

UNIVERSIDAD PERUANA UNIÓN

ESCUELA DE POSGRADO

Unidad de Posgrado de Ingeniería y Arquitectura



Modelo Integrado de Scrum y CMMI para la eficacia de la desviación del tiempo en el Ciclo de Vida de los Proyectos de Ingeniería de Software de la Consultora Assembly Solution SAC

Tesis

Presentada para optar el grado académico de Magíster en
Ingeniería de Sistemas

Por

Benjamín David Reyna Barreto

Asesor:

Dr. Guillermo Mamani Apaza

Lima, febrero de 2018

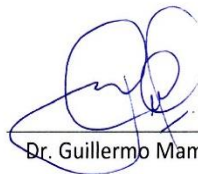
ANEXO 07 DECLARACIÓN JURADA DE AUTORIA DEL INFORME DE TESIS

Dr. Guillermo Mamani Apaza, de la Escuela de Posgrado, Unidad de Posgrado de Ingeniería y Arquitectura, de la Universidad Peruana Unión.

DECLARO:

Que el presente informe de investigación titulado: *"MODELO INTEGRADO DE SCRUM Y CMMI PARA LA EFICACIA DE LA DESVIACIÓN DEL TIEMPO EN EL CICLO DE VIDA DE LOS PROYECTOS DE INGENIERÍA DE SOFTWARE DE LA CONSULTORA ASSEMBLY SOLUTION SAC"* constituye la memoria que presenta el **Bachiller Benjamín David Reyna Barreto** para aspirar al Grado académico de Maestro ha sido realizada en la Universidad Peruana Unión bajo mi dirección. Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente constancia en Lima, a los 22 días del año 2018


Dr. Guillermo Mamani Apaza

*Modelo Integrado de SCRUM y CMMI para la eficacia de la
desviación del tiempo en el ciclo de vida de los proyectos de
Ingeniería de Software de la Consultora Assembly
Solution SAC*

TESIS

Presentada para optar el Grado Académico de Maestro en Ingeniería de
Sistemas con mención en Dirección y Gestión de Tecnologías de
Información

JURADO DE SUSTENTACIÓN



Dr. César Augusto Gálvez Vivanco
Presidente



Dra. Erika Acuña Salinas
Secretaria



Dr. Guillermo Mamani Apaza
Asesor



Dr. Jorge Sánchez Garcés
Vocal



Mg. Omar Loaiza Jara
Vocal

Lima, 22 de febrero de 2018

Dedicatoria

A Dios, por brindarme sabiduría y salud, a mis padres por su apoyo incondicional, a mis hijos por ser la razón de mi existir, y a todos los que de una u otra manera contribuyeron en este trabajo.

Agradecimientos

Agradecer en primer lugar a nuestro Dios, quien es la fuente de todo conocimiento, inteligencia y sabiduría.

Al Dr. Guillermo Mamani, por su disposición brindada para el desarrollo de esta tesis.

A mi familia que me acompañó dándome fuerzas, mucho amor y ánimo en cada momento de mi vida para conseguir este sueño.

Y a todos mis amigos que contribuyeron con sus ideas, consejos y motivación para el logro de este objetivo.

Contenido

DEDICATORIA.....	iii
AGRADECIMIENTOS	iv
CONTENIDO.....	v
INDICE DE TABLAS	viii
INDICE DE FIGURAS	ix
INDICE DE ANEXOS	x
RESUMEN	xi
ABSTRACT	xii
CAPITULO I	9
1. INTRODUCCION.....	9
1.1. Situación Problemática	9
1.2. Formulación del Problema	15
1.2.1. Problema General	15
1.3. Justificación de la investigación	16
1.4. Objetivos de la Investigación	19
1.4.1. Objetivo General	19
1.4.2. Objetivos Específicos	19
1.5. Hipótesis de Investigación	20
CAPITULO II	21
2. MARCO TEÓRICO.....	21
2.1. Marco Histórico de la Investigación	21
2.1.1. La historia de SCRUM.....	21
2.1.2. La historia de CMMI	24
2.2. Antecedentes de la Investigación	27
2.3. Bases teóricas	29

2.3.1.	Ingeniería de Software	29
2.3.2.	Metodologías Ágiles	39
2.3.2.1.	SCRUM.....	39
2.3.3.	CMMI.....	47
2.4	Ciclo de vida del proyecto de software	59
CAPITULO III		60
3.	METODOLOGÍA DE LA INVESTIGACIÓN	60
3.1.	Tipo y Diseño de la Investigación	60
3.1.1.	Tipo de Investigación.....	60
3.1.2.	Diseño de la Investigación.....	60
3.1.3.	Esquema de investigación.....	61
3.2.	Unidad de Análisis	62
3.3.	Población de Estudio	62
3.4.	Técnica y Tamaño de muestra.....	62
3.5.	Recolección de Datos	63
3.6.	Análisis e Interpretación de Investigación	63
3.7.	Delimitación del proyecto de investigación	63
CAPITULO IV		65
4.	CONSTRUCCION DEL MODELO INTEGRADO.....	65
4.1.	Diseño del Modelo Integrado SCRUM v CMMI	65
4.2.	Áreas de Proceso Scrum y CMMI-DEV Nivel 2	65
4.2.1.	Planificación del Proyecto (PP) y Scrum.	66
4.2.2.	Monitorización y Control del Proyecto (PMC) y Scrum.....	70
4.2.3.	Gestión de Requisitos (REQM) y Scrum	71
CAPITULO V		72
5.	RESULTADOS Y DISCUSIÓN	72
5.1.	Descripción de las características de la población de estudio	72

5.2. Parámetros estadísticos de la desviación de los planificado vs realidad.....	76
5.3. Prueba de normalidad de la muestra de desviación porcentual de los proyectos.	78
5.4. Diferencia de medias de las desviaciones porcentuales para grupos independientes.	79
5.5. Contrastación de la hipótesis de la influencia del SCRUM y CMMI en el ciclo de vida de los proyectos.	80
CONCLUSIONES.....	83
RECOMENDACIONES	86
REFERENCIAS BIBLIOGRÁFICAS	87
ANEXOS	¡Error! Marcador no definido.

Índice de tablas

Tabla 1. Área de Proceso Gestión de Requerimiento	50
Tabla 2. Área de Proceso: Planificación del Proyecto (PP).....	51
Tabla 3. Área de proceso: Monitorización y Control del Proyecto	52
Tabla 4. Área de Proceso: Gestión de Configuración	54
Tabla 5. Área de Proceso: Aseguramiento de la Calidad	55
Tabla 6. Área de Proceso: Gestión de acuerdos con proveedores (SAM)	56
Tabla 7. Área de Proceso: Medición y Análisis	57
Tabla 8. Monitorización y Control del Proyecto (PMC).....	58
Tabla 9. Planificación del Proyecto y Scrum	67
Tabla 10. Product Backlog del proyecto.....	68
Tabla 11. Estimación de los Sprint	69
Tabla 12. Monitoreo y Control del Proyecto y Scrum	70
Tabla 13. Gestión de Requisitos y Scrum	71
Tabla 14. Planificación y ejecución del desarrollo de los requerimientos del proyecto de Sistemas de Devueltas de facturación.....	73
Tabla 15. Planificación y ejecución del desarrollo de los requerimientos del proyecto de Medicina Física.....	74
Tabla 16. Planificación y ejecución del desarrollo de los requerimientos del proyecto del proyecto de sistema académico.....	75
Tabla 17. Resumen de casos analizados de los proyectos en estudio	76
Tabla 18. Parámetros descriptivos de la desviación porcentual de los proyectos	77
Tabla 19. Prueba estadística no paramétrica de Kolmogorov-Smirnov.....	78
Tabla 20. Medias de los grupos para muestras independientes de las desviaciones de los proyectos.....	79
Tabla 21. Prueba t-student para muestras independientes.....	80

Índice de figuras

Figura 1. Estadísticas de la gestión de Proyectos de Software.....	12
Figura 2. Proyectos más pequeños mayor éxito que los más grandes	12
Figura 3. Enfoques Agiles proyectos exitosos.....	14
Figura 4. Factores que influyen en el éxito de los proyectos.....	18
Figura 5. La historia de los CMMS [23]	24
Figura 6. Triangulo de la gestión de proyectos.....	31
Figura 7. Modelo Cascada	36
Figura 8. Modelo V	36
Figura 9. Modelo de Prototipos	37
Figura 10. Ciclo de vida Espiral.....	38
Figura 11. Mitología RUP	39
Figura 12. Principios Scrum	40
Figura 13. Organización Scrum.....	42
Figura 14. Procesos Scrum	43
Figura 15. Flujo Scrum para un Spring.....	44
Figura 16. Duración de las reuniones de Scrum	45
Figura 17. Scrum vs Cascada Tradicional.....	46
Figura 18. Esquema de área de proceso CMMI	48
Figura 19. Procesos Centrales CMMI	49
Figura 20. Modelo Integrado Scrum y CMMI	65
Figura 21. CMMI-DEV Nivel 2 y sus Áreas de proceso	66

Índice de anexos

Anexo 1. Requerimientos Sistema de Devueltas de Facturación.....	91
Anexo 2. Requerimientos del Sistema de Medicina Física.....	92
Anexo 3. Requerimientos del Sistema Académico.....	93
Anexo 4. Plantilla para Historias de Usuario	94
Anexo 5. Plantilla del Estado del Spring de los proyectos.....	95
Anexo 6. Plantilla para el control de reuniones de retrospectiva	96
Anexo 7. Interfaz del TRELLO donde se monitorearon los proyectos del grupo de control. Módulo GTH	97
Anexo 8. Interfaz del TRELLO donde se monitorearon los proyectos del grupo de control. Módulo GTH Vacaciones.....	98

Resumen

El presente estudio tiene como objetivo determinar la eficacia del modelo integrado Scrum y CMMI en el ciclo de vida de los proyectos de ingeniería de la consultora ASSAC.

Esta investigación es aplicada porque se utilizó las teorías de SCRUM y CMMI para optimizar el ciclo de vida del proyecto. Es tecnológica porque fue de carácter ejecutivo, es decir; se implementó el Modelo Integrado de Scrum y CMMI para la gestión de los Proyectos de Ingeniería de Software de la consultora ASSAC.

La población de estudio estuvo conformada por todos los requerimientos de los proyectos de ingeniería de software del centro de consultoría ASSAC que se han desarrollado en el año 2016 y 2017 por los equipos Scrum. La muestra fue intencional y estuvo conformado por tres proyectos: medicina física, sistema de devueltas, y sistema de matrícula y caja.

Para contrastar la hipótesis se aplicó la prueba t-student para muestras no relacionadas, y de acuerdo a los resultados se encontró que existe una diferencia de medias entre el grupo de proyectos con SCRUM y CMMI y el grupo de control de 31.558 con un nivel de significancia p-valor de 0.000 siendo este menor a 0.05. De este resultado se concluye que el Modelo de SCRUM y CMMI influyen de manera positiva en el ciclo de vida de los proyectos de Ingeniería de Software.

Palabras claves: SCRUM, CMMI y ciclo de vida del proyecto.

Abstract

The objective of this study is to determine the effectiveness of the integrated Scrum and CMMI model in the life cycle of the engineering projects of the ASSAC consultancy.

This research is applied because the SCRUM and CMMI theories were used to optimize the life cycle of the project. It is technological because it was of an executive nature, that is to say; the Integrated Scrum and CMMI Model was implemented for the management of the Software Engineering Projects of the ASSAC consultancy.

The study population was made up of all the requirements of the software engineering projects of the ASSAC consulting center that were developed in 2016 and 2017 by the Scrum teams. The sample was intentional and consisted of three projects: physical medicine, system of returns, and system of registration and cash.

To test the hypothesis, the t-student test was applied for unrelated samples, and according to the results it was found that there is a difference in means between the group of projects with SCRUM and CMMI and the control group of 31,558 with a level of significance p-value of 0.000, this being less than 0.05. From this result it is concluded that the SCRUM and CMMI Model have a positive influence on the life cycle of Software Engineering projects.

Keywords: SCRUM, CMMI and project life cycle.

Capítulo I

1. Introducción

1.1. Situación problemática

El desarrollo de software, es considerado como una importante actividad económica en muchas industrias de muchos países tales como: la India. Estados Unidos, Brasil, etc., siendo en algunos casos parte fundamental en el crecimiento de sus economías representando en un 85 por ciento de este rubro [1]. Las empresas cuando comienzan a crecer, la falta de tener un proceso claro de desarrollo de software termina creando problemas en toda la empresa [2]. Por tal motivo, existe una búsqueda constante de empresas de software en herramientas, modelos, metodologías, marcos de trabajo, etc. que ayuden a gestionar proyectos de software y den como resultado software a bajo costo, con calidad, que satisfaga las necesidades del cliente, que sea mantenible, desarrollado de acuerdo al tiempo planificado, es un desafío que las empresas de software deben afrontar.

La complejidad que existe al construir un software ha sido por los años 60 un desafío difícil de manejar. Cuando las empresas de software tenían poco tiempo de haber empezado en este rubro, se usó el término “crisis del software” para describir la situación por lo que estaba atravesando el cual fue manifiesto [3], algunas causas fueron: sobre costos, incumplimiento en los plazos, software ineficiente, software de baja calidad, requisitos incumplidos, software no mantenible, fracaso y no entrega del software.

Según estudios académicos realizados por la SEI, en una comparación entre dos metodologías aplicadas en diferentes proyectos dentro de 25 organizaciones sobre su costo y beneficio, tales como CMMI (tradicional) y Scrum (ágil), se tomaron en cuenta seis variables, tales como: costo, tiempo, productividad, calidad, satisfacción y ROI costos de calidad, las cuales en promedio dieron un resultado de que las metodologías ágiles reportan un 29% de menor costo, 91% de menor tiempo, 97% de mayor productividad, 50% de mayor calidad, 400% de mayor satisfacción y un 470% de mayor ROI que CMMI. [4]. Dos estudios similares de costo y beneficio de SW-CMM (Capability Maturity Model for Software) se realizaron en Israel por investigadores de desarrollo de software [5] y The Data and Analysis Center for Software (DACS) [6] , [7].

Para el año 2003 el 66% de los proyectos de software en el mundo estaban usando metodologías ágiles, de los cuales el 90% de estos usaban Extreme Programming [8] aunque el número de proyectos que usaban XP disminuyeron un 23% [9], es sorprendente como el mercado ha logrado abarcar las metodologías ágiles.

La cantidad de proyectos de software que están usando Scrum, se está acrecentando, llamando así la atención de grandes corporaciones como: google, yahoo y microsoft. Además, existen cerca de 50000 proyectos que están usando scrum. [6], [10]

Según el estudio realizado por Forrester Research, a inicios del 2009 a muchos desarrolladores fueron evaluados bajo la interrogante: ¿cuál es la metodología de software que usas actualmente? La encuesta mostró los siguientes resultados: scrum (10.9%), agile modeling (6%), feature

driven development (fdd, 3.8%), test driven development (tdd, 3.4%), extreme programming (xp, 2.9%), lean development (2.1%), microsoft solutions framework (msf) for agile (1.8%), agile data method (1.6%), adaptive software development (asd, 1.3%), six sigma (0.9%), crystal (0.3%), behavior driven development (bdd, 0.2%) y dynamic systems development method (dsdm, 0.2%). De este estudio podemos concluir que a la actualidad se han desarrollado muchas metodologías ágiles, de modo que el mercado ha optado por probar muchas de estas, debido a lo cual, se puede observar un descenso en el uso de XP, sin embargo, se destaca la presencia del marco de trabajo scrum como líder de estas [11].

En el siglo pasado se hicieron muchos esfuerzos para afrontar esta crisis. Se desarrollaron estrategias para contrarrestar esta crisis las cuales fueron: ingeniería de software, gestión predictiva de proyectos, producción basado en procesos.

En los años 80, se establecieron los objetivos para la gestión de proyectos de software debía cumplir. A fin de verificar si los proyectos se concluyeron con éxito. Estos objetivos estaban basados en el tiempo planificado, el alcance del presupuesto estimado, satisfacción de las necesidades del cliente las que fueron evidenciadas en las funcionalidades que el cliente necesita.

Desde 1994 Standish Group, realizó una encuesta sobre el éxito o fracaso de los proyectos de software, obteniendo como resultado que uno de cada tres proyectos es exitoso, como se muestra en la figura 1.

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

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

Figura 1. Estadísticas de la gestión de Proyectos de Software.

Fuente: CHAOS 2015

En la siguiente encuesta se muestra como los proyectos más pequeños tienen una probabilidad mucho mayor de éxito que los más grandes. Como se aprecia en la figura 2.

CHAOS RESOLUTION BY PROJECT SIZE			
	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	100%	100%	100%

The resolution of all software projects by size from FY2011-2015 within the new CHAOS database.

Figura 2. Proyectos más pequeños mayor éxito que los más grandes

Fuente: CHAOS 2015

Frente a estas estadísticas que presentan una realidad crítica surge una gran interrogante ¿por qué fracasan los proyectos? Frente a esta pregunta muchas empresas en sus procesos de construcción de software utilizan una metodología tradicional RUP que está centrado en la técnica, es formal y tiene un conjunto de artefactos a utilizar en cada fase. Sin embargo, para un desarrollo ágil de software aparece el modelo scrum que está centrado principalmente en la persona. Uno de los aspectos del modelo es desarrollar el Ciclo de Vida del Proyecto por ser un factor de éxito.

De acuerdo a la metodología scrum existe diferencia entre el compromiso e involucramiento. Compromiso es aquel personal que se identifica con el proyecto, tiene un sentido de pertenencia y lealtad y participa de manera proactiva. Un personal involucrado carece de estas características.

“A los fines de su informe, el Grupo Standish definió el éxito del proyecto como entregado a tiempo, manteniéndose dentro del presupuesto y alcanzando la finalización con todas las características planificadas intactas. En su informe CHAOS 2015, publicaron sus hallazgos sobre el estado de la cascada y los proyectos ágiles que se llevaron a cabo desde el año fiscal 2011 hasta el año fiscal 2015. Los resultados de ese informe se muestran en la tabla a continuación “ [12].

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%

The resolution of all software projects from FY2011-2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

Figura 3. Enfoques Agiles proyectos exitosos.

Fuente: CHAOS 2015

La consultora Assembly Solutions, no es ajena a toda esta problemática que se observa a nivel mundial. Maneja una cartera de proyectos de software y los problemas más comunes que se presentan son: falta de una definición clara del alcance y sus especificaciones, algunos usuarios no saben describir o explicar de manera objetiva cuáles son sus necesidades, no se priorizan los requerimientos del usuario, estimaciones inexactas en los tiempos de desarrollo, retrasos en la entrega del producto, incumplimiento habitual en la fecha de entrega establecida como límite para su elaboración, personal poco comprometido con el seguimiento y la documentación de cada una de las etapas del proyecto, no se hace un seguimiento permanente al desarrollo de los proyectos de desarrollo, ausencia en el control de cambios, no se realiza una evaluación de los riesgos frente a la implementación y dificultades en la obtención de las conformidades entre otros.

Los problemas que se presentan encajan dentro de las cuatro dimensiones del compromiso organizacional que son: participación, identidad, pertenencia y lealtad que se ven reflejados en los equipos de software. En las reuniones de los equipos de trabajo se observa poca colaboración entre ellos, muchas veces no se considera la opinión de los miembros y cuando existe algún problema no se pide opinión al personal; en síntesis, no existe una participación activa de los integrantes del equipo. En cuanto a la identidad de cada miembro de equipo realmente no sienten orgullo de trabajar en el área y difícilmente expresan a sus amigos en la universidad donde trabajan; es decir, no tienen identidad. Respecto a la pertenencia, se observa que existe un defraude respecto a la institución y la mayoría del personal no siente que pertenece a su área y/o institución. Finalmente, en la dimensión lealtad se manifiesta en el personal que cumple exactamente con su trabajo y no tiene la actitud de aportar más de lo debido, y algunos anhelan tener un trabajo diferente.

1.2. Formulación del Problema

1.2.1. Problema General

Frente a las diferentes casuísticas de la situación problemática de los proyectos de software que se ha expuesto en la sección anterior, se formula la siguiente interrogante:

¿Cuál es la eficacia del modelo integrado scrum y CMMI en la desviación del tiempo en el ciclo de vida de los proyectos de ingeniería de software de la consultora ASSAC?

1.3. Justificación de la investigación

“Con el avance de la tecnología hace que los proyectos de software sean poco exitosos ya que los cambios tecnológicos se realizan con tanta rapidez que es muy difícil planificar a largo plazo. El cliente requiere de soluciones de software de gestión con entregas cortas debido a la automatización de sus procesos. Cuando los procesos son más grandes y complejos el uso de métodos tradicionales para el desarrollo de proyectos de software de gestión tomaría más tiempo para la culminación del mismo y por ende una entrega a destiempo, con el riesgo de que al momento de entregar los procesos hayan cambiado” [13].

En [14] se menciona que la integración de metodologías ágiles con CMMI combinan la adaptabilidad la previsibilidad para atender mejor los requerimientos de los clientes. Esto le ayudó a desarrollar sistemáticamente una estrategia para la duplicar la productividad.

“Es útil para las organizaciones que crean un nuevo marco de gestión de proyectos basado en las prácticas de CMMI y scrum” [15].

La integración de la metodología scrum y CMMI será un factor de éxito ya que se tendrá un equipo multidisciplinario con amplia experiencia, así como herramientas adaptadas a los requerimientos del proceso y la empresa [16].

En [17] se menciona que los proyectos de software se gestionan usando marcos tradicionales o ágiles. La combinación de la metodología SRUM y CMMI-DEV puede ayudar a incluir las mejores prácticas de ambas

tecnologías, teniendo en cuenta los factores clave de la gestión de proyectos.

Las empresas que desarrollan software comprenden cada vez adoptar más herramientas, metodologías, etc., que ayuden a la gestión de desarrollo de un producto de software. El dilema de las empresas es cómo desarrollar software con éxito y gestionar la complejidad que existe en producir un software. La innovación y los cambios de requisitos por parte de los clientes, también se suma a esta complejidad. “Encontrar el equilibrio entre los valores de las metodologías ágiles y la robustez de los modelos tradicionales brinda una luz para enfrentar esta situación dado que la Gestión de la complejidad requiere disciplina en los procesos, mientras que la gestión del cambio exige capacidad de adaptación. Es así como, CMMI proporciona disciplina” [18] y la metodología Scrum mejora la adaptación a esta diversidad de circunstancias [1].

Al identificar el gran uso de las metodologías ágiles, se puede notar que para poder tener éxito se requiere de una disciplina en el proceso, esto será demostrado en el desempeño del equipo, en la planeación, en el seguimiento, en la retroalimentación continua, en la interacción permanente con los clientes, en las prácticas de pruebas, en el manejo de propiedad colectiva, así como con el seguimiento de los valores del manifiesto ágil [19], [20].

“Según el análisis del Grupo Standish en los últimos 21 años, ha sido la identificación y clasificación de los factores que trabajan juntos para hacer

que los proyectos sean más exitosos. Los resultados de este año muestran la siguiente lista y clasificación de factores” [21]. Figura 4

FACTORS OF SUCCESS	POINTS	INVESTMENT
Executive Sponsorship	15	15%
Emotional Maturity	15	15%
User Involvement	15	15%
Optimization	15	15%
Skilled Resources	10	10%
Standard Architecture	8	8%
Agile Process	7	7%
Modest Execution	6	6%
Project Management Expertise	5	5%
Clear Business Objectives	4	4%

Figura 4. Factores que influyen en el éxito de los proyectos.

Fuente: CHAOS 2015

Por lo expuesto, este modelo integrado de scrum y CMMI permitirá:

- Determinar de una manera formal las mejoras que se pueden introducir en los procesos.
- Mayor seguimiento a las decisiones de gestión de proyecto, en situaciones cuando los proyectos y la organización crecen y aumenta la presión.
- Involucrar a la organización y los actores externos, en el seguimiento, aprendizaje y mejoras.
- Definición clara de los roles del equipo a fin de facilitar el deslinde de responsabilidades.

- e. Normalización de las tareas, para un mejor aprovechamiento del conocimiento, eficiente y evitar problemas de calidad.
- f. Desarrollo de los requerimientos del cliente. Permite tener una mejor planificación del proyecto.

1.4. Objetivos de la Investigación

1.4.1. Objetivo General

Determinar la eficacia del modelo integrado scrum y CMMI en la desviación del tiempo en el ciclo de vida de los proyectos de ingeniería de software de la consultora ASSAC

1.4.2. Objetivos Específicos

- a. Analizar el modelo de gestión de proyectos scrum y los procesos del CMMI para establecer los elementos de integración y aplicarlo de manera conjunta.
- b. Diseñar el modelo integrado de SCRUM con CMMI en base a los artefactos de SCRUM y los procesos del modelo CMMI.
- c. Aplicar el modelo integrado de SCRUM con CMMI en base a los artefactos de SCRUM y los procesos de CMMI a los proyectos de ingeniería de software de la consultora ASSAC.
- d. Determinar la desviación porcentual del tiempo de desarrollo de los requerimientos en base a la planificación de los proyectos de ingeniería de software.
- e. Comparar la desviación porcentual del tiempo de desarrollo de los requerimientos en base a la planificación de los proyectos de

ingeniería de software que aplicó SCRUM y CMMI con los proyectos que no utilizaron SCRUM y CMMI.

1.5. Hipótesis de Investigación

La aplicación del modelo integrado scrum y CMMI mejora la eficacia en la desviación del tiempo ciclo de vida de los proyectos de ingeniería de la consultora ASSAC.

Capítulo II

2. Marco teórico

2.1. Marco histórico de la investigación

2.1.1. La historia de SCRUM

Jeff Sutherland y Ken Schwaber crearon el marco de trabajo scrum por los años 90. En el año de 1995 fue presentado en la conferencia OOPSLA en Austin, Texas (Estados Unidos) y publicado el documento "Proceso de Desarrollo de Software SCRUM".

Ken y Jeff heredaron el nombre 'scrum' desde el 1986 innovadora de papel 'The New Desarrollo de Nuevos Productos Juego' por Takeuchi y Nonaka, dos pensadores de gestión reconocida. Con el término 'scrum' Nonaka y Takeuchi hacen referencia al juego del rugby hacer hincapié en la importancia de los equipos y algunas analogías entre un deporte de equipo como el rugby y tener éxito en el juego de desarrollo de nuevos productos. La investigación que se describe en su artículo mostró que el desempeño sobresaliente en el desarrollo de nuevos productos complejos, se logra cuando los equipos, ya que las pequeñas y auto-organización de las unidades de la gente, se alimentan con los objetivos, no con las tareas. Los mejores equipos son aquellos que se dan dirección dentro del cual tienen espacio para idear sus propias tácticas sobre cómo mejor cabeza hacia su objetivo común. Los equipos requieren autonomía para alcanzar la excelencia.

Mientras que en el proceso de desarrollo y uso de las primeras versiones de scrum, Ken pidió al profesor Babatunde A. Ogunnaike Tunde, un importante investigador de control de procesos, observar el proceso de desarrollo de software. Tunde investigó varias metodologías de desarrollo de software comerciales para concluir que el proceso de cascada y predictivo, no es un buen ajuste para el trabajo de desarrollo de software. Se confirmó que el enfoque empírico de Scrum es el procedimiento preferido.

En febrero del 2001, Jeff y Ken fueron parte de los 17 líderes de desarrollo de productos de software que crearon el Manifiesto de desarrollo de software ágil.

Tras el manifiesto ágil, la Alianza Ágil fue fundada con Ken Schwaber siendo su primer presidente.

En el 2001, gran parte inspirado por Kent Beck, Ken Schwaber co-autor del primer libro sobre Scrum con Mike Beedle, desarrolló el software ágil con scrum.

En el 2002, Ken Schwaber fundó la Alianza Scrum con Mike Cohn y Esther Derby, crearon el curso de Certified Scrum Master con el objetivo de promocionar el uso de Scrum por todo el mundo. La certificación tenía el objetivo de ofrecer formación práctica en Scrum.

En el 2006, Jeff Sutherland creó su propia compañía, Scrum Inc, sin dejar de ofrecer y enseñar a los cursos de certificado de Scrum.

Ken salió de la Alianza Scrum en el otoño del 2009, y fundó Scrum.org para mejorar aún más la calidad y la eficacia de scrum, principalmente a través de la serie profesional scrum.

Con la primera publicación de la Guía de Scrum en el 2010, y sus actualizaciones incrementales en el 2011 y 2013, Jeff y Ken establecieron el organismo reconocido a nivel mundial del conocimiento de scrum.

Desde su primera publicación en 1995, y hasta ahora, scrum ha sido adoptado por una gran cantidad de compañías de desarrollo de software en todo el mundo. Hoy en día, es reconocido como el marco más aplicado para el desarrollo de software ágil. Más de 1000 libros han sido publicados en Scrum.

El método, sin embargo, también se ha aplicado con éxito en otros ámbitos, por ejemplo, fabricación, marketing, operaciones y la educación.

Después de un arduo trabajo y la incorporación de mejoras que se hicieron en el tiempo, Ken Schwaber y Jeff Stherland en el año 2011, decidieron publicar la guía oficial del Scrum lo cual estaba especificado por un conjunto de reglas y buenas prácticas aplicadas de acuerdo interpretación del profesional que lo implantaba.

Entre el año 2013 y 2016 se añadieron dos grandes cambios. Se dio énfasis al Scrum diario como elemento de planificación y también reunión de grooming a refinement. En el año 2016 se añadieron los valores de Scrum, que hacen que el marco de trabajo funcione en una organización.

2.1.2. La historia de CMMI

En el año de 1984 fue fundado el SEI (Software Engineering Institute), a través del financiamiento del Departamento de Defensa de los EEUU, hoy en día es administrado por la Universidad de Carnegie Mellon. Creado con el objetivo de desarrollar modelos de evaluación y mejora del proceso de desarrollo de software, que en sus inicios contribuyeron a la solución de problemas de planificación y desarrollo de software, en la construcción de sistemas militares del ejército de los Estados Unidos. El SEI, tiene como razón primordial proporcionar la plataforma para mejorar el proceso de construcción de software y la calidad del producto. El éxito está dado por procesos definidos, tareas de desarrollo de software que se pueden gestionar [22].

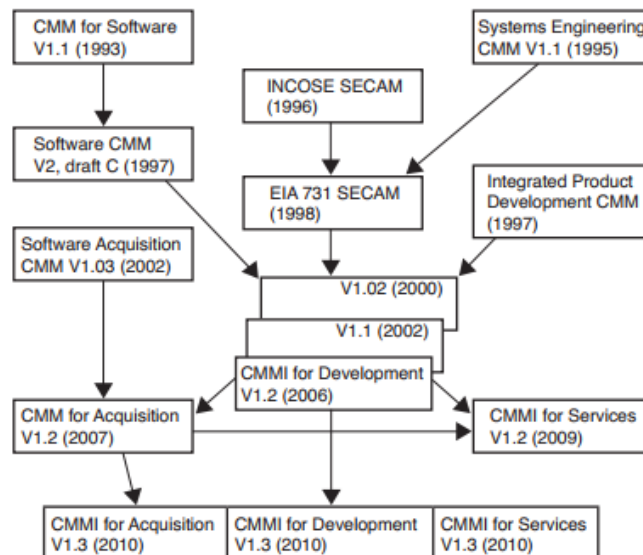


Figura 5. La historia de los CMMS [23]

CMMI: Historia y perspectivas

Los modelos CMM se concentran en la mejora de los procesos de una organización. Definen los elementos fundamentales para que se pueda lograr este objetivo, tienen una hoja de ruta de mejora evolutivo el cual consiste en pasar de procesos inmaduros ad hoc a procesos gestionados y maduros de mejor eficiencia y calidad.

El SEI creó el primer CMM para organizaciones de desarrollo de software y lo publicó en el libro, The Capability Maturity Model: Guidelines for improvement the Software Process [SEI 1995].

El libro detalla los principios de experiencias de buenas prácticas desarrolladas hace casi un siglo en la mejora de procesos. “Los resultados obtenidos en la aplicabilidad han sido de mucho valor a la aproximación de la mejora de procesos, el cual ha sido confirmado por la usabilidad de las buenas prácticas a través del tiempo. Las organizaciones que aplicaron estos modelos se han visto beneficiados en el crecimiento de la productividad y una mejora de la calidad, mejorando la duración del ciclo productivo y han logrado planificaciones y presupuestos más precisos y fiables” [24].

Evolución de CMMI

CMMI: Desde el pasado hacia el futuro

Existen CMM para innumerables disciplinas, los más notables son: modelos para ingeniería de software, para la adquisición de software, para el desarrollo de productos y procesos (IPPD).

A pesar de que estos modelos han probado ser útiles para diversas organizaciones de diferentes industrias, la usabilidad de muchos modelos ha sido un factor problemático. Un sin número de organizaciones tienen

la intención de extender sus esfuerzos de mejora a diversos grupos de procesos en sus organizaciones. Las diferencias entre cada modelo de disciplina específica utilizados, considerando su arquitectura, contenido y aproximación, han disminuido las capacidades de estas organizaciones para lograr el éxito en sus mejoras. Por otro lado, la aplicabilidad de diversos modelos no integrados en una organización es costosa en términos de formación, evaluaciones y de actividades de mejora. Los modelos CMMs permiten regular el problema de utilizar múltiples CMM [25].

CMMI: integración y mejora continúa

Desde sus inicios el SEI agrupa las mejores prácticas a lo que llamamos “constelaciones”. La constelación tiene una colección de componentes que se usa para crear modelos, los materiales de formación y los documentos de evaluación. Una constelación para la gestión de servicios CMMI for Services(CMMI-SVC) , CMMI for Acquisition (CMMI-ACQ) [23]. La tercera constelación, CMMI Development (CMMI-DEV) [18], alineado a los procesos de construcción de software.

CMMI para el desarrollo

El CMMI-DEV establece una guía de buenas prácticas CMMI en una organización de desarrollo de software. Estas prácticas, están centradas en actividades para productos y servicios de calidad a fin de satisfacer a los usuarios y clientes finales. La colección de estas buenas prácticas, son el resultado del desarrollo de la industria y el gobierno que ha obtenido de

la aplicabilidad del marco de CMMI-DEV. “Está basado en el CMMI Model Foundation o CMF el cual reúne el trabajo realizado por organizaciones de desarrollo para adaptar CMMI para su uso en el desarrollo de productos y servicios” [26].

2.2. Antecedentes de la Investigación

David F. Rico, 2004. What is the ROI of agile vs traditional methods?

“Se examinaron más de 300 artículos sobre métodos ágiles; los costos, el cronograma, la productividad, la calidad y la satisfacción del cliente se encontraron en 69 estudios; y los datos de ROI se identificaron en 29 estudios.

El ROI de los métodos ágiles era cuatro veces más que los métodos tradicionales costosos, dos veces menos que los económicos, y los mejores métodos tradicionales y ágiles tenían un ROI igual” [7].

María E. Rojas Izaquita, (2011). Agilizando lo ágil: Un Framework Para La Desarrollo de Software Bajo el Modelo CMMI En Compañías Que Usan Metodologías Ágiles De Desarrollo De Software Usando El Modelo Acelerado De Implementación (Aim).

“Con este trabajo se pretende presentar una guía para que las organizaciones cuyo ciclo de vida de desarrollo de software se enmarcan dentro de las metodologías ágiles de desarrollo puedan mapear las prácticas de dicho ciclo de vida contra los objetivos genéricos, practicas genéricas, objetivos específicos y prácticas específicas del modelo CMMI niveles 2 y 3” [27].

Aleli Sánchez Méndez, (2014). Model and practices of dac methodology integrating the agile methods, pmbok and cmmi-dev.

“El presente artículo tiene como objetivo elaborar una metodología para el desarrollo de software de gestión basado en componentes, integrando las prácticas ágiles, estándares internacionales de gestión de proyectos y calidad de software y el modelo CMMI, para la mejora de las evaluaciones en los indicadores que miden la ejecución del proyecto, la satisfacción del cliente, la calidad del proceso y del producto en proyectos de software” [13].

Gandomani, Taghi JavdaniZulzalil, HHazura, (2013). Compatibility of Agile Software Development Methods and CMMI.

“El objetivo principal de este estudio fue analizar este tema. Prácticas genéricas y prácticas específicas en CMMI se seleccionaron como criterios para verificar la compatibilidad de CMMI y métodos ágiles. Los resultados mostraron que CMMI y ágiles son compatibles en varias áreas de proceso. Sin embargo, hay serias incompatibilidades en los demás. En el nivel 3 y nivel 4 de CMMI, se han visto dos y un área de proceso incompatibles, respectivamente. Además, ágil no es compatible con la mayoría de las prácticas genéricas en los niveles 4 y 5 de CMMI. Los resultados también mostraron que estas incompatibilidades vuelven a la noción de organización y su rol en el desarrollo de software” [28].

“A nivel internacional la utilizan empresas como Amazon, Spotify, Ericsson y Adobe. Ahora también la adoptaron empresas locales, entre ellas, Globant, GlobalLogic, Mercado Libre, Despegar y Osde. Aunque se

originó en el mundo del desarrollo de software en los últimos años fue ganando terreno en otros rubros de productos y servicios” [29].

“El mundo del rugby se convirtió en una gran inspiración para los equipos de trabajo de las compañías. Con la idea de trasladar ese espíritu del deporte a la oficina, nació una metodología llamada scrum, que hoy utilizan grandes tecnológicas globales Amazon, Spotify, Ericsson y Adobe, entre otras y que se aplica cada vez más en empresas locales. Se trata de un conjunto de prácticas de trabajo en grupo que primero se expandió entre las desarrolladoras de software pero que luego se comenzó a imitar en firmas de productos y servicios de rubros variados” [29].

2.3. Bases teóricas

2.3.1. Ingeniería de software

a. Introducción

Son muchas las actividades por las cuales los desarrolladores codifican programas a fin de simplificar sus tareas, desde aplicaciones más simples hasta más complejas. Sin embargo, un mayor porcentaje de la construcción de un producto de software se realiza de manera profesional, a fin de automatizar los procesos de negocio de la empresa.

La ingeniería de software es la rama encargada del desarrollo profesional de software. Tal como lo expresó Barry Boehm [30] al

mencionar que los equipos informáticos se vuelven útiles para el hombre mediante programas informáticos, procedimientos y documentación asociada. De esta afirmación podemos extraer dos aspectos importantes que es el considerar el código escrito, la documentación y los procedimientos empleados. Por otro da énfasis de que el software debe satisfacer el propósito por lo cual fue desarrollado, es decir, prestar mucha importancia a que los objetivos y requisitos se cumplan.

Sommerville [31] extiende a un más lo antes mencionado porque considera a la ingeniería de software como una disciplina que engloba todo los aspectos relacionados a la construcción de un nuevo producto de, y todas las etapas del ciclo de vida.

La ingeniería de software con el objetivo de satisfacer las necesidades de acuerdo a un presupuesto, en un plazo determinado y con una calidad. Se conoce como el triángulo de la gestión de proyectos, como se muestra en la figura 6.

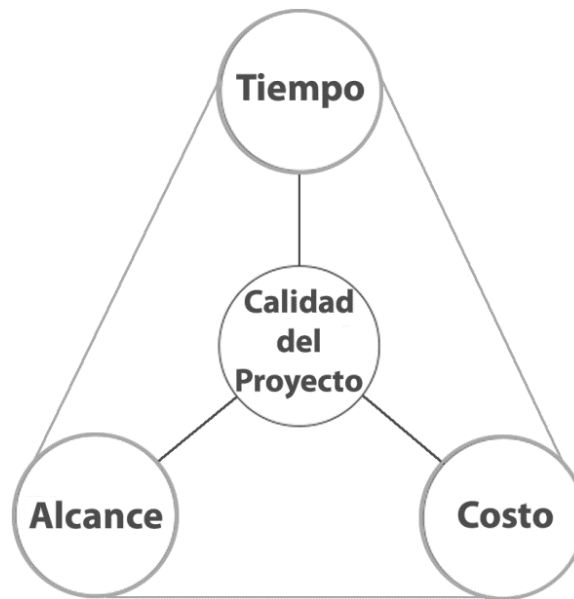


Figura 6. Triángulo de la gestión de proyectos.

Un enfoque sistémico ayudará a los ingenieros a construir productos de software de alta calidad. Parte de este esfuerzo consistirá en definir el método más adecuado y apropiado en cada situación.

Al enfoque sistémico se le denomina como proceso de software. En el cual se identifican cuatro actividades comunes a todo proceso de software:

- **Especificación.** En donde se define el propósito, las limitaciones y la operación del software que se va producir.
- **Desarrollo.** Está relacionado con el diseño técnico y la programación.
- **Validación.** Satisfacción del cliente.
- **Evolución.** La escalabilidad del software.

Cada software necesitará de diferentes tipos de desarrollo.

b. Proceso de Software

Consiste en describir el conjunto de actividades que lo componen, tales como: diseño de la interfaz, diseñar el modelo de datos, etc. A demás de estas actividades el proceso puede involucrar:

- **Productos.** Es el resultado del proceso.
- **Roles.** Responsabilidades del equipo.
- **Pre y pos-condiciones.** Declaraciones que se deben tomar en cuenta al desarrollar una actividad.

Los procesos de software son complejos y están basadas en las decisiones del equipo que toma parte. Existe una diversidad de situaciones que las empresas de software tienen que lidiar ya que no exista un aplantilla o estándar que indique los pasos que uno tiene que realizar para desarrollar cualquier producto de software.

c. Actividades para construir un software

Se idéntica cuatro actividades básicas para la construcción de un producto de software:

- **Especificación**

Es la ingeniería de requerimientos. Este proceso consiste en la definición, comprensión de la funcionalidad y las restricciones del proyecto de software.

Existen cuatro actividades de este proceso.

- Estudio de viabilidad.
- Análisis de los requisitos

- Especificación de los requisitos
- Validación de requisitos

- **Diseño e implementación**

Es donde las especificaciones se convierten en código ejecutable. Esta actividad considera el diseño y la programación.

Este proceso permite determinar la arquitectura de software, el modelo de base de datos, los algoritmos y las interfaces con otros sistemas. El diseño contempla 4 actividades básicas:

- Diseño de la arquitectura. Diseño general de las partes del sistema.
- Diseño de la interfaz. Diseño de los elementos visuales del sistema que permita la interacción con el usuario.
- Diseño de componentes. Es el diseño de la funcionalidad.
- Diseño de la base de datos. Estructura de los datos y su almacenamiento.

El tipo de proyecto será el factor determinante para la obtención de los resultados de esta actividad.

- **Validación y verificación**

Permite asegurar la funcionalidad del software de acuerdo a las expectativas del cliente. Está relacionado con el control de calidad [32].

La técnica que se usa para la validación es el testing o pruebas software. Esta técnica está basada en tres niveles:

- **Testing de software**

Procesos, métodos de trabajo y herramientas para identificar defectos en el software. El JUnit facilita esta tarea de pruebas unitarias [33].

Testing de integración

Esta prueba se enfoca en buscar defectos o fallos en las interfaces de comunicación o fallos en la interacción de componentes.

- **Testing de aceptación**

Estas pruebas se realizan con el sistema terminado y en las mismas condiciones que tendrá el sistema en producción. Estas pruebas también se le conoce como “Alpha Testing”.

d. El ciclo de vida del desarrollo de productos de software

El ciclo de vida, nos permite englobar una serie de procedimientos y etapas bien definidas para construir un producto de software. Este ciclo está dado por algunas de las siguientes fases:

Análisis. Permite determinar los objetivos, el estudio de viabilidad del sistema, tales como costes, beneficios, etc. A través de un plan preliminar con recomendaciones y alternativas posibles.

Definición de requisitos. Permite definir el objetivo, requerimientos y operaciones del producto. Se obtienen los requerimientos del cliente.

Desarrollo. Es la parte de la codificación del software.

Integración y pruebas. Permite testear el software antes de entrar a producción y ser usado por el usuario final.

Instalación y despliegue. Es la puesta en marcha del software en ubicación final.

Mantenimiento. A medida que el software se usa requiere de ciertos ajustes, evolucionar o corregir fallos de producción.

Evaluación. Consiste en evaluar el rendimiento del software. Durante la vida útil del software es necesario evaluar su rendimiento, las funcionalidades para la organización. Estas evaluaciones ayudan a la administración a tomar decisiones de mejora del software, reemplazo o sustitución.

No todas las metodologías de desarrollo consideran todas estas fases, algunas de ellas son alterados o modificadas. A continuación, se detallan algunas de ellas.

- **Modelo de Cascada.** Es el paradigma más simple está basado en un desarrollo lineal. Esto implica que cuando la primera fase termina continua la segunda y así hasta terminar la última fase. Como se observa en la Figura 7.

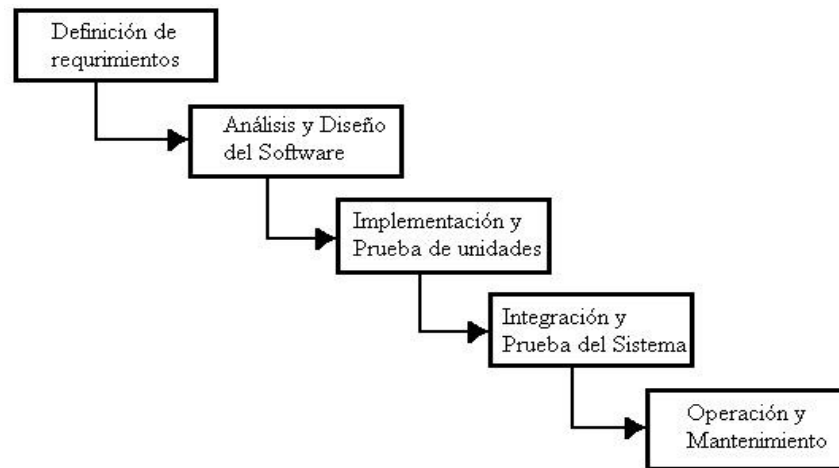


Figura 7. Modelo Cascada

La naturaleza del modelo no permite volver atrás y deshacer o volver a hacer acciones. Este modelo es factible cuando el desarrollador ha tenido experiencia con ese tipo de proyectos.

Modelo V. Basado en evaluaciones en cada fase del proyecto.

Para verificar y validar el producto.

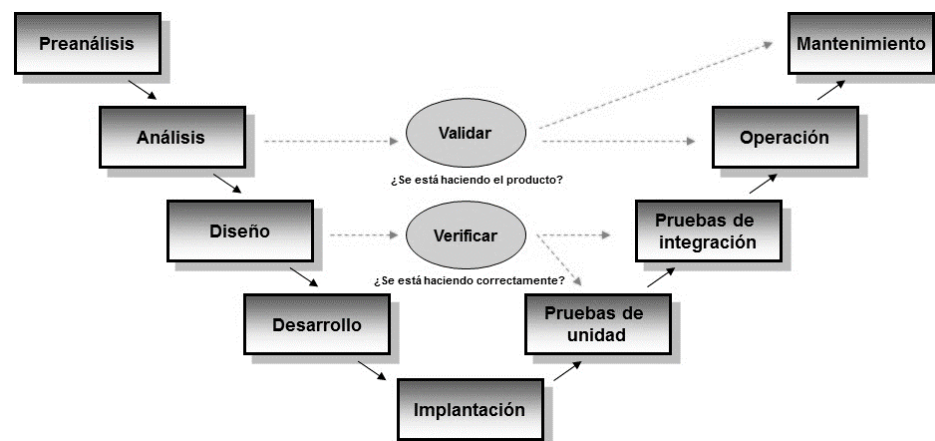


Figura 8. Modelo V

- **Ciclo de vida Prototipado.** Se basa en la realización de prototipos de forma continua. A fin de tener una mejor comprensión de los requerimientos que plantea el cliente. Si este no tiene idea de lo que desea. A continuación, en

la Figura 9 se muestra un diagrama de creación de prototipos.



Figura 9. Modelo de Prototipos

- **Ciclo de vida en Espiral.** El ciclo del software está definido en forma de un espiral en el cual los bucles representan el conjunto de actividades. Como se muestra en la Figura 10. Cada bucle está dividido en cuatro sectores:

Determinar objetivos. Se identifican los objetivos y se elaboran un plan de acción.

Análisis de riesgo. Consiste en identificar y analizar riesgos a fin de mitigar o reducirlos.

Desarrollar y probar. Después de haber analizado los riesgos se determina un modelo de desarrollo.

Costo

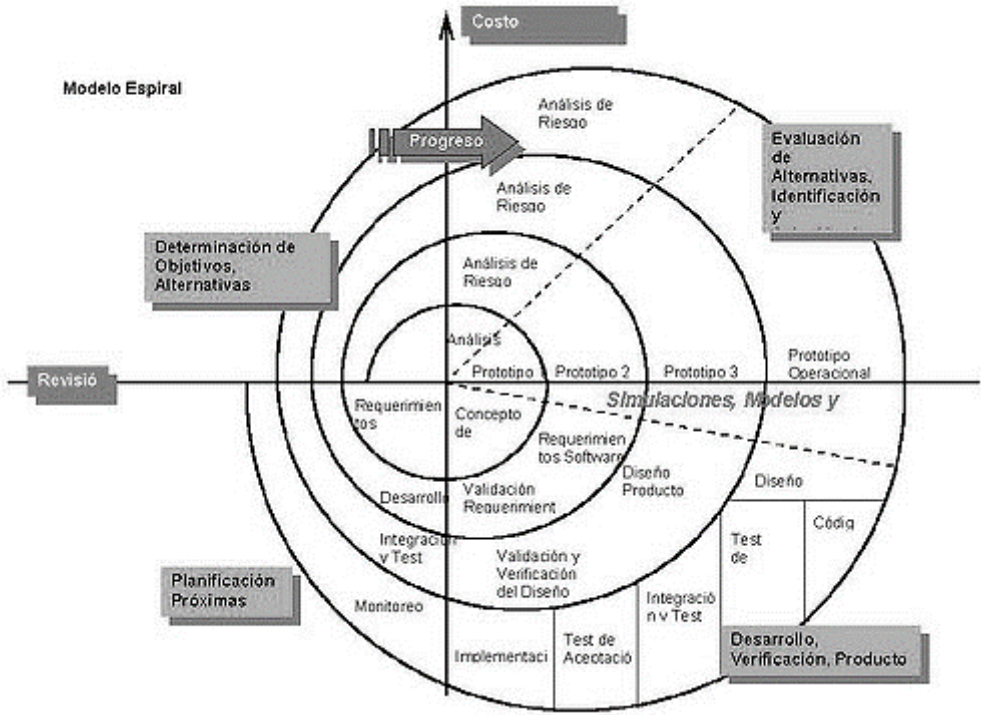


Figura 10. Ciclo de vida espira

- **Rational Unified Process (RUP)** Proceso de desarrollo de software moderno, creado por Rational Software. Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Está basado en dos dimensiones:
 - Eje horizontal representa el tiempo y muestra los aspectos del ciclo de vida del proceso durante toda su ejecución.

- Eje vertical representa las disciplinas, las cuales agrupan actividades de acuerdo a una manera lógica y naturaleza.
- A continuación, en la Figura 11 se muestra su representación.

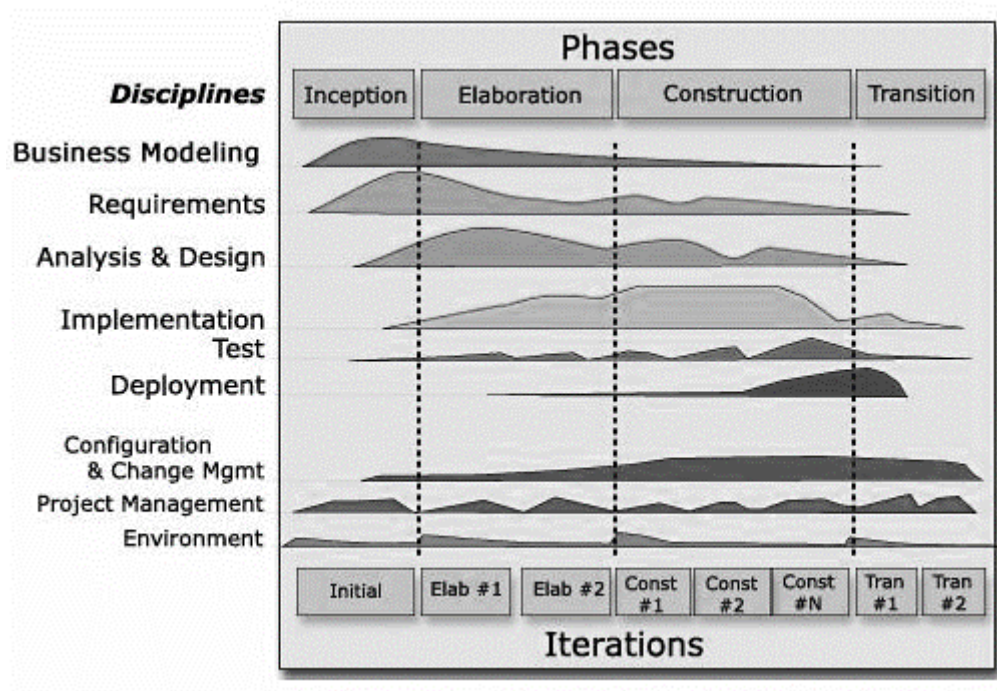


Figura 11. Mitología RUP

2.3.2. Metodologías ágiles

2.3.2.1. SCRUM

Es una metodología ágil muy utilizado en la actualidad para el desarrollo de software. Es un marco de trabajo adaptable flexible, iterativo, rápido y eficaz a fin de brindar un valor considerable a lo largo del proyecto [34].

- **Principios scrum**

Son directivas con respecto a la aplicación del framework scrum. Estos deben ser implementados de forma obligatoria en todos los proyectos. A fin de garantizar el éxito del marco de trabajo. Los seis principios se muestran en la figura 12.

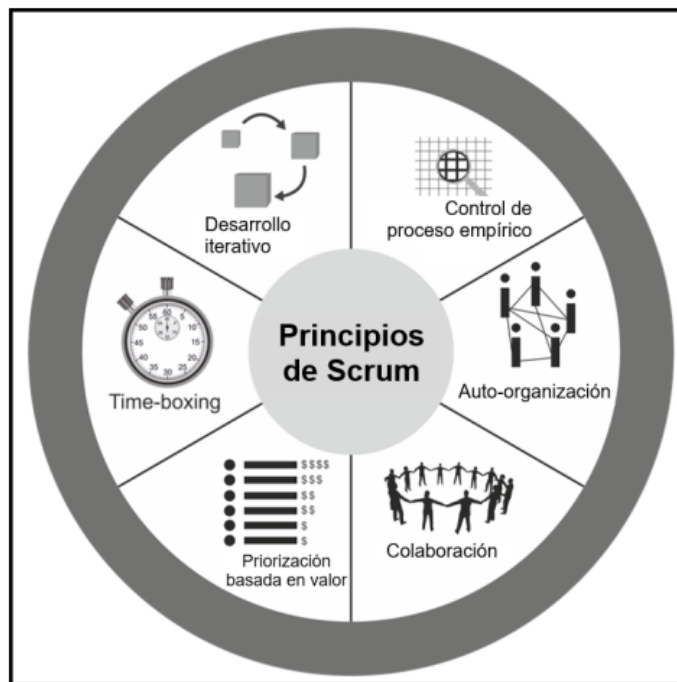


Figura 12. Principios Scrum

- **Aspectos scrum**

Los aspectos de Scrum deben aplicarse y gestionarse durante todo el proyecto.

- **Organización.** Comprender los roles y responsabilidades definidos en un proyecto Scrum.

- **Roles centrales.** Son de suma importancia para crear el producto o servicio. Las personas con este rol están muy comprometidas con el proyecto y responsables del éxito.
- **Product owner.** Maximizar el valor del trabajo y del producto por el equipo de desarrollo. Representa la voz del cliente.
- **Scrum master.** Tiene como función principal asegurar la comprensión y seguimiento del Scrum. Ayuda a alcanzar el máximo nivel en productividad del equipo.
- **Equipo scrum.** Comprender los requerimientos del product Owner y realizar los entregables del proyecto.
- **Roles no centrales.** Son roles no obligatorios para el proyecto Scrum. Los cuales son: Stakeholders, el Scrum Guidance Body y los vendedores.
- A continuación, en la figura se ilustra la estructura de la organización Scrum.

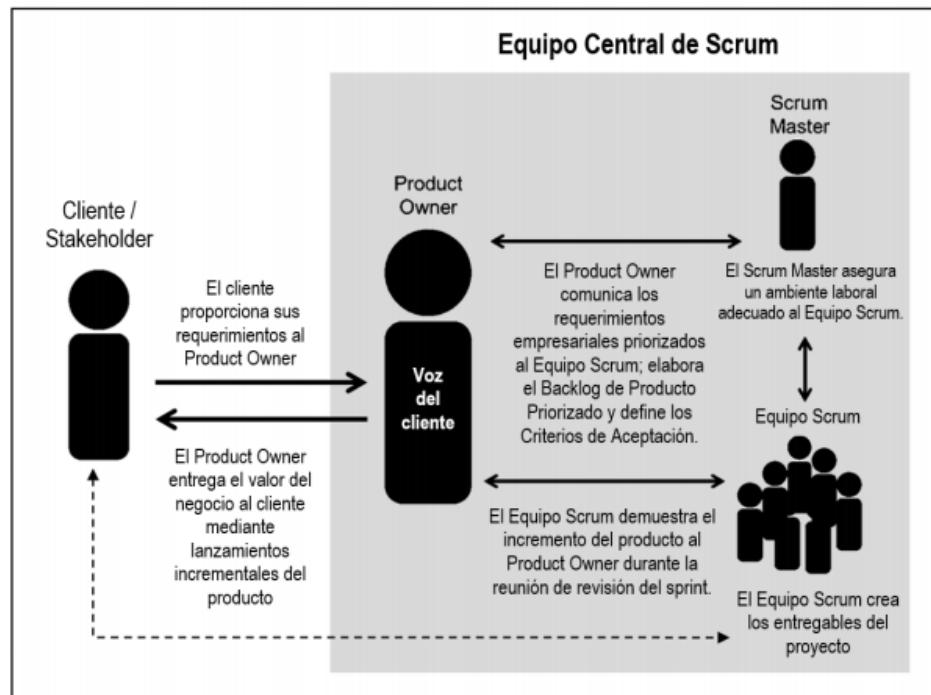


Figura 13. Organización Scrum

- **Procesos scrum.** Aquí se encuentran las actividades específicas y el flujo del proyecto Scrum. Está conformado por 19 procesos, los cuales están agrupados en 5 fases como se muestra en la figura 14.

Capítulo	Fase	Procesos fundamentales de Scrum
8	Inicio	1. Crear la visión del proyecto 2. Identificar al Scrum Master y Stakeholder(s) 3. Formar Equipos Scrum 4. Desarrollar épica(s) 5. Crear el Backlog Priorizado del Producto 6. Realizar la planificación de lanzamiento
9	Planificación y estimación	7. Crear historias de usuario 8. Estimar historias de usuario 9. Comprometer historias de usuario 10. Identificar tareas 11. Estimar tareas 12. Crear el Sprint Backlog
10	Implementación	13. Crear entregables 14. Realizar Daily Standup 15. Refinar el Backlog Priorizado del Producto
11	Revisión y retrospectiva	16. Demostrar y validar el sprint 17. Retrospectiva del sprint
12	Lanzamiento	18. Enviar entregables 19. Retrospectiva del proyecto

Figura 14. Procesos Scrum

○ **Time-boxing**

Consiste en la fijación de tiempo para cada proceso y actividad en un proyecto Scrum. Ventajas del Time boxing son las siguientes:

- ✓ Desarrollo de proceso de forma eficiente.
- ✓ Reducción de gastos generales.
- ✓ Velocidad por los integrantes de los equipos en el desarrollo de sus requerimientos.

- **Sprint.** Es una iteración de unas seis semanas de duración, durante el proceso el Scrum Master guía y protege al equipo. Durante este tiempo el equipo se esfuerza para convertir los requerimientos en funcionalidades de producto [34]. Como se muestra en la figura 15.

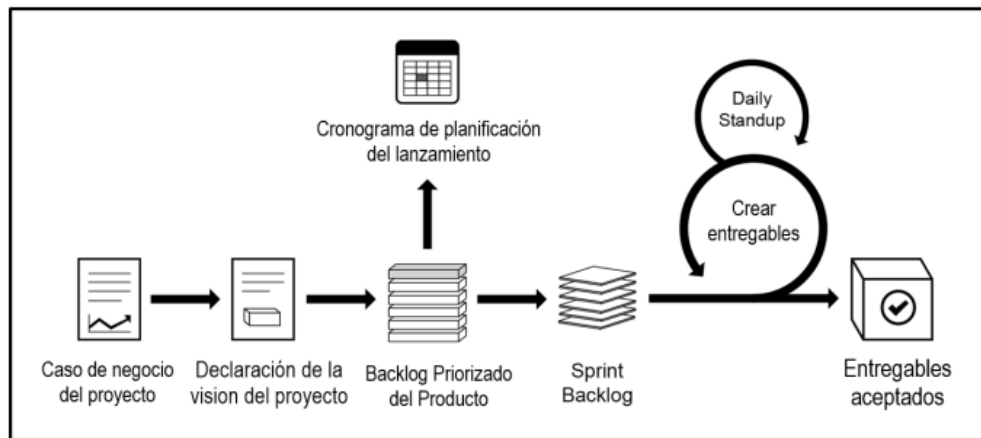


Figura 15. Flujo Scrum para un Sprint

- **Daily standup.** Es la reunión diaria de 15 minutos. Donde se informa los avances del proyecto, en base a un cuestionario de preguntas.
- **Reunión de planificación del sprint.** Esta reunión se lleva antes del Sprint y se divide en dos partes:
 - **Definición del objetivo.** El Product Owner define las historias de usuario en función de esto se define las metas del Sprint.
 - **Identificar y estimar las tareas.** Se identifica y estima como completar los elementos seleccionados en el Backlog.
 - **Reunión de retrospectiva del sprint.** En esta reunión los equipos revisan y reflexionan sobre el Sprint anterior. Lo que salió bien y lo que no salió bien; con el objetivo de aprender y mejorar.

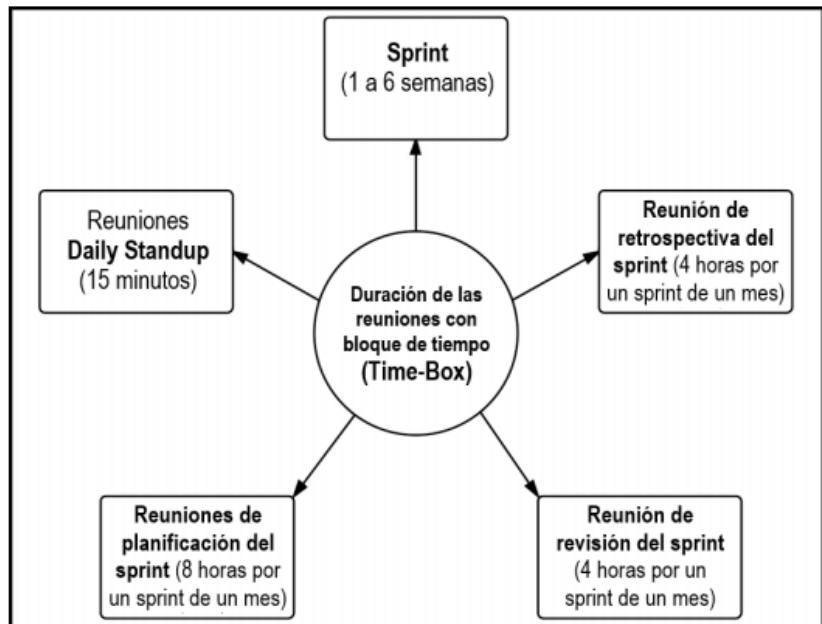


Figura 16. Duración de las reuniones de Scrum

- **Scrum y gestión tradicional de proyectos de software**

Scrum tiene como objetivo principal hacer más en menos tiempo. Para lograr esto Scrum considera el desarrollo iterativo de entregables.

“La gestión tradicional de proyectos está enfocada en el proceso y no en las personas. El enfoque Spring está basado en la entrega de productos que satisfagan los requisitos de los clientes en pequeños incrementos interactivos que sean entregable” [34]. Como se muestra en la figura 16.

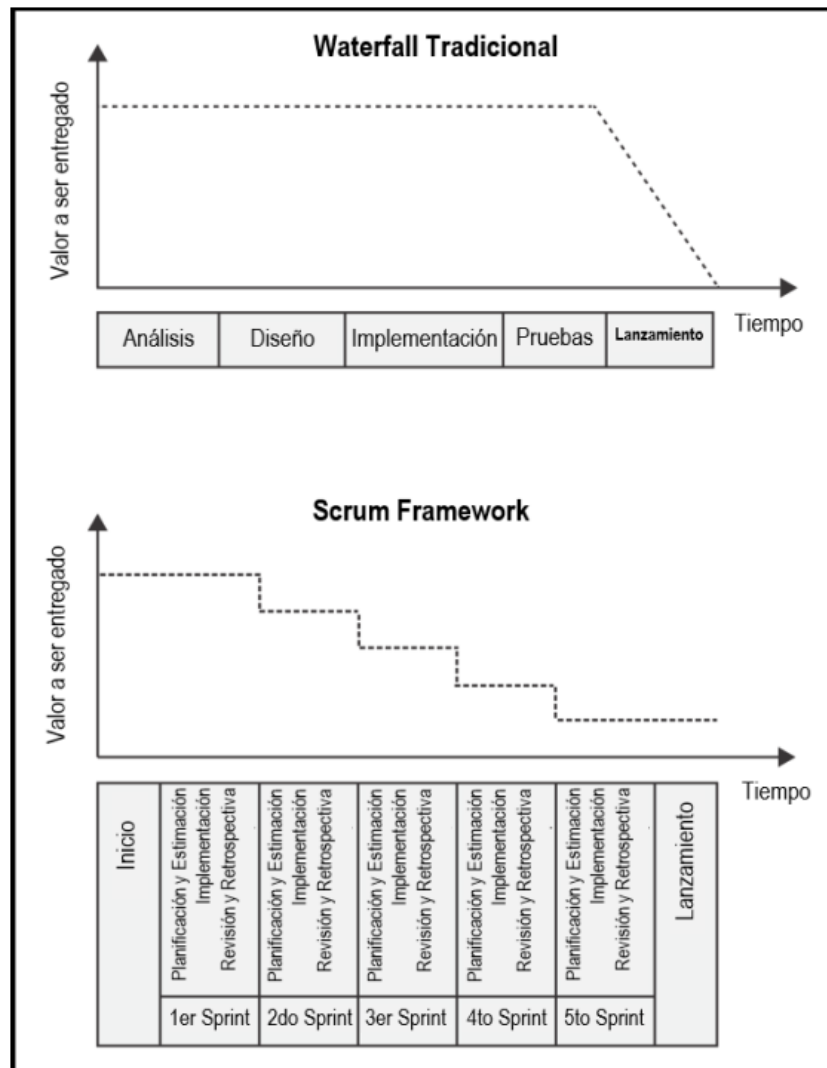


Figura 17. Scrum vs Cascada Tradicional

- **Artefactos Scrum**

Los artefactos son los siguientes:

- **Product Backlog**, es la lista de objetivos para la construcción de un producto de software y evoluciona a lo largo del desarrollo. Se construyen en base a historias de usuario [35].

- **Sprint Backlog**, Tareas que el equipo elabora en la reunión de planificación de la iteración, con el objetivo de completar los objetivos [36].
- **Increment**, Son los entregables de desarrolla el equipo Scrum [37].
- **Grafico Burn-Down**, representación gráfica del trabajo restante de lo avanzado en el proyecto, a nivel del Sprint y el proyecto [38].

2.3.3. CMMI

Fue desarrollado por el SEI (Software Engineering Institute). Este instituto desarrolló un producto que contiene el modelo de mejora, los métodos asociados al modelo y los métodos de evaluación [39].

Inicialmente fue un solo modelo, pero con el transcurrir del tiempo y por el éxito que venía teniendo por parte de las empresas. Se dividió en tres modelos diferentes:

- CMMI-DEV centrado en el desarrollo de software.
- CMMI-ACQ centrado en la adquisición de sistemas y servicios de software.
- CMMI-SVC centrado en la mejora de prestación de servicios de software.

Áreas de procesos CMMI

Son un conjunto de prácticas que se realizan de manera colectiva con el fin de alcanzar determinadas metas.

Estas áreas incluyen:

- Objetivo
- Prácticas
- Material informativo

En la Figura 6 se muestra la estructura como está organizado el modelo por cada área de proceso. Las áreas de proceso comienzan con una descripción del proceso.

El modelo CMMI se encuentra conformado por dos tipos de objetivos los genéricos y los específicos estos objetivos están asociados a un conjunto de sus prácticas a fin de ayudar a obtener una comprensión más profunda.

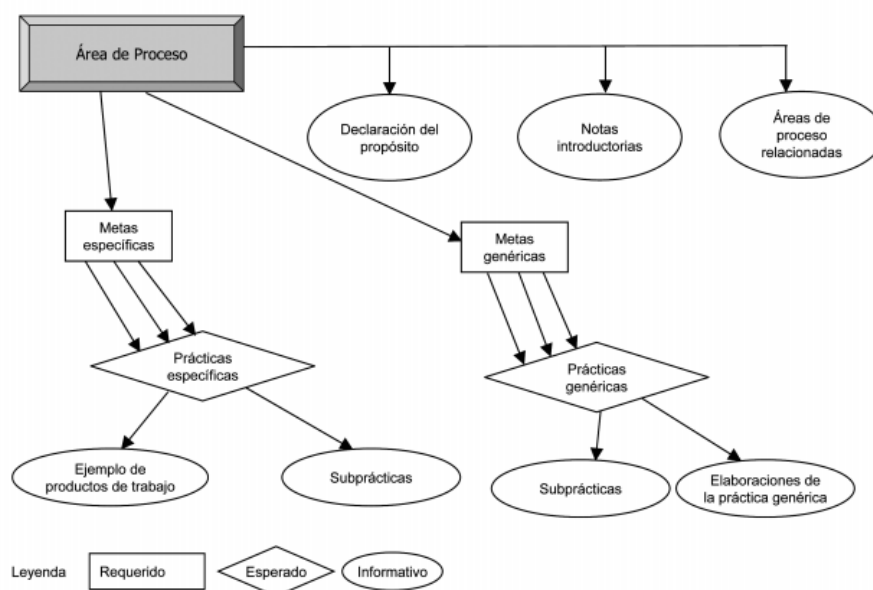


Figura 18. Esquema de área de proceso CMMI

Las áreas inician de la siguiente forma:

- **Propósito.** Establece el objetivo o propósito del área.
- **Notas introductorias.** expresan la idea tras el propósito expuesto.

- **Área del proceso relacionados.** Muestra la relación de un PA con otros.
- **Objetivo específico y resumen de prácticas.** Tanto el objetivo y la práctica tienen un título y un enunciado asociado.

Áreas de Procesos en los modelos CMMI

En los tres tipos de modelos CMMI existen un total de 16 áreas de procesos centrales. Como se muestra en la figura 16.

Abbreviation	Name	Area	Maturity Level
CAR	Causal Analysis and Resolution	Support	5
CM	Configuration Management	Support	2
DAR	Decision Analysis and Resolution	Support	3
IPM	Integrated Project Management	Project Management	3
MA	Measurement and Analysis	Support	2
OPD	Organizational Process Definition	Process Management	3
OPF	Organizational Process Focus	Process Management	3
OPM	Organizational Performance Management	Process Management	5
OPP	Organizational Process Performance	Process Management	4
OT	Organizational Training	Process Management	3
PMC	Project Monitoring and Control	Project Management	2
PP	Project Planning	Project Management	2
PPQA	Process and Product Quality Assurance	Support	2
QPM	Quantitative Project Management	Project Management	4
REQM	Requirements Management	Project Management	2
RSKM	Risk Management	Project Management	3

Figura 19. Procesos Centrales CMMI

Descripción de áreas de proceso

A continuación, se detalla cada área de proceso perteneciente al modelo CMMI-DEV Nivel 2.

a. Gestión de requisitos (REQM)

Propósito. Es identificar, asignar y hacer seguimiento de los requisitos a fin de satisfacer los objetivos del proyecto [39].

Esta Área de proceso tiene un solo objetivo y cuenta con cinco áreas específicas asociadas a este objetivo, ver Tabla 1.

Tabla 1. Área de Proceso Gestión de Requerimiento

Gestionar Requisitos	
	Los requisitos son administrados y las inconsistencias son identificadas con el plan de proyecto y los productos de trabajo identificados
SP 1.1	Comprender los requisitos implica entender su significado
SP 1.2	Obtener el compromiso de todos los participantes del proyecto
SP 1.3	Gestionar los cambios de requisitos a medida que evolucionan en el proyecto
SP 1.4	Mantener una trazabilidad entre los requisitos bidireccional con los productos de trabajo
SP 1.5	Alinear el trabajo del proyecto y los requisitos

b. Planificación del proyecto (PP)

Propósito: Tiene como propósito establecer y mantener planes que definan las actividades del proyecto.

Esta área incluye el desarrollo del plan de proyecto, identificar e involucrar a todos los interesados del proyecto, el compromiso para llevar dicho plan y tener actualizado cuando existan cambios.

Esta área establece tres objetivos específicos para el PP, ver Tabla 2

Tabla 2. Área de Proceso: Planificación del Proyecto (PP)

SG1	Establecer estimadores	
	Se establecen una serie de estimadores de parámetros de proyecto que se mantendrán a lo largo del ciclo del proyecto	
	SP 1.1	Realizar la Estimación del alcance del proyecto
	SP 1.2	Establecer una estimación del trabajo y las actividades a realizar
	SP 1.3	Definir las etapas del ciclo de vida del proyecto
	SP 1.4	Definir el esfuerzo y costo del proyecto
	Crear el plan de proyecto	
	Establecer un plan del proyecto como base para la administración del proyecto	
	SP 2.1	Establecer el calendario y los costes del proyecto
	SP 2.2	Se debe Identificar los riesgos del proyecto
	SP 2.3	Planificar la administración de datos
	SP 2.4	Planificar los recursos del proyecto
	SP 2.5	Planificar las habilidades y conocimientos necesarios
	SP 2.6	Planificar la participación de los involucrados
	SP 2.7	Establecer el plan de proyecto
	Obtener compromisos con el plan de proyecto	
	Los compromisos con el proyecto son establecidos y mantenidos durante el ciclo de vida del proyecto	
	SP 1.1	Revisar los planes que puedan afectar al proyecto
	SP 1.2	Poner de acuerdo el trabajo y los recursos compartidos
	SP 1.3	Obtener compromiso de acuerdo para el plan de proyecto

En el SP 1.1 se establece el EDT o WBS con el objetivo de estimar el alcance del proyecto. El SP 1.2 permite determinar cuánto trabajo debe ser empleado en cada paquete del WBS. El SP 1.3 se definen las fases del

proyecto en función del número de requisitos. En el SP 2.1 se define el presupuesto y plazos en función del WBS y lo establecido en el SG 1. En el SP 2.4 se definen los recursos del proyecto. En el SP 2.6 son identificados los involucrados y su participación en el proyecto. El SG 3 busca el compromiso de todos los involucrados y la viabilidad del proyecto.

c. Monitorización y Control del Proyecto (PMC)

Propósito

Proporcionar una comprensión del progreso del proyecto para que se puedan tomar las acciones correctivas apropiadas, cuando el proyecto sale del objetivo.

Esta área de proceso establece dos objetivos específicos para realizar la monitorización y control. A estos objetivos están asociados prácticas específicas tal como se aprecia en la Tabla 3.

Tabla 3. Área de proceso: Monitorización y Control del Proyecto

SG1	Monitorizar el proyecto contra su plan	
	Se monitoriza y compara el progreso y el desempeño del proyecto real y se compara contra el plan inicial	
	SP 1.1	Monitorizar los parámetros del proyecto o trabajo
	SP 1.2	Monitorizar los compromisos
	SP 1.3	Monitorizar riesgos
	SP 1.4	Monitorizar gestión datos
	SP 1.5	Monitorizar la participación de los involucrados
	SP 1.6	Llevar a cabo revisiones del progreso
	SP 1.7	Llevar a cabo revisiones de hitos
	Gestionar acciones correctivas	

	Las acciones correctivas son gestionadas hasta ser completadas cuando el proyecto o sus resultados se desvían significativamente del plan trazado	
	SP 2.1	Analizar Problemas
	SP 2.2	Llevar medidas correctivas
	SP 2.3	Gestionar las medidas correctivas

El objetivo específico SG 1 hace referencia a la monitorización de los elementos definidos en el plan del proyecto. En el SP 1.1 se monitoriza el trabajo completado, el esfuerzo gastado, el presupuesto consumido, los plazos completados y los recursos empleados.

EL SG 2 establece las practicas necesarias para corregir las desviaciones del proyecto. Se analizan los problemas y se definen las acciones a tomar en la corrección de dichos problemas.

d. Gestión de configuración (CM)

Propósito

Desarrollar y mantener la capacidad de medición utilizada para dar soporte a las necesidades de información de la gerencia [39].

Esta área de proceso cuenta con tres objetivos específicos, tal como se muestra en la Figura 4.

El primer objetivo específico SG 1 establece la línea base y el sistema que será utilizado por los demás objetivos específicos SG2, SG 3. SG 2 asegura que los cambios a los productos de trabajo están controlados y el SG 3 que las entradas sean guardadas y su integridad conservada.

Tabla 4. Área de Proceso: Gestión de Configuración

SG1	Establecer línea base	
	Se establecen las líneas base de los productos de trabajo.	
	SP 1.1	Se identifican los elementos de la configuración
	SP 1.2	Se establece y mantiene un sistema de gestión de configuración
	SP 1.3	Se crea y se libera la línea base
	Control de cambios de configuración	
	Se monitorea y controla los productos de trabajo a través de la gestión de la configuración.	
	SP 2.1	Se realizan peticiones de cambio a los elementos de configuración.
	SP 2.2	Se controlan los cambios a los elementos de la configuración.
	Mitigar los riesgos	
	Se establece y mantiene la integridad de la línea base.	
	SP 3.1	Se establecen y mantienen los registros que especifican los elementos de configuración
	SP 3.2	Se auditan los elementos de la configuración a fin de mantener la integridad de las líneas base.

e. Aseguramiento de la calidad del proceso y del Producto (PPQA)

Propósito

Proporcionar al personal y a la gerencia visión objetiva de los procesos y de los productos de los trabajos asociados [39].

Esta área de proceso cuenta con dos objetivos específicos. SG 1 establece que los procesos y los resultados deben ser medibles. SG 2 se enfoca en los temas de no conformidad detectados durante la validación y verificación del proceso. La

estructura de dicho proceso es tal como se muestra en la Tabla 5.

Tabla 5. Área de Proceso: Aseguramiento de la Calidad

SG1	Revisión de cumplimiento	
	La adherencia de los procesos realizados, y de los productos de trabajo asociados a las descripciones de proceso estándares y procedimientos aplicables es evaluada objetivamente	
	SP 1.1	Evaluación objetiva de procesos
	SP 1.2	Evaluación objetiva de productos de trabajo
	Comunicación y seguimiento	
	Los productos de trabajo bajo gestión de configuración son seguidos y controlados.	
	SP 2.1	Comunicar y resolver no conformidades
	SP 2.2	Establecer registros

f. Gestión de acuerdos con proveedores (SAM)

Propósito

Gestionar la adquisición de productos y servicios de proveedores [39].

Esta área de proceso va orientado a la selección establecimiento de acuerdos con los proveedores y también a revisar el cumplimiento del acuerdo.

Esta área de proceso define dos objetivos específicos tal como se muestra en la Tabla 6. El SG 1permite seleccionar a los proveedores y le SG 2. Los acuerdos con ellos, es establecido de igual a igual.

Tabla 6. Área de Proceso: Gestión de acuerdos con proveedores (SAM)

SG1	Establecer acuerdos con los proveedores	
	Se establecen y mantienen acuerdos con los proveedores	
	SP 1.1	Establecer el tipo de adquisición
	SP 1.2	Seleccionar los proveedores
	SP 1.3	Establecer acuerdos con los proveedores
	Satisfacer los acuerdos con los proveedores	
	Los acuerdos con proveedores deben ser satisfechos por ambas partes, el proveedor y quien ejecuta el proyecto.	
	SP 2.1	Ejecutar el acuerdo con proveedores
	SP 2.2	Aceptar el producto adquirido
	SP 2.3	Asegurar la transacción de productos

En el SP 1. Estable la necesidad ya sea de producto o servicio a adquirir. El SP 1.2. se evalúan el abanico de proveedores con respecto a esa necesidad. SP 1.3 después de haber seleccionado el proveedor que va proveer el requerimiento y luego se establecen alianzas y acuerdos comerciales. Los SP relacionados al SG 2, está relacionado con la ejecución y evaluación antes de recibir el requerimiento.

g. Medición y análisis (MA)

Propósito

“Desarrollar y mantener la capacidad de medición utilizada para dar soporte a las necesidades de información de la gerencia” [39].

La medición es un factor importante, ya que permite la toma de decisiones en función de la foto actual. El uso del instrumento adecuado es un factor determinante para cumplir a cabalidad

este proceso y el estado y rumbo del proyecto. Establecer medidas de los procesos permite el registro de datos históricos que posterior serán utilizados para establecer el nivel de madurez de la organización.

Esta área de proceso cuenta con dos objetivos específicos. El SG 1. Que permite establecer los objetivos a medir. En el SG 2. Es la aplicación de dichas medidas y su evaluación para la toma de decisiones. Como se muestra en la Tabla 7.

Tabla 7. Área de proceso: medición y análisis

SG1	Alinear las actividades de medición y análisis	
	Los objetivos de medición y las actividades son alineadas con las necesidades y objetivos de información identificados	
	SP 1.1	Establecer objetivos de medida
	SP 1.2	Especificar medidas
	SP 1.3	Especificar procedimientos de recopilación y almacenamiento de datos
	SP 1.4	Especificar procedimientos de análisis
	Promover los resultados de las medidas	
	Se proporcionan los resultados de las medidas llevadas a cabo de acuerdo a las necesidades y objetivos identificados previamente	
	SP 2.1	Obtener los datos de la medición
	SP 2.2	Analizar los datos de la medición
	SP 2.3	Almacenar los datos y resultados
	SP 2.4	Comunicar los resultados

h. Monitorización y control del proyecto (PMC)

Propósito

Dotar de un conocimiento acerca del progreso del proyecto de tal forma que se puedan tomar acciones correctivas en caso de

que el proyecto se desvíe significativamente del plan inicial. Se establecen dos objetivos específicos para llevar a cabo la monitorización y el control. Estos objetivos se detallan en la Tabla 8.

SG 1. Las cinco primeras prácticas hacen referencia a la monitorización del proyecto con respecto al plan del proyecto. El SP 1.6 y 1.7 se enfocan en las revisiones del proyecto y los hitos cumplidos. Si se detectan desviaciones del proyecto el SG 2. Establece las prácticas necesarias para corregir dichas desviaciones. Es importante que las acciones correctivas tengan fechas límites para su finalización y que la monitorización se haga hasta el final.

Tabla 8. Monitorización y Control del Proyecto (PMC)

SG1	Monitorizar el proyecto contra su plan	
	Se monitoriza y compara el progreso y el desempeño del proyecto real y se compara contra el plan inicial	
	SP 1.1	Monitorizar los parámetros del proyecto o trabajo
	SP 1.2	Monitorizar los compromisos
	SP 1.3	Monitorizar riesgos
	SP 1.4	Monitorizar gestión datos
	SP 1.5	Monitorizar la participación de los involucrados
	SP 1.6	Llevar a cabo revisiones del progreso
	SP 1.7	Llevar a cabo revisiones de hitos
	Gestionar Acciones correctivas	
	Las acciones correctivas son gestionadas hasta ser completadas cuando el proyecto o sus resultados se desvían significativamente del plan trazado	
	SP 2.1	Analizar problemas
	SP 2.2	Llevar a cabo medidas correctivas
	SP 2.3	Gestionar las medidas correctivas

2.4 Ciclo de vida del proyecto de software

La aplicación del modelo integrado Scrum y CMMI permite mayor eficacia en el ciclo de vida planificado de los proyectos de ingeniería de la consultora ASSAC.

2.5 Marco conceptual

Scrum

Es el esfuerzo de trabajo colaborativo para crear un nuevo producto o servicio u otro resultado. Es adaptable, interactivo, ágil, flexible a los cambios, eficaz en el alcance de los objetivos. A fin de maximizar el retorno de inversión [40].

CMMI

Está conformado por un conjunto de modelos basadas en las mejores prácticas en la gestión de procesos, que ayudan a las organizaciones a proporcionar una orientación referente a fin de alcanzar sus objetivos de negocio [39].

Modelo integrado Scrum y CMMI

Es la integración del modelo ágil de SCRUM y el modelo de gestión de proyectos de CMMI. La integración está en base a las áreas de procesos que contempla el CMMI-DEV considerando las áreas en las que se puede integrar o aplicar la metodología ágil Scrum.

Ciclo de vida de los proyectos de ingeniería de software

El ciclo de vida de los proyectos de ingeniería de software es el tiempo en que dura el desarrollo de un requerimiento de negocio hasta que esté listo para que el usuario pueda utilizarlo o en algunas ocasiones hasta que esté

implementado. Se considera las fases de análisis funcional, diseño del requerimiento, programación y testeo.

Capítulo III

3 Metodología de la investigación

3.4 Tipo y diseño de la investigación

3.4.1 Tipo de investigación

Esta investigación es aplicada, descriptiva y tecnológica [41], porque se han utilizado las teorías de SCRUM y CMMI para optimizar el ciclo de vida del proyecto. Es descriptiva porque se ha realizado una descripción de la información obtenida del objeto de estudio para comparar los resultados con proyectos previos que la misma consultora ha realizado con otras metodologías. Es tecnológica porque ha sido de carácter ejecutivo, es decir; se ha implementado el Modelo Integrado de Scrum y CMMI para la gestión de los Proyectos de Ingeniería de Software de ASSAC.

3.4.2 Diseño de la Investigación

El diseño de investigación de acuerdo a Hernández Sampieri es cuasi experimental [42], puesto que se hacen experimentos y/o testeo para validar la adecuada implementación del framework Scrum para optimizar la Influencia del Modelo Integrado de Scrum y CMMI en el Ciclo de Vida del Proyecto de Ingeniería de Software de la ASSAC y luego se realiza una comparación con otros proyectos donde no hubo intervención.

A continuación, se presenta el esquema de investigación:

GE: X ----- O₁

Gc: ----- O₂

Donde:

GE: Grupo Experimental conformado por los requerimientos de los proyectos de ingeniería de software que aplicaron el modelo.

GC: Grupo de Control conformado por los requerimientos de los proyectos de ingeniería de software que no aplicaron el modelo.

O₁: Diagnóstico Inicial del Ciclo de Vida de los Proyectos del Grupo Experimental.

X: Modelo Integrado de Scrum y CMMI.

O₂: Diagnóstico Final del Ciclo de Vida de los Proyectos del Grupo de Control.

3.4.3 Esquema de investigación

La presente investigación tuvo las siguientes fases para su ejecución que son:

1. Análisis y diseño del modelo SCRUM y CMMI: en esta etapa se han analizado los componentes o elementos de cada modelo para encontrar características comunes y puntos de integración para aplicar de manera conjunta estas dos tecnologías.
2. Análisis de la planificación de los requerimientos: en esta fase se ha obtenido la información del tiempo de planificación y el tiempo de desarrollo real de los proyectos de ingeniería de software.

3. Aplicación del modelo SCRUM y CMMI: En esta etapa se aplicó el modelo integrado considerando la documentación necesaria y las buenas prácticas de SCRUM.
4. Análisis de la planificación del grupo de control: en esta fase se recopiló la información de la planificación de los proyectos de ingeniería de software de los proyectos que han sido desarrollados de manera tradicional.
5. Comparación del grupo de control con los resultados del grupo experimental. Finalmente, en esta fase se ha comparado los resultados con el grupo experimental y el grupo de control para ver la eficacia del modelo integrado.

3.5 Unidad de Análisis

La unidad de análisis son los requerimientos de los proyectos de desarrollo de software.

3.6 Población de Estudio

Está conformado por los 99 requerimientos de los proyectos de ingeniería de software del centro de consultoría ASSAC que se han desarrollado en el año 2016 y 2017 por los equipos Scrum.

3.7 Técnica y Tamaño de muestra

Por la naturaleza de la investigación la muestra no probabilística e intencional, es decir, es intencional porque se han seleccionado los

proyectos que reunían las condiciones para aplicar el modelo integrado. Y el tamaño de la muestra estuvo conformado los 99 requerimientos de los proyectos seleccionados.

3.8 Recolección de datos

La técnica de recopilación de datos fue documental y análisis de los sistemas de gestión de proyectos. Fue documental porque se han revisado los documentos de trabajo que se ha utilizado en los proyectos, y fue de análisis de sistemas porque se ha extraído la información del TFS en donde estabas configurados los requerimientos de los proyectos en estudio.

3.9 Análisis e interpretación de investigación

Se ha preparado una plantilla en Excel para recopilar los diferentes tiempos que tuvieron cada requerimiento y luego se ha exportado al paquete estadístico SPSS 23. Para el procesamiento de la información se ha utilizado la estadística descriptiva, y para la contrastación de la hipótesis de investigación se ha utilizado el modelo estadístico de la t-de student para muestras no relacionadas.

3.10 Delimitación del proyecto de investigación

En el ciclo de vida del proyecto de software se tiene diferentes tipos de planificación que se tiene que realizar para sacar un producto de software. Dentro de los procesos de planificación y desde la perspectiva de la gestión de proyectos se tiene cinco procesos que son: inicio, planificación, ejecución, control y cierre de proyecto. Dentro de cada proceso se

plantean un conjunto de actividades a realizar por cada proyecto y se estiman los tiempos por cada actividad. En la presente investigación solo se analizaron los tiempos de ejecución que dentro de ingeniería de software es la programación de los requerimientos de software. No se ha considerado el análisis o la identificación de los requerimientos ni el tiempo de implementación del software.

CAPITULO IV

4 CONSTRUCCION DEL MODELO INTEGRADO

4.4 Diseño del Modelo Integrado SCRUM v CMMI

En inicio se realizó una revisión documental sobre la aplicabilidad de estos dos enfoques, partiendo de las metodologías ágiles de desarrollo de software [43], [44], [45], [46], [47]. Así como el modelo de buenas prácticas de calidad de software [39], a partir de la investigación encontrado se planteó el modelo integrado de Scrum y CMMI. Tal como se muestra en la Figura 20.

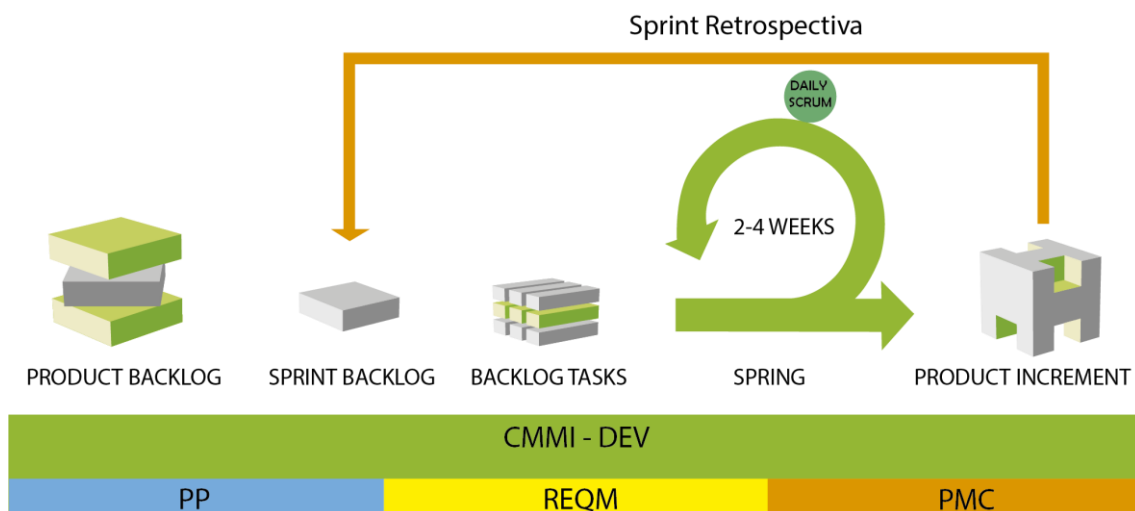


Figura 20. Modelo Integrado Scrum y CMMI

4.5 Áreas de Proceso Scrum y CMMI-DEV Nivel 2

No existe una metodología que aborde todos los retos y casuísticas que puedan surgir a lo largo de un proyecto de software [48]. Las guías implementadas en una organización no pueden ser lo suficiente flexible para adaptarse a diversas situaciones o un proceso en particular del

proyecto. Esta es una de las razones por lo cual aparecieron las metodologías ágiles. Para ayudar a las organizaciones a adaptarse a las situaciones de su entorno.

En esta sección se realiza la integración de Scrum y las áreas de procesos CMMI. De las áreas de procesos que contempla el CMMI-DEV no en todas las áreas se puede integrar o aplicar la metodología ágil Scrum. Para este estudio se realizó la integración en las áreas de proceso del nivel dos, tal como se muestra en la Figura 11.

CMMI-DEV						SCRUM
	2		REQM	PP	PMC	
		SAM	MA	PPQA	CM	
1						

Figura 21. CMMI-DEV Nivel 2 y sus Áreas de proceso

4.5.1 Planificación del Proyecto (PP) y Scrum.

A continuación, en la tabla 9 se puede apreciar la alineación del proceso (PP) y la metodología Scrum. Este proceso se inicia con la recepción del proyecto, como resultado se obtiene el plan del proyecto.

Tabla 9. Planificación del Proyecto y Scrum

Planificación del Proyecto (PP)			SCRUM	Artefacto
SG1	Establecer estimadores			
	SP 1.1	Realizar la Estimación del alcance de	Definición del Product Backlog y los Sprints	Documento de alcance del proyecto
	SP 1.2	Establecer una estimación del trabajo y las actividades a realizar	Estimación inicial en el pregame	
	SP 1.3	Definir las etapas del ciclo de vida del proyecto	Ciclo de vida Scrum	
	SP 1.4	Definir el esfuerzo y costo del proyecto	Pregame, inicio de cada Sprint, historias de usuario	
SG2	Crear el plan de proyecto			
	SP 2.1	Establecer el calendario y los costes del proyecto	Dayly Scrum Product backlog	Documento con las fases del proyecto y sus funcionalidades
	SP 2.2	Se debe Identificar los riesgos del proyecto	Sprint Planning Meeting, Sprint Review Meeting	Documento con los riesgos del proyecto y su seguimiento
	SP 2.3	Planificar la administración de datos		Documento con las definiciones para la planeación de los datos
	SP 2.4	Planificar los recursos del proyecto	Sprint Planning Meeting	Documento con los recursos del proyecto
	SP 2.5	Planificar las habilidades y conocimientos necesarios	Sprint Planning Meeting, Sprint Review Meeting	Documento con los conocimientos y habilidades de cada integrante del equipo.
	SP 2.6	Planificar la participación de los involucrados		Documento con el registro de participantes del proyecto
	SP 2.7	Establecer el plan de proyecto	Daily Scrum	
SG3	Obtener compromisos con el plan de proyecto			
	SP 3.1	Revisar los planes que puedan afectar al proyecto	Sprint Planning Meeting, Sprint Review Meeting, Daily Scrum	Documento con el registro de compromisos asociados al plan del proyecto
	SP 3.2	Poner de acuerdo el trabajo y los recursos compartidos		Documento con los recursos solicitados frente a los disponibles
	SP 3.3	Obtener compromiso de acuerdo para el plan de proyecto		Documento de inicio del proyecto

Con la realización del primer Release Planning. Se determinó y priorizo el Product Backlog tal como se muestra a continuación, en la Tabla 10 en el cual se especifica los requerimientos del proyecto a nivel macro. Los cuales son definidos por el equipo del proyecto a través de historias de usuario.

Tabla 10. Product Backlog del proyecto

#	Requerimiento	Prioridad
01	La administración del área de Talento Humano necesita implementar un módulo de configuración para controlar datos generales del software de Gestión del talento Humano.	1
02	El área de Talento Humano necesita implementar el módulo de contratación de personal no docente.	2
03	El área de Talento Humano necesita implementar el módulo de vacaciones.	3
04	El área de Talento Humano necesita implementar el módulo de Renuncias y abandono de Trabajo	4

Después de haber definido el Product Backlog tomando como base los módulos de sistema de GTH. Se procedió a analizar la complejidad de los módulos con el objetivo de determinar la prioridad, tomando dos criterios para dicha estimación.

- Las historias de usuario nos brindan el flujo más básico posible, se procedió a otorgar tres pesos preferenciales (1, 5, 10) según su implicancia.
- Para establecer la estimación, se hizo en base de la experiencia del equipo usando una escala de 1 al 10. Tal como se muestra en la tabla 11.

Tabla 11. Estimación de los Sprint

IT	Código	Requerimiento Funcional		
01	CFG	Módulo de Configuración	Importancia	Estimación

En base a la complejidad del proyecto utilizando juicio de expertos y el Planning Pocker se elabora el documento de estimación de costos. También se obtuvo el ciclo de vida y la estimación del proyecto.

4.5.2 Monitorización y Control del Proyecto (PMC) y Scrum

Tabla 12. Monitoreo y Control del Proyecto y Scrum

Monitorización y Control del Proyecto (PMC)			SCRUM	Artefacto
SG1	Monitorizar el proyecto contra su plan			
	SP 1.1	Monitorizar los parámetros del proyecto o trabajo	Daily Scrum	Registro de seguimiento del avance del proyecto
	SP 1.2	Monitorizar los compromisos		Registro de cumplimiento de los compromisos
	SP 1.3	Monitorizar riesgos		Registro de seguimiento a riegos
	SP 1.4	Monitorizar gestión datos		Registro del monitoreo sobre los datos del proyecto con respecto al plan.
	SP 1.5	Monitorizar la participación de los involucrados		Registro de la participación de los involucrados.
	SP 1.6	Llevar a cabo revisiones del progreso	Sprint Review Meeting, Daily Scrum	Registro de la participación de los involucrados.
SP 1.7	Llevar a cabo revisiones de hitos	Registro de los eventos importantes del proyecto		
SG2	Gestionar acciones correctivas			
	SP 2.1	Analizar Problemas	Sprint Review Meeting, Daily Scrum	Registro de los eventos importantes del proyecto
	SP 2.2	Llevar medidas correctivas		Registro de las acciones tomadas para solucionar los problemas
	SP 2.3	Gestionar las medidas correctivas		Registro de los resultados obtenidos a partir de las decisiones tomadas.

En la tabla 12 se puede apreciar la alineación del proceso PMC y el modelo Scrum. Se lleva a cabo el control del cronograma considerando el tiempo definido en el plan del proyecto. Cuando hay desviaciones se agregan horas de trabajo para cubrir en lo posible lo ofrecido. Hay reuniones que se desarrollan en el área de desarrollo del proyecto donde se exponen los proyectos en curso, posibles problemas o imprevistos del proyecto. Existe un procedimiento

establecido para el control de cambios realizados en el proyecto el cual involucra registro de cambios y versiones. También se realizan reuniones diarias para evaluar el estado del proyecto.

4.5.3 Gestión de Requisitos (REQM) y Scrum

Tabla 13. Gestión de Requisitos y Scrum

Gestion de Requisitos (REQM)			SCRUM	Artefacto
SG2	SP 1.1	Comprender los requisitos implica entender su significado	Historia de Usuario	Registro de historias de Usuario
	SP 1.2	Obtener el compromisos de todos los participantes del proyecto	Spring Planning, Product Backlog	Registro de compromisos para los requisitos
	SP 1.3	Gestionar los cambios de requisitos a medida que evolucionan en el proyecto	Sprint Review Meeting, Spring Planning.	Registro de cambios de los requisitos
	SP 1.4	Mantener una trazabilidad entre los requisitos bidireccional con los productos de trabajo	Historias de Usuario	Registro de historias de Usuario
	SP 1.5	Alinear el trabajo del proyecto y los requisitos	Sprint Planning.	Registro del trabajo realizado y los requisitos

En la tabla 13 se muestra la alineación del Proceso REQM. Se cuenta con un registro para la lista de requisitos que están definidos como historias de usuarios. A través del Sprint planning se comunica las especificaciones del proyecto instando al compromiso de los participantes. Las modificaciones se detallan en un registro de cambios de los requisitos.

CAPITULO V

5 RESULTADOS Y DISCUSIÓN

5.4 Descripción de las características de la población de estudio

La población de estudio está conformada por todos los requerimientos de los proyectos que se han considerado en la presente investigación que son: proyecto de sistema de devueltas de facturación, proyecto de medicina física y proyecto de sistema académico.

En la tabla 24 se presenta la lista de requerimientos del proyecto del sistema de devueltas de facturación. En la tabla se observa las iteraciones que tuvo el proyecto, la descripción del requerimiento, la planificación y el tiempo real en que se demoró en realizar cada requerimiento y finalmente la desviación en base a lo planificado. De acuerdo a los resultados se puede observar que en todo los requerimientos planificados existe una desviación excepto en el requerimiento 9 que cuya desviación es cero lo cual significa que el programador lo ha planificado en realizar en 9 horas dicho requerimiento y los ha terminado en 9 horas de acuerdo a lo planificado; a diferencia del requerimiento 31 que se ha planificado realizarlo en 3 horas y el tiempo que se ha demorado en desarrollar dicho requerimiento fue de 6 horas por lo tanto existe una desviación del 100% ya que se ha duplicado el tiempo de desarrollo.

Tabla 14. Planificación y ejecución del desarrollo de los requerimientos del proyecto de Sistemas de Devueltas de facturación

Iteracion	N° de Req	Lista de Requerimientos	Planificado	Real	%Diferencia
1	Req - 1	RS501- Resolver posicion de comentarios cuando esta publicado en el servidor	3	5	67%
1	Req - 2	RM02- Reportes personalizados	7	9	29%
1	Req - 3	RF01- Agregar comprobantes a la lista de comprobantes de la bandeja de plantilla de envio	10	8	20%
1	Req - 4	Eliminar comprobantes de la lista de comprobantes de la bandeja de planilla de envio	2	3	50%
1	Req - 5	Crear plantilla de forma manual y automatica	9	9	0%
1	Req - 6	Agregar comprobantes deslizando hacia una planilla	10	12	20%
1	Req - 7	Eliminar comprobantes de la planilla al deslizar la planilla hacia la lista de comprobantes	8	6	25%
1	Req - 8	Enviar planilla deslizando a la sub - Bandeja de recibidos	2	3	50%
1	Req - 9	Rechazar planillas	9	9	0%
1	Req - 10	Enviar planilla recibida deslizando a la mesa de trabajo	5	4	20%
1	Req - 11	Analizar expediente	8	5	38%
1	Req - 12	Mostrar ubicación del comprobante	15	12	20%
1	Req - 13	Configuracion de usuarios	10	11	10%
1	Req - 14	Configuracion de roles	8	7	13%
1	Req - 15	Acceso al sistema	12	13	8%
1	Req - 16	Editar comprobante	9	11	22%
1	Req - 17	Reportes	5	4	20%
2	Req - 18	Agregar comprobantes por serie, numero o codigo de barra a una planilla	10	9	10%
2	Req - 19	Agregar por codigo de barras comprobantes analizados a una planilla	13	10	23%
2	Req - 20	Enviar planillas por codigo de barras	6	7	17%
2	Req - 21	Recibir planillas por codigo de barras	8	7	13%
2	Req - 22	Enviar planilla a la mesa de trabajo por codigo de barras	7	11	57%
2	Req - 23	Ejecutar comprobante mediante codigo de barras	11	9	18%
2	Req - 24	Mostrar el tiempo de la plantilla en ejecucion	12	13	8%
2	Req - 25	Configuracion de rutas	9	7	22%
2	Req - 26	Configuracion de motivos	3	4	33%
2	Req - 27	RF025- Configuracion de areas	6	8	33%
2	Req - 28	RP01- Elaborar presentacion del SIDDV	5	7	40%
3	Req - 29	RM05- Modificar requisito desvincular comprobante de la planilla	3	5	67%
3	Req - 30	RT01- Pruebas de regresion a sistema SIDDV	12	14	17%
3	Req - 31	RP01- Configurar requisitos en el TFS	3	6	100%
3	Req - 32	RM04- Mostrar el destino del comprobante en analisis expediente	4	6	50%
3	Req - 33	RM03- Cambiar el color del componente	9	8	11%
3	Req - 34	RM01- Modificar agregar motivos	3	5	67%
3	Req - 35	RM02- Reportes personalizados	5	8	60%

Fuente: Elaboración propia del autor

En la tabla 25 se presenta la lista de requerimientos del proyecto de medicina física. En la tabla se observa las iteraciones que tuvo el proyecto, la descripción del requerimiento, la planificación y el tiempo real en que se demoró en realizar cada requerimiento y finalmente la desviación en base a lo planificado. De acuerdo a los resultados se puede observar que en todo los requerimientos planificados existe una desviación excepto en el requerimiento 28 y 29 que cuya desviación es cero lo cual significa que el programador ha desarrollado dicho

requerimiento en el tiempo que lo ha planificado; en las desviaciones de este proyecto se puede observar que la desviación más alta es de 67% que es el requerimiento 5 y el resto tiene una desviación de 40% o menos de aquí se puede deducir que el equipo en este segundo proyecto tuvo menos porcentaje de desviación en los diferentes requerimientos del cual se deduce que han sido más asertivos en la planificación.

Tabla 15. Planificación y ejecución del desarrollo de los requerimientos del proyecto de Medicina Física.

Iteracion	Nº de Req	Lista de Requerimientos	Planificado	Real	%Diferencia
1	Req - 1	El usuario del sistema necesita visualizar, registrar, editar y eliminar cabinas	5	6	20%
1	Req - 2	El usuario necesita administrar la Asignacion de accesos al rol	5	7	40%
1	Req - 3	El usuario necesita registrar, editar y eliminar Roles	8	11	38%
1	Req - 4	El usuario necesita registrar, editar y eliminar Accesos	8	9	13%
1	Req - 5	Listar Admisiones	3	5	67%
1	Req - 6	Eliminar Admision (Particular)	5	6	20%
1	Req - 7	Atender Admision	5	3	40%
1	Req - 8	Registrar Admision	5	3	40%
2	Req - 9	Registrar orden de admision	5	4	20%
2	Req - 10	Se registrara el cobro - Listar los cobros pendientes	8	7	13%
2	Req - 11	Se registrara el cobro - Registrar cobro	8	7	13%
2	Req - 12	Cuadrar caja por ingresos del dia	8	10	25%
2	Req - 13	Registrar Modalidad de Ingreso	8	11	38%
2	Req - 14	Registrar Contrato - Formulario de Registro de Contrato	16	18	13%
2	Req - 15	Registrar el Pre-contrato - Programacion de Carga	16	15	6%
2	Req - 16	Generar contrato - Matrícula	8	11	38%
2	Req - 17	El administrador necesita visualizar las atenciones	5	7	40%
2	Req - 18	Registrar Carga Académica	8	9	13%
2	Req - 19	El administrador del sistema necesita visualizar, agregar, editar y eliminar usuarios del sisten	8	9	13%
3	Req - 20	Visualizar contratos Pendientes de pago y pagados	13	10	23%
3	Req - 21	El administrador del sistema necesita visualizar, agregar, editar los clientes matriculados	8	7	13%
3	Req - 22	El coordinador imprime tres copias del contrato y una es entregada al cliente	8	10	25%
3	Req - 23	El administrador del sistema necesita visualizar, agregar, editar y eliminar los programas aca	13	15	15%
3	Req - 24	El administrador del sistema necesita visualizar, agregar, editar y eliminar tarifarios para reali	13	15	15%
3	Req - 25	El administrador, secretarías y/o usuarios del sistema necesitan visualizar, agregar, editar y	8	9	13%
3	Req - 26	Los usuarios necesitan acceder al sistema mediante una contraseña para realizar las opera	5	7	40%
3	Req - 27	El sistema mostrara el rol que cumple el usuario en el programa	5	7	40%
3	Req - 28	El coordiandor podra visualizar la lista de programas académicos que debe gestionar	8	8	0%
3	Req - 29	El administrador necesita visualizar los reportes: docentes y su carga, alumnos y su carga.	8	8	0%

Fuente: Elaboración propia del autor

En la tabla 3 se presenta la lista de requerimientos del proyecto del sistema de matrícula y caja de una institución educativa. En la tabla se observa las iteraciones que tuvo el proyecto, la descripción del requerimiento, la planificación y el tiempo real en que se demoró en

realizar cada requerimiento y finalmente la desviación en base a lo planificado. De acuerdo a los resultados se puede observar que existen 11 requerimientos que no se han desviado de la planificación y representan casi la tercera parte del total de requerimientos. Además, la máxima desviación que se tuvo en el proyecto es del 33%. De estos resultados se puede deducir que el equipo en este tercer proyecto tuvo menos porcentaje de desviación en los diferentes requerimientos y fue más asertivo en la planificación en la tercera parte del proyecto.

Tabla 16. Planificación y ejecución del desarrollo de los requerimientos del proyecto del proyecto de sistema académico

Iteracion	Nº de Req	Lista de Requerimientos	Planificado	Real	%Diferencia
1	Req - 1	El administrador necesita visualizar los reportes: docentes y su carga, alumnos y su carga.	7	6	14%
1	Req - 2	Generar contrato - Matrícula	9	9	0%
1	Req - 3	Registrar Carga Académica	3	4	33%
1	Req - 4	El coordinador podrá visualizar la lista de programas académicos que debe gestionar	8	8	0%
1	Req - 5	Imprimir comprobante de pago (boleta o factura generada) para cuadrar caja y entregarle ur	8	8	0%
1	Req - 6	Se abrirá caja al inicio del día	8	9	13%
1	Req - 7	Cuadrar caja por ingresos del día	6	6	0%
1	Req - 8	Registrar el Pre-contrato - Programación de Carga	10	8	20%
1	Req - 9	El administrador del sistema necesita visualizar, agregar, editar y eliminar los programas aca	9	11	22%
1	Req - 10	El administrador del sistema necesita visualizar, agregar, editar y eliminar tarifarios para reali	5	5	0%
1	Req - 11	El coordinador necesita visualizar la lista de solicitudes de cambio o retiro de cursos	6	6	0%
1	Req - 12	El administrador necesita visualizar las atenciones	7	8	14%
1	Req - 13	El coordinador podrá reprogramar nuevos grupos de cursos de acuerdo a la demanda	12	11	8%
2	Req - 14	Registrar Contrato - Formulario de Registro de Contrato	16	16	0%
2	Req - 15	El coordinador imprime tres copias del contrato y una es entregada al cliente	12	12	0%
2	Req - 16	Eliminar Admisión (Particular)	8	8	0%
2	Req - 17	Se registrará el cobro - Listar los cobros pendientes	4	4	0%
2	Req - 18	Registrar Modalidad de Ingreso	10	11	10%
2	Req - 19	Visualizar contratos Pendientes de pago y pagados	5	6	20%
2	Req - 20	El administrador del sistema necesita visualizar, agregar, editar los clientes matriculados	12	9	25%
2	Req - 21	Atender Admisión	10	12	20%
2	Req - 22	Se registrará el cobro - Registrar cobro	8	8	0%
2	Req - 23	Solicitudes de cambio - Listar solicitudes Pendientes	5	6	20%
2	Req - 24	Registrar orden de admisión	12	10	17%
3	Req - 25	Registrar Admisión	12	10	17%
3	Req - 26	El administrador del sistema necesita visualizar, agregar, editar y eliminar usuarios del sisten	6	7	17%
3	Req - 27	El administrador, secretarías y/o usuarios del sistema necesitan visualizar, agregar, editar y	10	11	10%
3	Req - 28	Solicitudes de cambio - Registrar solicitudes Pendientes	9	10	11%
3	Req - 29	El usuario necesita registrar, editar y eliminar Accesos	6	7	17%
3	Req - 30	El usuario necesita administrar la Asignación de accesos al rol	4	5	25%
3	Req - 31	Los usuarios necesitan acceder al sistema mediante una contraseña para realizar las opera	13	15	15%
4	Req - 32	El usuario necesita registrar, editar y eliminar Roles	6	8	33%
4	Req - 33	El usuario del sistema necesita visualizar, registrar, editar y eliminar cabinas	3	4	33%
4	Req - 34	Listar Admisiones	7	8	14%
4	Req - 35	El sistema mostrará el rol que cumple el usuario en el programa	6	7	17%

Fuente. Elaboración propia del autor

5.5 Parámetros estadísticos de la desviación de los planificado vs realidad.

En la tabla 27 se presenta el resumen de los casos analizado de los proyectos en estudio. Se observa que los proyectos del grupo 1 que representan el grupo que aplicó SCRUM y CMMI tiene un total de 99 requerimientos analizados. El grupo 2 que representa los proyectos que no utilizaron el SCRUM ni el CMMI se tuvo 84 casos analizados.

Tabla 17. Resumen de casos analizados de los proyectos en estudio

Grupos	Casos					
	Válido		Perdidos		Total	
	N	Porcentaje	N	Porcentaje	N	Porcentaje
1	99	100,0%	0	0,0%	99	100,0%
2	84	100,0%	0	0,0%	84	100,0%

Fuente: Resultados de la Investigación

En la tabla 28 se presenta los parámetros descriptivos de la desviación porcentual de los proyectos estudiados de acuerdo a sus requerimientos analizados en base a la planificación versus el tiempo real de desarrollo de dicho requerimiento. De los resultados se observa que la media del grupo 1 es 22.19% del cual se deduce que en promedio la desviación de la planificación de los requerimientos de los proyectos que utilizaron SCRUM con CMMI es solo en un 22.19%, a diferencia del grupo 2 que tiene una media de 53.75% de desviación; es decir que en promedio la desviación sobre lo base de lo planificado está por encima del 50%. De aquí se puede deducir que, si en total un proyecto tiene un tiempo planificado de duración de 100 horas, si se utiliza SCRUM y CMMI el proyecto se puede extender o desviar hasta 123 horas en promedio, y si no se

utiliza SCRUM y CMMI el proyecto se puede extender o desviar en promedio hasta 154 horas. Por otra parte, se puede observar que la desviación estándar del grupo 1 que es el grupo experimental es de 18.512% y del grupo de control 2 es de 29.791%; de estos resultados se deduce que el grupo experimental en sus resultados es mucho más homogéneo frente al grupo de control y este resultado refuerza el impacto del modelo SCRUM con CMMI.

Tabla 18. Parámetros descriptivos de la desviación porcentual de los proyectos

Grupos		Estadístico	Error estándar
1	Media	22,19	1,861
	95% de	Límite inferior	18,50
	intervalo de	Límite superior	25,88
	Media recortada al 5%	20,57	
	Mediana	18,00	
	Varianza	342,708	
	Desviación estándar	18,512	
	Mínimo	0	
	Máximo	100	
	Rango	100	
	Rango intercuartil	22	
	Asimetría	1,403	,243
	Curtosis	2,740	,481
2	Media	53,75	3,250
	95% de	Límite inferior	47,28
	intervalo de	Límite superior	60,22
	Media recortada al 5%	51,83	
	Mediana	50,00	
	Varianza	887,515	
	Desviación estándar	29,791	
	Mínimo	0	
	Máximo	175	
	Rango	175	
	Rango intercuartil	34	
	Asimetría	1,192	,263
	Curtosis	2,642	,520

Fuente: Resultados de la investigación

5.6 Prueba de normalidad de la muestra de desviación porcentual de los proyectos.

Para proceder a contrastar la hipótesis, los datos de la muestra de la desviación porcentual deben ajustarse a una curva normal. Entonces, para probar la normalidad se aplica la prueba estadística de Kolmogorov-Smirnov a ambos grupos.

Tabla 19. Prueba estadística no paramétrica de Kolmogorov-Smirnov

Grupo	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
1	,173	99	,060	,880	99	,065
2	,114	84	,059	,931	84	,051

a. Corrección de significación de Lilliefors

Fuente. Resultados de la investigación

En la tabla 19 se presenta la prueba estadística no paramétrica de Kolmogorov-Smirnov para muestras independientes. Y se observa que en P-Valor del grupo experimental es 0.060 el cual es mayor al nivel de significancia 0.05, de este resultado se deduce que la distribución de los datos del grupo experimental se ajusta a una curva normal. De manera similar se observa que en el grupo de control el P-Valor es 0.059 y es menor al nivel de significancia de 0.05; por lo tanto, se concluye que los datos del grupo de control se ajustan a una curva normal.

5.7 Diferencia de medias de las desviaciones porcentuales para grupos independientes.

Puesto que la unidad de análisis son las desviaciones que tienen los requerimientos respecto a la planificación frente al tiempo real de desarrollo de un requerimiento, la contrastación de la hipótesis será en función a la variable de desviación porcentual de planificación versus lo real.

Tabla 20. Medias de los grupos para muestras independientes de las desviaciones de los proyectos.

Participación	N	Media	Desviación estándar	Media de error estándar
Grupo Experimental	99	22,19	18,512	1,861
Grupo de Control	84	53,75	29,791	3,250

Fuente. Resultados de la investigación.

En la tabla 20 se presenta el estadístico de tendencia central que es la media aritmética para ver las diferencias de las evaluaciones de desviación porcentual entre grupos. En efecto, se observa que el promedio de desviación porcentual de los proyectos con SCRUM y CMMI en el grupo experimental fue de 22.19%, a diferencia del grupo de control que está muy por encima con un promedio de 53.75.

5.8 Contrastación de la hipótesis de la influencia del SCRUM y CMMI en el ciclo de vida de los proyectos.

Para responder la primera pregunta específica de investigación que es ¿Cuál es el grado de influencia del Modelo de Gestión de Proyectos SCRUM en el ciclo de vida del proyecto?; se planteó la hipótesis de: El Modelo de Gestión de Proyectos SCRUM y CMMI influye de manera positiva en el ciclo de vida de los proyectos de ingeniería de software de la consultora BP-QL. En efecto, se procede a contrastar la hipótesis con la prueba t-student para muestras independientes.

Tabla 21. Prueba t-student para muestras independientes

		Prueba de Levene de		prueba t para la igualdad de medias						
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	95% de intervalo de confianza de la diferencia	
									Inferior	Superior
Diferencias porcentuales	Se asumen varianzas iguales	12,612	,000	-8,739	181	,000	-31,558	3,611	-38,683	-24,433
	No se asumen varianzas iguales			-8,426	134,105	,000	-31,558	3,745	-38,966	-24,151

Fuente. Resultados de la investigación.

En la tabla 21 se presenta Prueba t-student para muestras independientes de las desviaciones porcentuales de los proyectos de ingeniería de software. De los resultados se observa que la prueba de Levene que analiza si las varianzas son iguales o no, se observa que el p-valor es de 0.000 y es menor a 0.05 del cual se concluye que los grupos de comparación no tienen varianzas iguales.

Además, se observa que al aplicar la prueba t-student, existe una diferencia de medias entre el grupo experimental y el grupo de control de 31.558 el cual es significativa con un p-valor de 0.000 siendo este menor

a 0.05. De este resultado se deduce que el Modelo de SCRUM y CMMI influyen de manera positiva en el ciclo de vida de los proyectos de Ingeniería de Software; es decir, que las actividades practicadas al implementar el modelo SCRUM en combinación con el CMMI en el grupo experimental ha influido en la planificación del desarrollo de los requerimientos logrando ser más asertivos y en consecuencia controlando la desviación del ciclo de vida del proyecto de ingeniería de software.

DISCUSIÓN

María E. Rojas Izaquita, (2011) como resultado del estudio ha logrado presentar una guía para que las organizaciones cuyo ciclo de vida de desarrollo de software se enmarcan dentro de las metodologías ágiles de desarrollo puedan mapear las prácticas de dicho ciclo de vida contra los objetivos genéricos, practicas genéricas, objetivos específicos y prácticas específicas del modelo CMMI en los niveles 2 y 3.

A diferencia de los resultados de la investigación anterior, en el presente estudio se ha integración el Scrum y las áreas de procesos CMMI. De las áreas de procesos que contempla el CMMI-DEV no en todas las áreas se puede integrar o aplicar la metodología ágil Scrum. Para este estudio se realizó la integración en las áreas de proceso del nivel dos considerando las áreas que mantienen los principios del manifiesto de tal forma que el modelo integrado no pierda su esencia de los principios. Finalmente se puede afirmar que si se puede mapear las metodologías ágiles con el modelo del CMMI; sin embargo no en todas las áreas se puede realizar una integración práctica de tal forma que conserve los principios fundamentales del manifiesto ágil que son los pilares de las metodologías ágiles.

CONCLUSIONES

Primero. Con respecto al modelo de gestión de proyectos SCRUM y CMMI. Al analizar los modelos de gestión de proyectos, se ha encontrado que el CMMI tienen elementos y procesos que llevan a una alta formalización en el proceso de desarrollo de software debido a la documentación que exige en los diferentes procesos, a diferencia de SCRUM tiene una baja formalización en la cual está más centrada en el equipo, en los principios y en el compromiso de cada miembro de los equipos. Al integrar o trabajar de manera conjunta estos modelos se logra un equilibrio que ayuda a una buena gestión del proyecto logrando una formalización media; es decir, la documentación básica y necesaria, y centrado en compromiso y la velocidad del equipo de tal forma que se pueda cumplir con los tiempos de entrega del producto.

Segundo. En el modelo integrado se observa de manera conjunta como las áreas de planificación de proyecto, la monitorización y el control del proyecto y la gestión de requisitos se alinean con el modelo SCRUM en sus diferentes componentes. Este modelo gráfico ha permitido una comprensión rápida del equipo de desarrollo para internalizar los cambios a realizar en su forma de trabajo.

Tercero. Al planificar con la técnica de Planinng Pocker considerando a todo el equipo, en el primer proyecto el 97% de los requerimientos presentaron desviaciones con respecto a lo planificado, en el segundo proyecto solo el 85% de los requerimientos presentaron desviaciones y en el tercer proyecto solo el 65% de los requerimientos presentaron desviaciones. Las desviaciones en general estuvieron por debajo del 50% en todos los proyectos en los que se aplicó el modelo SCRUM y CMMI. De estos resultados se concluye que la aplicación del SCRUM y el CMMI influyen en la asertividad de la planificación de los requerimientos de los proyectos de ingeniería de software.

Cuarto. Con respecto al ciclo de vida del proyecto. De acuerdo a los resultados se tiene que la media de la desviación porcentual del grupo experimental es 22.19% del cual se deduce que en promedio la desviación de la planificación de los requerimientos de los proyectos que utilizaron SCRUM con CMMI es solo en un 22.19%, a diferencia del grupo de control se tiene una media de 53.75% de desviación porcentual; es decir que en promedio la desviación sobre lo base de lo planificado está por encima del 50%. De aquí se concluye que, si en total un proyecto tiene un tiempo planificado de duración de 100 horas, si se utiliza SCRUM y CMMI el proyecto se puede extender o desviar hasta 123 horas en promedio, y si no se utiliza SCRUM y CMMI el proyecto se puede extender o desviar en promedio hasta 154 horas.

Quinto. Finalmente, al aplicar la prueba t-student, existe una diferencia de medias entre el grupo experimental y el grupo de control de 31.558 con un nivel de significancia p-valor de 0.000 siendo este menor a 0.05. De este resultado se concluye que el Modelo de SCRUM y CMMI influyen de manera positiva en el ciclo de vida de los proyectos de Ingeniería de Software; esto significa que las actividades practicadas al implementar el modelo SCRUM en combinación con el CMMI influyen en el ciclo de vida de los proyectos de ingeniería de software medido en base a la desviación porcentual con respecto a la planificación de los requerimientos de cada proyecto.

RECOMENDACIONES

Primero. Con respecto al modelo SCRUM y CMMI se recomienda para futuras investigaciones analizar los diferentes componentes del modelo SCRUM que se han integrado con las áreas del CMMI considerando los formatos utilizados para añadir indicadores de gestión en cada componente integrado y así recopilar evidencias cuantitativas para el modelo.

Segundo. Para los equipos SCRUM que utilizan la técnica planning pocker, de acuerdo a las experiencias en cada proyecto, se recomienda realizar una tabla de equivalencia considerando las variables de esfuerzo versus tiempo en que lleva en desarrollar un requerimiento. Esto resultados de permiten definir con mayor asertividad el ciclo de vida de los proyectos de ingeniería de software a medida que transcurre el tiempo.

Tercero. La variable dependiente de la presente investigación fue la desviación porcentual en base a lo planificado y el tiempo real en que se hizo un requerimiento del proyecto permite hacer una retroalimentación al equipo SCRUM y en cada sprint ir mejorando hasta llegar a un mínimo aceptable de desviaciones porcentuales. En efecto, se recomienda hacer una investigación longitudinal considerando como unidad de análisis los sprint y analizar el comportamiento de las desviaciones en el tiempo con el objetivo de que la tendencia sea negativa ya que esto significa que son menores las desviaciones a medida que transcurre el tiempo.

REFERENCIAS BIBLIOGRÁFICAS

- [1] M. Beedle *et al.*, “Manifesto for Agile Software Development,” pp. 2–3, 2001.
- [2] A. Iglesias, A. Messino, P. Pedroza, and R. Llanos, “Applicable Model To Process Management Agile Software Development Based on Cmmi 1 . 3 – Dev and Scrum Modelo Aplicable a La Gestión De Procesos Ágiles De Desarrollo De Software Basado En Cmmi – Dev,” vol. 3, no. 3, pp. 17–29, 2015.
- [3] E. W. Dijkstra and E. W., “The humble programmer,” *Commun. ACM*, vol. 15, no. 10, pp. 859–866, Oct. 1972.
- [4] M. Schmand *et al.*, “Performance results of a new DOI detector block for a high resolution PET-LSO research tomograph HRRT,” *IEEE Trans. Nucl. Sci.*, vol. 45, no. 6, pp. 3000–3006, 1998.
- [5] D. Galin and M. Avrahami, “Are CMM Program Investments Beneficial? Analyzing Past Studies,” *IEEE Softw.*, vol. 23, no. 6, pp. 81–87, Nov. 2006.
- [6] D. Rico, “What Is the Roi of Agile Vs . Traditional Methods ?,” *Www.Gilb.Com/DI244*, no. Cmmi, pp. 1–10, 2008.
- [7] D. F. Rico, *ROI of software process improvement : metrics for project managers and software engineers*. J. Ross Pub, 2004.
- [8] M. Johnson, *Agile methodologies: Survey results*. Victoria, Australia: Shine Technologies., 2002.
- [9] S. W. Ambler, “Agile adoption rate survey,” *Retrieved September 17, 2006*.
- [10] J. M. Navarro and J. Garzás, “Experiencia en la implantación de CMMI-DEV v1. 2 en una micropyme con metodologías ágiles y software libre,” *REICIS Rev. Española Innovación, Calid. e Ing. del Softw.*, 2010.
- [11] D. West and T. Grant, “Agile Development: Mainstream Adoption Has Changed Agility,” *Forrester Res. Inc.*, pp. 1–20, 2010.
- [12] James Sywilok, “Agile Transition Strategy,” 2017. [Online]. Available: <https://www.frontrowagile.com/blog/posts/93-agile-transition-strategy>.
- [13] A. S. Méndez, “MODELO Y PRÁCTICAS ESENCIALES DE LA METODOLOGÍA DAC INTEGRANDO LOS MÉTODOS ÁGILES, PMBOK Y CMMI-DEV,” *Medellín-Colombia*, no. 2013, pp. 13–24, 2014.
- [14] C. R. Jakobsen and J. Sutherland, “Scrum and CMMI Going from Good to Great,” in *2009 Agile Conference*, 2009, pp. 333–337.
- [15] Z. Lina and S. Dan, “Research on Combining Scrum with CMMI in Small and Medium Organizations,” in *2012 International Conference on*

Computer Science and Electronics Engineering, 2012, pp. 554–557.

- [16] J. Navarro, José Manuel, Garzás, “Experiencia en la implantación de CMMI-DEV v1.2 en una micropyme con metodologías ágiles y software libre. REICIS,” *Rev. Española Innovación, Calid. e Ing. del Softw.*, vol. 6, 2010.
- [17] J. A. Hincapie, G. P. Gasca-Hurtado, and A. F. Bustamante, “Multimodel catalogue heuristics for software project managemet,” in *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, 2016, pp. 1–6.
- [18] C. P. Team, “CMMI ® for Development , Version 1 . 2,” no. August, 2006.
- [19] V. Escobar-Sarmiento and M. Linares-Vasquez, “A model for measuring agility in small and medium software development enterprises,” in *2012 XXXVIII Conferencia Latinoamericana En Informatica (CLEI)*, 2012, pp. 1–10.
- [20] N. C. Paulk, “Extreme programming from a CMM perspective,” *IEEE Softw.*, vol. 18, no. 6, pp. 19–26, 2001.
- [21] Shane Hastie and Stéphane Wojewoda, “Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch,” 2015. [Online]. Available: <https://www.infoq.com/articles/standish-chaos-2015>.
- [22] Y. Nohemí, G. Meneses, J. Juan, and H. Mora, “Análisis del Estado Actual de Certificaciones Mundial y en México,” vol. 79, pp. 121–134, 2014.
- [23] Software Engineering Institute, *CMMI for Acquisition, version 1.2*. 2007.
- [24] J. G. Parra and E. a Irrazábal, “Guía práctica de supervivencia en una auditoría CMMI,” p. 33, 2011.
- [25] Pearson Educación, “CMMI. Integración y mejora continua (1/2),” 2010.
- [26] SEI, “CMMI ® para Desarrollo, Versión 1.3,” *C. Para Dsarrolo, Version 1.3*, p. 23, 2010.
- [27] M. Eugenia and R. Izaquita, “AGILIZANDO LO ÁGIL: UN FRAMEWORK PARA LA DESARROLLO DE SOFTWARE BAJO EL MODELO CMMI EN COMPAÑÍAS QUE USAN METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE USANDO EL MODELO ACELERADO DE IMPLEMENTACIÓN (AIM),” Universidad Nacional de Colombia, 2011.
- [28] T. J. Gandomani and Hh. Zulzalil, “Compatibility of Agile Software Development Methods and CMMI,” *Indian J. Sci. Technol. Compat.*, vol. 6, no. 8, pp. 5090–5094, 2013.
- [29] Ximena Casas, “Llega Scrum, el método de trabajo inspirado en el rugby | El Cronista,” 2014. [Online]. Available: <https://www.cronista.com/negocios/-Llega-Scrum-el-metodo-de-trabajo-inspirado-en-el-rugby-20140219-0006.html>.
- [30] B. W. Boehm, “Software Engineering As It Is.,” *Int Conf Softw. Eng*, 4th,

- Proc Tech Univ Munich*, pp. 11–21, 1979.
- [31] I. Sommerville, *Software Engineering*, 9th ed. 2010.
 - [32] B. W. Boehm, “Software Risk Management : Principles and practices,” *Software, IEEE*, no. January, pp. 32–40, 1991.
 - [33] V. Massol, *JUnit IN ACTION*. United States of America: Manning Publications Co, 2004.
 - [34] SCRUMstudy™, *CUERPO DE CONOCIMIENTO DE SCRUM (Guía SBOK™)*. 2016.
 - [35] K. Schwaber and J. Sutherland, “La Guía de Scrum,” p. 18, 2016.
 - [36] José Manuel Navarro, “Blog - Page 8 of 13 - Por el equipo de colaboradores de Scrum Manager,” 2011. [Online]. Available: <http://www.scrummanager.net/blog/page/8/>.
 - [37] P. Deemer, G. Benefield, and C. Larman, “the Primer,” *Development*, pp. 1–22, 2010.
 - [38] J. Herranz and G. Sáez, “New Identity-Based Ring Signature Schemes,” 2004, pp. 27–39.
 - [39] Software Engineering Institute and C. Desarrollo, *CMMI ® para Desarrollo, Versión 1.3*, no. November. 2010.
 - [40] SCRUMstudy™, *Cuerpo de conocimiento de scrum (Guía SBOK)*. 2017.
 - [41] J. Cegarra Sánchez, *Metodología de la investigación científica y tecnológica*. Ediciones Díaz de Santos, 2004.
 - [42] C. Hernández Hernández, *Metodología de la Investigación*, 5th ed. México: MCGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V, 2010.
 - [43] “Manifiesto por el Desarrollo Ágil de Software,” 2001. [Online]. Available: <http://agilemanifesto.org/iso/es/manifesto.html>.
 - [44] K. Beck, *Extreme Programming Explained , Second Edition*, 2nd ed. Massachusetts. United States: Pearson Education, In, 2012.
 - [45] K. Beck, “Test Driven Development: By Example,” 2002.
 - [46] K. Schwaber, *The Enterprice and Scrum*. United States of America., 2007.
 - [47] K. Schwaber, “SCRUM Development Process,” *Bus. Object Des. Implement.*, no. April 1987, pp. 117–134, 1997.
 - [48] J. P. Elm, D. R. Goldenson, K. El Emam, N. Donitelli, A. Neisa, and N. S. E. Committee, “A Survey of Systems Engineering Effectiveness - Initial Results (with detailed survey response data),” *Office*, no. December, 2008.

ANEXOS

Anexo 1. Requerimientos Sistema de Devueltas de Facturación.

N° DE PROYECTO	PLATAFORMA	ITERACCION	N° DE REQ.	DESCRIPCION DEL REQUERIMIENTO	ASIGNADO	FUNCION	PLANIFICADO	REAL	DIFERENCIA	% DIFERENCIA
PROYECTO DE SISTEMA DE DEVUELTAS DE FACTURACIÓN	APLICACIÓN WEB	1	Req - 1	RS01- Resolver posicion de comentarios cuando esta publicado en el servidor	Luis Rojas	Desarrollador	3	5	2	67%
		1	Req - 2	RM02- Reportes personalizados	Luis Rojas	Desarrollador	7	9	2	29%
		1	Req - 3	RF01- Agregar comprobantes a la lista de comprobantes de la bandeja de plantilla de envio	Sully Salazar	Desarrollador	10	8	-2	20%
		1	Req - 4	Eliminar comprobantes de la lista de comprobantes de la bandeja de planilla de envio	Sully Salazar	Desarrollador	2	3	1	50%
		1	Req - 5	Crear plantilla de forma manual y automatica	Sully Salazar	Desarrollador	9	9	0	0%
		1	Req - 6	Agregar comprobantes deslizando hacia una planilla	Ruben Huaman	Desarrollador	10	12	2	20%
		1	Req - 7	Eliminar comprobantes de la planilla al deslizar la planilla hacia la lista de comprobantes	Ruben Huaman	Desarrollador	8	6	-2	25%
		1	Req - 8	Enviar planilla deslizando a la sub - Bandeja de recibidos	Ruben Huaman	Desarrollador	2	3	1	50%
		1	Req - 9	Rechazar planillas	Kelvin Meza	Desarrollador	9	9	0	0%
		1	Req - 10	Enviar planilla recibida deslizando a la mesa de trabajo	Kelvin Meza	Desarrollador	5	4	-1	20%
		1	Req - 11	Analizar expediente	Ruben Huaman	Desarrollador	8	5	-3	38%
		1	Req - 12	Mostrar ubicación del comprobante	Ruben Huaman	Desarrollador	15	12	-3	20%
		1	Req - 13	Configuracion de usuarios	Sully Salazar	Desarrollador	10	11	1	10%
		1	Req - 14	Configuracion de roles	Luis Rojas	Desarrollador	8	7	-1	13%
		1	Req - 15	Acceso al sistema	Sully Salazar	Desarrollador	12	13	1	8%
		1	Req - 16	Editar comprobante	Luis Rojas	Desarrollador	9	11	2	22%
		1	Req - 17	Reportes	Luis Rojas	Desarrollador	5	4	-1	20%
		2	Req - 18	Agregar comprobantes por serie, numero o codigo de barra a una planilla	Kelvin Meza	Desarrollador	10	9	-1	10%
		2	Req - 19	Agregar por codigo de barras comprobantes analizados a una planilla	Kelvin Meza	Desarrollador	13	10	-3	23%
		2	Req - 20	Enviar planillas por codigo de barras	Kelvin Meza	Desarrollador	6	7	1	17%
		2	Req - 21	Recibir planillas por codigo de barras	Kelvin Meza	Desarrollador	8	7	-1	13%
		2	Req - 22	Enviar planilla a la mesa de trabajo por codigo de barras	Kelvin Meza	Desarrollador	7	11	4	57%
		2	Req - 23	Ejecutar comprobante mediante codigo de barras	Kelvin Meza	Desarrollador	11	9	-2	18%
		2	Req - 24	Mostrar el tiempo de la plantilla en ejecucion	Kelvin Meza	Desarrollador	12	13	1	8%
		2	Req - 25	Configuracion de rutas	Ruben Huaman	Desarrollador	9	7	-2	22%
		2	Req - 26	Configuracion de motivos	Luis Rojas	Desarrollador	3	4	1	33%
		2	Req - 27	RF025- Configuración de áreas	Kelvin Meza	Desarrollador	6	8	2	33%
		2	Req - 28	RP01- Elaborar presentación del SIDV	Sully Salazar	Desarrollador	5	7	2	40%
		3	Req - 29	RM05- Modificar requisito desvincular comprobante de la planilla	Sully Salazar	Scrum Master	3	5	2	67%
		3	Req - 30	RT01- Pruebas de regresion a sistema SIDV	Sully Salazar	Scrum Master	12	14	2	17%
		3	Req - 31	RP01- Configurar requisitos en el TFS	Sully Salazar	Scrum Master	3	6	3	100%
		3	Req - 32	RM04- Mostrar el destino del comprobante en analisis expediente	Sully Salazar	Scrum Master	4	6	2	50%
		3	Req - 33	RM03- Cambiar el color del componente	Ruben Huaman	Scrum Master	9	8	-1	11%
		3	Req - 34	RM01- Modificar agregar motivos	Ruben Huaman	Desarrollador	3	5	2	67%
		3	Req - 35	RM02- Reportes personalizados	Ruben Huaman	Desarrollador	5	8	3	60%

Anexo 2. Requerimientos del Sistema de Medicina Física

N° DE PROYECTO	PLATAFORMA	ITERACCION	N° DE REQ.	DESCRIPCION DEL REQUERIMIENTO	ASIGNADO	FUNCION	PLANIFICADO	REAL	DIFERENCIA	% DIFERENCIA
PROYECTO MEDICINA FÍSICA	APLICACIÓN WEB	1	Req - 1	El usuario del sistema necesita visualizar, registrar, editar y eliminar cabinas	Steve Campos	Scrum Master	5	6	1	20%
		1	Req - 2	El usuario necesita administrar la Asignacion de accesos al rol	Sully Salazar	Scrum Master	5	7	2	40%
		1	Req - 3	El usuario necesita registrar, editar y eliminar Roles	Jose Arias	Desarrollador	8	11	3	38%
		1	Req - 4	El usuario necesita registrar, editar y eliminar Accesos	Jose Arias	Desarrollador	8	9	1	13%
		1	Req - 5	Listar Admisiones	Marco C.	Desarrollador	3	5	2	67%
		1	Req - 6	Eliminar Admision (Particular)	Marco C.	Desarrollador	5	6	1	20%
		1	Req - 7	Atender Admision	Helen C.	Desarrollador	5	3	-2	40%
		1	Req - 8	Registrar Admision	Helen C.	Desarrollador	5	3	-2	40%
		2	Req - 9	Registrar orden de admision	Hector	Desarrollador	5	4	-1	20%
		2	Req - 10	Se registrara el cobro - Listar los cobros pendientes	Hector	Desarrollador	8	7	-1	13%
		2	Req - 11	Se registrara el cobro - Registrar cobro	Hector	Desarrollador	8	7	-1	13%
		2	Req - 12	Cuadrar caja por ingresos del dia	Steve Campos	Scrum Master	8	10	2	25%
		2	Req - 13	Registrar Modalidad de Ingreso	Sully Salazar	Scrum Master	8	11	3	38%
		2	Req - 14	Registrar Contrato - Formulario de Registro de Contrato	Jose Arias	Desarrollador	16	18	2	13%
		2	Req - 15	Registrar el Pre-contrato - Programacion de Carga	Jose Arias	Desarrollador	16	15	-1	6%
		2	Req - 16	Generar contrato - Matricula	Marco C.	Desarrollador	8	11	3	38%
		2	Req - 17	El administrador necesita visualizar las atenciones	Marco C.	Desarrollador	5	7	2	40%
		2	Req - 18	Registrar Carga Académica	Helen C.	Desarrollador	8	9	1	13%
		2	Req - 19	El administrador del sistema necesita visualizar, agregar, editar y eliminar usuarios del sistena	Helen C.	Desarrollador	8	9	1	13%
		3	Req - 20	Visualizar contratos Pendientes de pago y pagados	Hector	Desarrollador	13	10	-3	23%
		3	Req - 21	El administrador del sistema necesita visualizar, agregar, editar los clientes matriculados	Hector	Desarrollador	8	7	-1	13%
		3	Req - 22	El coordinador imprime tres copias del contrato y una es entregada al cliente	Hector	Desarrollador	8	10	2	25%
		3	Req - 23	El administrador del sistema necesita visualizar, agregar, editar y eliminar los programas aca	Steve Campos	Scrum Master	13	15	2	15%
		3	Req - 24	El administrador del sistema necesita visualizar, agregar, editar y eliminar tarifarios para real	Sully Salazar	Scrum Master	13	15	2	15%
		3	Req - 25	El administrador, secretarias y/o usuarios del sistema necesitan visualizar, agregar, editar y	Hector	Desarrollador	8	9	1	13%
		3	Req - 26	Los usuarios necesitan acceder al sistema mediante una contraseña para realizar las opera	Hector C.	Desarrollador	5	7	2	40%
		3	Req - 27	El sistema mostrara el rol que cumple el usuario en el programa	Hector C.	Desarrollador	5	7	2	40%
		3	Req - 28	El coordiandor podra visualizar la lista de programas académicos que debe gestionar	Marco C.	Desarrollador	8	8	0	0%
		3	Req - 29	El administrador necesita visualizar los reportes: docentes y su carga, alumnos y su carga.	Helen C.	Desarrollador	8	8	0	0%

Anexo 3. Requerimientos del Sistema Académico

N° DE PROYECTO	PLATAFORMA	ITERACCION	N° DE REQ.	DESCRIPCION DEL REQUERIMIENTO	ASIGNADO	FUNCION	PLANIFICADO	REAL	DIFERENCIA	% DIFERENCIA
PROYECTO DE SISTEMA ACADÉMICO	APLICACIÓN WEB	1	Req - 1	El administrador necesita visualizar los reportes: docentes y su carga, alumnos y su carga.	Fernando Lazo	Desarrollador	7	6	-1	14%
		1	Req - 2	Generar contrato - Matrícula	Rut Aguirre	Desarrollador	9	9	0	0%
		1	Req - 3	Registrar Carga Académica	Rut Aguirre	Desarrollador	3	4	1	33%
		1	Req - 4	El coordinador podrá visualizar la lista de programas académicos que debe gestionar	Fernando Lazo	Desarrollador	8	8	0	0%
		1	Req - 5	Imprimir comprobante de pago (boleta o factura generada) para cuadrar caja y entregarle al usuario	Hubert Baltazar	Desarrollador	8	8	0	0%
		1	Req - 6	Se abrirá caja al inicio del día	Rut Aguirre	Desarrollador	8	9	1	13%
		1	Req - 7	Cuadrar caja por ingresos del día	Hubert Baltazar	Desarrollador	6	6	0	0%
		1	Req - 8	Registrar el Pre-contrato - Programación de Carga	Rut Aguirre	Desarrollador	10	8	-2	20%
		1	Req - 9	El administrador del sistema necesita visualizar, agregar, editar y eliminar los programas académicos	Christian Lopez	Desarrollador	9	11	2	22%
		1	Req - 10	El administrador del sistema necesita visualizar, agregar, editar y eliminar tarifarios para real	Christian Lopez	Desarrollador	5	5	0	0%
		1	Req - 11	El coordinador necesita visualizar la lista de solicitudes de cambio o retiro de cursos	José Carlos Choquehuan	Desarrollador	6	6	0	0%
		1	Req - 12	El administrador necesita visualizar las atenciones	Leydi Torres	Testing	7	8	1	14%
		1	Req - 13	El coordinador podrá reprogramar nuevos grupos de cursos de acuerdo a la demanda	José Carlos Choquehuan	Desarrollador	12	11	-1	8%
		2	Req - 14	Registrar Contrato - Formulario de Registro de Contrato	Rut Aguirre	Desarrollador	16	16	0	0%
		2	Req - 15	El coordinador imprime tres copias del contrato y una es entregada al cliente	Rut Aguirre	Desarrollador	12	12	0	0%
		2	Req - 16	Eliminar Admisión (Particular)	José Carlos Choquehuan	Desarrollador	8	8	0	0%
		2	Req - 17	Se registrará el cobro - Listar los cobros pendientes	Hubert Baltazar	Desarrollador	4	4	0	0%
		2	Req - 18	Registrar Modalidad de Ingreso	Carlos Baltazar	Desarrollador	10	11	1	10%
		2	Req - 19	Visualizar contratos Pendientes de pago y pagados	Rut Aguirre	Desarrollador	5	6	1	20%
		2	Req - 20	El administrador del sistema necesita visualizar, agregar, editar los clientes matriculados	Leydi Torres	Testing	12	9	-3	25%
		2	Req - 21	Atender Admisión	José Carlos Choquehuan	Desarrollador	10	12	2	20%
		2	Req - 22	Se registrará el cobro - Registrar cobro	Hubert Baltazar	Desarrollador	8	8	0	0%
		2	Req - 23	Solicitudes de cambio - Listar solicitudes Pendientes	José Carlos Choquehuan	Desarrollador	5	6	1	20%
		2	Req - 24	Registrar orden de admisión	José Carlos Choquehuan	Desarrollador	12	10	-2	17%
		3	Req - 25	Registrar Admisión	José Carlos Choquehuan	Desarrollador	12	10	-2	17%
		3	Req - 26	El administrador del sistema necesita visualizar, agregar, editar y eliminar usuarios del sistema	Hubert Baltazar	Desarrollador	6	7	1	17%
		3	Req - 27	El administrador, secretarías y/o usuarios del sistema necesitan visualizar, agregar, editar y	Fernando Lazo	Desarrollador	10	11	1	10%
		3	Req - 28	Solicitudes de cambio - Registrar solicitudes Pendientes	José Carlos Choquehuan	Desarrollador	9	10	1	11%
		3	Req - 29	El usuario necesita registrar, editar y eliminar Accesos	Fernando Lazo	Desarrollador	6	7	1	17%
		3	Req - 30	El usuario necesita administrar la Asignación de accesos al rol	Hubert Baltazar	Desarrollador	4	5	1	25%
		3	Req - 31	Los usuarios necesitan acceder al sistema mediante una contraseña para realizar las opera	Hubert Baltazar	Desarrollador	13	15	2	15%
		4	Req - 32	El usuario necesita registrar, editar y eliminar Roles	Hubert Baltazar	Desarrollador	6	8	2	33%
		4	Req - 33	El usuario del sistema necesita visualizar, registrar, editar y eliminar cabinas	Christian Lopez	Desarrollador	3	4	1	33%
		4	Req - 34	Listar Admisiones	José Carlos Choquehuan	Desarrollador	7	8	1	14%
		4	Req - 35	El sistema mostrará el rol que cumple el usuario en el programa	Hubert Baltazar	Desarrollador	6	7	1	17%

Anexo 4. Plantilla para Historias de Usuario

[illegible]

Anexo 5. Plantilla del Estado del Spring de los proyectos

Nombre de la tarea	Responsable	Fecha de inicio	Fecha final	Días	Estado
Sprint 1	<Nombre>	9/3	9/13	10	Finalizado
Característica 1		9/3	9/7	4	Finalizado
Característica 2		9/7	9/12	5	Finalizado
Característica 3		9/9	9/13	4	Vencido
Sprint 2	<Nombre>	9/25	10/5	10	En progreso
Característica 4		9/25	9/29	4	En progreso
Característica 5		9/24	10/2	8	Sin empezar
Característica 6		10/2	10/5	3	Sin empezar
Sprint 3	<Nombre>	9/25	10/5	10	Sin empezar
Característica 7		9/25	9/29	4	Sin empezar
Característica 8		9/24	10/2	8	Sin empezar
Característica 9		10/2	10/5	3	Sin empezar

Anexo 6. Plantilla para el control de reuniones de retrospectiva

¿Qué salió bien en la Iteración?	¿Qué salió bien en la Iteración?	¿Qué mejoras vamos a implementar en la próxima Iteración?

Anexo 7. Interfaz del TRELLO donde se monitorearon los proyectos del grupo de control. Módulo GTH

GTH

Personal

Privado

AR

J

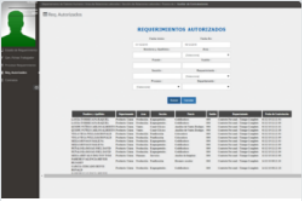
5

Requerimientos

RF08: El sistema debe mostrar el tiempo de sesión. Tiempo de sesión inactiva 20 minutos

RF09: Mantenimiento de DGP

RF11: Mantenimiento de Ficha de Datos del trabajador



Duplicidad en reporte de Requerimientos Autorizados

Carga masiva de requerimientos

Despues de autorizar la solicitud de Requerimiento, debe pasar a Estado de Requerimiento (no a req interrumpidos)

Fecha de suscripción = a la fecha de inicio por defecto



+ Añada otra tarjeta

Desarrollo

RF07: El sistema debe generar reportes

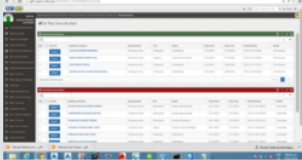
0/3

AR

ANULACION DE REQUERIMIENTO



El paso 1 sigue apareciendo con un solo nombre (a pesar de que lo realizan distintas secretarias)



Sol Requerimientos Fuera de Plazo - Agregar la columna plazo y en solicitudes autorizadas ordenar del ultimo autorizado (debe salir primero)

Autorización de DGP en masivo, para gerente financiero, gerente general, otros administrativos

Para Asignación Familiar, agregar

+ Añada otra tarjeta

Testing

+ Añada una tarjeta

Terminado (host local)

Datos del Trabajador

Nombre : MANUEL ALCIDES

Apellido Paterno : CONDORI

Apellido Materno : GARCIA

Fecha de Nacimiento : 16/12/1988

Tipo de documento : DNI

Numero de documento : 45451423

Quitar tipo de documento, por que está duplicado y mostrando datos errados



NO carga el centro de costo, condicion al usuario remuneraciones



Actualizar "subir fotos de trabajador"

19 de ene. de 2016

+ Añada otra tarjeta

Implementacion (server)

RF06: Actualizar contrato.

OBS: El centro de costo se esta listando en duplicado



Centro de costo duplicado - BERROCAL BALVIN RONALD MISIAEL, HERRERA BLANCAS GABY, LAUCATA ZAMORA GREGORIA , OTROS

REVISAR LA SOLICITUD DE DGPS

MOstrar SISTEMA PENSIONARIO Y NOMBRE AFP (EN DATOS CONTRACTUALES) AREA LABORAL



RF10: Actualizar Formulario de contrato que visualiza remuneraciones: Mostrar en Rojo, religión

+ Añada otra tarjeta

97

Anexo 8. Interfaz del TRELLO donde se monitorearon los proyectos del grupo de control. Módulo GTH Vacaciones

RRHH - Módulo de Vacaciones | Equipo privado | Visible para el equipo

To Do

- RFMV006 - El sistema debe permitir listar las vacaciones programadas
- RFMV007 - El sistema debe permitir aprobar o rechazar las programaciones individual o masivamente al Jefe de Departamento
- RFMV009 - El sistema debe permitir generar el cuaderno de vacaciones a partir del listado del RFMV008
- RFMV008 - El sistema debe permitir listar a los trabajadores con la programación de vacaciones autorizada por el jefe del departamento
- RFMV010 - El sistema debe notificar a los trabajadores que ya tienen sus vacaciones programadas
- RFMV011 - El sistema debe permitir listar a los trabajadores cuya programación de vacaciones fué

Doing

- RFMV001 - El sistema debe permitir listar los trabajadores a quienes les corresponden vacaciones
- RFMV005 - El sistema debe permitir programar las vacaciones masivamente
- RFMV002 - El sistema debe permitir listar a los trabajadores seleccionados para vacaciones para su posterior programación de vacaciones

Testing

- RFMV004 - El sistema debe permitir subir archivos en el formulario para la programación de vacaciones

Done

- Crear el proyecto con framework Spring
- RFMV003 - Crear un formulario para la programación de vacaciones de cada trabajador

Resources

- Diagrama de actividad del módulo de vacaciones
- Ejemplo del cuaderno de vacaciones

+ Añada otra tarjeta