪ Productividad

El marco de trabajo de Scrum mejora la productividad de los equipos de desarrollo de software, así lo afirmaron Sutherland y Schwaber (2013, p. 6) de indicaron que: “El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad”, Los especialistas de la Consultora Española SofTeng (2017, párr. 4) indicaron que uno de los beneficios del impacto de Scrum en los desarrollos de software radica en la productividad pues se realiza menor burocracia y mayor auto-organización. En esta misma línea Erin indicó que “los métodos ágiles de desarrollo de software, como Scrum, son métodos de desarrollo de software que están destinados a mejorar la productividad” (2013, p. 86). Según Mitre, Ortega y Lemus (2014, p. 404) dijeron que para conocer el rendimiento de un equipo ágil es conociendo su velocidad sobre lo desarrollado en el Product Backlog, en ese sentido la medida para hallar el rendimiento para el desarrollo, para esto las medidas más comunes en los métodos ágiles son el esfuerzo y el tamaño.

▪ Cobertura de Requisitos

Para la cobertura de los requisitos los especialistas de la consultora SofTeng (2017, párr. 4) explicaron que Scrum tiene flexibilidad por su alta reacción a los cambios por necesidad del Cliente o del negocio, según (Palacio, p. 58) Scrum tiene estrategia de gestión para producir resultados rápidos en contraste con la gestión tradicional, por ejemplo; la entrega de manera temprana los incrementos funcionales del software, estas son las que son de mayor valor para el cliente y se podrán usar en un menor tiempo, finalmente el solapamiento en las fases de desarrollo.

▪ Eficiencia del Costo

Referente al costo los especialistas de la empresa SofTeng (2017, párr. 4) dijeron que Scrum minimiza el retorno de la inversión (ROI), ya que generan la producción de funcionalidades de software que proporcionan mayor valor al negocio, en tanto, Mitre (2014, p. 10) dijo que para calcular la eficiencia del costo en un método ágil se puede aplicar el índice del desempeño del costo (CPI), esta métrica permite visualizar la fluctuación del Costo por SP contra el costo por SP real durante cada iteración. Este índice permite conocer si el equipo desarrolla los SP bajo lo presupuestado en la planificación o está excediendo el presupuesto y pone en riesgo el éxito del proyecto.

2.2.2 Principios de Scrum

Respecto a los principios de Scrum, Álvarez et al. (2012, p. 39) dijo que Scrum propone un marco de trabajo que tiene mucha relación o está vinculado necesariamente a equipos auto-gestionados. Asimismo, indican que se pueden dar resultados de calidad en corto tiempo.

- 1. Inspección y Adaptación:** Álvarez et al. (2012 p. 39) dijo que Scrum trabaja en interacciones denominadas Sprint, estas duran entre una a cuatro semanas, cada interacción da como resultado un producto entregable funcional, este es dado al cliente para su apreciación y finalmente se realiza una reunión para el análisis del trabajo realizado.
- 2. Auto-organización y Colaboración:** Álvarez, et al. (2012, p. 39) indicó que el equipo de trabajo es auto organizado, se requiere un alto nivel de responsabilidad, respeto y compromiso entre los colaboradores del equipo.

3. **Priorización:** Álvarez, et al. (2012, p. 39) suscribió que la importancia de no tener tiempos muertos es maximizar los resultados, por ello es importante tener los requisitos priorizados y que reflejen la necesidad del negocio.
4. **Mantener un latido:** Según Álvarez, et al. (2012, p. 39) Mantener un ritmo con todos los integrantes del equipo es de gran importancia, puesto que ayudará a pensar en un mismo objetivo y centrarse en optimizar el trabajo, se realizará con mayor eficiencia las estimaciones de las actividades a realizar para el posterior compromiso de las fechas de entrega o reuniones.

2.2.3 Los Roles que maneja el equipo Scrum

1. **El Cliente:** En la definición del Cliente Álvarez et al. (2012, p. 63) refirió lo siguiente: “En Scrum, más que cliente se debe hablar de los Stakeholders, es decir, de todas las personas y organizaciones que tienen algún interés en el trabajo.
2. **El Scrum Master:** Para Álvarez, et al. (2012, p. 66) el Scrum Master no es un Jefe de Proyecto sino más bien se le considera un facilitador, su principal objetivo es mejorar la productividad de los participantes del proyecto, abre el camino y elimina interferencias, fomenta también las prácticas ágiles. Otra definición dada por los especialistas de Scrum Alliance Inc. indicaron que:

El papel Scrum Master tiene dos elementos distintos. En primer lugar [...] actúa como protector del equipo, asegurándose de que todo el mundo en el proyecto [...] pueda concentrarse en su trabajo [...] El segundo elemento de la función Scrum Master es proteger el proceso de Scrum en sí [...] es el experto en cómo funciona Scrum y la forma en que se debe aplicar (2016, párr 6).

El Scrum Master en Scrum cambia su definición respecto de los proyectos que se desarrollan bajo el método tradicional Mayfield (2010 p.

28) dijo que “en lugar de enfocarse en la ejecución de un plan detallado, como era la norma en la gestión de proyectos tradicionales, el Scrum Master asegura que el equipo está colaborando efectivamente entre ellos y con el dueño del producto”.

3. El Product Owner: La definición para el Product Owner según Álvarez et al. (2012,p. 65) dijo que este “forma parte del cliente y actúa como intermediario entre este y el equipo. Por ello debe ser capaz de hablar el lenguaje del negocio o de los requisitos del cliente y estar familiarizado con los métodos y conceptos empleados por el equipo”.

4. El Equipo: “Los componentes del equipo de trabajo no tiene un rol específico asignado, pero sin ellos es imposible llevar a buen puerto el trabajo [...] con la ayuda del Scrum Master, el equipo deberá ser capaz de realizarán seguimiento de la evolución de su productividad y mejorarla” (Álvarez et al., 2012 p. 68).

Los especialistas de Scrum Alliance Inc (2016, párr 9) indicaron que las características principales de un equipo de desarrollo Scrum son de fácil adaptación, o por lo menos deberían acoplarse al medio de trabajo a fin de tener una mejor organización, dijeron que “[...] el Team Developer (equipo de desarrollo): Encargado de auto organizarse y construir el producto”, está es una característica que es impulsada con mayor fuerza en los equipos ágiles.

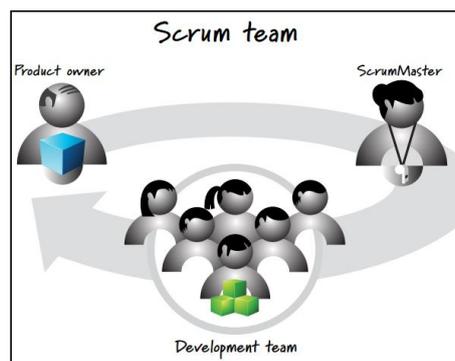


Figura 4: Scrum Team – Kenneth S, Rubín and Innolution

2.2.4 Los Artefactos de Scrum

1. Product Backlog

El Producto Backlog es uno de los artefactos de mayor importancia que manejan los equipos de trabajo ágil como Scrum, Alvarez, et al. dijo lo siguiente:

“El Product Back log es una lista de los requisitos que debe cumplir el producto que se quiere construir [...] los requisitos surgen para aportar valor al negocio a lo que se está desarrollando [...] es una lista ordenada por prioridad. De mayor prioridad a menor” (2012. p. 117).

1.1 Técnicas de priorización del Product Backlog

Priority Poker: Para Álvarez, et al. (2012, p. 126) esta es una técnica en la cual se reúne a las personas involucradas, cada una toma una baraja con valores del 1 al 9 siendo 9 la de mayor prioridad, luego se enuncia los ítems del Product Backlog y cada participante asigna un valor, finalmente se hace una sumatoria y se obtiene un resultado el cual será ordenado de mayor a menor o viceversa.

MosCoW: Álvarez, et al. (2012, p. 127) indicó que este método consiste en estratificar el Backlog en partes, esta división será en referencia al nombre del método el cual tiene los siguientes significados:

- **Must:** El producto debe tener este requerimiento.
- **Should:** No es tan importante como el anterior, pero de mucho valor para el producto.
- **Could:** El producto podría o no tenerlo.
- **Won't:** No es necesario tener las características en el producto.

Modelo Kano: Este modelo según Álvarez, et al. (2012, p. 128) es usado también para priorizar el Product Backlog, el modo de uso es categorizando cada ítem del Backlog en las diferentes categorías:

- Básicas u obligatorias: Representas las necesidades que son fundamentales en un producto software de tal forma que sin ellas sería de total insatisfacción para el Usuario.
- Rendimiento Lineal: Representa aquellos requisitos que produzcan satisfacción al cliente, pero si se presiden de estas no causan mayor insatisfacción.
- Inesperadas o Emocionantes: Representan los requisitos neutrales que no son objeto de primordial uso, pero si están presente generar gran satisfacción por el Usuario.

Criba de Temas: Para Álvarez, et al. (2012, p. 130) en esta técnica se debe establecer criterios como una especie de misión a donde va apuntar el objetivo del Backlog, esto es para poder ordenar en base a las épicas de cada conjunto de elementos la nueva misión a seguir, nos explica que la consideración de la cantidad de criterios a seguir deberá ser no mayor a 5 para un trabajo adecuado.

Puntuación de elementos: Esta técnica es muy parecida al Cribado de Temas, así lo explicó Álvarez, et al. (2012 p. 132), pues su principal método es asignarle pesos a cada Criterio u objetivo y asignar a estos criterios los elementos del Backlog para poder diferenciarlos y priorizarlos en base al objetivo del negocio.

Peso Relativo: El siguiente método según Álvarez, et al. (2012, p. 133) es el más completo pues se apoya en los demás métodos anteriores, este usa la asignación de pesos por cada elemento del Backlog que son valores que van de 1 al 9 de tal forma que se evalúa cada requerimiento en 2 aspectos, el primero es, que impacto tiene el requisito de tenerlo en el producto software, el otro es, el impacto de no tenerlo, en base a estas asignaciones se obtendrá como resultado el valor porcentual y total de la siguiente manera:

- Valor total: Beneficio Relativo + Penalización relativa
- Valor porcentual = Valor total del elemento / Suma (Valores Totales)
- Coste porcentual = Estimación de elemento / Suma (Estimaciones)

“Analizando la relación entre el valor porcentual y el coste porcentual se podrá valorar de forma objetiva cual sería la prioridad para poder primar los elementos que con menor coste porcentual se puedan obtener en mayor valor porcentual” (Álvarez, et al., 2012, p. 134).

2. Spring Backlog

El Sprint Backlog “se trata de un repositorio que recoge los trabajos que van a realizarse en un iteración o Sprint determinado. Es decir, cada Sprint tiene un Sprint Backlog distinto”. Según Álvarez et al. (2012, p. 151) dijo que:

Sprint Backlog es el conjunto de ítems del Backlog del Producto seleccionados para Sprint, más un plan para entregar el producto Incrementar y realizar el objetivo de Sprint. Sprint Backlog es un pronóstico por el Equipo de Desarrollo sobre qué funcionalidad será en el próximo Incremento y el trabajo necesario para entregar esa funcionalidad en un Incremento "Hecho" (Schwaber, 2017, p. 14).

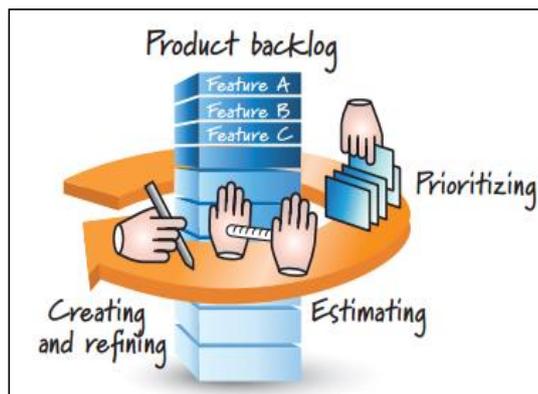


Figura 5: Product Backlog - Kenneth S, Rubín and Innolution

3. Burdown Chart

Para Álvarez, et al. (2012, p. 81) el Burdown Chart es una gráfica que representa el trabajo pendiente del equipo. La diferencia entre el trabajo real y la planificada ira mostrando si se está avanzando a un ritmo adecuado o si

se necesitará corregir el proceso. Los especialistas de Scrum.Org manifestaron el siguiente concepto del Burdown Chart:

Es un gráfico que muestra la cantidad de trabajo que se piensa que permanecerá en el Backlog. El tiempo se muestra en el eje horizontal y el trabajo restante en el eje vertical. A medida que pasa el tiempo y los elementos se extraen de la acumulación y se completan, es de esperar que disminuya una línea de trama que muestra el trabajo restante. La cantidad de trabajo se puede evaluar de varias maneras, como puntos de historia de usuario u horas de tarea (s.f, párr. 3).

2.2.5 Reuniones de Scrum

1. Sprint Planning

El Sprint Planning es una de las principales reuniones de Scrum, Ken Schwaber, et al. dijo lo siguiente:

El trabajo que se realizará en el Sprint está planificado en Sprint Planning. Este plan es creado por el trabajo colaborativo de todo el equipo de Scrum. La planificación de Sprint está sincronizada en el tiempo hasta un máximo de ocho horas para un Sprint de un mes (2017 p. 10).

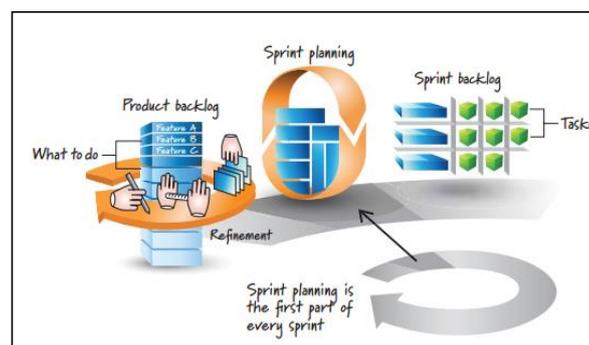


Figura 6: Spring Planning - Kenneth S, Rubín and Innolution

2. Daily Meeting

Según Álvarez et al. El Daily Meeting es una “reunión que, con frecuencia idealmente diaria realiza el equipo Scrum para anunciar los avances

realizados desde la última, las actividades que se planean realizar hasta la siguiente, y los impedimentos encontrados” (2012, p. 334). Para Schwaber et al. (2017, p. 11) “The Daily Scrum es un evento de 15 minutos en el que el Equipo de desarrollo sincroniza actividades y crear un plan para las próximas 24 horas”.

3. Sprint Review

Una Revisión de Sprint se lleva a cabo al final del Sprint para inspeccionar el Incremento y adaptar el Producto atrasos si es necesario. Durante la Revisión de Sprint, el equipo de Scrum y las partes interesadas colaboran sobre lo que se hizo en el Sprint [...] esta es una reunión informal, no una reunión de estado, y la presentación del Incremento es con la intención de obtener retroalimentación y fomentar la colaboración (Schwaber et al., 2017 p. 13).

4. Sprint Retrospective

Según lo que explicaron Álvarez et al. Acerca de la definición de la retrospectiva:

Reunión que tiene lugar al finalizar cada iteración del ciclo Scrum. Estas reuniones tienen como objetivo analizar la manera en que el equipo está trabajando para detectar todo aquello que no es útil para eliminarlo o modificarlo, así como para potenciar y maximizar aquello que si lo es (2012, p. 336).

Schwaber et al. dijo:

La Retrospectiva Sprint ocurre después de la Revisión del Sprint y antes de la próxima planificación del Sprint. Esta es una reunión de tres horas en el tiempo para Sprints de un mes. Para sprints más cortos, el evento es generalmente más corto. El Scrum Master asegura que el evento tenga lugar y que los asistentes entiendan su propósito (2017, p. 12).

Otra definición de la retrospectiva dadas por Schwaber et al. indicó que :

La Retrospectiva de Sprint es una oportunidad para que Scrum Team se inspeccione a sí mismo y cree un plan para las mejoras que se promulgarán durante el próximo Sprint. La Retrospectiva Sprint ocurre después de la Revisión Sprint y antes de la próxima Planificación Sprint. Esta es como máximo una reunión de tres horas para Sprints de un mes. Para sprints más cortos, el evento es generalmente más corto. El Scrum Master asegura que el evento tenga lugar y que los asistentes entienden su propósito (2017 p. 14).

Proceso general de Scrum

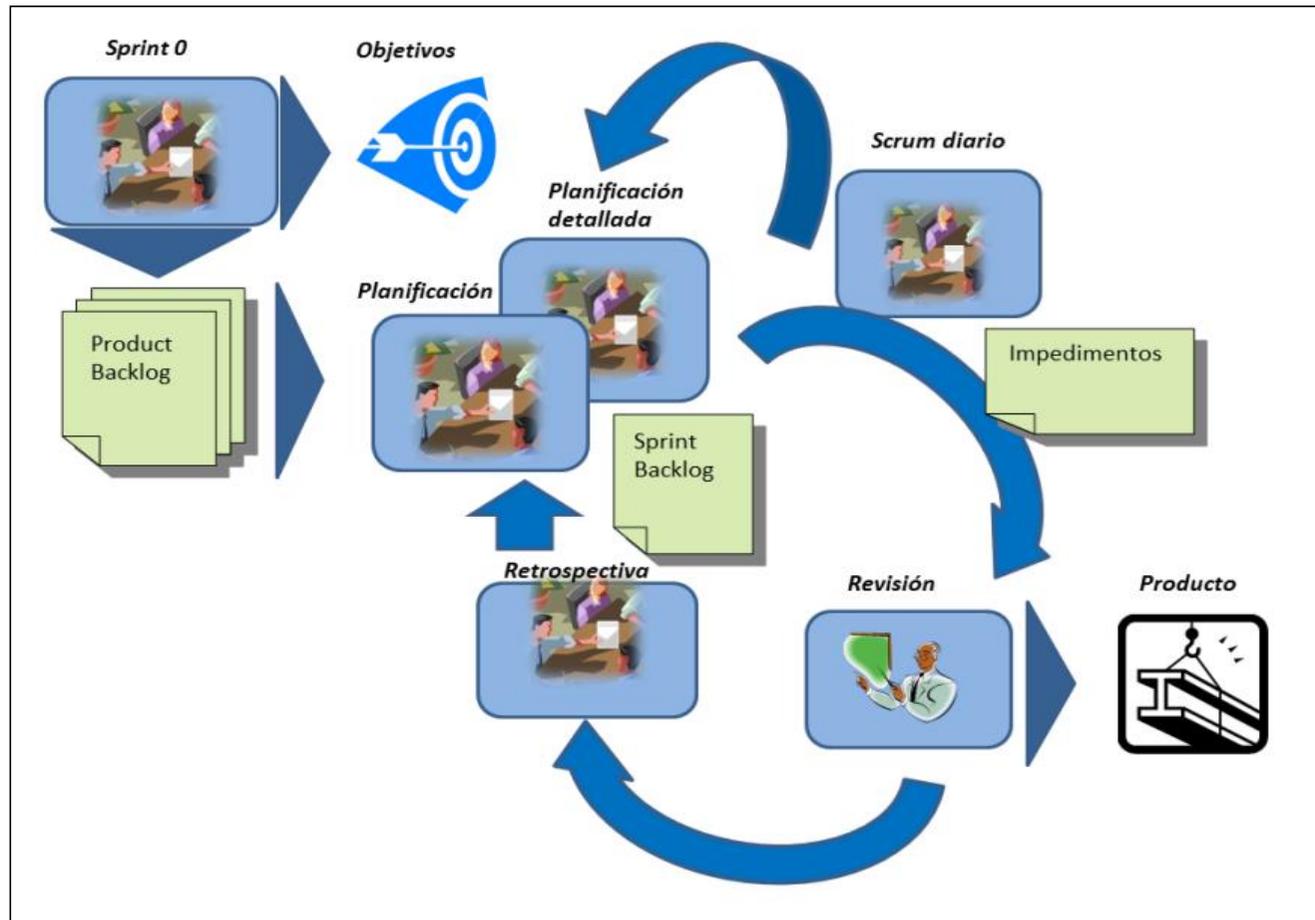


Figura 7: Scrum – Ciclo de Aplicación (Álvarez, et al., 2012 p. 86)

2.2.6 Ingeniería del Software

Software

En el libro Ingeniería de Software de Sánchez (2012 p. 12) describió que el “software es el conjunto completo de programas, procedimientos y documentación relacionada que se asocia con un sistema, y especialmente con un sistema de computadora. En un sentido específico, software son los programas de computadora”. Indicó también que: “El proceso de software es un conjunto coherente de políticas, estructuras organizativas, tecnológicas, procedimientos y artefactos que se necesitan para concebir, desarrollar, implantar y mantener un producto software “. (Sánchez, 2012 p. 34).

Mejora de procesos de software

Según Marie (2012, p. 1) la mejora de procesos en software consiste en la modificación sistemica y continua de un proceso software para reducir costos de producción, mejorar el cronograma y aumentar la calidad del producto, estas mejoras se guían por pautas de mejora predefinidas que se usan para provocar la mejora en los procesos con medidas aplicadas durante cada iteración.

Proceso del Ciclo de Vida

Según explicó Sánchez (2012, p. 22) El ciclo de vida de un “[...] producto o proyecto software es la evolución del mismo desde el momento de su concepción hasta el momento en que deja de usarse, y puede describirse en función de las actividades que se realizan dentro de él”. Referente al mismo tema, Lledó (2009, p. 35) explicó que: “El ciclo de vida del proyecto se refiere a las distintas fases del proyecto [...] cada fase del proyecto por lo general termina con un entregable que habilita o no a continuar con la siguiente fase”.

Modelos del Ciclo de Vida de Software

1. Modelo en Cascada

Salvador Sánchez (2012, p. 40) dijo que el ciclo de vida en cascada es el más ampliamente difundido dentro de las metodologías clásicas, su característica única es llevar a cabo diferentes fases en forma secuencial, cada una de ellas comienza en cuanto termina la anterior. Los especialistas del (Project Management Institute Inc, 2013, párr 18) explicaron el siguiente concepto acerca del modelo en cascada, o comúnmente llamadas metodologías tradicionales:

[...] Las metodologías tradicionales se centran en la planificación, mientras que las ágiles ponen su foco en la ejecución [...] la una posee líneas base de tiempo, coste y alcance, la otra es adaptable al cambio, y su restricción principal es la calidad del producto y la rapidez.

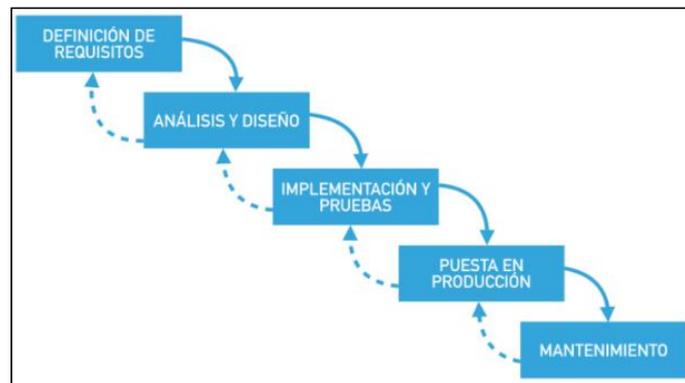


Figura 8: Modelo en cascada – Sánchez (2012, p.40)

Algunas ventajas del modelo cascada explicados por Salvador Sánchez (2012, p. 40) indicaron que el estado del proyecto es visible, por la progresión secuencial de las fases, los entregables se asocian a las fases, esto facilita la gestión, las desventajas del modelo en cascada descritas por Sánchez (2012, p. 40) fueron que las limitaciones de las fases son rígidos, las revisiones de los entregables entre fases se complica y la problemática del análisis se traslada a la fase de desarrollo al igual que la fase de pruebas donde estas etapas

ya no pueden repetirse. Referente a estas afirmaciones del modelo tradicional, Marie indicó que “el desarrollo de software siempre ha estado plagado de costos y calendarios excesivos y comprometió la calidad del producto (2012, p. 16).

2. Modelo en “V”

Este modelo del ciclo de vida es la mejora del tradicional modelo en cascada, Sánchez (2012, p. 41) nos explicó que aquí se enfatizan mucho lo que son actividades de validación y verificación, además de sus ventajas, es que le da mucho énfasis a la parte de la prueba de funcionalidades, de todos modos, las fases continúan su ritmo secuencial en adelante, en conclusión, el modelo en “V” es solo una variante del modelo en cascada.

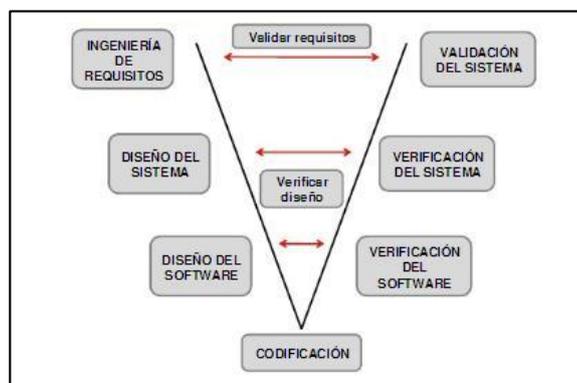


Figura 9: Modelos en “V” – Sánchez (2012, p.41)

3. Modelo Espiral

Sánchez (2012, p. 41) consideró que este modelo de ciclo de vida es considerado por poseer mayores relaciones con el uso de prototipos y de los modelos de ciclo de vida en cascada, este modelo está enfocado en evaluar de manera constante los riesgos en el proceso de desarrollo de software para poder mitigar y a su vez tratar de evitar los las posibles complicaciones en el desarrollo del producto.

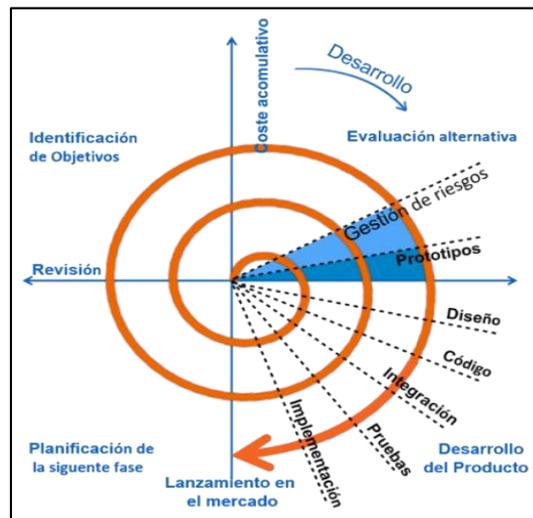


Figura 10: Modelo espiral – Sánchez (2012, p.45)

Procesos de Desarrollo de Software

Un proceso software determina, guía y define como se deben realizar las actividades del desarrollo del producto o software, una actividad, según Sánchez: “es la unidad mínima de trabajo, la cual tiene una duración definida, está relacionada lógicamente con otras actividades del proyecto, consume recursos y tiene generalmente un coste asociado” (2012, p. 46). Los procesos también se pueden describir bajo 3 elementos principales:

- **Actividades.** Para Sánchez (2012, p. 47) Las actividades y sus diferentes niveles de detalle, la descomposición de actividades en sub actividades y las actividades puntuales que son los denominados hitos.
- **Quien hacen las actividades.** Sánchez (2012, p. 47) en esta parte se refiere a que cada actividad debe ser asignada a un colaborador o persona ya que cada persona en algún momento podría actuar con un rol distinto, depende de la fase en la que se encuentre.
- **Uso de las actividades y sus beneficios.** Sánchez (2012, p. 47) Considera que aquí se especifican las salidas y entradas de cada actividad, esto marca también los parámetros y delimitaciones, se define el fin de un trabajo o producto concluido.

1. Iteraciones o Incrementos

Sánchez (2012, p. 52) indicó que estos procesos se suelen asociar mucho con el modelo de ciclo de vida en espiral, se puede decir que cada vuelta de la espiral es una iteración y cada incremento es un agregado de tipo funcional para el software final. Algunos de los beneficios de los procesos incrementales que mencionó Sánchez (2012, p. 52) tienen estas características, el software se entrega en etapas mucho antes que un proyecto en cascada, la calidad del producto mejora en cada iteración y hay oportunidades para la mejora, además, ayuda en proyectos donde se esté en fase de aprendizaje o uso de tecnologías nuevas.

Algunas desventajas de proceso incremental que menciona Sánchez (2012, p. 52) fueron por ejemplo; que no es de mucha utilidad en procesos cortos ni donde la funcionalidad está completamente definida puesto que el tiempo de planificación sería desaprovechado, acá existe riesgo que la funcionalidad no termine por ser un proceso abierto a cambios sin límites, además la pieza clave de los desarrollos incrementales es saber usar y definir bien los incrementos en los que se va enfocar el equipo de desarrollo.

2. Procesos estructurados y procesos orientados a objetos

Sánchez (2012, p. 56) dijo que estos procesos están basados en el concepto de abstracción funcional en contraposición con la tradicional abstracción orientada a objetos, esta diferencia solo muestra un sesgo entre los métodos usados por la ingeniería de software mas no afecta al proceso en sí mismo como ordenación de esas actividades.

3. Procesos Ágiles

Los procesos ágiles según Sánchez (2012, p. 56) son aquellos procesos que enfatizan en gran manera los procesos iterativos y además de un proceso adaptable al negocio, se caracterizan especialmente porque reducen gran parte de la documentación y procedimientos, Según Sánchez (2012, p. 56) algunas características de los procesos ágiles comprobaron la entrega rápida y oportuna al Cliente de una aplicación o programa funcional después de cada iteración, además, no hay documentación de especificaciones o análisis, solo en el código o la documentación es mínima.

Sánchez (2012, p. 56) explicó que en los procesos ágiles la planificación es bastante más corta que en los procesos tradicionales, esto es porque el marco de trabajo trabaja con iteraciones cortas, las principales ideas o principios de la agilidad ha quedado suscrito en una declaración de manifiesto ágil, algunos de estos son:

- Satisfacción del cliente con entregas cortas.
- El software se entrega en semana en lugar de meses.
- Los cambios nuevos en el progreso del desarrollo son bienvenidos.
- Colaboración y comunicación cercana entre el equipo del proyecto y los Usuarios.
- Equipos auto organizados.
- Equipos auto gestionados.
- El proceso es cambiante y además se adapta al negocio.

4. Procesos basados en componentes

La definición de estos procesos para Sánchez (2012, p. 58) tienden a diferenciar las partes funcionales de software y categorizarlas de la misma forma como la industria electrónica categoriza sus componentes por funcionalidades, en software también los sistemas están preparados de esa forma para trabajar con otras aplicaciones. Sánchez también explicó que: “Un componente es una pieza de software auto-contenida, que ofrece unos servicios definidos en sus

interfaces y que está preparado para integrarse en otras aplicaciones”. (Sánchez, 2012 p. 58).

Finalmente, para Sánchez (2012, p. 59) explicó que los componentes son fuentes o paquetes compilados de funcionalidades, estos componentes tienen varios beneficios puesto que se pueden vender por separado para que sean usados por otros sistemas, en ese sentido cada funcionalidad ya ha sido probada y puede mejorarse, así como también la confiabilidad de cada componente es mayor.

2.6.7 Medidas en la Ingeniería de Software

1. Medida en el tamaño de los sistemas

“La cuenta de las líneas de código (Loc – Lines of Code) es una de las técnicas más usadas, especialmente por su facilidad y simplicidad de cálculos” (Sánchez, 2012 p. 75).

2. Medida en la complejidad del Software

Para hallar la complejidad del software Sánchez se basa en la secuencia o sucesión de caminos lógicos referidos especialmente al código fuente, lo explicó de la siguiente manera:

La complejidad ciclomática de McCabe $v(g)$ se basa en la cuenta del número de caminos lógicos individuales contenidos en un código de programa. Para calcular la complejidad ciclomática, el programa (o fragmento de programa) se representa como un grafo cuyos nodos son las instrucciones, y los posibles caminos sus aristas [...] donde e representa el número de aristas y n el número de nodos, esto es el número de posibles caminos del código (2012 p. 76).

$$v(g) = e - n + 2$$

3. Medida de la documentación

En este apartado Sánchez explicó ciertos criterios para poder medir la documentación en el proceso de desarrollo de software.

Las métricas de la documentación, por ejemplo, permiten precisar la documentación generada en cada una de las distintas fases del ciclo de vida. [...] Existen además otras métricas provenientes de la lingüística, más generales y no específicas de la ingeniería de software, que permiten medir por ejemplo la legibilidad de un documento (2012 p. 80).

3.1 Índice de Gunning Fog

$$Fog = 0.4 \times \frac{Nro\ Palabras}{Nro\ Frases} + 100 \times \frac{(Nro\ Palabras\ Complejas)}{Nro\ Palabras}$$

Acerca de este índice de Fog Sánchez dijo:

[...] donde se consideran las *Palabras Complejas*, las de 3 o más sílabas. Las métricas toman valores entre 1 y 12 para representar el nivel de educación necesario –en el sistema norteamericano- para comprender un texto. Aun no conociendo este sistema la interpretación es sencilla: cuanto menor el índice, más fácil de entender el texto (Sánchez, 2012 p. 80).

3.2 Índice de facilidad de lectura de Flesh

Según Sánchez (2012, p. 80) acerca de este índice dice que lo señala como de comprensión fácil de un escrito a través de la siguiente ecuación:

$$Flesh = 206,835 - 1,015 \times \frac{total\ palabras}{total\ frases} - 84,6 \times \frac{(total\ sílabas)}{total\ palabras}$$

Para el cálculo de este índice Sánchez explicó lo siguiente:

Este índice, cuyo rango oscila entre 0 y 100, se interpreta del siguiente modo: cuanto menor es el índice, más difícil será la comprensión del texto. Valores superiores a un 80% o 90% indican que el texto puede ser comprendido por la práctica totalidad de la población (2012 p. 81).

4. Medida de Reutilización

Acerca de la medida de reutilización Sánchez indicó lo siguiente:

La reutilización de documentación, diseños, códigos, casos de prueba, etc., es de primordial importancia a la hora de desarrollar nuevos proyectos. [...] Es frecuente clasificar los módulos, clases, etc., mediante rangos nominales con valores como completamente reutilizado, ligeramente modificado (si el número de líneas es menos del 25%), muy modificado (si el número de líneas modificadas es más del 25%), y nuevo (cuando el módulo, función o clase es completamente nuevo). Sin embargo, es más común emplear como métrica general de la reutilización el porcentaje de reutilización de líneas de código, que se calcula de la siguiente manera: (Sánchez, 2012 p. 81).

$$\text{Porcentaje de reutilización} = \frac{\text{Loc reutilizados}}{\text{Total LoC}} \times 100$$

5. Medidas de la eficiencia

La eficiencia del software se puede medir según lo establece Sánchez, lo describió de la siguiente manera:

Si lo que se desea es medir la eficiencia examinando únicamente el código del programa (es decir, mediante, métricas internas), podemos contar el número de operaciones que efectúa el algoritmo dada una entrada. Esta métrica se ha formalizado matemáticamente en lo que se denomina orden (O Grande) de un algoritmo (2012 p. 81).

6. Medidas con los recursos

Sánchez (2012, p. 84) manifestó que cuando se habla de medidas de recursos son directamente relacionadas con medidas de productividad, se define en muchos casos de manera cuantificable, la productividad para la ingeniería de software se muestra de la siguiente manera:

$$\text{Productividad} = \frac{\text{Tamaño}}{\text{Esfuerzo}}$$

Sánchez (2012, p. 84) indicó que el tamaño normalmente se da por el número de líneas de código o por el número de puntos de función, el esfuerzo casi siempre se representará por la cantidad de personas por mes, existen métricas relacionadas con la estructura y comunicación, así lo mencionó Sánchez (2012, p. 85), para ello enunció la técnica de Hewlett-Packard para medir la complejidad de las comunicaciones. Estas métricas representan a los miembros como nodos y las comunicaciones como aristas, esta combinación sirve para definir las siguientes métricas.

- El tamaño de personas por equipo.
- La densidad de las comunicaciones, proporción del tamaño del equipo y número de comunicaciones.
- Líneas de comunicación.
- El nivel de comunicación.

$$m(G) = \frac{2 \cdot (e - n + 1)}{(n - 1) \cdot (n - 2)}$$

Sánchez (2012, p. 85) explicó que de acuerdo a los resultados de estas métricas se puede visualizar cual es el nivel de complejidad de las comunicaciones de los integrantes del equipo, para la toma de decisión se puede determinar qué acciones tomar para mejorar la comunicación dentro del equipo de proyecto.

7. Métricas de los requisitos

Sánchez (2012, p. 158) explicó que es posible tomar medidas de los requisitos de un proyecto para que nos muestren el estado del alcance de un determinado proyecto software resulta útil para la realización de planes de proyecto, predicción de alternativas, riesgos e identificación de resultados a obtener, algunas medidas para requisitos se dan a continuación:

Numero de requisitos nuevos: Según Sánchez (2012, p. 158) estos se calculan entre el número de cambios realizados y el total de requisitos, se resulta normalmente en términos porcentuales.

$$\text{Porcentaje de requisitos nuevos} = \frac{\text{Cambios en los requisitos}}{\text{Requisitos totales}}$$

Cobertura de requisitos: Para Sánchez (2012, p. 159) indicó que es el porcentaje de requisitos incluidos en la línea base y que fueron soportados por algún componente que se ha incluido en el diseño detallado del sistema.

$$\text{Cobertura de requisitos} = \frac{\text{Requisitos soportados}}{\text{Requisitos en la línea base}}$$

Requisitos derivados: Para Sánchez (2012, p. 159) este indicador muestra la desviación de los requisitos que nacen a partir de los requerimientos iniciales y que por motivos de algún cambio en el negocio fueron modificados o añadidos,

$$\text{Tasa de requisitos derivados} = \frac{\text{Requisitos derivados}}{\text{Requisitos iniciales}}$$

Requisitos incompletos: Sánchez (2012, p. 160) explicó que este indicador nos muestra la tasa de requisitos que no han sido completados, la falta de cumplimiento de los requisitos puede causar efectos negativos en el transcurso del proceso de desarrollo ya que no tendríamos un diseño completado.

$$\text{Tasa de requisitos incompletos} = \frac{\text{Requisitos TDB}}{\text{Requisitos totales}}$$

Requisitos probados: Sánchez (2012, p. 160) describió que esta métrica se puede usar para determinar cuánto del alcance ya ha sido completado en un estado determinado del proyecto.

$$Tasa\ de\ requisitos\ probados = \frac{Requisitos\ probados}{Requisitos\ totales}$$

Sánchez (2012, p. 161) dijo que los motivos para poder hacer uso de las métricas de los requisitos varían, puesto que todas las prácticas de este tipo pueden ayudar a la gestión de procesos de desarrollo en los siguientes puntos:

- Ahorre de coste de especificación y desarrollo minimizando el impacto de errores.
- Mejora en la calidad mediante un adecuado seguimiento de los requerimientos.
- Aumento de la productividad.
- Facilidad de administración sobre las especificaciones de Usuario.
- Apoyo en el cumplimiento de los estándares de calidad.

8. Métricas de construcción

Sánchez (2012 pág. 267) indicó que para el proceso de construcción del producto software se puede usar las siguientes métricas:

Código desarrollado: Según Sánchez:

Se puede englobar dentro de las medidas de productividad. La más común es la cuenta del número de las líneas de código fuente entregadas. También usa el número de instrucciones de código objeto entregado o el número de páginas de la documentación del sistema (2012, p. 267).

Código reutilizado: Según Sánchez:

Conocer cuánto código ha sido creado de nuevo y cuanto procede de desarrollo anteriores [...] permite mejorar las estimaciones sobre el esfuerzo necesaria en la construcción de cara a desarrollos posteriores (2012, p. 267).

Código destruido: “El código eliminado o comentado puede utilizarse si está utilizando una herramienta de control de versiones” (Sánchez, 2012, p. 267).

Esfuerzo de construcción: “suele estimarse a partir del tamaño del código desarrollado (habitualmente líneas de código) y el tiempo a emplear en la construcción” (Sánchez, 2012, p. 268).

9. Medidas durante las pruebas

Sánchez (2012, p. 318) dijo que estas medidas se realizan durante la etapa de las pruebas de desarrollo de software y sirven para analizar errores, medidas de densidad de fallos, esto determinará si es necesario seguir con las pruebas o se toma otra decisión.

- **Dirección de las pruebas:** Sánchez (2012, p. 318) explicó que para poder ayudar en la planificación y diseño de las pruebas se pueden usar esas medidas, entran dentro de las métricas del tamaño, por ejemplo el número de líneas de código, o las que miden la estructura de un producto software (su complejidad).
- **Tendencia que muestran los casos de prueba:** Sánchez (2012, p. 318) describió que en esta medida se pueden usar los casos de prueba ejecutados o el número de defectos pendientes de resolver.
- **Aumento de la efectividad:** Sánchez (2012, p. 318) refirió que estas métricas permiten hacer más precisos los futuros desarrollos y se pueden hacer mediante la clasificación de sus causas, su severidad, la etapa del desarrollo en la que se detectó, etc.

- **Densidad de fallos:** Para Sánchez (2012, p. 318) dijo que la densidad de fallos se puede determinar mediante la cantidad de fallos encontrados, clasificarlos por su tipo y calcular el valor de su densidad.
- **Conclusión de las pruebas:** Para esta medida según Sánchez (2012, p. 318) dijo que se pueda hacer uso determinando si la prueba se concluyó satisfactoriamente o no, para esto es importante hacer uso de la fiabilidad del software que se está usando.

10. Métrica del producto

Para las métricas, Sánchez (2012, p. 357) explicó que desde el punto de vista del mantenimiento del software se puede hacer uso de la medida para el producto, en esta se pueden tomar datos como el tamaño, la complejidad, o algunas características de su diseño, asimismo, Sánchez (2012, p. 357) también dijo que se puede hacer uso de la medida para el proceso en el mantenimiento, en ese ámbito se pueden hacer uso de factores como la eficacia de eliminar defectos en etapa de desarrollo, patrones de aparición de defectos durante las pruebas o el tiempo fijo de respuesta del proceso. Sánchez (2012, p. 318) dijo que para calcular el índice de madurez del software es necesario antes tener algunas medidas previas, las cuales son:

M_t = número de módulos de la versión actual.

F_m = número de módulos de la versión actual que han sido modificados.

F_a = número de la versión actual que han sido añadidos

F_e = número de módulos de la versión anterior eliminados de la versión actual

Finalmente Sánchez (2012, p. 318) suscribió que luego de este producto, el número de la madurez (IMS) se obtiene de la siguiente manera:

$$IMS = \frac{M_t - (F_a + F_m + F_e)}{M_t}$$

11. Método del valor conseguido

Según Sánchez (2012, p. 451) este método es conocido por sus siglas en inglés EMV (Earned Value Management) que se usó en el departamento de defensa de los estados unidos para el control de proyectos, la forma como se realiza el seguimiento es integrando información de plazos y coste, de esta manera mediante una gráfica se observa la evolución del proyecto en el tiempo. Sánchez (2012, p. 451) dijo que el “Coste presupuestado o coste presupuestado del trabajo planeado. (Budgeted cost of work Schedule, BCWS). Se calcula sumando los costes planificados del proyecto”.

- Valor actual o Costo real del trabajo realizado (Actual Cost of work Performance, ACWP). Representa el esfuerzo invertido hasta la fecha, es decir, los costes realmente gastados en las tareas llevadas a cabo en el proyecto hasta la fecha” (Sánchez, 2012 p. 451).
- Valor conseguido (Earned Value) o coste presupuestado del trabajo realizado (Budgeted Cost of Work Performed, BCWP): Representa el grado de finalización de las distintas tareas. Se calcula sumando los costos planificados de las tareas que realmente se han llevado a cabo hasta la fecha. (Sánchez, 2012, p. 451).

a) Ratio del Costo:

$$CR = \frac{(BAC + MR)}{BAC}$$

b) Ratio del plazo:

$$SR = \frac{(BAC + SR)}{BAC}$$

Índices: Según Sánchez (2012, p. 451) dijo que estos indicadores nos muestran cuando estamos tan cerca de los valores planificados, refiriéndonos al costo y plazo. La ventaja de estos índices es que pueden

comparar proyectos independientemente de su tamaño, algunos que podemos mencionó son:

- Índice de eficiencia del costo (Cost Performance Index, *CPI*)

$$CPI = \frac{BCWP}{ACWP}$$

- Índice de eficiencia del plazo (Schedule performance Index, *SPI*)

$$SPI = \frac{BCWP}{BCWS}$$

- Índice de coste y plazo (Cost-Schedule Index, *CSI*)

$$CSI = CPI * SPI$$

Se incluyen también los indicadores para hallar la eficacia, según Sánchez el “índice de eficacia para la conclusión (To Complete Performance Index, *TCPI*): es el cociente entre el trabajo restante y el presupuesto restante (equivale a *CSI*)” (2012, p. 453).

$$TCPI = \frac{(BAC - BCWP)}{(EAC - ACWP)}$$

2.6.8 Análisis comparativo Metodologías Ágiles vs. Tradicionales

Standish Group

“El Standish Group es una organización consultiva investigación primaria que se centra en el rendimiento del proyecto de software. Utilizando nuestra extensa investigación primaria que puede mejorar sus inversiones en proyectos de software” (Standish Group, 2015, párr 3).

Informe del Chaos Report

Lynch respecto del reporte del Caos enunció que este ha venido desarrollando a lo largo del tiempo para dar a conocer los resultados de los diferentes proyectos de software, se destacan resultados importantes que ayudarán a futuro en las diferentes investigaciones:

El informe del Caos viene siendo publicado por Standish group desde 1994 dando una visión sobre el fracaso o éxito de los proyectos. En el informe del año 2015 han estudiado unos 50.000 proyectos de todo el mundo desde mantenimientos pequeños hasta gigantescos proyectos de reingeniería (2015, párr. 2).

Entrando a detalle de la comparativa, Lynch (2015, párr. 14) indicó que hay diferencia entre “[...] la comparativa del éxito de los proyectos en función de la metodología seguida para su desarrollo: ágil vs predictiva [...] por los datos presentados, los proyectos ágiles son mucho más éxitos que los no ágiles”.

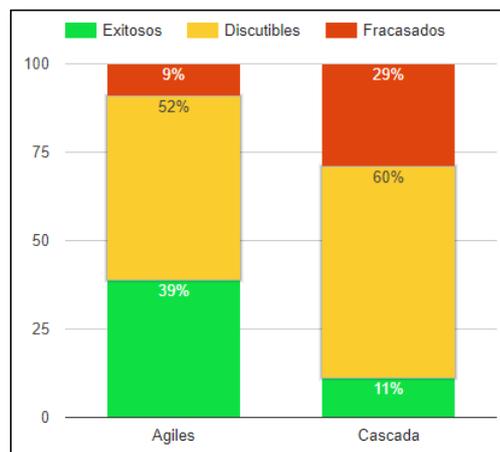


Figura 11: Ágiles vs cascada – Lynch (2015, párr. 14)

a) Ágiles vs cascada - División por tamaños

“Si ahora desglosamos los proyectos no sólo por metodología sino además por tamaño, el dato obtenido es realmente interesante” (Lynch, 2015 párr. 22).

1. Proyectos pequeños

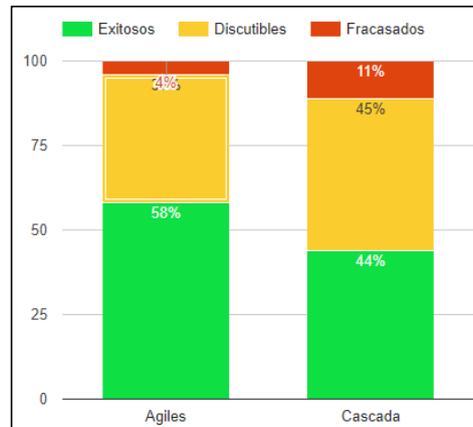


Figura 12: Pequeños - ágiles vs cascada - Lynch (2015, párr. 22)

2. Proyectos grandes

“Aquí la diferencia no es tan grande pero es evidente. Los proyectos ágiles siguen siendo los más exitosos aunque por un margen menor. Ahora bien cuando subimos en tamaño hacia proyectos grandes la diferencia se vuelve mucho más grande” (Lynch, 2015, párr. 23).

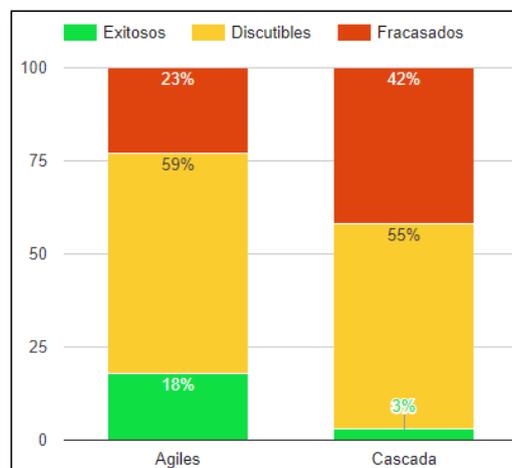


Figura 13: Grandes - ágiles vs cascada - Lynch (2015, párr. 23)

En la figura 14 se muestra la comparación de resultados obtenido por proyectos de software que fueron gestionados con metodología ágiles y tradicionales, Según el Chaos Report Lynch explicó lo siguiente:

Con la adopción de métodos de desarrollo ágiles en los últimos años, fue posible comparar los resultados de proyectos entre proyectos de cascada, ágiles y tradicionales. En todos los tamaños de proyectos, los enfoques ágiles dieron como resultado proyectos más exitosos y fallas menos rotundas, como se muestra en esta tabla (2015, párr. 4).

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

Figura 14: Metodologías ágiles versus Cascada - Lynch, 2015 Chaos Report.

Respecto de las diferencias entre metodologías ágiles y tradicionales, Maida et al explicó en su Tesis de Metodología de Desarrollo de Software lo siguiente:

En las metodologías tradicionales el principal problema es que nunca se logra planificar bien el esfuerzo requerido para seguir la metodología. Pero entonces, si logramos definir métricas que apoyen la estimación de las actividades de desarrollo, muchas prácticas de metodologías tradicionales podrían ser apropiadas. En la tabla que se muestra a continuación aparece una comparativa entre estos dos grupos de metodologías (2015, p. 18).

En la figura 15 se puede apreciar las diferencias que Maida considera entre las metodologías ágiles y tradicionales, en el detalle de la imagen se aprecia doce diferencias importantes que caracterizan una metodología de la otra.

Metodologías ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos
Poca documentación	Documentación exhaustiva
Muchos ciclos de entrega	Pocos ciclos de entrega

Figura 15: Comparativa Ágiles vs Tradicionales. Maida et al. (2015, p. 19)

2.6.9 Elección del marco de trabajo Scrum

En la revista Científica de la Universidad San Marcos, Tinoco Gómez, et al. (2010 p. 73) realizó un estudio que muestra la preparación de un cuadro resumen con los resultados de la selección, se han evaluado 6 metodologías ágiles para su posible uso en la gestión o desarrollo de proyectos informáticos. Para cada metodología evaluada, se ha colocado la puntuación que se ha obtenido de la clasificación. El sumatorio total de cada clasificación determina que Scrum, es la metodología que se debería usar, por tener una mejor puntuación.

Metodología	Mayor presencia en Internet	Mejor documentación	Certificadas y con training	Comunidades	Presencia empresarial	Proyectos de software	Total
Agile Project Management (APM)	2	1	3	5	1	1	11
Dynamic Systems development methods (DSDM)	1	3	5	5	4	4	22
Scrum	5	2	5	5	5	5	27
Test Driven Development	3	4	3	2	2	2	16
Extreme Programming (XP)	4	5	3	2	3	3	19
Total	15	15	19	19	15	15	95

Figura 16: Análisis Metodologías ágiles (Tinoco Gómez, et al., 2010 p. 73)

Finalmente, Tinoco et al. (2010, p. 736) concluyó que Scrum es la metodología que tiene mayor presencia, información documentaria, mayor presencia empresarial y es usado por proyectos de software, por lo tanto en el presente trabajo de investigación se utilizó como solución para el cumplimiento de los objetivos aquí descritos.

2.3 Formulación del Problema

General

¿Cuál es el efecto de la aplicación del marco de trabajo Scrum en el proceso de desarrollo de software en TeamSoft S.A.C?

Específicos

- a) ¿Cuál es el efecto de la aplicación del marco de trabajo Scrum en la productividad en TeamSoft S.A.C?
- b) ¿Cuál es el efecto de la aplicación del marco de trabajo SCRUM en la cobertura de los requisitos en TeamSoft S.A.C?
- c) ¿Cuál es el efecto de la aplicación del marco de trabajo SCRUM en la eficiencia del costo en TeamSoft S.A.C?

2.4 Justificación del Estudio

El presente trabajo de investigación se justifica en el ámbito teórico, práctico y metodológico, se describen a continuación:

1. Teórica

El presente trabajo de investigación se realizó con el propósito de ratificar las teorías referentes al marco de trabajo Scrum y cómo impactó en los procesos de la ingeniería de software para mejorar la productividad, cobertura de requisitos y eficiencia del costo. Lo que dijo Bernal (2010, p. 106) para esta justificación es que “cuando en una investigación se busca mostrar las soluciones de un modelo, está haciéndose una justificación teórica”, en ese sentido el modelo propuesto para la solución es el marco de trabajo Scrum.

2. Práctica

Este trabajo de investigación se realizó porque se tuvo la necesidad de mejorar los procesos del desarrollo de software en TeamSoft en base a los métodos que ofrece el marco de trabajo Scrum, Bernal (2010, p. 106) indicó que “se considera que una investigación tiene justificación práctica cuando su desarrollo ayuda a resolver un problema o, por lo menos, propone estrategias que al aplicarse contribuirían a resolverlo”, bajo este lineamiento, en el trabajo de investigación se aplicaron los principios, eventos, métodos, técnicas y artefactos que Scrum propone para resolver la problemática de la empresa.

3. Metodológica

En este ámbito, la aplicación del marco Scrum fue una nueva forma de trabajo que se dio en TeamSoft para el equipo de desarrollo de software, este nuevo método generó valor para la empresa y ayudó a mejorar aspectos en los equipos de trabajo como la auto-organización y auto-gestión los cuales son principios fundamentales de Scrum. Sobre este aspecto Bernal (2010, p. 107) nos dijo que “en investigación científica, la justificación metodológica del estudio se da cuando el proyecto que se va a realizar propone un nuevo método o una nueva estrategia para generar conocimiento valido y confiable”.

2.5 Hipótesis

General

La aplicación del marco de trabajo SCRUM mejoró el proceso de desarrollo de software en TeamSoft S.A.C. Los que Scrum propone según Álvarez et al. Es una forma de trabajar que tenga como base principal la mejora continua de procesos y llevarlos a la aplicación del trabajo en equipo (2012, p. 61). De la misma manera Malone sostuvo que Scrum ofrece un drástico cambio en los procesos de desarrollo de software para el costo-beneficio de las empresas obteniendo mejores resultados en el software (2014, p. 27).

Específicos

- a) La aplicación del marco de trabajo SCRUM mejoró la productividad en TeamSoft S.A.C. Álvarez et al. Sostuvo que Scrum propone el llamado Time-boxing que representa la manera estricta de abocarse específicamente a la tarea que se va desempeñar sin abordar otros temas, con esto se llega a completar el esfuerzo por alcanzar la mayor productividad (2012, p. 61).
- b) La aplicación del marco de trabajo SCRUM mejoró la cobertura de los requisitos en TeamSoft S.A.C, referente a este punto Álvarez et al. Indicó que Scrum deja de lado la idea de un proyecto con requerimientos estáticos, sino que por el contrario se adapta a los cambios con requisitos flexibles y variados y se aplica a todo el proceso (2012, p. 61).
- c) La aplicación del marco de trabajo SCRUM mejoró la eficiencia del costo en TeamSoft S.A.C. En el análisis de la metodología ágil Arana, Ruiz y La Serna (2015, p. 159) afirmaron que Scrum disminuyó el costo en el proyecto de software planteado con una eficiencia del costo de hasta un 65% en su realización.

2.6 Objetivos

General

Determinar cuál es el efecto de la aplicación del marco de trabajo Scrum en el proceso de desarrollo de software en TeamSoft S.A.C.

Específicos

- a) Determinar cuál es el efecto de la aplicación del marco de trabajo Scrum en la productividad en TeamSoft S.A.C.
- b) Determinar cuál es el efecto de la aplicación del marco de trabajo Scrum en la cobertura de los requisitos en TeamSoft S.A.C.
- c) Determinar cuál es el efecto de la aplicación del marco de trabajo Scrum en la eficiencia del costo en TeamSoft S.A.C.

II. MÉTODO

2.1 Diseño de la Investigación

El diseño de la investigación para este estudio es Experimental de tipo Pre-Experimental porque obtendremos resultados de causa y efecto por manipulación la variable independiente, asimismo trabajaremos con grupos que nos permitirán tener una visión clara y futurista del estudio realizado en contraste con el pre y post-test.

Hernández et al. Para el diseño experimental dijo:

[...] se refiere a un estudio en el que se manipulan intencionalmente una o más variables independientes (supuestas causas-antecedentes), para analizar las consecuencias que la manipulación tiene sobre una o más variables dependientes (supuestos efectos-consecuentes), dentro de una situación de control para el investigador. Esta definición quizá parezca compleja; sin embargo, conforme se analicen sus componentes se aclarara el sentido de la misma (2010, p. 121).

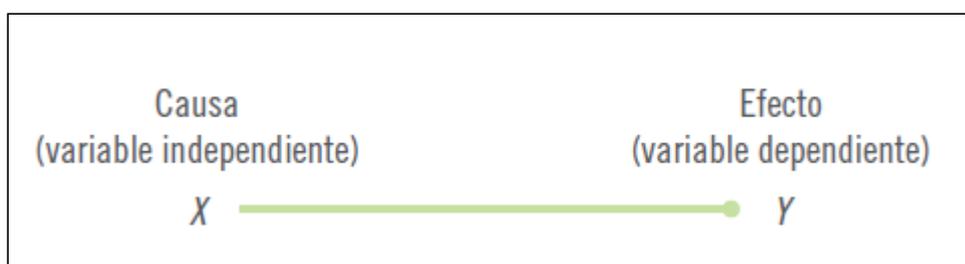


Figura 17: Causa – Efecto (Hernández, et al., 2010 p. 121)

El tipo de la investigación es Aplicada porque se contrastarán las hipótesis mencionadas en el presente estudio en base a las causas y efectos de las variables relacionadas, según Vargas (2009, p. 159) la investigación de tipo aplicada es también “entendida como la utilización de los conocimientos en la práctica, para aplicarlos en provecho de los grupos que participan en esos procesos y en la sociedad en general, [...] tiene como propósito hacer un uso inmediato del conocimiento existente”.

2.2 Variables – Operacionalización

Variable	Definición conceptual	Definición operacional	Dimensiones	Indicadores	Escala de medición
Aplicación de Scrum en el desarrollo de software en TeamSoft S.A.C.	El proceso de software es un conjunto coherente de políticas, estructuras organizativas, tecnológicas, procedimientos y artefactos que se necesitan para concebir, desarrollar, implantar y mantener un producto software". (Sánchez, 2012, p. 34)	Esta dimensión se refiere a la cantidad de Recursos que pueden lograr entregar al Usuario final un producto de software funcional en un determinado tiempo, mientras sea mayor la cantidad de entregables y la cantidad de recursos disminuya quiere decir que existe una mayor productividad.	Productividad	<ul style="list-style-type: none"> • Cantidad de entregables de software. • Cantidad de personas. $Productividad = \frac{Tamaño}{Esfuerzo}$	Razón
		El cumplimiento de los requerimientos funcionales para un proyecto de software es la base fundamental para obtener un software que cubra las necesidades del negocio. Los requisitos de software varían en función del beneficio del negocio y sus objetivos.	Cobertura de requisitos	<ul style="list-style-type: none"> • Requisitos soportados. • Requisitos en línea base. $Cob. de requisitos = \frac{Req. soportados}{Req. en la línea base}$	Razón
		El costo de los proyectos se mide en cada proyecto bajo un cronograma, los proyectos gestionados deben mantener el costo establecido desde el inicio del proyecto siendo su variabilidad mínima para poder tener una eficiencia esperada.	Eficiencia del costo	<ul style="list-style-type: none"> • Sumatoria de los costos planificados. • Sumatoria de los costos reales. $CPI = \frac{Costo Planificado}{Costo Real}$	Razón

Figura 18: Cuadro de Operacionalización de las variables

2.3 Población y Muestra

La población para la investigación estuvo constituida por nueve proyectos de software, los que formaron parte de la cartera de proyectos que tiene el área actualmente bajo su gestión. Según Francia (1988, p. 36) citado por Bernal (2010, p. 161): “la población es el conjunto de todos los elementos a los cuales se refiere la investigación. Se puede definir también como el conjunto de todas las unidades de muestreo”.

La muestra utilizada estuvo constituida por el total de la población. “La muestra es, en esencia, un subgrupo de la población. Digamos que es un subconjunto de elementos que pertenecen a ese conjunto definido en sus características al que llamamos población [...] con frecuencia leemos y escuchamos hablar de muestra representativa, muestra al azar, muestra aleatoria, como si con los simples términos se pudiera dar más seriedad a los resultados”. (Hernández, et al., 2010 p. 175).

2.4 Técnicas e Instrumentos de recolección De Datos

- Técnica utilizada: La Entrevista.
Para Tamayo y Silva (s.f, p. 5) “La entrevista es una situación de interrelación o dialogo entre personas, el entrevistador y el entrevistado”.
- Instrumento: Ficha de Registro.
Para Hernández (2011, p. 3) La Ficha de Registro “son los instrumentos de la investigación documental que permiten registrar los datos significativos de las fuentes consultadas”.

2.5 Validez y Confiabilidad del Instrumento

La validez utilizada en el presente estudio de investigación fue la validez de contenido, el objetivo fue focalizar las dimensiones objeto del estudio dentro del instrumento basándonos en las teorías relacionadas al tema para poder lograr el objetivo, según Hernández (2014, p. 201) indica que “La validez de contenido se refiere al grado en que un instrumento refleja un dominio

específico de contenido de lo que se mide. Es el grado en el que la medición representa al concepto o variable medida”, es ese sentido el constructo utilizado en el presente trabajo (**Ver. Anexo Nro. 2**) hacer referencia de manera contundente las dimensiones usadas las cuales son Productividad, Cobertura de requisitos y eficiencia del costo, cabe destacar que las pruebas estadísticas se realizaron con un nivel de confianza del 95%.

La confiabilidad no aplica en instrumento de tipo fichas de registro utilizado en el presente estudio de investigación (**Ver. Anexo Nro. 2**), así lo estableció Corral (2009, p. 18) donde indicó “[...] que existen instrumentos para recabar datos que por su naturaleza no ameritan el cálculo de la confiabilidad, como son entrevistas, escala de estimación, listas de cotejo, guías de observación, hojas de registros, inventarios, rubricas, otros”.

2.6 Métodos de Análisis de Datos

En la presente investigación se hace uso de enfoque cuantitativo como método de análisis de datos, este enfoque permitió centrarse en el tratamiento de datos numéricos para el postrer tratamiento estadístico de los resultados plasmado en el presente estudio, así lo estableció Hernández (2014, p. 4) el cual indicó que el enfoque cuantitativo “utiliza la recolección de datos para probar hipótesis con base en la medición numérica y el análisis estadístico, con el fin establecer pautas de comportamiento y probar teorías”.

2.6.1 Prueba de normalidad

Para el estudio se ha trabajado las pruebas e normalidad usando la información del Shapiro-Wilk porque nuestra muestra es menor a 50, así lo estableció Ortega donde dijo:

Cuando la muestra es menor, $n < 50$ se aplica la prueba de Shapiro-Wilk para la prueba de normalidad. Esta prueba ayuda a calcular la estadística de prueba W , que si llega a ser mayor al nivel de significancia o se asume que la distribución es normal, si no la distribución es no normal (2009, p.181).

2.6.2 Prueba de la hipótesis estadística

En esta prueba se realizó en el programa SPSS y se utilizó del método Wilcoxon cuando la significancia del test de normalidad arrojó un nivel menor a 0.05, más por el contrario cuando la significancia fue mayor a ese valor se usó el método T de Student.

2.6.3 Definición de las variables

- **la** = Indicador actual del proceso.
- **lp** = Indicador propuesto del proceso.

2.6.4 Hipótesis Estadística

Para el análisis de las hipótesis estadísticas se consideraron los siguientes niveles de significancia y confianza:

- Nivel de significancia máximo permitido: (α): 0.05
- Nivel de confianza: ($\gamma = 1 - \alpha$): 0.95

1. Hipótesis General

HG₀: La aplicación del marco de trabajo SCRUM no mejoró el proceso de desarrollo de software en TeamSoft S.A.C.

HG₁: La aplicación del marco de trabajo SCRUM mejoró el proceso de desarrollo de software en TeamSoft S.A.C.

2. Hipótesis Específicas

2.1 Productividad

- **HE1₀**: La aplicación del marco de trabajo SCRUM no mejoró la productividad en TeamSoft S.A.C.
- **HE1₁**: La aplicación del marco de trabajo SCRUM mejoró la productividad en TeamSoft S.A.C.

$$HE1 = I1a \leq I1p$$

2.2 Cobertura de Requisitos

- **HE2₀**: La aplicación del marco de trabajo SCRUM no mejoró la cobertura de los requisitos en TeamSoft S.A.C.
- **HE2₁**: La aplicación del marco de trabajo SCRUM mejoró la cobertura de los requisitos en TeamSoft S.A.C.

$$HE2 = I2a \leq I2p$$

2.3 Eficiencia del Costo

- **HE3₀**: La aplicación del marco de trabajo SCRUM no mejoró la eficiencia del costo en TeamSoft S.A.C.
- **HE3₁**: La aplicación del marco de trabajo SCRUM mejoró la eficiencia del costo en TeamSoft S.A.C.

$$HE3 = I3a \leq I3p$$

2.7 Aspectos Éticos

El investigador se compromete a respetar la veracidad de los resultados, la confidencialidad de los datos suministrados por la empresa donde se han recogido toda la información y la identidad de las personas que participan en el estudio de investigación.

III. RESULTADOS

3.1 Resultados de la estadística descriptiva

Las siguientes tablas muestran las medias de los indicadores de productividad, el resultado muestra un aumento de los indicadores por cada dimensión en el resultado del Post-Test, la interpretación se da de la siguiente manera:

3.1.1 Resultados del indicador de Productividad

El efecto de la productividad realizado en el Post-Test tiene un promedio de 0.79 y es mayor a la productividad realizada en Pre-Test el cual tiene un promedio de 0.39 efectuado en el presente estudio.

Tabla 1: Efecto de la productividad de Scrum

	N	Mínimo	Máximo	Media	Desviación estándar
Pre_Productividad	9	,20	,50	,3889	,10138
Post_Productividad	9	,40	1,20	,7944	,27889
N válido (por lista)	9				

Fuente: Elaboración propia

En la siguiente grafica se muestra los resultados porcentuales de la dimensión de la productividad en las etapas de post-test y pre-test.

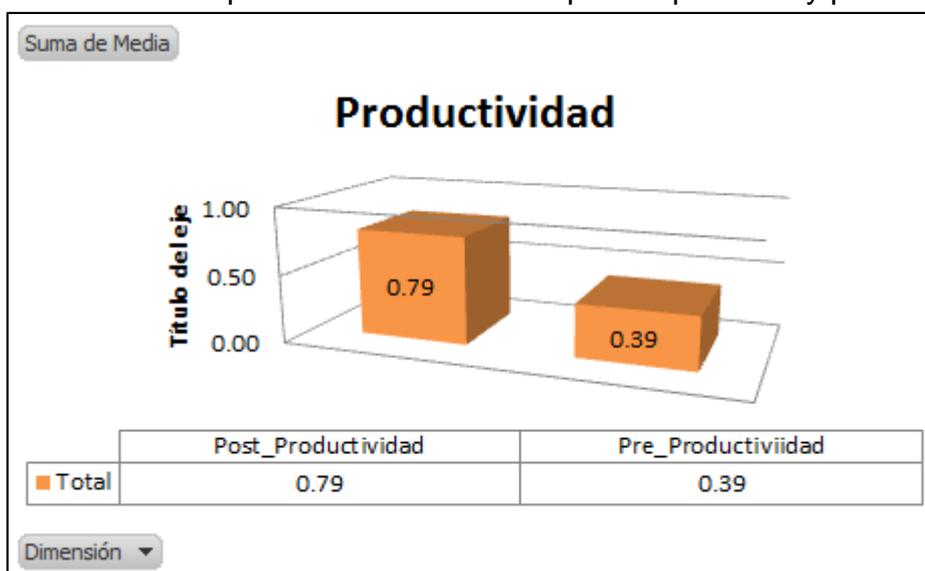


Figura 19: Resultados de la productividad pre y post test

3.1.2 Resultados del indicador de Cobertura de Requisitos

El efecto de la cobertura de requisitos realizado en el Post-Test tiene un promedio de 0.95 y es mayor a la cobertura de requisitos realizados en Pre-Test el cual tiene un promedio de 0.34 efectuado en el presente estudio, a continuación, se detalla los resultados estadísticos con el análisis para cada grupo.

Tabla 2: Efecto de la cobertura de requisitos de Scrum

	N	Mínimo	Máximo	Media	Desviación estándar
Pre_Requisitos	9	,00	1,00	,3356	,30657
Post_Requisitos	9	,80	1,00	,9528	,07546
N válido (por lista)	9				

Fuente: Elaboración propia

En la siguiente grafica se muestra los resultados porcentuales de la dimensión de la cobertura de requisitos en las etapas de post-test y pre-test.

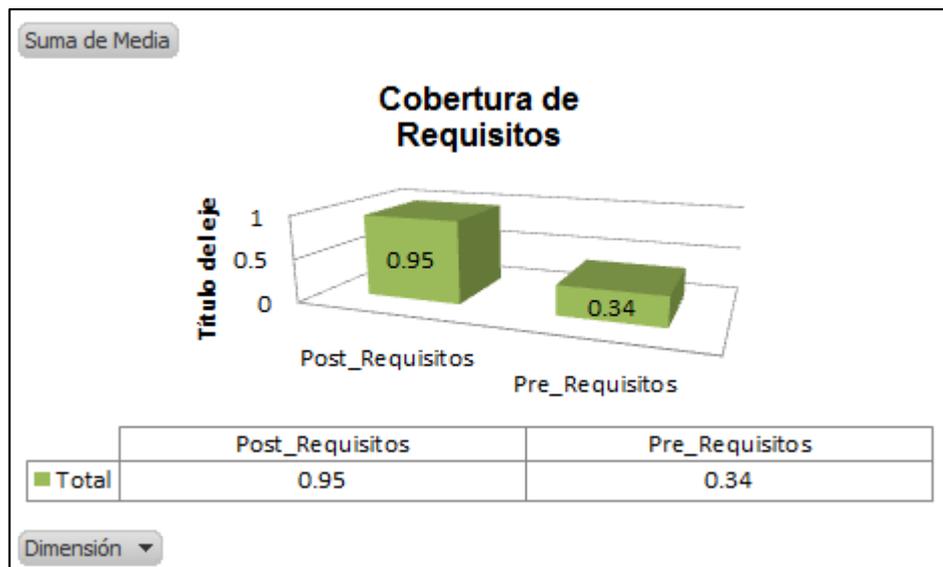


Figura 20: Resultados de la productividad pre y post test

3.1.3 Resultado del indicador de la eficiencia del Costo

El efecto de la eficiencia del costo realizado en el Post-Test tiene un promedio de 0.94 y es mayor a la eficiencia del costo realizada en Pre-Test el cual tiene un promedio de 0.83 efectuado en el presente estudio, en la siguiente tabla se puede observar los resultados estadísticos de las medias por cada grupo de datos procesados.

Tabla 3: Efecto de la eficiencia del costo de Scrum

	N	Mínimo	Máximo	Media	Desviación estándar
Pre_Costo	9	,57	1,00	,8259	,13173
Post_Costo	9	,64	1,00	,9385	,12015
N válido (por lista)	9				

Fuente: Elaboración propia

La siguiente grafica muestra los resultados porcentuales de la dimensión de eficiencia del costo, en las etapas de post-test y pre-test.

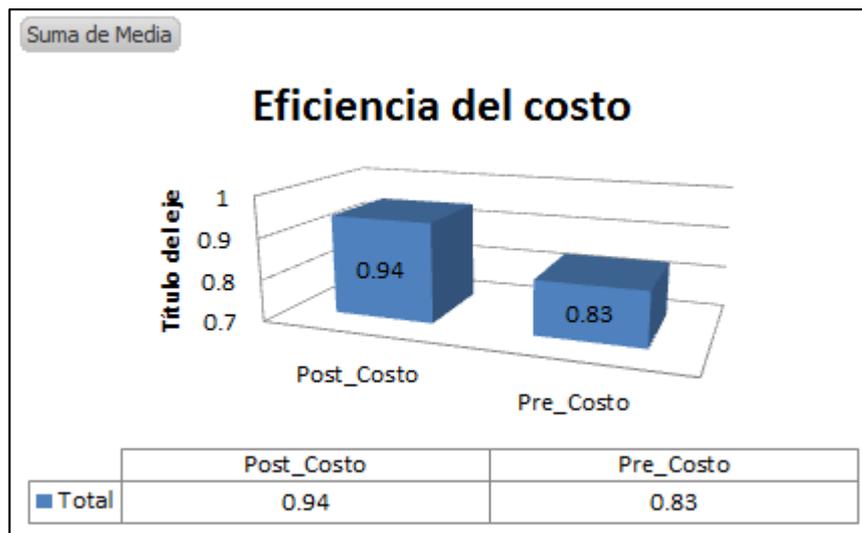


Figura 21: Resultados de la eficiencia del costo post y pre-test

3.1.4 Resultado general de las dimensiones

A continuación, se muestran los resultados de la estadística, se puede observar, en la figura 22 el resultado de las tres dimensiones con su respectivo indicador y luego en la figura 23 el resultado global de análisis.

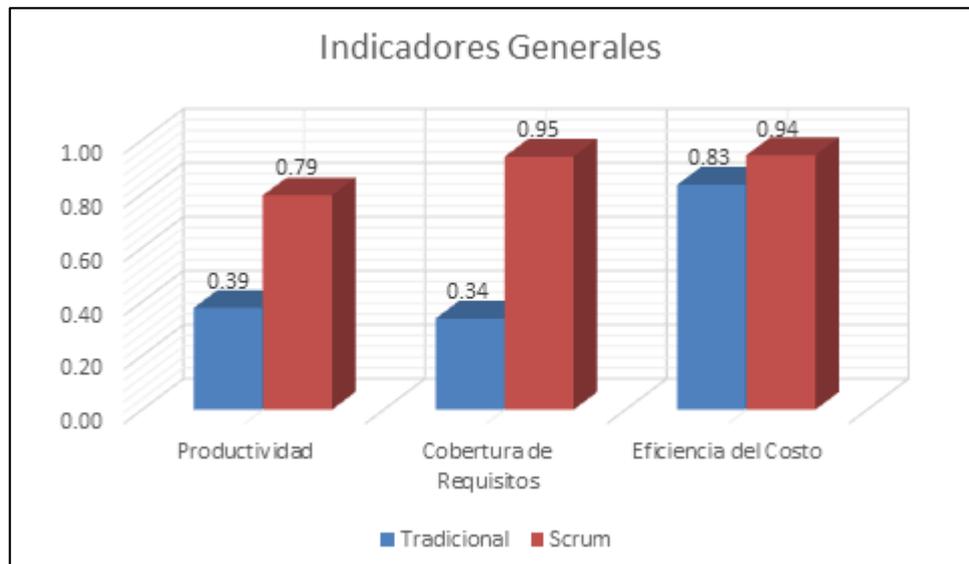


Figura 22: Resultados pre y post – Tradicional Vs. Scrum

La tabla 4 muestra el resultado estadístico de las tres dimensiones donde se aplicó las medias a los promedios para identificar en términos general los resultados de manera global.

Tabla 4: Resultado de las dimensiones generales y promedios estadísticos

	N	Mínimo	Máximo	Media	Desviación estándar
Tradicional	3	,34	,83	,5166	,27557
Scrum	3	,79	,94	,8922	,08433
N válido (por lista)	3				

Fuente: Elaboración propia

La grafica 23 muestra los resultados porcentuales de las tres dimensiones en las etapas de post-test y pre-test.

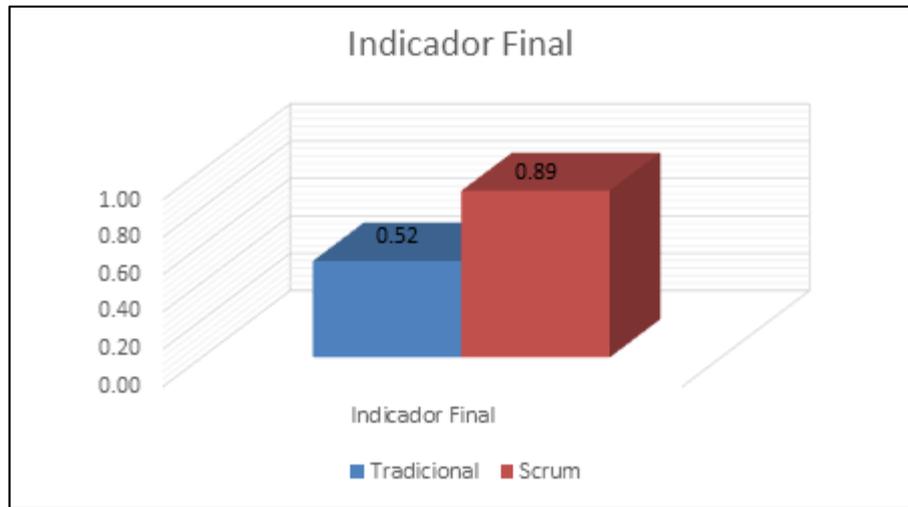


Figura 23: Indicadores finales de las dimensiones del proceso.

3.2 Prueba de Hipótesis

Antes de aplicar las pruebas de hipótesis se aplicó la prueba de normalidad de los datos a los indicadores para determinar el grado de significancia que tiene y así poder definir la correspondiente prueba de hipótesis que se aplicará, los resultados que se analizaron corresponden al método Shapiro-Wilk y no al Kolmogrov- Smirnov porque nuestra muestra es menor a 50.

3.2.1 Resultados del indicador de Productividad

Según los resultados obtenidos los niveles de significancia son mayores a 0.05 esto quiere decir que los datos son tipificados como “normales”, en este sentido se hará uso de la prueba paramétrica T-Student para determinar la hipótesis.

Tabla 5: Pruebas de normalidad - Productividad

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Pre_Productividad	,887	9	,185
Post_Productividad	,918	9	,379

Fuente: Elaboración propia

Según el resultado de la prueba T de Student el nivel de significancia es de 0.006 este es un valor menor a 0.05 por ende no se considera la Hipótesis Nula.

Tabla 6: Prueba de muestras emparejadas - Productividad

	t	gl	Sig. (bilateral)
Pre_Productividad - Post_Productividad	-3,699	8	,006

Fuente: Elaboración propia

La tabla 7 muestra el resumen de la prueba de hipótesis realizada para la dimensión de Productividad que determinó que hipótesis se aceptó en el presente estudio de investigación.

Tabla 7: Resultado de la hipótesis - Productividad

Resumen de prueba de Hipótesis para el indicador de Productividad.	Resultado de Hipótesis
HE1 = I1a ≤ I1p HE1 = 0.05 ≤ 0.006	No cumple
HE1 ₀ : La aplicación de Scrum no mejoró la productividad en TeamSoft S.A.C.	No se acepta la Hipótesis Nula
HE1 ₁ : La aplicación de Scrum mejoró la productividad en TeamSoft S.A.C.	Se acepta la hipótesis Alternativa.

Fuente: Elaboración propia

3.2.2 Resultados del indicador de Cobertura de Requisitos

Según el resultado obtenido, un nivel de significancia (0.001) es menor a 0.05 esto quiere decir que los datos son tipificados como “no normales”, en este sentido se hará uso de la prueba no paramétrica Wilcoxon para determinar la hipótesis.

Tabla 8: Prueba de normalidad – Cobertura de Requisitos

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Pre_Requisitos	,926	9	,443
Post_Requisitos	,696	9	,001

Fuente: Elaboración propia

Según el resultado de la prueba de Wilcoxon el nivel de significancia es de 0.012 este es un valor menor a 0.05 por ende no se considera la Hipótesis Nula.

Tabla 9: Estadísticos de prueba – Cobertura de Requisitos

	Post_Requisitos
	-
	Pre_Requisitos
Z	-2,521 ^b
Sig. asintótica (bilateral)	,012

Fuente: Elaboración propia

La tabla 10 muestra el resumen de la prueba de hipótesis realizada para la dimensión de Cobertura de requisitos que determinó que hipótesis se aceptó en el presente estudio de investigación

Tabla 10: Resultado de la hipótesis – Cobertura de Requisitos

Resumen de prueba de Hipótesis para el indicador de Requisitos.	Resultado de Hipótesis
$HE2 = I2a \leq I2p$ $HE2 = 0.05 \leq 0.012$	No cumple
HE2 ₀ : La aplicación de SCRUM no mejoró la cobertura de los requisitos en TeamSoft S.A.C.	No se acepta la Hipótesis Nula
HE2 ₁ : La aplicación de SCRUM mejoró la cobertura de los requisitos en TeamSoft S.A.C.	Se acepta la hipótesis Alterna.

Fuente: Elaboración propia

3.2.3 Resultado del indicador de la eficiencia del Costo

Según el resultado obtenidos un nivel de significancia (0.000) es menor a 0.05 esto quiere decir que los datos son tipificados como “no normales”, en este sentido se hará uso de la prueba no paramétrica Wilcoxon para determinar la hipótesis.

Tabla 11: Prueba de normalidad – Eficiencia del costo

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Pre_Costo	,903	9	,272
Post_Costo	,612	9	,000

Fuente: Elaboración propia

Según el resultado de la prueba de Wilcoxon el nivel de significancia es de 0.006 este es un valor menor a 0.05 por ende no se considera la Hipótesis Nula.

Tabla 12: Prueba de muestras emparejadas – Eficiencia del costo

	t	gl	Sig. (bilateral)
Pre_Productividad - Post_Productividad	-3,699	8	,006

Fuente: Elaboración propia

La tabla 13 muestra el resumen de la prueba de hipótesis realizada para la dimensión de eficiencia del costo que determinó que hipótesis se aceptó en el presente estudio de investigación

Tabla 13: Resultado de hipótesis – Eficiencia del costo

Resumen de prueba de Hipótesis para el indicador de Productividad.	Resultado de Hipótesis
HE3 = $I3a \leq I3p$ HE3 = $0.05 \leq 0.006$	No cumple
HE3 ₀ : La aplicación de Scrum no mejoró la eficiencia del costo en TeamSoft S.A.C.	No se acepta la Hipótesis Nula
HE3 ₁ : La aplicación de Scrum mejoró la eficiencia del costo en TeamSoft S.A.C.	Se acepta la hipótesis Alterna.

Fuente: Elaboración propia

3.2.4 Resultado general de las dimensiones

Según el resultado obtenidos el nivel de significancia (0.000) es menor a 0.05 esto quiere decir que los datos son tipificados como “no normales”, en este sentido se hará uso de la prueba no paramétrica Wilcoxon para determinar la hipótesis.

Tabla 14: Prueba de normalidad – dimensiones de la investigación

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Tradicional	,667	9	,001
Scrum	,658	9	,000

Fuente: Elaboración propia

Según el resultado de la prueba de Wilcoxon el nivel de significancia es de 0.007 este es un valor menor a 0.05 por ende no se considera la Hipótesis Nula.

Tabla 15: Estadísticos de prueba – dimensiones de la investigación

	Scrum - Tradicional
Z	-2,694 ^b
Sig. asintótica (bilateral)	,007

Fuente: Elaboración propia

La tabla 16 muestra el resumen de la prueba de hipótesis realizada para la todas las dimensiones de la investigación, según este resultado se determinó que hipótesis se aceptó en el presente estudio de investigación de manera general.

Tabla 16: Resultado general de la hipótesis – dimensiones de la investigación

Resumen de prueba de Hipótesis para los procesos de desarrollo de software.	Resultado de Hipótesis
$HE3 = I3a \leq I3p$ $HE3 = 0.05 \leq 0.007$	No cumple
<p>H₀: La aplicación de SCRUM no mejoró el proceso de desarrollo de software en TeamSoft S.A.C.</p>	<p>No se acepta la Hipótesis Nula</p>
<p>H₁: La aplicación de SCRUM mejoró el proceso de desarrollo de software en TeamSoft S.A.C.</p>	<p>Se acepta la hipótesis Alternativa.</p>

Fuente: Elaboración propia

IV. DISCUSIÓN

4.1 En base a los resultados de la Productividad

En base al estudio realizado se muestra los diferentes escenarios comparativos por cada dimensión, estos fueron impactados por el marco de trabajo aplicado en la empresa. El primer resultado muestra un indicador favorable del 40% en la mejora de la productividad, esto es un dato de suma importancia el cual se vio reflejado en el transcurso de la aplicación del proceso de investigación en cada uno de los proyectos que fueron objeto del estudio, así también se analizó el trabajo desarrollado por Malpica (2014) el cual indicó que la aplicación de Scrum influyó positivamente sobre el incremento de la productividad logrando que se cumplan los plazos estimados teniendo un desfase de 0 días, en su estudio Malpica incrementó la productividad en un 30%. Esto refleja una correlación de resultados muy parecidos en ambos estudios, se observa que en el presente trabajo se evaluaron nueve proyectos de software mientras que en la investigación de Malpica solo se aplicó en un proyecto, la tendencia del resultado en todo sentido sigue representando una mejora producida por Scrum en la productividad del equipo de desarrollo aun cuando las muestras son distintas pero el resultado acompaña de manera muy parecida ambas investigaciones.

4.2 En base a los resultados de la cobertura de requisitos

El resultado presentado en este trabajo muestra un resultado favorable en la cobertura de requisitos de los proyectos de software que se han desarrollado a lo largo de la aplicación de Scrum, los proyectos evaluados tuvieron un promedio final de 95% de efectividad en la cobertura de los requisitos de software lo cual muestra que se pudo cubrir casi toda la totalidad de requerimientos que fueron objeto de algún cambio dentro del proceso de desarrollo, en comparación con estos resultados, en el trabajo de investigación de Rey (2017) el cual tuvo como objetivo comparar la eficiencia y efectividad de los proyectos de desarrollo de software haciendo uso de metodologías ágiles, este concluyó que aplicando Scrum fue más eficiente y mucho mayor organizado alcanzando un mayor involucramiento

con los requerimientos de software, por ende alcanzó un resultado del 98% de efectividad. Las diferencias entre los logros alcanzados muestran una tendencia similar, sin embargo, se puede apreciar que en la presente investigación el porcentaje de requerimientos cubiertos no es tan alto como la investigación de Rey, pero se considera un porcentaje bastante bueno respecto de la cobertura que se tenía con las metodologías tradicionales las cuales fueron de 34% versus su postrer 95% obteniendo un margen positivo de 61%.

4.3 En base a los resultados de la eficiencia del costo

Finalmente, para la eficiencia del costo se obtuvo un margen positivo del 11% superior al pre-test, este margen no presenta una tendencia tan alta como los anteriores mencionados en los cuales se alcanzó un aumento importante, sin duda pese a esta observación hubo una mejora y se logró el aumento en este indicador pero no se generó una brecha mayor posiblemente por los constantes cambios que ocurrieron a lo largo de los proyectos, no para todos los casos se controlaron al 100%, hubieron casos que se reasignaron analistas y esto causó el aumento del costo, pero en términos generales se produjo una mejora adecuada, en contraste con este resultado, en el estudio de Vargas (2013) donde se mejoró la eficiencia del costo en un 5% en su resultado se adjudicó solo un proyecto de software en comparación con el presente estudio donde se aplicó Scrum a toda una cartera de nueve proyectos. Finalmente se observó que el margen en ambos estudios fue positivo en relación de los costos, esto hace notar una efectiva correlación de resultados salvo por su diferencia en el valor ganado, se puede sumar a ello el hecho que las consultoras y fábricas de software como TeamSoft S.A.C tienen mejor cotizados sus desarrollos de proyectos de software que otra empresa que no se dedica al rubro mencionado.

V. CONCLUSIONES

5.1 General

En el ámbito general de la investigación, se concluye que Scrum mejoró los procesos de desarrollo de software sobre los cuales se basó el presente estudio, antes de la aplicación se obtuvo un indicador de 0.52, después de la aplicación el indicador fue de 0.89, esto se tradujo en un margen positivo de 37% que reflejó la mejora en los procesos de desarrollo de software después de la aplicación del marco de trabajo Scrum.

5.2 Específicos

Se concluye que el marco de trabajo Scrum mejoró la productividad del equipo de desarrollo en base al resultado de la prueba de hipótesis realizada, antes de la aplicación de Scrum la productividad alcanzó un indicador de 0.39 donde los recursos utilizados en promedio eran 5 personas y los entregables de software fueron de 1.89 por proyecto, con la aplicación de Scrum se alcanzó un indicador de 0.79, en este escenario se logró usar en promedio cuatro personas por proyecto, asimismo se entregó 3.44 entregables de software, con estos resultados el indicador final marcó un aumento del 40% en productividad, algunas características importantes de la mejora fue en cuanto a la integración de las diferentes etapas del ciclo de vida del proyecto, análisis, diseño, desarrollo, planificación, esto permitió que los integrantes compartan y distribuyan el trabajo en forma equitativa, además se dividió complejidad y tiempo lo que hizo más ligero los proyectos de software, algo de suma importancia fue la motivación de las personas que mostraron un mayor compromiso por alcanzar los objetivos en los proyectos debido a la nueva metodología aplicada y por los reconocimientos que se entregaron en cada etapa de la investigación.

Se concluye que el marco de trabajo Scrum mejoró la cobertura los requisitos solicitados por el Usuario en los proyectos de software, antes de la aplicación de Scrum el indicador obtenido fue de 0.34, luego de la aplicación de la metodología el indicador fue de 0.95, el resultado final mostró un margen positivo del 61% en la cobertura de requisitos de Usuario en los diferentes proyectos de software, algunas características importantes de la investigación respecto de los resultados fue debido a la flexibilidad que permitió Scrum en cada iteración, el manejo de los Sprints en cada proyecto hizo factible que se pueda gestionar mejor los requerimientos y su respectiva inclusión en las diferentes iteraciones, algunos más resaltantes fueron cambios por parte del Usuario debido a nuevas funcionalidades, mejoras técnicas y cambios funcionales del negocio que se vieron solapadas en cada Sprint pudiendo finalizar el requerimiento antes de cada entregable al Usuario del negocio.

Se concluye que el marco de trabajo Scrum mejoró la eficiencia del costo para los proyectos de software, antes de la aplicación de Scrum el indicador mostró indicador de 0.83, posterior a la aplicación de la metodología el indicador fue de 0.94, esto representó un margen positivo en la eficiencia del costo de 11%, esto representó un ahorro de S/. 20,272.78, algunas consideraciones importantes de la mejora radicó en el cumplimiento del alcance del cual no se desprendieron ampliaciones de tiempo, se verificó que seis de los nueve proyectos no generaron re asignaciones de recursos, esto debido a que los cambios que se daban eran cubiertos en el siguiente Sprint, es por ello que el control fue menor en cada iteración en el ciclo de vida del proyecto por la división de etapas, esto provocó también que no aplicaran penalizaciones por entregas tardías del software a los Usuarios.

VI. RECOMENDACIONES

Las recomendaciones para futuras investigaciones se basaron en el conjunto de experiencias recabadas en la presente investigación y que son de suma importancia como fuente de información para futuras investigaciones que realizarán un análisis de metodologías ágiles en el desarrollo de software, las recomendaciones son las siguientes:

- 6.1 Se recomienda para futuros trabajos de investigación ampliar los ámbitos del presente estudio de investigación, se deberían considerar las dimensiones de comunicación, riesgos y calidad, las cuales pueden sumar como beneficiosas a las áreas de gestión para la ingeniería del software, se debería dar mayor amplitud en estos aspectos puesto que implicaría un mejor análisis, el hecho de poder evaluar las relaciones entre estas dimensiones, por ejemplo, si aumenta la productividad respecto de la calidad entregada al Cliente, o cómo influye en la comunicación en la productividad del equipo de software, con este análisis se podría obtener un mayor beneficio y poder delimitar que aspectos se pueden mejorar sin desestimar otros.
- 6.2 Se recomienda ampliar la muestra de proyectos de software y hacer uso de la estratificación de los proyectos por el nivel de complejidad para efecto de poder tener diferenciado de qué manera influye la aplicación por cada nivel o tamaño de proyecto, esto ayudará a la toma de decisiones de las empresa por el hecho de tener un detalle mayor y poder decidir con mayor seguridad el camino que se deberá tomar para las futuras implementaciones de software.

VII. REFERENCIAS

Alfaro Paredes, Emigdio. 2012. *Identificación de beneficios financieros concretos de la implementación de tecnologías de información.* s.l. : Lidera, 2012.

Álvarez García, Alonso, del las Heras del Dedo, Rafael y Lasa Gómez, Carmen. 2012. *Métodos ágiles y Scrum.* Madrid : Grupo Anaya SA, 2012. pág. 39. 978-84-415-3104-8.

Alvarez Vergara, Ricardo Miguel y Fernandez Soto, José Mauro. 2012. *Implementación de una solución integrada para la gestión de proyectos en una compañía de Auditoría financiera.* Lima : s.n., 2012.

Arana López, Liz Melissa, Ruiz Rivera, Maria Elena y La Serna Palomino, Nora. 2015. *Análisis de aplicaciones empleando la computación en la nube de tipo PaaS y la metodología ágil Scrum.* 2015. Revista de la Facultad de Ingeniería Industrial . 1810-9993.

Avila Baray, Hector Luis. 2006. *Introducción a la metodología científica.* Chihuahua : s.n., 2006. 84-690-1999-6.

Bernal Torres, César Augusto. 2010. *Metodología de la investigación.* [ed.] Orlando Fernández Palma. Tercera edición. 2010. pág. 320. 978-958-699-128-5.

Camacho Carrillo, Oscar. 2015. America Sistemas. [Citado el: 15 de Octubre de 2017.] Disponible en: <http://www.americasistemas.com.pe/por-que-fracasan-los-proyectos-de-ti-en-el-estado-peruano/>.

Carreño R., Alejandro y Jiménez V, Lira Andrea. 2016. Elaboración de una guía para auditoría a la gerencia de proyectos de desarrollo de software con enfoque PMI®, aplicable a las áreas de alcance, tiempo y costo. Junio de 2016. [Citado el: 22 de Octubre de 2017.] Disponible en: <https://repositorio.escuelaing.edu.co/bitstream/001/413/3/HB-Maestr%C3%ADa%20en%20Desarrollo%20y%20Gerencia%20de%20Proyectos-80134194-Sustentaci%C3%B3n.pdf>.

Corral, Yadira. 2009. Validez confiabilidad de los instrumentos de investigación para la recolección de datos. 2009. [Citado el: 11 de Julio de 2017.] <http://servicio.bc.uc.edu.ve/educacion/revista/n33/art12.pdf>.

Daniela Panizzi , Marisa. 2015. Revista latinoamericana de Ingeniería de Software. Febrero de 2015. [Citado el: 28 de Noviembre de 2017.] Disponible en: <http://sistemas.unla.edu.ar/sistemas/redisla/ReLAIS/relais-v3-n1-revista.pdf>. 2314-2642.

Erin Brown, Mary. 2013. *Data-Driven Decision Making as a Tool to Improve Software Development Productivity.* s.l. : ProQuest LLC, 2013. 3591716.

Escobar Pérez, Jazmine y Cuervo Martínez, Ángela. 2008. Validez de Contenido y Juicio de Expertos: Una aproximación a su utilización. [Citado el: 20 de 10 de 2017]. Disponible en: http://www.humanas.unal.edu.co/psicometria/files/7113/8574/5708/Articulo3_Juicio_de_expertos_27-36.pdf.

Flores Santos, Ericka Raiza. 2016. *Framework de trabajo para proyectos de tesis aplicando la metodología Scrum en la ingeniería de software.* 2016.

Gamboa Carrascal, Jessica Patricia. 2015. Diseño de un método ágil de desarrollo de software basado en XP, Scrum, Openup y validado con la herramienta de análisis 4-Dat para mejorar la calidad de los proyectos desarrollados por los grupos de gestión de software de la UFPSO. 2015. [Citado el: 08 de Julio de 2017.] Disponible en: <http://repositorio.ufpso.edu.co:8080/dspaceufpso/handle/123456789/818>.

Gamboa Manzaba, Julio. 2014. Aumento de la productividad en la gestión de proyectos, utilizando una metodología ágil aplicada en una fábrica de software en la ciudad de Guayaquil. Diciembre de 2014.

Gómez-García Palao, Vicent y Palao Castañeda, Jorge Antonio. 2010. *12 Pasos para el éxito - Concrete su empresa.* Primera. Lima : Editorial Septiembre SAC, 2010. pág. 56. 978-612-301-206-9.

Guneth de Lourdes, Katherine. 2010. *Implementación de un método ágil para el desarrollo de software en el centro dedicado de desarrollo del banco de crédito.* 2010.

Hanadi Salameh, Loay Alnaji. 2015. *Performance-Measurement Framework to Evaluate Software Engineers for Agile Software-Development Methodology.* 2015. Vol. 7. 2222-2839.

Hernández Sampieri, Roberto. 2014. *Metodología de la Investigación.* Sexta edición. 2014. 978-1-4562-2396-0.

Hernández Sampieri, Roberto, Fernández Collado, Carlos y Baptista Lucio, Pilar. 2010. *Metodología de la investigación.* [ed.] Jesús Mares Chacón. Quinta. s.l. : McGRAW-HILL / Interamericana Editores, S.A., 2010. 978-607-15-0291-9.

Hernández Vázquez, Marina Adriana. 2011. Técnicas de Investigación. <https://es.slideshare.net>. Septiembre de 2011. [Citado el: 30 de Noviembre de 2017.] <https://es.slideshare.net/herreramarina4/tecnicas-de-investigacin-fichas-de-registro>.

Iglesias, Jesús. 2009. Proyectos Ágiles. *Proyectos Ágiles.* 14 de Noviembre de 2009. [Citado el: 09 de Junio de 2017.] Página Oficial de Xavier Albaladejo Scrum Master certificado por Acrum Alliance. <https://proyectosagiles.org/2009/11/14/scrum-dos-equipos-distintas-ciudades/>.

International Scrum Institute™. 2017. <http://www.scrum-institute.org>. s.l. : International Scrum Institute™, 2017. Scrum is a lightweight agile project management framework mainly used for software development. It describes an iterative and incremental approach for project work..

Kniberg, Henrik y Skarin, Mattias. 2010. Kanban y Scrum – obteniendo lo mejor de ambos. 2010. [Citado el: 25 de Junio de 2017.] http://www.proyectalis.com/documentos/KanbanVsScrum_Castellano_FINAL-printed.pdf. 978-0-557-13832-6.

Lipkin, Ilya. 2011. *Testing Software Development Project Productivity Model.* ProQuest LLC (2017). 2011. 10631147.

Lledó, Pablo. 2009. *Director Profesional de Proyectos*. Segunda. Mendoza : s.n., 2009. pág. 428. 978-987-05-5681-7.

Lynch, Jennifer . 2015. Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch. *Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch*. 04 de Octubre de 2015. [Citado el: 2017 de Noviembre de 2017.] <https://www.infoq.com/articles/standish-chaos-2015>.

Maida, Esteban Gabriel y Pacienza, Julián. 2015. Metodologías de desarrollo de software. Diciembre de 2015. [Citado el: 24 de 10 de 2017.] <http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf>.

Maldonado Arango, Miguel Ángel. 2017. Factores que afectan la productividad en equipos Scrum analizados con Pensamiento sistémico. 2017. [Citado el: 27 de Noviembre de 2017.] Disponible en: <http://www.bdigital.unal.edu.co/57465/1/71275549.2017.pdf>.

Malone, Michael W. 2014. *Process Subversion In Agile Scrum Software Development: A Phenomenological Approach*. s.l. : ProQuest LLC, 2014. 3632735.

Malpica Velazquez, Carlos Jesús. 2014. Aplicación de la metodología Scrum para incrementar la productividad del proceso de desarrollo de software en la empresa CCJ S.A.C. Lima. 2014. [Citado el: 08 de Julio de 2017.] Disponible en: <http://repositorio.uncp.edu.pe/handle/UNCP/1138>.

Marie Mitchell, Susan. 2012. *Software Process Improvement through the Removal of Project-level Knowledge Flow Obstacles: The Perceptions of Software Engineers*. s.l. : ProQuest LLC, 2012. 3516319.

Mariño, Sonia y Alfonso, Pedro. 2014. *Implementación de Scrum en el diseño del proyecto del Trabajo Final de Aplicación*. Pereira : s.n., 2014. Vol. 19. 0122-1701.

Martel, Antonio. 2014. *Gestión Práctica de Proyectos con Scrum*. [ed.] illustrated. 2014. Vol. Volume 1 of Aprender a Ser Mejor Gestor De Proyectos. 1517192366.

Mayfield, Kathleen M. 2010. *Project managers experience and description of decision uncertainty associated with the agile software development methodology: A phenomenological study*. s.l. : ProQuest LLC., 2010. 3427057.

Ticona Yanqui, Fidel Ernesto. 2014. *Metodología Scrum para el desarrollo de software y gestión de proyectos en las pequeñas y medianas empresas de la ciudad de Juliaca*. : s.n., 2014, Investigación Andina, Vol. Volumen 13 Nro 1.

Mitre-Hernández , Hugo A, Ortega-Martínez, Edgar y Lemus-Olalde, Cuauhtémoc. 2014. *Estimación y control de costos en métodos ágiles para desarrollo de software: un caso de estudio*. 2014. Vol. XV, Ingeniería Investigación y Tecnología. 1405-7743 FI-UNAM.

Murillo, Javier. s.f. Métodos de investigación de enfoque experimental. s.f. [Citado el: 09 de Julio de 2017.] Disponible en: <http://www.postgradoune.edu.pe/documentos/Experimental.pdf>.

Ortega, Carlos. 2009. *Estadística General*. Lima : s.n., 2009. 978-84-15271-70-3.

P. Landry, Jeffrey y McDaniel, Rachel. 2015. Agile Preparation Within a Traditional Project Management Course. 2015. [Citado el: 25 de Noviembre de 2017.] Conference on Information Systems and Computing Education. Disponible en: <http://proc.iscap.info/2015/pdf/3429.pdf>.

Palacio, Juan. 2008. *Flexibilidad con Scrum*. 2008. Principios de diseño e implantación de campos de Scrum.

Pérez Pérez, Caudia Marcela. 2015. Una metodología Ágil para el desarrollo de software en una compañía financiera. 2015. [Citado el: 08 de Julio de 2017.] Disponible en: <http://repository.unimilitar.edu.co/bitstream/10654/13235/1/Proyecto%20investigacion%20-%20Claudia%20Perez.pdf>.

Project Management Institute Inc. 2013. *Guía de los Fundamentos para la dirección de proyectos*. Pensilvania : s.n., 2013. pág. Chapter 1.3. 978-1-62825-009-1.

Rey Guevara, Christian Felipe. 2017. *Estudio de la efectividad de la aplicación de la metodología Ágil de desarrollo Scrum*. 2017.

Rivas, Carlos Ignacio, y otros. 2015. *Metodologías actuales de desarrollo de software*. 2015. Vol. 2.

Rodríguez, César y Dorado, Rubén. 2015. ¿Por qué implementar Scrum?, Journal. 15 de Abril de 2015. [Citado el: 28 de Noviembre de 2017.] Disponible en: <http://journal.ean.edu.co/index.php/Revistao/article/view/1253/1218>.

Rodriguez, José Ramón. 2016. *¿Cómo planificar un proyecto de inteligencia de negocio?* Barcelona : UOC Publishing, 2016. 978-84-9116-324-4.

Romero Roldán, José Ramón. *Gestión de proyectos desde la propuesta al cierre*. Madrid : ESIC EDITORIAL. 978-84-1670-130-8.

Sanchez, Salvador. 2012. *Ingeniería del Software desde un punto de vista de la SWEBOOK*. 2012. 978-84-9281-240-0.

Schwaber, Ken y Sutherland, Jeff. 2017. The Scrum Guide. 2017. [Citado el: 2017 de 11 de 25.] Disponible en: <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>.

Scrum Alliance Inc. 2016. Scrumalliance. 2016. [Citado el: 25 de Junio de 2017.] Disponible en: <https://www.scrumalliance.org/agile-resources/scrum-roles-demystified>.

Scrum Org. Scrum.Org. [Citado el: 28 de Noviembre de 2017.] Disponible en: <https://www.scrum.org/resources/scrum-glossary>.

Singh Samra, Taranjit. 2012. *Software Risk Management: an Exploration of software life cycle methodologies, best practices and tools for their application to medical device software risk management*. s.l. : ProQuest LLC., 2012. 3514303.

SofTeng. 2017. SofTeng. 2017. [Citado el: 29 de Junio de 2017.] Disponible en: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>.

Standish Group. 2015. <https://www.standishgroup.com/>. <https://www.standishgroup.com/>. 2015. [Citado el: 27 de Junio de 2017.] Disponible en: <https://www.standishgroup.com/about>.

Tamayo, Carla y Siesquén, Irene Silva. Técnicas e instrumentos de recolección de datos. *departamento académico de metodología de la investigación*. [Citado el: Noviembre de 30 de 2017.] Disponible en: http://www.postgradoune.edu.pe/documentos/tecnicas_Instrumentos.pdf.

Tinoco Gómez, Oscar, Rosales López, Pedro Pablo y Salas Bacalla, Julio. 2010. *Criterios de selección de metodologías de software*. Lima : Industrial Data, Julio de 2010. Vol. 13. 1560-9146.

Universidad Nacional Abierta. 1991. Medios, Instrumentos, Técnicas y Métodos en la Recolección de Datos e Información. 1991. [Citado el: 11 de Julio de 2017.] Disponible en: <http://postgrado.una.edu.ve/metodologia2/paginas/cerda7.pdf>.

Vargas Cordero, Zoila Rosa. 2009. *La Investigación aplicada: una forma de conocer las realidades con evidencia científica*. 2009. Vol. 33. 0379-7082.

Vargas Rosales, Carlos Iván. 2013. *Mejora de Procesos de Desarrollo de Software mediante metodologías ágiles*. 2013.

Yanqui, Ticona. 2014. *Metodología Scrum para el Desarrollo de Software y Gestión de Proyectos en las pequeñas y medianas empresas de la ciudad de Juliaca, 2014*. Juliaca : s.n., 2014.

Zuker, Alan. 2016. Lo que realmente sabemos sobre proyectos exitosos. *Scrumalliance*. 03 de Octubre de 2016. [Citado el: 27 de Noviembre de 2017.] Disponible en: <https://www.scrumalliance.org/community/articles/2016/october/what-we-really-know-about-successful-projects>.

VIII. ANEXOS

Anexo 1: Matriz de Consistencia

PROBLEMAS	OBJETIVOS	HIPÓTESIS	VARIABLE	DIMENSIONES	INDICADORES
General	General	General			
¿Cuál es el efecto de la aplicación del marco de trabajo Scrum en el proceso de desarrollo de software en TeamSoft S.A.C?	Determinar cuál es el efecto de la aplicación del marco de trabajo Scrum en el proceso de desarrollo de software en TeamSoft S.A.C.	La aplicación del marco de trabajo Scrum mejoró el proceso de desarrollo de software en TeamSoft S.A.C.			
Específicos	Específicos	Específicos			Indicadores
¿Cuál es el efecto de la aplicación del marco de trabajo Scrum en la productividad en TeamSoft S.A.C?	Determinar cuál es el efecto de la aplicación del marco de trabajo Scrum en la productividad en TeamSoft S.A.C.	La aplicación del marco de trabajo Scrum mejoró la productividad en TeamSoft S.A.C.	Aplicación de Scrum en el desarrollo de software en TeamSoft S.A.C.	Productividad Salvador (Sánchez, 2012 p. 84)	Porcentaje de la productividad $Productividad = \frac{Tamaño}{Esfuerzo}$ Tamaño: Número de entregables Esfuerzo: Cantidad de personas
¿Cuál es el efecto de la aplicación del marco de trabajo Scrum en la cobertura de los requisitos en TeamSoft S.A.C?	Determinar cuál es el efecto de la aplicación del marco de trabajo Scrum en la cobertura de los requisitos en TeamSoft S.A.C.	La aplicación del marco de trabajo Scrum mejoró la cobertura de los requisitos en TeamSoft S.A.C.		Cobertura de los requisitos (Sánchez, 2012, p. 159)	Porcentaje de requisitos cubiertos $= \frac{Cob.de\ requisitos}{Req. soportados}$ $= \frac{Req. en\ la\ línea\ base}{Req. en\ la\ línea\ base}$
¿Cuál es el efecto de la aplicación del marco de trabajo Scrum en la eficiencia del costo en TeamSoft S.A.C?	Determinar cuál es el efecto de la aplicación del marco de trabajo Scrum en la eficiencia del costo en TeamSoft S.A.C.	La aplicación del marco de trabajo Scrum mejoró la eficiencia del costo en TeamSoft S.A.C.		Eficiencia del costo Salvador (Sanchez, 2012 p. 451)	Porcentaje de eficiencia del costo. $CPI = \frac{BCWP}{ACWP}$ CPI: Índice de rendimiento del costo BCWP: Costo Planificado ACWP: Costo Real

Figura 24: Matriz de consistencia de los procesos de ingeniería de software

Anexo 2: Ficha de Registro

FR-00001	TeamSoft SAC
Ficha de Registro	
Objetivo:	
La presente Ficha de Registro nos ayudará a poder determinar cuáles son las problemáticas dentro del área Operacionales de TeamSoft para el proceso de desarrollo de sistemas informáticos.	
El uso de la información será sometida a tratamientos matemáticos con el único fin de conseguir una línea base de la situación actual del área respecto de sus procesos internos y como se maneja el desarrollo de software, será de gran utilidad para poder inicial el trabajo de investigación.	
Consideramos que su participación será de ayuda importante para poder mejorar los procesos de desarrollo de software dentro del área y también para la empresa.	
Área de sistemas operacionales de Teamssoft	
Datos del encuestado	
Nombres: <u>Giovani</u>	
Apellidos: <u>Vizquez</u>	
Cargo: Líder de proyecto <input checked="" type="checkbox"/> Analista Funcional <input type="checkbox"/>	
Código del proyecto: <u>PROY-18096</u>	
El desarrollo de la Ficha de registro consta de un tiempo de 1 hora.	
Marque con una "X"	
1. ¿Qué proceso metodológico usa para desarrollar los sistemas informáticos?	
a) Metodología tradicional <input checked="" type="checkbox"/> b) Metodología ágil <input type="checkbox"/> c) Ninguna <input type="checkbox"/>	
2. ¿Qué modelo de ciclo de vida se usó en el proceso de desarrollo de software?	
a) Cascada <input checked="" type="checkbox"/> b) Modelo "V" <input type="checkbox"/> c) Modelo de prototipos <input type="checkbox"/> d) Espiral <input type="checkbox"/> e) Ninguna <input type="checkbox"/>	
3. ¿Qué complejidad tiene el proyecto?	
a) ALTA <input checked="" type="checkbox"/> b) MEDIA <input type="checkbox"/> c) BAJA <input type="checkbox"/>	
4. ¿Cuántos líderes gestionan el proyecto?	
Líder de proyecto: <u>1</u>	
5. ¿Cuántos Analista funcionales gestionan el proyecto?	
Analista Funcional: <u>1</u>	
6. ¿Cuántos Analista desarrolladores gestionan el proyecto?	
Analista programador: <u>4</u>	
7. ¿Cuántos recursos en total trabajaron en el presente proyecto?	
Total: <u>6</u>	
1	

Figura 25: Ficha de Registro Nro. 1

8. ¿Cuántos entregables funcionales se hicieron al Usuario establecidos en la etapa de planificación inicial?

Cantidad: 1

9. ¿Cuántos entregables funcionales se hicieron al Usuario en la re planificación?

Cantidad: 1

10. ¿Cuál fue el tiempo en días planificado?

Días: 158

11. ¿El tiempo planificado del proyecto se mantuvo hasta el final?

SI NO

12. ¿Los requerimientos planificados del proyecto se mantuvieron hasta el final?

SI NO

13. ¿Cuántos cambios funcionales fueron solicitados en el trámite del proyecto?

Cantidad: 10

14. ¿Cuántos cambios funcionales fueron efectuados por el equipo de desarrollo?

Cantidad: 2

15. ¿Cuál es el tiempo en días que se adicionó al proyecto?

Días: 25

16. ¿Cuál fue el tiempo real del proyecto?

Días: 183

17. ¿El costo planificado del proyecto se mantuvo hasta el final?

SI NO

18. ¿Cuál fue el costo planificado del proyecto?

Costo: 123767

19. ¿Cuál fue el costo adicionado del proyecto?

Costo adicional: 19583

20. ¿Cuál fue el costo real del proyecto?

Costo real: 143350

Figura 26: Ficha de Registro Nro. 2

Anexo 3: Desarrollo del proyecto de Investigación

En la presente investigación se realizó la aplicación de la metodología Scrum en los proyectos que fueron objeto de la investigación, para el caso del desarrollo del proyecto se mostrará a continuación la aplicación de Scrum en un proyecto, en base a esta forma de trabajo presentada se aplicó a los demás proyectos objeto de la muestra mencionada en el capítulo II Población y muestra. El proyecto de software fue catalogado con el nombre de: **“Rediseño de débitos automáticos”**, el Cliente ha solicitado poder rediseñar algunas funcionalidades de su proceso de débitos automáticos de su área de Recaudación, este cambio le permitirá poder mejorar la calidad de servicio de atención al su Cliente final y reducir el tiempo de trabajo por trabajos realizados manualmente. Scrum define 2 etapas para el desarrollo de un proyecto basado en esta metodología, dentro de cada etapa una serie de consideraciones que se deben abordar y se ven desarrolladas a continuación.

1. Etapa 1: Definiendo las condiciones

Inicialmente el Cliente solicitó a TeamSoft su consultoría como fábrica de software, posterior a ello se derivó un requerimiento al gerente de cuenta de TeamSoft quien lo derivó al equipo de “Proyecto Operacionales” de TeamSoft, en el área se definieron las condiciones del proyecto cuando este fue entregado por el Cliente y los acuerdos quedaron documentados de la siguiente manera.

1.1 Alcance del proyecto:

Se requiere por parte de la Gerencia de recaudación del Cliente poder rediseñar los formularios del aplicativo SIOP para el sistema de afiliaciones y débitos para que pueda ser capaz de informar al Cliente cada vez que ha realizado una transacción de afiliación o cobro, así como automatizar el proceso de afiliación mediante un proceso automático que permita la intervención del Usuario final.

1.2 Recursos

En TeamSoft se estableció conformar el equipo de trabajo que cumpla con las necesidades del proyecto a implantar, para usar la metodología Scrum se incluyeron las siguientes condiciones:

- El personal que cumpla el rol de Scrum Master deberá tener certificación Scrum.
- El personal que cumpla el rol de Product Owner deberá tener el puesto de Analista Funcional o de Negocio, así como también tener capacitación del marco de trabajo Scrum.

El equipo de desarrollo deberá tener experticia en el proceso de negocio, especialista en lenguaje de programación y base de datos, además ser capacitado en el marco de trabajo Scrum.

1.3 Herramientas

Para poder dar seguimiento al Proyecto y poder soportar el proceso de Scrum se estableció utilizar las siguientes herramientas.

- Microsoft Excel. Plantillas compartidas que ayudarán a flexibilizar los procesos de Scrum.
- Pizarra con cintas adhesivas y post-it para la construcción del Sprint Backlog.
- ScrumDesk. Aplicativo que soporta los procesos de Scrum.

1.4 Reglas de entrega de resultados

En el área de TeamSoft se definieron las siguientes reglas:

1. Para medir la velocidad del equipo se utilizará como técnica el Planning póker.
2. Para la priorización de las historias de Usuario se utilizará como método el Peso relativo.

3. Para la reunión de retrospectiva se hará uso de la técnica “Bien, Mejorable, Mejoras”. Duración máxima 45 min.
4. Las entregas se realizarán de manera progresiva asumiendo que cada entregable funcional tendrá 1 o más Sprint y que un entregable funcional al Cliente se hará en el máximo de 1 mes.
5. Asimismo, se estableció que las entregas se harán cubriendo primero cada Épica y luego su priorización calculada.
6. El Daily Meeting será máximo 20min.

1.5 Factibilidad

Se estableció factible el proyecto porque 1 integrante tiene certificación Scrum Master, el personal tiene experiencia en el proceso del negocio y todos están capacitados en la metodología Scrum.

2. Etapa 2: Contenido del trabajo

Se estableció el equipo de trabajo Scrum que trabajó en el proyecto, se asignó un rol a cada integrante del equipo.

- Scrum Master
- Product Owner
- Equipo Scrum

2.1 Product Backlog

2.1.1 El Artefacto

Para crear el siguiente artefacto se determinó una estructura que almacenaría los requisitos del Usuario, es ese documento se guardará todas las historias de Usuario que se recopilen de los requisitos de las áreas solicitantes.

Tabla 17: Plantilla del Product Backlog

Código de Historia de Usuario	Código de Requerimiento Funcional	Historia de Usuario	Priorización	Criterio de Aceptación

Fuente: Elaboración propia

2.1.2 El Proceso de desarrollo

Luego de la reunión con los Usuarios se estableció el Product Backlog del producto, en el cual plasmó los requerimientos de los usuarios que según Scrum lo nombramos como Historias de Usuario. Este proceso se logró realizando una entrevista con los usuarios donde se registró los requisitos del proyecto en una ficha de registro y luego se plasmó en el repositorio del equipo de desarrollo, el Product Backlog quedó plasmado de la siguiente manera:

Tabla 18: Desarrollo del Product Backlog

CHU	CRF	HISTORIA	PRI	CRI
HU-RF01	RF01	Como Analista de recaudación me gustaría poder agregar al formulario de clientes SGA de la fija una columna de teléfono de referencia de la manera que me permita poder enviar un mensaje de texto al Cliente cuando se haya registrado correctamente.		Se deberá registrar mediante el front el número de teléfono en BD y se deberá poder ver el cambio de estado para afiliaciones masivas.
HU-RF02	RF02	Como Analista de recaudación me gustaría tener una sola plataforma de consulta de afiliaciones de Clientes de tal manera que pueda centralizar las consultas del estado de las afiliaciones.		Se debe visualizar todas las opciones ordenadamente según se especifica en el documento detallado.
HU-RF03	RF03	Como Operador CAC me gustaría agregar al formulario de afiliación el campo de teléfono de referencia en SIAPOST de tal manera que se pueda enviar mensajes de		Los campos deben estar validados en su nivel de integridad de datos. Los formularios deben tener la claridad y facilidad de registrar la

		texto al Cliente.	información. Los registros de afiliaciones deben estar en BD registrados correctamente.
HU-RF04	RF04	Como Operador CAC me gustaría agregar al formulario de afiliación el campo de teléfono de referencia en SIACHFC de tal manera que se pueda enviar mensajes de texto al Cliente.	Los campos deben estar validados en su nivel de integridad de datos. Los formularios deben tener la claridad y facilidad de registrar la información. Los registros de afiliaciones deben estar en BD registrados correctamente.
HU-RF05	RF05	Como Analista de recaudación me gustaría tener un proceso automático que envíe cada cierta hora del día los archivos de afiliación a un correo personalizado de tal manera que pueda tener de forma automática la base de afiliaciones para verificar y enviar	Los registros ingresados deben verse en BD incluidos el número de teléfono.
HU-RF06	RF06	Como Analista de recaudación me gustaría poder generar archivos de afiliación leyendo desde la base de datos de tal manera que me permita agilizar el envío de las tramas a las entidades bancarias.	Los registros de la BD origen deberá tener la misma consistencia y la misma integridad que la BD de destino, para afiliados en todas las plataformas.
HU-RF07	RF07	Como Analista de recaudación me gustaría tener un proceso automático que envíe cada cierta hora del día los archivos de débito a un correo personalizado de tal manera que pueda tener de forma automática la base de cobros para verificar y enviar	Los registros ingresados deben verse en BD incluidos el número de teléfono.
HU-RF08	RF08	Como Analista de recaudación me gustaría poder importar los archivos de respuesta de débito que envían las entidades de tal manera que pueda realizar la cobranza masiva de todos los Clientes y a su vez se envíe la notificación a su teléfono de referencia.	Los procesos automáticos deben registrar toda la información de los archivos de afiliación y débito en BD.

Fuente: Elaboración propia

La siguiente imagen muestra el cuadro de Product Backlog que fue desarrollado para este proyecto, se observa la descripción de cada requerimiento de Usuario con las diferentes medidas tales como Beneficio Relativo, Porcentaje Relativo, Valor Total, Valor porcentual, estimación, costo porcentual y el promedio de puntos de historia.

PROYECTO "PROY-18096 REDISEÑO DE DÉBITOS AUTOMÁTICOS"										
HIST.	RF.	Descripción	Beneficio Relativo	Penalización Relativa	Valor Total	Valor porcentual	Estimación	Costo porcentual	Prioridad	PROMEDIO POK
HU-RF03	RF03	Como Operador CAC me gustaría agregar al formulario de afiliación el campo de teléfono de referencia en SIAPOST de tal manera que se pueda enviar mensajes de texto al Cliente.	9	9	18	16%	5	6%	259	17
HU-RF04	RF04	Como Operador CAC me gustaría agregar al formulario de afiliación el campo de teléfono de referencia en SIACHFC de tal manera que se pueda enviar mensajes de texto al Cliente.	9	9	18	16%	5	6%	259	19
HU-RF01	RF01	Como Analista de recaudación me gustaría poder agregar al formulario de clientes SGA de la fila una columna de teléfono de referencia de la manera que me permita poder enviar un mensaje de texto al Cliente cuando se haya registrado	9	9	18	16%	5	6%	259	17
HU-RF05	RF05	Como Analista de recaudación me gustaría tener un proceso automático que envíe cada cierta hora del día los archivos de afiliación a un correo personalizado de tal manera que pueda tener de forma automática la base de afiliaciones para verificar y enviar.	8	4	12	11%	7	9%	123	88
HU-RF07	RF07	Como Analista de recaudación me gustaría tener un proceso automático que envíe cada cierta hora del día los archivos de débito a un correo personalizado de tal manera que pueda tener de forma automática la base de cobros para verificar y enviar.	8	6	14	12%	10	12%	101	88
HU-RF08	RF08	Como Analista de recaudación me gustaría poder importar los archivos de respuesta de débito que envían las entidades de tal manera que pueda realizar la cobranza masiva de todos los Clientes y a su vez se envíe la notificación a su teléfono de referencia.	9	4	13	11%	15	18%	62	88
HU-RF02	RF02	Como Analista de recaudación me gustaría tener una sola plataforma de consulta de afiliaciones de Clientes de tal manera que pueda centralizar las consultas del estado de las afiliaciones.	7	3	10	9%	15	18%	48	36
HU-RF06	RF06	Como Analista de recaudación me gustaría poder generar archivos de afiliación leyendo desde la base de datos de tal manera que me permita agilizar el envío de las tramas a las entidades bancarias.	7	4	11	10%	20	24%	40	72
					114		82			425

Figura 27: Desarrollo del Product Backlog

2.1.3 Priorización del Product Backlog

Posterior a la definición de las historias de Usuario se procedió a establecer la prioridad de cada historia para poder hacer las entregas en función de lo que establece la metodología que es la entrega de valor al Cliente. Para este proceso se hizo uso del método "Peso relativo" y se ordenó cada historia por el requisito que involucra menor costo y mayor valor al Cliente, para esto se hizo un focus group con el equipo de trabajo donde cada participante en orden empezó a dar su apreciación tomando el un valor del 1 al 9 aplicándolo a cada historia de usuario en base a 2 aspectos:

1. Cuál es el impacto positivo de la funcionalidad sobre el producto software si se llega a completar.
2. Cuál es el impacto negativo de la funcionalidad sobre el proyecto el producto software si no se llega a completar.

Para este método se desarrolló una matriz para definir el peso relativo de cada historia de Usuario y se muestra a continuación:

Tabla 19: Peso relativo

HU	Beneficio Relativo	Penalización Relativa	Valor Total	Valor porcentual	Estimación	Costo porcentual	Prioridad
HU-RF01	9	9	18	16%	5	6%	259
HU-RF02	7	3	10	9%	15	18%	48
HU-RF03	9	9	18	16%	5	6%	259
HU-RF04	9	9	18	16%	5	6%	259
HU-RF05	8	4	12	11%	7	9%	123
HU-RF06	7	4	11	10%	20	24%	40
HU-RF07	8	6	14	12%	10	12%	101
HU-RF08	9	4	13	11%	15	18%	62

Fuente: Elaboración propia

Se ordenó el Product Backlog en función a la prioridad establecida por el método del Peso Relativo, se actualizaron en todas las fuentes de archivos que administraba cada equipo, la lista de Backlog quedó priorizada para su postrer desarrollo en función al valor de negocio que cada Historia de Usuario proponía, el cuadro priorizado quedó de la siguiente manera:

Tabla 20: Desarrollo del Peso Relativo

HU	Beneficio Relativo	Penalización Relativa	Valor Total	Valor porcentual	Estimación	Costo porcentual	Prioridad
HU-RF01	9	9	18	16%	5	6%	259
HU-RF03	9	9	18	16%	5	6%	259
HU-RF05	8	4	12	11%	7	9%	123
HU-RF07	8	6	14	12%	10	12%	101
HU-RF08	9	4	13	11%	15	18%	62
HU-RF02	7	3	10	9%	15	18%	48
HU-RF06	7	4	11	10%	20	24%	40

Fuente: Elaboración propia

Luego de la priorización del Product Backlog se dio lugar a establecer el peso de cada historia de usuario, decidimos usar el método del Planning póker para cada historia donde cada participante dio su estimación en una reunión, cada participante estimó la complejidad de cada historia mediante la asociación de los valores usados por la serie Fibonacci. Se registraron las estimaciones que hicieron cada participante del equipo en el siguiente cuadro:

Tabla 21: Priorización del Product Backlog

Código de Historia	Req. Funcional	Scrum Master	Product Owner	Scrum Team 1	Scrum Team 2	Scrum Team 3	Total PoH
HU-RF01	RF01	13	13	20	20	20	86
HU-RF02	RF02	40	40	40	20	40	180
HU-RF03	RF03	13	13	20	20	20	86
HU-RF04	RF04	20	13	20	20	20	93
HU-RF05	RF05	40	100	100	100	100	440
HU-RF06	RF06	40	100	100	20	100	360
HU-RF07	RF07	40	100	100	100	100	440
HU-RF08	RF08	40	100	100	100	100	440

Fuente: Elaboración propia

2.2 Sprint Planning

Se tuvo lugar luego de la priorización del Product Backlog y su respectiva priorización donde hubo una reunión del equipo de trabajo:

Participantes:

- Scrum Master
- Product Owner
- Team Developer

2.2.1 El Artefacto

Para poder obtener el Sprint Backlog se creó la siguiente estructura que almacenaría la información del proyecto donde se tomaron los siguientes campos principales:

Tabla 22: Plantilla del Sprint Backlog

Código del Sprint	Responsable	Código de Historia de Usuario	Descripción de la tarea	Tiempo estimado

Fuente: Elaboración propia

2.2.2 El Proceso de desarrollo

Se tomó la historia de Usuario con código de historia de acuerdo a su orden de priorización y se empezó a realizar el análisis para generar historias de usuario a nivel técnico por cada historia del Product Backlog priorizado. Luego del análisis con el equipo de trabajo se definieron los siguientes Sprint para el proyecto de “Rediseño de débitos automáticos” y se asignó el tiempo estimado en horas para cada tarea, el artefacto se desarrolló de la siguiente manera:

Tabla 23: Desarrollo del Sprint Backlog – Sprint 1

Código del Sprint	Responsable	Código de Historia de Usuario	Descripción de la tarea	Tiempo estimado
Sprint.1	Scrum Team 1	RF06	Importar afiliados a BD OAC	16
Sprint.1	Scrum Team 1	RF03	Agregar un campo de teléfono al formulario del Cliente	0.5
Sprint.1	Scrum Team 2	RF03	Modificar el SP de registro de afiliación	8
Sprint.1	Scrum Team 2	RF03	Crear un método del WS de Transacción - insertar	8
Sprint.1	Scrum Team 2	RF03	Crear un método del WS de Transacción - modificar	8
Sprint.1	Scrum Team 1	RF03	Pruebas unitarias	8

Fuente: Elaboración propia

En la siguiente figura se muestra los cambios funcionales realizados por el equipo de desarrollo cumpliendo con lo establecido en el Sprint 1, se representan las funcionalidades de caer al usuario final.

Figura 28: Formulario de afiliación de clientes de fija

En la tabla 24 se muestra el Sprint Backlog con los requerimientos técnicos desarrollados para el sprint 2 planificado para entregar como producto final.

Tabla 24: Desarrollo del Sprint Backlog - Sprint 2

Código del Sprint	Responsable	Código de Historia de Usuario	Descripción de la tarea	Tiempo estimado
Sprint.2	Scrum Team 1	RF04	Agregar un campo al formulario del Cliente	0.5
Sprint.2	Scrum Team 2	RF04	Modificar el SP de registro de afiliación	8
Sprint.2	Scrum Team 3	RF04	Reutilizar un método del WS de Transacción	8
Sprint.2	Scrum Team 1	RF04	Modificar la tabla de afiliaciones para el nuevo campo	8
Sprint.2	Scrum Team 1	RF04	Pruebas unitarias	4

Fuente: Elaboración propia

A continuación, se muestra los entregables del Sprint 1 y sus funcionalidades del sistema de cara al Usuario final, se muestra en la figura 28 los aspectos funcionales que se desarrollaron en sprint con el equipo de trabajo.

The screenshot shows a web browser window with the title 'Transacciones SIAC Postpago - Internet Explorer provided by IBM Corporation'. The main heading is 'Afiliación a Débito Automático'. Below the heading, there is a summary bar with the following information: 'Nro Teléfono/Servicio: 991691523. Tipo de Cliente: Consumer. Contacto : \$NYS QKNTRYRXS QXRRXSQK', 'Código del cliente: 7.1081410.00.00.100000', and 'Nombre del Cliente: NOMBRE DEL CLIENTE'. Below this, a field for 'Límite de Crédito:' is set to '269'. The main form area is divided into sections: 'Datos de la Transacción' with dropdowns for 'Medio:' (Débito en Cuenta), 'Entidad:' (Interbank), 'Moneda:' (SOLES), and 'Tipo de Cuenta:' (Cuenta de Ahorros); 'Número de Cuenta:'; 'Nombre del Titular:' (\$NYS QKNTRYRXS QXRRXSQK) and 'DNI del Titular:' (14133214); 'Teléfono de referencia:' (2211442) and 'Correo electrónico:'. There are radio buttons for 'Monto a debitar: Recibo del Mes' (selected) and 'Monto Máximo', and checkboxes for 'SMS Afiliación' (checked) and 'SMS Débito'. 'Origen de Solicitud:' has radio buttons for 'CLARO' (selected) and 'Entidad Financiera'. Other fields include 'Estado:' (Pendiente), 'Fecha de Solicitud:' (14/12/2015), 'Fecha de Proceso:' (14/12/2015), 'Fecha de Afiliación:', 'Fecha de Rechazo:', and 'Fecha de Desafiliación:'. A 'Notas:' field is at the bottom. At the very bottom are buttons for 'Grabar', 'Cerrar', and 'Imprimir Formulario'.

Figura 29: Formulario de afiliación al débito

Al formulario de afiliación se agregó un nuevo campo para el número telefónico con las validaciones de necesarias para mantener la integridad de los datos, se agregó el método al servicio web para poder registrar la información de las afiliaciones y se realizaron las pruebas unitarias.

En la siguiente tabla se muestra los requisitos técnicos del sprint 4 para el desarrollo de los procesos automático de las afiliaciones al débito automático, no hay imágenes de referencia porque el tratamiento del cambio fue a nivel de código fuente.

Tabla 25: Desarrollo del Sprint Backlog - Sprint 3

Código del Sprint	Responsable	Código de Historia de Usuario	Descripción de la tarea	Tiempo estimado
Sprint.4	Scrum Team 2	RF05	Crear la estructura del Shell	16
Sprint.4 a	Scrum Team 2	RF05	Crear un SP para obtener las afiliaciones	16
Sprint.4 h	Scrum Team 1	RF05	Incluir el WS de envió de SMS	8
Sprint.4 9	Scrum Team 1	RF05	Diferenciar las tramas de las entidades	16
Sprint.4	Scrum Team 3	RF05	Pruebas unitarias	4
Sprint.4 D	Scrum Team 1	RF05	Incluir método de envió de correo	8

Fuente: Elaboración propia

En la siguiente tabla se observa los requerimientos para poder realizar un formulario que consulte las afiliaciones, de esta manera el Usuario podrá ver los diferentes estados de los registros.

Tabla 26: Desarrollo del Sprint Backlog – Sprint 5

Código del Sprint	Responsable	Código de Historia de Usuario	Descripción de la tarea	Tiempo estimado
Sprint.5	Scrum Team 1	RF02	Crear un SP de que consulte los afiliados	16
Sprint.5	Scrum Team 1	RF02	Diseñar el formulario	8
Sprint.5	Scrum Team 3	RF02	Desarrollar un método de consulta en el WS de afiliación	16
Sprint.5	Scrum Team 3	RF02	Pruebas unitarias	4

Fuente: Elaboración propia

Las características del formulario fueron diseñadas y probadas con los insumos que se crearon en el sprint, esto incluye los procedimientos almacenados y los métodos del servicio.

Figura 30: Formulario de afiliación al débito

En la siguiente tabla se muestra los requerimientos del sprint que incluye los procesos de generación de archivos de afiliación para el proceso de débito automático.

Tabla 27: Desarrollo del Sprint Backlog – Sprint 6

Código del Sprint	Responsable	Código de Historia de	Descripción de la tarea	Tiempo estimado
-------------------	-------------	-----------------------	-------------------------	-----------------

Usuario				
Sprint.6	Scrum Team 1	RF06	Crear SP de consulta para afiliaciones	24
Sprint.6	Scrum Team 1	RF06	Crear método y lógica en la arquitectura del proyecto	24
Sprint.6	Scrum Team 3	RF06	Diseñar pantalla de generación de archivo	4
Sprint.6	Scrum Team 3	RF06	Pruebas unitarias	16

Fuente: Elaboración propia

Los formularios diseñados para cumplir el objetivo del sprint se muestran a continuación, las funcionalidades del requerimiento se diseñaron y programaron con las características acordadas con el usuario.

GENERACION DE ARCHIVOS DE AFILIACION CONSULTA

Detalle archivo: BCP_ENVIO_AFIL_SGA_SOLES_01022016.txt

Fecha de generación: 01/02/2016

Entidad: BCP Moneda: Soles

Operación: SGA

Operación	Codigo cliente/	N° Cta. / N° Tarjeta	Moneda	Importe máx.
SGA	6426860	19133617570139	Soles	200.00
SGA	6426861	19133617570140	Soles	150.00
SGA	6426865	19133617570141	Soles	300.00
SGA	6426870	19133617570150	Soles	350.00
SGA	6426878	19133617570153	Soles	150.00
SGA	6426854	19133617570133	Soles	200.00
SGA	6426850	19133617570101	Soles	300.00
SGA	6426848	19133617570110	Soles	200.00

Descargar Archivo Generar Reporte Salir

Figura 31: Formulario de afiliación al débito

En la tabla 28 se visualiza el trabajo planificado para un conjunto de procesos a nivel de código fuente que estuvieron relacionada con el proceso automático del débito para la generación de archivos y para la mensajería por correo.

Tabla 28: Desarrollo del Sprint Backlog – Sprint 7

Código del Sprint	Responsable	Código de Historia de Usuario	Descripción de la tarea	Tiempo estimado
Sprint.7	Scrum Team 2	RF07	Crear la estructura del Shell	16
Sprint.7	Scrum Team 2	RF07	Crear un SP para obtener las deudas	24
Sprint.7	Scrum Team 1	RF07	Incluir el WS de envío de SMS	8
Sprint.7	Scrum Team 3	RF07	Incluir método de envío de correo	8

Fuente: Elaboración propia

En la tabla 29 se puede observar los requisitos para poder importar los archivos de las entidades a fin de leer las configuraciones y guardar la información de clientes afiliados en la base de datos.

Tabla 29: Desarrollo del Sprint Backlog – Sprint 8

Código del Sprint	Responsable	Código de Historia de Usuario	Descripción de la tarea	Tiempo estimado
Sprint.8	Scrum Team 2	RF08	Crear estructura en BD para soportar diferentes entidades	32
Sprint.8	Scrum Team 1	RF08	Diseñar un formulario para poder cargar los archivos de respuesta	0.5
Sprint.8	Scrum Team 1	RF08	Diferenciar la tramas de las entidades	32
Sprint.8	Scrum Team 3	RF08	Incluir mensaje SMS	16
Sprint.8	Scrum Team 3	RF08	Pruebas unitarias	32

Fuente: Elaboración propia

En la siguiente figura siguiente se muestra las funcionalidades desarrolladas por el equipo de desarrollo donde incluyeron la recepción de los archivos de afiliación que provenían de las entidades recaudadoras.

Figura 32: Formulario de recepción de la afiliación

El Sprint Backlog se desarrolló en una plantilla en Excel donde se administraba todos los requerimientos técnicos que fueron derivados a cada recurso del equipo Scrum, se asignaron responsables por cada tarea y se revisaba constantemente para el seguimiento de las tareas. Ver figura 32.

Codigo de Release	Entidad Sprint	Responsable	Historia de Usuario	Descripción de la tarea	HE
Release 1	Sprint 1	Miguel Matos	RF01	Registrar afiliación a RED SAC	16
Release 1	Sprint 1	Miguel Matos	RF02	Agregar un campo al formulario del Cliente	0.5
Release 1	Sprint 1	Emma Guzmán	RF03	Modificar el SP de registro de afiliación	6
Release 1	Sprint 1	Frady Farnanda	RF04	Crear un método del WS de Transacción -Insertar	6
Release 1	Sprint 1	Miguel Matos	RF05	Crear un método del WS de Transacción -modificar	6
Release 1	Sprint 1	Miguel Matos	RF06	Modificar la tabla de afiliaciones para el nuevo campo	4
Release 1	Sprint 1	Frady Farnanda	RF07	Pruebas unitarias	4
Release 1	Sprint 1	Frady Farnanda	RF08	Incluir SMS de afiliación	4
Release 1	Sprint 2	Miguel Matos	RF09	Agregar un campo al formulario del Cliente	Horas 0.5
Release 1	Sprint 2	Emma Guzmán	RF04	Modificar el SP de registro de afiliación	6
Release 1	Sprint 2	Frady Farnanda	RF04	Validar un método del WS de Transacción	6
Release 1	Sprint 2	Miguel Matos	RF04	Modificar la tabla de afiliaciones para el nuevo campo	4
Release 1	Sprint 2	Miguel Matos	RF04	Incluir SMS de afiliación	4
Release 1	Sprint 3	Frady Farnanda	RF04	Agregar un campo al formulario del Cliente	Horas 0.5
Release 1	Sprint 3	Miguel Matos	RF04	Validar un método del WS de Transacción	6
Release 1	Sprint 3	Emma Guzmán	RF04	Modificar la tabla de afiliaciones para el nuevo campo	4
Release 1	Sprint 3	Frady Farnanda	RF04	Incluir SMS de afiliación	4
Release 1	Sprint 3	Frady Farnanda	RF04	Incluir SMS de afiliación	Horas 40.5
Release 2	Sprint 4	Emma Guzmán	RF06	Crear la estructura del Shell	16
Release 2	Sprint 4	Miguel Matos	RF06	Crear un SP para obtener las afiliaciones	16
Release 2	Sprint 4	Miguel Matos	RF06	Incluir al WS de envío de SMS	6
Release 2	Sprint 4	Frady Farnanda	RF06	Definir el flujo de las entidades	4
Release 2	Sprint 4	Frady Farnanda	RF06	Pruebas unitarias	6
Release 2	Sprint 4	Miguel Matos	RF06	Incluir método de envío de correo	6
Release 2	Sprint 5	Miguel Matos	RF02	Crear un SP de que consulta los afiliados	Horas 16
Release 2	Sprint 5	Miguel Matos	RF02	Crear el formulario	16
Release 2	Sprint 5	Frady Farnanda	RF02	Desarrollar un método de consulta en el WS de afiliación	16
Release 2	Sprint 5	Frady Farnanda	RF02	Pruebas unitarias	4
Release 3	Sprint 6	Miguel Matos	RF04	Crear SP de consulta para afiliaciones	Horas 6
Release 3	Sprint 6	Miguel Matos	RF06	Crear método y login en la arquitectura del proyecto	24
Release 3	Sprint 6	Frady Farnanda	RF06	Crear estructura de gestión de archivos	4
Release 3	Sprint 6	Frady Farnanda	RF06	Pruebas unitarias	6
Release 3	Sprint 7	Emma Guzmán	RF02	Crear la estructura del Shell	Horas 16
Release 3	Sprint 7	Miguel Matos	RF02	Crear un SP para obtener los afiliados	16
Release 3	Sprint 7	Miguel Matos	RF02	Incluir al WS de envío de SMS	16
Release 3	Sprint 7	Miguel Matos	RF02	Definir el flujo de las entidades	6
Release 3	Sprint 7	Frady Farnanda	RF02	Incluir método de envío de correo	16
Release 3	Sprint 7	Frady Farnanda	RF02	Pruebas unitarias	Horas 6
Release 4	Sprint 8	Emma Guzmán	RF06	Crear estructura en RED para registrar diferentes entidades	Horas 36
Release 4	Sprint 8	Miguel Matos	RF06	Crear el método de la recepción	16
Release 4	Sprint 8	Emma Guzmán	RF06	Crear un SP para actualizar el estado de los usuarios	6
Release 4	Sprint 8	Miguel Matos	RF06	Crear un formulario para poder cargar los archivos de respaldo	0.5
Release 4	Sprint 8	Miguel Matos	RF06	Definir el flujo de las entidades	16
Release 4	Sprint 8	Miguel Matos	RF06	Incluir método SMS	16
Release 4	Sprint 8	Frady Farnanda	RF06	Pruebas unitarias	Horas 16
				Total Dias	65

Figura 33: Desarrollo del Sprint Backlog

Luego de obtener el Sprint Backlog, para el seguimiento se definió una pizarra donde se monitoreó el avance de las tareas del equipo, los avances del Sprint mediante un cuadro o tablero de tareas donde de categorizaron en 5 dimensiones. Estas son las siguientes:

- Impediment Backlog: Tareas que impiden que el Sprint se desarrolle correctamente
- Task: Tareas del Sprint.
- In Progress: Tareas en curso o en progreso de ejecución.
- Stopped: Tareas paralizadas.
- Completed: Tareas terminadas al 100%

Para cada iteración se trabajó el seguimiento del Sprint Planning de la forma tradicional con un cuadro en la pizarra, esto en simultáneo con la plantilla de Excel con el objetivo de mostrar los requerimientos técnicos por cada tarea de tal manera que se mantenga la visibilidad para todo el equipo, esta forma de trabajos se usó de forma diaria en las reuniones de Daily Scrum y Retrospectiva. Ver Figura 33.



Figura 34: Desarrollo del Sprint Backlog en la pared

El cuadro de seguimiento del Sprint Planning se estableció para todos los proyectos de la cartera, se pudo conseguir un mejor seguimiento pues era de manera visible a todos los recursos y las tareas estaban a la vista de los equipos.

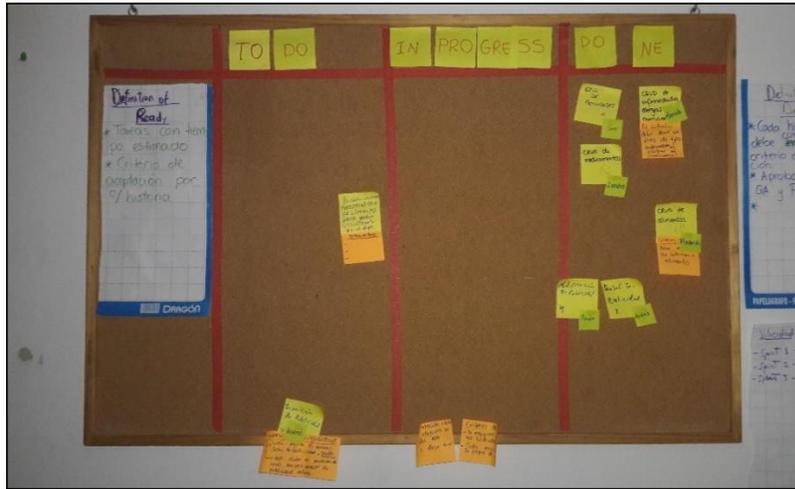


Figura 35: Seguimiento del Sprint Backlog en la pizarra

2.3 El Burdown Chart

Se estableció en monitor de trabajos para poder dar seguimiento al proyecto, el primer cuadro Burdown Chart para inicializar las tareas del sprint se desarrolló para monitorear el avance de los diferentes sprints de los proyectos. En la columna de las Y se tomó como referencia los puntos de historia vs. La línea de tiempo en el eje de las X.

- Y = Puntos de historia.
- X = Tiempo en días del total del proyecto.

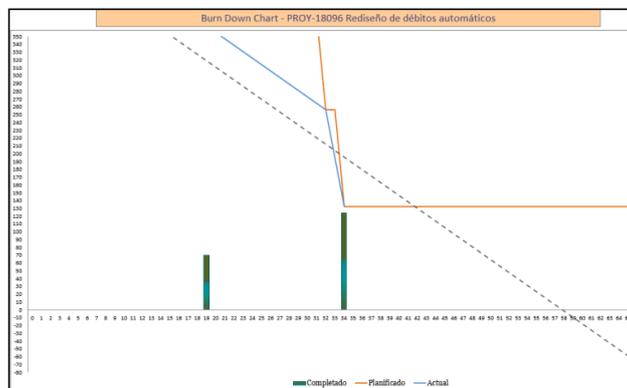


Figura 36: Desarrollo del Burdown Chart

2.4 Daily Meeting

En la primera reunión diaria el objetivo fue se asignar las tareas a cada participante para poder dar inicio al primer sprint, y posterior se estableció el primer documento Impediment Backlog. Ver figura 36.

Codigo	Impediment BL	Equipo	Responsable
HU-RF03	No se tiene información del WS que valida si el número de teléfono es de CLARO	Team Scrum	Sven Riveros
	Gestionar permisos en producción para migrar la data a OAC	Team Scrum	Sven Riveros
	No se tiene el documento detallado de WS de TransaccionesAfiliacionesDEAU	Team Scrum	Miguel Matos
Codigo	Impediment BL	Equipo	Responsable
HU-RF04	El Web Service de envío de SMS no está disponible a ciertas horas	Scrum Master	Marleny Llamocca
HU-RF01	No se tiene data de prueba para seguir con el proceso	Producto Owner	Luis Villalva
	Problemas al instalar el SGA en el ambiente de prueba	Scrum Master	Marleny Llamocca
	El procedimiento para obtener la lista de Cliente de la fija está lento.	Team Scrum	Miguel Matos

Figura 37: Objetivo del desarrollo del Daily Meeting

Los diferentes problemas que ocurrieron en el transcurso del desarrollo se plasmaron en el documento en Excel, estos fueron priorizados por el Scrum Master en coordinación con todo el equipo y lo que se buscaba era la solución pronta de cada punto incluido en el documento de impedimentos.

2.5 Review

Al finalizar del primer sprint se tuvo una reunión para poder revisar el resultado del trabajo realizado, en esta reunión se tuvo la participación de las siguientes personas:

- Scrum Master: Marleny Llamocca
- Product Owner: Luis Villalva
- Equipo Scrum: Miguel Matos, Fredy Fernández, Emma Guzmán
- Usuario de recaudación: Pedro Arrunátegui
- Usuario de recaudación: Guillermo Flores

Se realizó la primera presentación de resultados referente al sprint trabajado el cual se vieron los siguientes puntos:

- La migración de la base de clientes afiliados.
- La funcionalidad de las pantallas del sistema afectado.
- Las funcionalidades de los servicios afectados

Finalmente se completó el registro con los objetivos alcanzados, mediante un cuadro de seguimiento de tareas.

RF	Criterio de aceptación	Objetivos	Avance	Estado
RF04	Los campos deben estar validados en su nivel de integridad de datos. Los formularios deben tener la claridad y facilidad de registrar la información.	Se validaron todos los registros migrados a la BD OAC	100%	Done
		Se agregó el campo de teléfono al formulario	100%	Done
		Se guardó correctamente la información desde el formulario	100%	Done
		Se validaron todos los campos.	100%	Done
RF01	Se deberá registrar mediante el front el numero de telefono en BD y se deberá poder ver el cambio de estado para afiliaciones masivas.	No se está replicando el estado en la base de SGA lo que se registra en la base de OAC	90%	El SP no tiene réplica

Figura 38: Review del sprint de desarrollo – Fase 1

En los Sprint se registraron las actividades en estado Done y aquellas que no se pudieron terminar las cuales pasaron luego para su revisión y posterior resolución.

2.6 Retrospective

Luego de terminar los 2 primeros sprint se realizó el primer entregable (Reléase) donde se determinó la reunión para la retrospectiva, para ellos se usó la técnica “Bien, Mejorable, Mejora”, para lo cual se convocó a una reunión donde estaban todos los participantes del equipo, cada uno dio un aspecto que le pareció bueno, malo o regular, cada participación de los integrantes fue categorizada por el Scrum Master y luego analizada para evitar cometer el mismo incidente en el siguiente sprint, de igual manera se hizo con los aspectos regulares y buenos de la experiencia del sprint a fin de mejorar a medida que avancen los sprints hasta finalizar todo el proyecto.

Tabla 30: Tabla de retrospectiva del proyecto

<i>Equipo Scrum</i>	<i>Bien</i>	<i>Mejorable</i>	<i>Mejora</i>
Scrum Master	La comunicación del equipo es buena.	El manejo del tiempo de las reuniones del Review y Daily Meeting	Hora de iniciar las labores
Product Owner	Se investiga con rapidez.	Se debe tener mayor conocimiento de los servicios de CLARO. El resultado de las pruebas funcionales	El diseño de los formularios debe tener una mejor presentación. Las aprobaciones demoran mucho.
Scrum Team 2	Apoyo de otros equipos en técnicas de desarrollo	El resultado de las pruebas unitarias	Muchos cambios en la arquitectura
Scrum Team 1	Las maquinas tienen buen rendimiento de hardware	El resultado de las pruebas unitarias	Demora en la gestión de documentación para el pase.
Scrum Team 3	Los mockups para el usuario están bien detallados	La data de prueba debe ser mejor generada	Validar el ambiente de producción, los accesos y permisos

Fuente: Elaboración propia

Luego de tener la información se estableció un cuadro de prioridades para poder mitigar estos aspectos y se definió de la siguiente manera asociando cada Responsable a cada tarea.

Tabla 31: Tabla de respuesta a la retrospectiva del proyecto

<i>Responsable</i>	<i>Mejora</i>	<i>Importancia</i>	<i>Acción a tomar</i>
Product Owner/ Scrum Team 1	Generar data de prueba al inicio de cada Sprint.	ALTA	La data de prueba debe ser mejor generada
Scrum Team 3	Crear más escenarios para cada método nuevo o modificado	ALTA	El resultado de las pruebas unitarias/funcionales
Product Owner	Definir con mayor detalle el alcance inicial	MEDIA	Muchos cambios en la arquitectura
Product Owner	Solicitar con anticipación las aprobaciones	MEDIA	Demora en la gestión de documentación para el pase.
Scrum Team 2	Verificar los estándares de otros proyectos	BAJA	El diseño de los formularios debe tener una mejor presentación.

Fuente: Elaboración propia

Con la priorización de la retrospectiva se desarrolló de manera más ágil el aprendizaje de Scrum en los diferentes procesos del desarrollo del software, bajo este enfoque se desarrollaron los diferentes proyectos y a medida que se iba obteniendo la agilidad y experiencia se establecieron metas en cada uno de los proyectos, la forma de llevar a cabo la gestión de los proyectos bajo el marco Scrum fue a través de los formatos creados en las tablas y figuras del presente anexo.

3. Documentación del Proyecto

Documento de Acta de constitución del proyecto “PROY-18096 Rediseño de débitos automáticos”, este documento demuestra el inicio del proyecto a solicitud del Cliente, esta es una parte documentaria que siempre se establecen en todos los proyectos que desarrollará TeamSoft S.A.C.

1. DATOS DEL PROYECTO					
Código del Proyecto	Nombre del Proyecto				
PROY-18096	Rediseño de Débito Automático				
Dependencia de otras ideas	Descripción de la dependencia				
No Aplica					
¿Se está atendiendo a este proyecto con más de una PS?	No Aplica				
2. AFECTADOS RELEVANTES					
No.	Rol	Nombre completo	Área	Responsabilidad	Plazos de compromisos
01	Gerente de Recaudación y Cobranza	Manuel Perez Diaz	Recaudación	Análisis de Recaudación	
02	Supervisor de Recaudación Canales Externos	Fernando Custodio	Recaudación	Análisis de Recaudación	
03	Analista de Recaudación	Guillermo Flores	Recaudación	Análisis de Recaudación	
04	Analista de Recaudación	Pedro Arrunátegui	Recaudación	Análisis de Recaudación	
05	Analista de Recaudación	Elizabeth Valentin	Recaudación	Análisis de Recaudación	
Sistemas / Aplicaciones / Productos / Procesos involucrados o impactados					
Sistema/Aplicación/Producto/Procesos	Descripción del impacto				
SIOP	Se deben crear nuevos formularios que permitan al usuario: <ul style="list-style-type: none"> • Generar solicitudes de afiliación o débito, así como subir las respuestas a estas solicitudes a demanda, ya sea para uno o varios registros. • Crear una nueva afiliación y también modificarla, a través de la creación del formulario de Gestión de afiliación. • Visualizar todas las afiliaciones registradas por el cliente ingresando su código o su teléfono a través de 				

Figura 39: Acta de constitución del proyecto

Propuesta de solución de los requerimientos funcionales que el área de recaudación aprobó para el inicio del proyecto informático, fue el documento base que sirvió a lo largo de todo el proceso de aplicación de Scrum como el Backlog del producto que posterior fueron desarrollados por el equipo de desarrollo.

4. REQUERIMIENTOS FUNCIONALES- ALCANCE		
Lista de requerimientos de usuario		
Transformar el pedido del usuario en una lista de requerimientos donde se ha completado información faltante, consolidado/fusionado diversos puntos de vista, resuelto posibles conflictos y documentado el conjunto completo de requerimientos del usuario.		
Código de RU	Requerimiento de Usuario	Situación actual
RU01	Todas las solicitudes de afiliación que realicen los usuarios a través de un CAC deben ingresar con estado pendiente.	Actualmente en el SIACPOST solo se permiten generar solicitudes de afiliación con estado aprobado, pese a que dicha solicitud aún no ha sido confirmada por la entidad financiera o banco.
RU02	Se debe registrar, de forma obligatoria, un número móvil de contacto al momento de realizar la solicitud de afiliación a través de un CAC.	Actualmente no se está registrando un número móvil de contacto cuando se realiza una solicitud de afiliación a través de un CAC.
RU03	Se debe enviar mensajes de texto, al número registrado al momento de la solicitud de afiliación, donde se le informe al cliente si su solicitud de afiliación al débito automático fue aprobada o rechazada por la entidad financiera o bancaria.	Actualmente no se le notifica al cliente la respuesta a su solicitud de afiliación.
RU04	Hay que automatizar los procesos de generación y envío de archivos de solicitudes de afiliación así como las bases de débito que se envía a las entidades financieras y bancos. De la misma forma se debe automatizar la recepción de los archivos de respuesta a las solicitudes enviadas.	Actualmente, el usuario modifica, según el formato indicado por la entidad financiera o banco, los archivos y bases que son generados y descargados desde el SIOP. Luego estos son enviados a un correo o subidos a la web de la entidad financiera para su procesamiento. De la misma forma, las respuestas son enviadas por correo o descargadas desde la web de dicha entidad financiera para ser subidas de forma manual al SIOP.
RU05	El SIOP debe permitir generar archivos de solicitud de afiliación y bases a demanda para uno o varios registros de tal forma que no se tenga que esperar al proceso que correrá en automático al finalizar el día. Registro de nuevas afiliaciones y su modificación además de las consultas del estado de las solicitudes de afiliación y la visualización de las solicitudes registradas por los clientes móviles.	Actualmente, el usuario modifica, según el formato indicado por la entidad financiera o banco, los archivos y bases que son generados y descargados desde el SIOP. Luego estos son enviados a un correo o subidos a la web de la entidad financiera para su procesamiento. De la misma forma, las respuestas son enviadas por correo o descargadas desde la web de dicha entidad financiera para ser subidas de forma manual al SIOP.

Figura 40: Propuesta de solución del proyecto

Detalle de los requerimientos de Usuario asociados a requerimientos funcionales que posteriormente fueron establecidos en el Product Backlog.

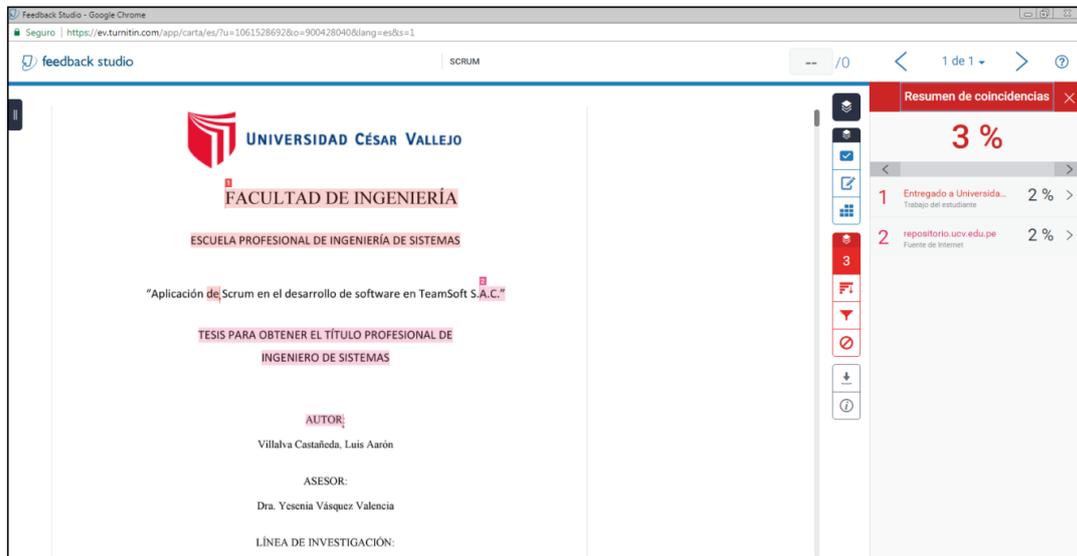
Lista de requerimientos funcionales
 Crear los Requerimientos Funcionales (RF) que establecen la solución funcional que se dará a los Requerimientos de Usuario. Por ejemplo: Pantalla de ingreso de datos, alerta de correo electrónico, proceso para determinar los empleados a enviar correo electrónico, alerta de correo electrónico masivo, etc.

RU	RF	Descripción corta de RF	Valor de Negocio	Complejidad	Es Interface	Equipo Claro / Proveedor que atenderá	Orden de Atención	Dependencia entre RF's
RU04 RU05	RF01	Migración de los objetos relacionados al DEAU desde la BD del SIOP a la BD del OAC, así como la migración de datos de las BD SGA y BSCS.	M	M	NO	CLARO	1ª	
RU01 RU02	RF02	Modificación del proceso de afiliaciones al débito automático de la aplicación SGA.	M	M	SI	CLARO	2ª	RF01
	RF03	Modificación del proceso de afiliaciones al débito automático de la aplicación SIACPOST.	M	M	SI	CLARO	2ª	RF01
RU05	RF04	Modificación del proceso de afiliaciones al débito automático de la aplicación SIACHEFC.	M	M	SI	CLARO	2ª	RF01
	RF05	Creación de formularios en SIOP para crear y modificar solicitudes de afiliación así como la implementación de consultas.	M	M	SI	CLARO	3ª	RF01
	RF06	Modificación de módulo de DEAU de la aplicación del SIOP.	M	M	SI	CLARO	3ª	RF01
RU03 RU04	RF07	Implementación de Shells que generen y envíen solicitudes de afiliación al débito automático así como las bases de débito en TIM-FTP1. Shell que realicen la depuración de los repositorios de los archivos generados y recepcionados. Validación del operador del teléfono móvil que registrarán los clientes.	M	M	NO	CLARO	4ª	RF01, RF06

Figura 41: Requerimientos de Usuario – Nivel 0

Anexo 4: Resultados de Autenticidad

A continuación, se muestra la imagen que certifica la originalidad de la tesis donde se puede apreciar como resultado del software Turnitin un 3% de coincidencias, esto constata que el presente trabajo cumple con los requisitos de autenticidad autorizados.



The screenshot displays the Turnitin Feedback Studio interface. The main content area shows the title page of a thesis from Universidad César Vallejo, Facultad de Ingeniería, Escuela Profesional de Ingeniería de Sistemas. The thesis title is "Aplicación de Scrum en el desarrollo de software en TeamSoft S.A.C.". The author is Villalva Castañeda, Luis Aarón, and the advisor is Dra. Yesenia Vásquez Valencia. The research line is listed as "LÍNEA DE INVESTIGACIÓN:". On the right side, a red sidebar titled "Resumen de coincidencias" (Summary of Similarities) shows a total similarity score of 3%. Below the score, two sources are listed: "Entregado a Universida..." (2%) and "repositorio.ucv.edu.pe" (2%).

Source	Similarity Percentage
Entregado a Universida... (Trabajo del estudiante)	2 %
repositorio.ucv.edu.pe (Fuente de Internet)	2 %

Figura 42: Autenticidad de Tesis