

ADAPTACIÓN DE UN PROCESO DE DESARROLLO DE SOFTWARE BASADO EN
BUENAS PRÁCTICAS

JAIME ANDRÉS BRITTO MONTOYA

UNIVERSIDAD AUTONOMA DE MANIZALES
FACULTAD DE INGENIERÍA
MAESTRIA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE
MANIZALES
2014

ADAPTACIÓN DE UN PROCESO DE DESARROLLO DE SOFTWARE BASADO EN
BUENAS PRÁCTICAS

JAIME ANDRES BRITTO MONTOYA

Tesis de grado para optar al título de magister en Gestión y desarrollo de proyectos de
software

ASESOR

Sandra Victoria Hurtado Gil

UNIVERSIDAD AUTONOMA DE MANIZALES

FACULTAD DE INGENIERÍA

MAESTRIA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE

MANIZALES

2014

Dedicatoria

*A mis hijas a quienes recibí en mi vida
durante este proceso y a mi esposa
quién me motivó y ayudó
a obtener este logro.*

TABLA DE CONTENIDO

LISTA DE FIGURAS	4
LISTA DE TABLAS	5
RESUMEN.....	7
INTRODUCCIÓN	8
1. REFERENTE CONTEXTUAL	11
1.1. Formulación del problema	11
1.2. Antecedentes	11
1.3. Justificación.....	13
1.4. Objetivos	16
1.4.1. Objetivo general.....	16
1.4.2. Objetivos Específicos.....	16
1.5. Alcance.....	17
2. REFERENTE TEÓRICO.....	18
2.1. Definiciones	18
2.1.1. Proceso de desarrollo de software.....	18
2.1.2. Métodos de Ingeniería de Software.....	19
2.2. Metodologías de Desarrollo de Software	19
2.2.1. Scrum	19
2.2.2. ICONIX.....	21
2.2.3. PROGRAMACIÓN EXTREMA (XP).....	23

2.3.	Modelos de referencia.....	25
2.3.1.	CMMI (Capability Maturity Model Integration)	26
2.3.2.	SPEM (Software Process Engineering Meta-model).....	29
2.4.	Compatibilidad entre CMMI y Ágil.....	31
2.5.	Matrices de comparación	36
2.6.	Frameworks mixtos.....	39
3.	APORTE INVESTIGATIVO	41
4.	METODOLOGIA PROPUESTA	42
4.1.	Enfoque	42
4.2.	Unidad de trabajo	44
4.3.	Instrumentos de recolección de información	45
5.	RESULTADOS ESPERADOS.....	46
6.	IMPACTOS ESPERADOS A PARTIR DEL USO DE LOS RESULTADOS	47
7.	DESARROLLO	48
7.1.	Descripción y evaluación del proceso de desarrollo actual del ADSCR	48
7.1.1.	Descripción del proceso de desarrollo actual del ADSCR.....	48
7.1.2.	Evaluación del proceso de desarrollo actual del ADSCR.....	51
7.1.3.	Fortalezas y debilidades del proceso de desarrollo actual	55
7.2.	Análisis comparativo de las prácticas ágiles seleccionadas.....	59
7.2.1.	Construcción del instrumento de comparación.....	59
7.2.2.	Comparación prácticas ágiles y el proceso del ADSCR.....	62
7.2.3.	Priorización de prácticas	72
7.3.	Elaboración del nuevo proceso ajustado para el ADSCR.....	76

7.3.1.	Selección de prácticas	76
7.3.2.	Diseño del nuevo proceso de desarrollo.....	83
7.3.3.	Documentación del proceso en EPF	101
7.4.	Evaluación del proceso estructurado propuesto	104
7.4.1.	Prueba Piloto	104
7.4.2.	Instrumento de evaluación	105
7.4.3.	Resultado de la evaluación.....	105
7.4.4.	Análisis de resultados de la prueba	107
7.5.	Resumen gerencial para Comfamiliar.....	111
7.6.	Análisis de resultados.....	111
8.	CONCLUSIONES	113
8.1.	Evaluación.....	113
8.2.	Comparación	114
8.3.	Diseño proceso estructurado de software.....	115
8.4.	Validación del proceso	115
9.	RECOMENDACIONES	117
	REFERENCIAS	118

LISTA DE FIGURAS

Figura 1. Resumen elementos de Scrum.....	20
Figura 2. Resumen del proceso ICONIX.....	22
Figura 3. XP en capas.....	24
Figura 4. CMMI, SCAMPI y la organización.....	26
Figura 5. Constelaciones CMMI.....	27
Figura 6. Diferencia Método y Proceso SPEM.....	31
Figura 7. Nivel de satisfacción de XP a las áreas de proceso de CMM.....	33
Figura 8. XP y las prácticas de institucionalización	34
Figura 9. Ubicación del proceso de sistemas en el organigrama de Comfamiliar Risaralda	48
Figura 10. Resultado porcentual cumplimiento por área de proceso evaluación inicial del área de desarrollo de SW.....	57
Figura 11. Comparación radial del cubrimiento de CMMI por las metodologías y el proceso actual del ADSCR.....	68
Figura 12. Comparación del cubrimiento frente a CMMI del conjunto de prácticas ágiles y el proceso actual del ADSCR.....	71
Figura 13. Role, tareas, productos de trabajo y guías documentadas en EPF.....	102
Figura 14. Ejemplo vista generada del rol Dueño del producto.....	103
Figura 15. Proceso modelado en EPF	104

LISTA DE TABLAS

Tabla 1. Comparación de calificaciones usadas por Paulk y Fritzsche & Keli.....	34
Tabla 2. Calificaciones usadas por Diaz y Garbajosa.....	36
Tabla 3. Comparación del porcentaje de cubrimiento de estudios previos.....	38
Tabla 4. Resumen evaluación del proceso actual por áreas de proceso.....	52
Tabla 5. Resumen resultado del proceso actual evaluación por prácticas específicas.....	53
Tabla 6. Homologación de calificaciones para los 2 instrumentos.....	54
Tabla 7. Comparación resultados evaluación.....	54
Tabla 8. Fortalezas y debilidades del proceso actual identificadas mediante la evaluación realizada.....	55
Tabla 9. Criterios de cubrimiento entre prácticas CMMI y Metodologías ágiles.....	59
Tabla 10. Ejemplo de comparación de Scrum frente a CMMI.....	60
Tabla 11. Calificaciones ajustadas a 5 niveles de detalle.....	61
Tabla 12. Homologación de calificaciones entre el instrumento de caracterización y la matriz de cumplimiento.....	62
Tabla 13. Homologación final para la calificación de la matriz de cumplimiento.....	62
Tabla 14. Homologación de calificaciones y cálculo del porcentaje de cubrimiento de la metodología frente a CMMI para el área PMC con respecto a la metodología XP.....	63
Tabla 15. Reevaluación del proceso actual.....	64
Tabla 16. Resumen cubrimiento porcentual por área de proceso de cada metodología.....	65
Tabla 17. Máximo potencial de cubrimiento de las prácticas ágiles seleccionadas frente a CMMI.....	68
Tabla 18. Potencial de mejora de cada área de proceso.....	70

Tabla 19. Áreas con mayor posibilidad de mejora.....	72
Tabla 20. 9 Áreas de proceso seleccionadas	73
Tabla 21. Ejemplo de ponderación para las prácticas específicas de REQM en cada uno de los métodos a evaluar	74
Tabla 22. Resumen de los resultados de la ponderación para las áreas de proceso seleccionadas.....	75
Tabla 23. Visión general del aporte de las prácticas ágiles a cada área de proceso.....	82
Tabla 24. Ejemplo evaluación realizada del nuevo proceso para la meta específica 3 de CM.	106
Tabla 25. Resultado evaluación porcentaje cumplimiento frente a CMMI del nuevo proceso de software.	107
Tabla 26. Comparación entre la mejora esperada y la real obtenida luego de la prueba piloto.	110

RESUMEN

Este proyecto presenta la construcción de un proceso de desarrollo de software para el área de sistemas de Comfamiliar Risaralda mediante la selección de las mejores prácticas obtenidas de las metodologías de Scrum, XP e ICONIX, tomando como marco de referencia al modelo de mejora y evaluación *Capability maturity model integration* para desarrollo (CMMI-DEV 1.3). Para este fin se evaluó el proceso actual, comparándolo con las prácticas ágiles provenientes de las metodologías, seleccionando las de mayor posibilidad de mejora, luego se diseñó el nuevo proceso usando estas prácticas seleccionadas. El nuevo proceso propuesto fue aplicado en una prueba piloto dentro de la institución y luego fue evaluado con respecto a CMMI como se hizo inicialmente para las metodologías ágiles mencionadas. Se obtuvo finalmente una matriz de comparación frente a CMMI en el cual se puede evaluar cualquier método o proceso de desarrollo, y un proceso estructurado de software formalmente definido mediante la herramienta Eclipse Process Framework. Se observa una mejora en el cumplimiento de prácticas frente a CMMI del 25% con respecto al proceso anterior aunque se encuentran dificultades al aplicar prácticas relacionadas con documentación extensiva y toma de medidas.

Palabras clave: EPF, CMMI, Scrum, XP, ICONIX, proceso de desarrollo, proceso estructurado de software.

INTRODUCCIÓN

En el área de desarrollo de software de Comfamiliar Risaralda (de ahora en adelante ADSCR) se han encontrado dificultades como sobrecostos y retrasos de cronogramas¹, la cobertura de la documentación de sistemas de información no es suficiente², lo que acarrea problemas de mantenimiento de las aplicaciones. Tal como lo encontró Pino, en el ADSCR, algunos programadores piensan que están haciendo las cosas bien (Pino, Vidal, Garcia, & Piattini, 2007) en uno de sus trabajos. El grupo de desarrollo se desenvuelve en un ambiente cambiante donde se requieren resultados rápidos, los equipos son unipersonales o pequeños y donde el conocimiento de cada solución de software desarrollado lo tienen unas pocas personas.

En (Pino, Vidal, Garcia, & Piattini, 2007) se plantea que posiblemente las propuestas extranjeras están enfocadas para grandes compañías con otras características, la misma duda que impedía implementar en el ADSCR una metodología de desarrollo de software tradicional.

En el 2010 el área de desarrollo en cuestión descubre ICONIX³ una metodología muy sencilla y que usa diagramas UML⁴ que los desarrolladores reconocen fácilmente; y

1 En el Informe Seguimiento Proyecto Nomina y Gestión Humana del año 2008 radicado 8819 se encontraron diferencias en costo del proyecto frente a lo presupuestado y lo efectivamente ejecutado equivalente sobrecosto del 38%. Dicho documento también evidencia un retraso de 45 días en el cronograma propuesto.

2 Se realizó un control de documentación de sistemas de información en el año 2010 donde se evidencia que de 28 proyectos a los que se le realizó el seguimiento un 32% no tenían ningún tipo de documentación oficial y solo la mitad cumplían con al menos el 60% de todos los ítems definidos en el instructivo.

3 Es una metodología abierta y gratis basada en UML con el fin de desarrollar el análisis y el diseño que incluye un análisis de robustez en la mitad que conecta ambos, el llamado proceso ICONIX es descrito en el libro UML Theory and Practice.

4 El Lenguaje unificado de modelamiento UML es una especificación desarrollada por la OMG que permite describir estructuras de aplicaciones, de comportamiento, de arquitectura, de negocios de proceso y de estructura de datos.

mediante una investigación de integrantes del equipo conocen a Disciplined Agile Delivery (DAD)⁵ un “proceso” híbrido publicado por IBM.

En este momento surge la inquietud, ¿Qué proceso de desarrollo de software podría resolver las necesidades de esta área de desarrollo? Las metodologías de desarrollo ágil se ajustan mejor para equipos pequeño y para entornos cambiantes (Highsmith, 2002) ¿Podría adecuarse una metodología de desarrollo ágil? En este punto se investigó sobre Scrum (Deemer, Benefield, Larman, & Vodde, 2009) un marco de trabajo para la gestión de equipos de desarrollo ágil, y sobre Programación Extrema (XP) (Beck & Andres, 2004) una metodología ágil de desarrollo completamente enfocada en la habilidad de los desarrolladores. Esto generaba más preguntas que respuestas, ¿es Scrum suficiente y adecuado para la empresa? ¿O lo es XP? ¿O un proceso híbrido como DAD (Ambler & Lines, 2012) más adecuado?

Este trabajo nace de estas inquietudes y tuvo como objetivo establecer un proceso estructurado de software (PES) para el ADSCR, basado en el análisis de CMMI⁶ y los métodos ágiles de ingeniería de software seleccionados Scrum, XP e ICONIX.

Este proyecto inicia con una evaluación del estado actual de la empresa aprovechando el instrumento basado en CMMI desarrollado dentro del grupo de investigación de la UAM.

Luego se compara el proceso actual del ADSCR con Scrum, XP e ICONIX a la luz de CMMI para encontrar las posibilidades de mejora y lograr relacionar y seleccionar las prácticas de estas metodologías que mejor se ajusten a la compañía. Con el resultado de esta comparación se define un proceso estructurado con estas metodologías y se documenta formalmente mediante SPEM⁷. Finalmente se valida este nuevo proceso mediante una prueba de concepto utilizando un proyecto piloto del ADSCR.

5 Metodología híbrida de gran influencia ágil descrita por Ambler y Lines en el libro *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise* publicado por IBM.

6 CMMI es una colección de mejores prácticas que ayudan a mejorar los procesos de la organización creado por el Software Engineering Institute. Este se usará como punto de referencia y como base de medida para la evaluación de las diferentes prácticas encontradas en la organización y en las metodologías ágiles seleccionadas.

7 Un lenguaje de modelado específico para procesos y métodos de software creado por la OMG. Mediante una herramienta que se ajuste a la especificación SPEM, es posible definir un modelo de proceso mediante un lenguaje de modelamiento no ambiguo usando generalmente tres elementos básicos: rol, producto de trabajo y tarea.

Con el fin de lograr los objetivos propuestos se propone una metodología empírica desarrollada en 5 fases que inicia con la evaluación del proceso actual, luego realizando una comparación de las metodologías, elaborando el nuevo proceso, evaluando este nuevo proceso en Comfamiliar y finalizando con una fase de conclusión y cierre final.

En este documento se encontrará la definición básica del proyecto incluyendo la metodología empleada y los instrumentos a utilizar para continuar con el desarrollo del mismo (numeral 4).

En el desarrollo se muestran los resultados obtenidos de describir y evaluar el proceso actual del ADSCR encontrando sus fortalezas y debilidades (numeral 7.1).

Luego se muestra el proceso de construcción del instrumento de comparación y el resultado de su aplicación para Scrum, XP e ICONIX (numeral 7.2).

Estas comparaciones se unen a la del proceso obteniendo los mejores nichos de crecimiento según las fortalezas de las metodologías evaluadas y las debilidades del proceso actual (numeral 7.2).

Seguidamente se muestra la ponderación de las prácticas de las metodologías ágiles y la construcción del nuevo proceso de desarrollo basado en esta ponderación (numeral 0).

Finalmente se prueba este proceso en un piloto dentro del ADSCR y se evalúa mediante el mismo instrumento describiendo los beneficios reales obtenidos (numeral 7.4).

1. REFERENTE CONTEXTUAL

1.1. FORMULACIÓN DEL PROBLEMA

¿Cuál es el proceso estructurado de software (PES) que se adapta a las necesidades y debilidades encontradas en el proceso actual en Comfamiliar Risaralda?

1.2. ANTECEDENTES

Varios estudios han realizado matrices de comparación entre metodologías ágiles y CMMI, por ejemplo, (Fritzsche & Keil, 2007) (Paulk, 2001) plantearon la inquietud sobre si los métodos ágiles en especial XP serían compatibles con CMMI o al contrario estarían en conflicto; En este estudio se analizaron todas las áreas de proceso para CMMI 1.2 y las prácticas genéricas, el método usado fue subjetivo a nivel de áreas de proceso y en algunos casos de prácticas específicas pero no a un nivel muy detallado. Este estudio permitió definir algunos lineamientos generales para realizar la comparación entre CMMI y las metodologías ágiles, además de presentar la viabilidad de este tipo de propuestas. Sin embargo, para la comparación en este proyecto se usó la propuesta de otros autores (que se presentan a continuación), complementada con elementos cuantitativos para facilitar el análisis de los resultados.

La relación de SCRUM y CMMI en las áreas de PP, PMC, SAM, IPM, RSKM, QPM fueron analizadas en el trabajo de (Marcal, de Freitas, Furtado Soares, & Belchior, 2007). Allí se analizó el propósito de cada práctica específica de dichas áreas y se contrastó con las prácticas definidas en SCRUM, estableciendo en qué medida la práctica de CMMI se encontraba satisfecha. Luego este estudio fue profundizado analizando la correspondencia de SCRUM y XP pero solo en las áreas de proceso PP, PMC y REQM por (Díaz Fernandez, 2009), en este caso el método usado para analizar la correspondencia con CMMI fue mediante una evaluación a nivel de subprácticas. Esta comparación a nivel de subpráctica fue usada en el proyecto para determinar el nivel de cumplimiento de cada práctica de CMMI por parte de una metodología ágil, pero ampliándolo a más áreas de proceso. Para este trabajo se usó en una primera fase la calificación propuesta en este trabajo, de tres valores: cumple totalmente, cumple parcialmente o no cumple; aunque luego se hizo una equivalencia con una escala cuantitativa.

En la version 1.3 de CMMI para desarrollo (CMMI Product Team, 2010) se incluyen notas en 10 de las áreas de proceso con el fin de facilitar la interpretación de las prácticas cuando se desean aplicar éstas en ambientes ágiles, dichas notas también refuerzan la noción de que CMMI es independiente de la implementación y de que hay cabida para su uso en entornos donde son valorados los principios del manifiesto ágil.

En el libro *Disciplined agile Delivery* (Ambler & Lines, 2012) se describe un marco de trabajo basado en Scrum, Programación Extrema (XP), Agile Modeling (AM), el Proceso Unificado (UP), Agil Data(AD) y Kanban. Este enfoque fue diseñado por los autores con el fin expandir el alcance de las metodologías ágiles en entornos empresariales, con proyectos riesgosos y equipos más grandes donde se requiere la institucionalización de los procesos. Esta propuesta presenta elementos prácticos que permiten demostrar la fiabilidad de integrar metodologías disciplinadas y metodologías ágiles. En el caso del presente proyecto, se incluyeron Scrum, XP e ICONIX, pero basadas en el proceso actual de ADSCR y no en UP.

Por otro lado las investigaciones como son los trabajos de (Pino, Vidal, Garcia, & Piattini, 2007) y (Leticia, Astorga Vargas, Olguín Espinoza, & Andrade Peralta, 2008) y sus conclusiones, sirven como antecedentes a la implementación de procesos de mejora en las áreas de desarrollo de software para empresas pequeñas y latinoamericanas, que muestran elementos claves relacionados con la cultura que son necesarios para el ajuste de metodologías y que no se contemplan en publicaciones estadounidenses como la mencionada anteriormente (Ambler & Lines, 2012).

Sin embargo (Fayad, Laitinen, & Ward, 2000) ya habían resaltado los retos de las pequeñas empresas desarrolladoras de software donde se indica sobre posibles problemas al tratar de escalar estas metodologías hacia abajo. De acuerdo con la propuesta de estos autores, el presente proyecto tendrá en cuenta estos aspectos relacionados con equipos pequeños así como la experiencia previa del ADSCR y usará una prueba de concepto para comenzar la implantación del proceso propuesto en la empresa, dejando así sentadas las bases para su implementación posterior por parte de la empresa.

1.3. JUSTIFICACIÓN

Durante el desarrollo de un proyecto de software en el año 2008 fueron detectados sobrecostos y retrasos en el cronograma⁸.

Más tarde en el año 2010 también se identificaron claramente dificultades en la completitud de la documentación de los sistemas de información desarrollados por esta área⁹. Este mismo año se adquirió el software Enterprise Architect con el fin de mejorar el procedimiento de documentación de sistemas de información y se recibió capacitación en su uso donde se dio a conocer la metodología ICONIX¹⁰.

⁸ En un informe seguimiento se encontraron diferencias en costo del proyecto frente a lo presupuestado y lo efectivamente ejecutado equivalente sobrecosto del 38%. Dicho documento también evidencia un retraso de 45 días en el cronograma propuesto.

⁹ En el año 2010 se realizó un control de documentación de sistemas de información en donde se evidencia que de 28 proyectos a los que se le realizó el seguimiento un 32% no tenían ningún tipo de documentación oficial y solo la mitad cumplían con al menos el 60% de todos los ítems definidos en el instructivo.

¹⁰ Es una metodología abierta basada en UML con el fin de desarrollar el análisis y el diseño que incluye un análisis de robustez en la mitad que conecta ambos, el llamado proceso ICONIX es descrito en el libro UML Theory and Practice.

Esto llevó a que el líder de sistemas, su coordinador de desarrollo y a un grupo de ingenieros a poner en consideración el apropiarse de otra metodología o mejorar la existente.

Iniciar un proceso de mejoramiento no es algo que deba ser tomado a la ligera. Este toma tiempo y recursos importantes, requiere de la participación de todos los involucrados; desde el usuario final hasta la alta gerencia se ven de una u otra manera afectados por una iniciativa como esta (Pino, Vidal, Garcia, & Piattini, 2007) (Leticia, Astorga Vargas, Olguín Espinoza, & Andrade Peralta, 2008).

Es por esto que es importante conocer el proceso a mejorar, medirlo y reconocer en proporción sus debilidades y fortalezas.

Estos argumentos son los que permiten finalmente tomar una decisión soportada por las evidencias más que por la intuición y permite de forma clara establecer el valor real del estado actual del proceso. Un proceso de mejora sustentado en la plataforma de un conocimiento verídico de nuestra condición actual tendrá un fundamento claro y una mejor acogida entre los involucrados.

Al poder validar qué está funcionando bien y qué no, se podrá evitar problemas como los que descubrieron Pino y Vidal en su estudio (Pino, Vidal, Garcia, & Piattini, 2007) al encontrar que en algunos casos “los desarrolladores tienen la convicción de que tal y como están haciendo las cosas está bien”.

Esto es parte de lo que sucede en el área de desarrollo de software Comfamiliar Risaralda, donde la tradición ha impuesto una forma de trabajo que es considerada acertada por el personal operativo, pero que empieza a plantear dudas por su líder y coordinador. Algunos de los problemas que evidencia la jefatura de sistemas son la dificultad en el mantenimiento de software ya que solo los desarrolladores familiarizados con el producto inicial están en capacidad de mantener el código eficientemente y la dificultad para cumplir con los cronogramas debido a que se encuentran dificultades para estimar, para tratar eventos y proyectos imprevistos.

Basado en esta problemática, este proyecto de aplicación busca establecer un proceso estructurado de software adecuado para el área de desarrollo de software de Comfamiliar Risaralda.

La idea general de construcción del proceso es tomado de la experiencia de (Ambler & Lines, 2012) condesada en el libro *Disciplined Agile Development* donde se describe un marco de trabajo híbrido basado en Scrum, Programación Extrema (XP), Agile Modeling (AM), el Proceso Unificado (UP), Agil Data(AD) y Kanban. También se tomaron elementos de diversos estudios que comparaban CMMI con metodologías ágiles (los cuales se presentaron en los antecedentes), y las consideraciones que se deben tener en entornos de pequeños equipos de desarrollo como es el caso de ADSCR.

Con respecto a los grupos de investigación de la UAM este proyecto aplica la herramienta de evaluación de CMMI en el cual ha trabajado el Grupo de Ingeniería de Software, en la línea de calidad y métricas de software. Los resultados obtenidos de estas evaluaciones servirán para la actualización y mejora del instrumento.

Se han seleccionado metodologías de desarrollo ágil por su habilidad de responder a ambientes cambiantes, con requisitos de entrega rápida, con equipos que utilizan nuevas tecnologías recorriendo caminos desconocidos anteriormente (Highsmith, 2002), ambiente que actualmente se vive en Comfamiliar Risaralda.

Se ha seleccionado SCRUM dado que aporta elementos de gestión de proyectos desde revisiones diarias, por iteración, hasta la revisión general del proceso. Además, empresas que han usado SCRUM han encontrado una reducción de defectos de hasta el 40% (Kulpa & Johnson, 2008, pág. 343)

Sin embargo como Scrum por si solo puede no proveer todas las cosas que se necesitan hacer en un proyecto (Kulpa & Johnson, 2008, pág. 352) se ha decidido incluir prácticas específicas de otras metodologías como XP para el personal de desarrollo que ayudan a mejorar la calidad del software desde la construcción del mismo (refactorización), así como prácticas que facilitan el mantenimiento del código (Estándares de codificación, propiedad colectiva de código). (Ambler & Lines, 2012)

Iconix por su parte aporta un nivel de formalismo y al mismo tiempo permite unir las fases de análisis y desarrollo guiando a los programadores en el descubrimiento de las necesidades del negocio y puede ser fácilmente adaptado a cualquier ciclo de vida. (Stephens & Rosenberg, 2007)

1.4. OBJETIVOS

1.4.1. Objetivo general.

Establecer un proceso estructurado de software (PES) para el área de desarrollo de software de Comfamiliar Risaralda, basado en el análisis de CMMI, buenas prácticas y métodos ágiles de ingeniería de software.

1.4.2. Objetivos Específicos.

- Evaluar el proceso de desarrollo de software actual utilizado en el área de sistemas de Comfamiliar Risaralda mediante el instrumento CMMI de la UAM.
- Comparar el proceso actual con Scrum, XP e ICONIX usando como referencia las prácticas incluidas en CMMI.
- Definir y documentar un proceso estructurado de software para la empresa basado en las mejores prácticas de Scrum, XP e ICONIX.
- Validar el proceso estructurado de software propuesto mediante una prueba de concepto en un proyecto piloto dentro de Comfamiliar Risaralda.

1.5. ALCANCE

La implementación final de este proceso en Comfamiliar Risaralda no se encuentra dentro del alcance de este proyecto y dependerá de variables y decisiones del personal directivo del área de TI.

2. REFERENTE TEÓRICO

Los modelos de proceso para el desarrollo de software vienen evolucionando desde varias décadas atrás. Cada par de años estos modelos son revisados y los investigadores desarrollan y prueban nuevas prácticas que prometen resolver los problemas de la industria. (Pressman, 2010)

2.1. DEFINICIONES

2.1.1. Proceso de desarrollo de software

“Se define como un marco de trabajo para las tareas que se requieren en la construcción de software de alta calidad” (Pressman, 2010)

“Un conjunto de pautas y un conjunto de actividades de trabajo parcialmente ordenadas pensadas para producir software de manera controlada, reproducible. El propósito de un proceso de desarrollo de software es asegurar el éxito y la calidad de un sistema acabado.” (Rumbaugh, Jacobson, & Booch, 2000, pág. 415)

“Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software” (Jacobson & Booch Grady, 2000, pág. 4)

“Un proceso de desarrollo de software guía los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto.” (Jacobson & Booch Grady, 2000, pág. 13)

“Un proceso de ingeniería de software es una definición del conjunto completo de actividades necesarias para transformar los requisitos de usuario en un producto. Un proceso es una plantilla para crear proyectos.” (Jacobson & Booch Grady, 2000, pág. 13)

En general los procesos de ingeniería de software nos ayudan a mejorar el desarrollo, proponiéndonos una ruta a seguir y encaminando los esfuerzos con el fin de obtener un resultado de calidad.

2.1.2. Métodos de Ingeniería de Software

“Los métodos de la ingeniería de software proporcionan los “cómo” técnicos para construir software. Los métodos abarcan un amplio espectro de tareas que incluyen la comunicación, el análisis de requisitos, el modelado del diseño, la construcción del programa, la realización de pruebas y el soporte. Los métodos de la ingeniería del software se basan en un conjunto de principios básicos que gobiernan cada área de la tecnología e incluye actividades de modelado y otras técnicas descriptivas.” (Pressman, 2010)

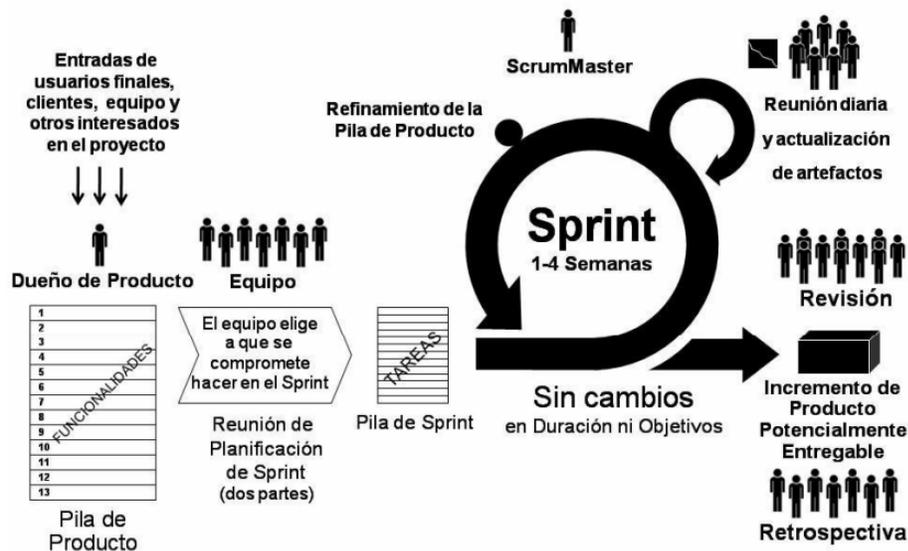
2.2. METODOLOGÍAS DE DESARROLLO DE SOFTWARE

A partir de este punto se resumirán los principales elementos de los marcos de trabajo y las metodologías de desarrollo de software relevantes para el presente proyecto, iniciando con Scrum, luego ICONIX y finalmente Programación Extrema (XP).

2.2.1. Scrum

Es un marco de trabajo para desarrollo de software y de proyectos. En 1993 fue creado el primer equipo de Scrum por Jeff Sutherland y el marco de trabajo fue definido formalmente en 1995 por Ken Schwaber. Hoy en día es usado por empresas como Google, Microsoft, Yahoo!, Cisco entre otras. (Deemer, Benefield, Larman, & Vodde, 2009)

Figura 1. Resumen elementos de Scrum



Fuente: *The Scrum Primer* (Deemer, Benefield, Larman, & Vodde, 2009, pág. 5)

Scrum se basa en el modelo iterativo e incremental, cada ciclo de trabajo o iteración es llamado Sprint, estos Sprint pueden durar entre 1 a 4 semanas, son fijos y no se extienden incluso cuando no se ha terminado todo el trabajo que estaba planeado.

El dueño del producto identifica y prioriza las funcionalidades que se desea del software. El equipo (interdisciplinario) se compone de todas las personas con las habilidades que se requieren para completar el desarrollo. El equipo es acompañado por una persona que los acompaña y asesora en el uso de Scrum (ScrumMaster).

Al inicio de cada Sprint el equipo selecciona las funcionalidades de la pila de producto que se incluirán en este ciclo (Reunión de Planificación de Sprint). Una vez inicia el Sprint no se pueden cambiar las funcionalidades seleccionadas (Pila de Sprint). Cada día se realiza una reunión muy corta (Reunión diaria) donde se informa lo que se hizo el día anterior, lo

que se está planeado hacer el día presente y qué inconvenientes se han presentado. Al terminar el Sprint se presenta el producto obtenido a los interesados recibiendo retroalimentación de los mismos para mejoras o ajustes en los próximos ciclos (Revisión). El producto resultante es un software usable, que ha sido probado y podría empezar a usarse inmediatamente. El ciclo se cierra con una reunión donde se identifica qué fue mal y qué podría mejorar en el proceso (Retrospectiva).

Scrum puede dividirse en 3 fases: el *pre-game* donde se realiza la planeación y la arquitectura; *el game* donde se realiza el desarrollo mediante los Sprints y el *post-game* o fase de cierre que incluye temas como integración, pruebas, entrenamiento y material de mercadeo.

Scrum se compone principalmente de 3 roles, 4 reuniones y 3 artefactos.

Roles

- Dueño del producto
- Equipo
- *ScrumMaster*

Reuniones

- Planificación del Sprint
- Reunión Diaria
- Revisión del Sprint
- Retrospectiva del Sprint

Artefactos

- Pila del Producto
- Pila del Sprint
- Gráfica de trabajo restante

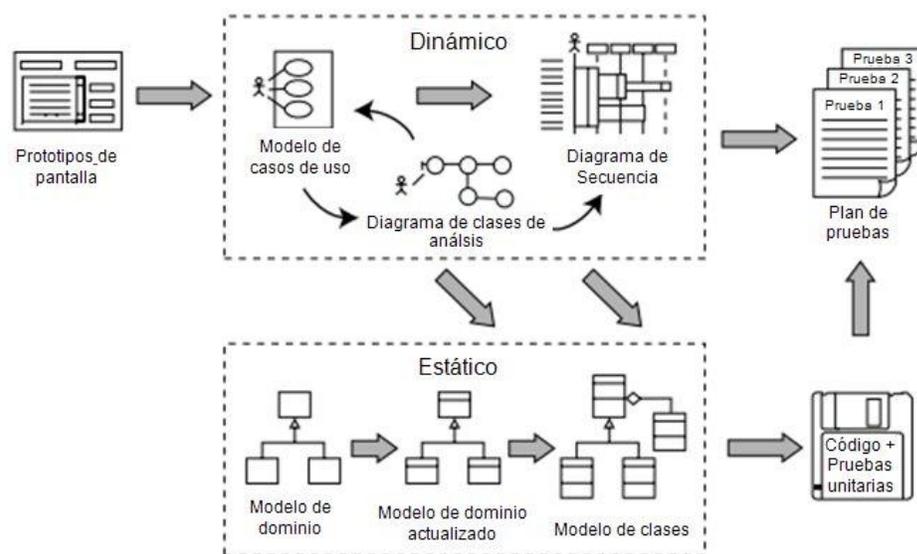
2.2.2. ICONIX

Matt y Doug en su libro *Use Case Driven Object Modeling with UML* (Stephens & Rosenberg, 2007), definen el que ellos llama “El Proceso ICONIX” como un proceso

“cookbook” en la que describen en una serie específica de pasos que ellos han encontrado que funciona muy bien en diferentes proyectos. ICONIX no prescribe ningún ciclo de vida como lo hacen otras metodologías, lo que permite según él ser usado de manera ágil o de la manera tradicional. Por esta razón ICONIX puede ajustarse a cualquier ciclo de vida (Stephens & Rosenberg, 2007).

ICONIX utiliza un enfoque dirigido por casos de uso, se enfoca en las etapas de análisis y diseño y define un grupo concreto de diagramas UML para formalizarlas. Esta dividido en dos flujos de trabajo uno dinámico y otro estático.

Figura 2. Resumen del proceso ICONIX.



Fuente: Del original en inglés (Stephens & Rosenberg, 2007, pág. 1)

Puede describirse en 4 fases, 3 hitos y 4 artefactos:

- Fase 1. Requerimientos. Artefactos: modelo del dominio, modelo de casos de uso.
- Hito 1. Revisión de requerimientos
- Fase 2. Análisis / Diseño preliminar. Artefacto: Diagrama de clases de análisis.
- Hito 2. Revisión del diseño preliminar

- Fase 3. Diseño Detallado. Artefacto: Diagrama de secuencia.
- Hito3. Revisión crítica de diseño
- Implementación

En (Stephens & Rosenberg, 2007, pág. 330) se incluye un capítulo sobre la realización de pruebas mediante *Design-Driven Testing (DDT)* sin embargo este elemento de pruebas no se considera parte del proceso ICONIX.

2.2.3. PROGRAMACIÓN EXTREMA (XP)

Kent Beck (2004) la describe como una forma de desarrollar software que es liviana, eficiente, de bajo riesgo, flexible, predecible, científica y divertida. Como una disciplina de software, la describe de esta manera porque si se evita realizar alguno de sus componentes simplemente no se está siendo extremo. Está diseñado para equipos pequeños de 2 a 10 programadores.

Según Wake (2002) XP es un enfoque liviano de desarrollo de software que cubre desde el levantamiento de requerimientos hasta la entrega final. XP usa un contacto más cercano con el cliente mejorando la comunicación y la retroalimentación, tiene un enfoque de planeación particular y pruebas continuas.

XP es descrito en términos de 12 prácticas (Beck & Andres, 2004), estas prácticas puede agruparse en tres capas:

Programación: Diseño simple, pruebas, refactorización, estándares de codificación.

Prácticas de equipo: Propiedad colectiva, integración continua, metáfora, estándares de codificación, 40 horas por semana, programación en parejas, entregas pequeñas.

Proceso: Cliente en casa, pruebas, entregas pequeñas, juego de planeación.

Figura 3. XP en capas



Fuente: Del original en inglés (Wake, 2002, pág. 4)

XP como estilo de programación:

Los programadores XP escriben código de manera incremental, siempre escribiendo primero los casos de prueba y luego el código que los resuelve. Todo código debe corresponder a una prueba. El código se revisa periódicamente para mejorarlo (refactorización) y se verifica para que siga cumpliendo con las pruebas.

XP como práctica de equipo:

XP promueve el trabajo en equipo y la igualdad y cooperación entre sus integrantes. La propiedad compartida del código permite a cualquier persona del equipo modificar código, mejorarlo o extenderlo, para facilitar esta tarea los programadores cumplen con un estándar de codificación que al mismo tiempo mejorará el mantenimiento del código en el futuro. La integración continua provee un método para que todos los integrantes siempre trabajen en la última versión del código de manera que puedan aprovechar los nuevos desarrollos y garantizar que las mejoras y adiciones aún cumplen con las pruebas.

Una de las prácticas más características y controversiales es la programación en pareja, esta permite el diseño y revisión del código al mismo tiempo que se va generando. Es un mecanismo para asegurar la calidad del software.

La arquitectura del sistema en XP es llamada metáfora, la metáfora ayuda a todas las personas involucradas en el proyecto a entender los elementos básicos y sus relaciones. Es una historia coherente sobre la cual trabajar y que puede ser fácilmente comunicada a los demás.

Las 40 horas de trabajo se refiere a comprometerse a no trabajar horas extras ya que es una práctica que no puede sostenerse efectivamente en el tiempo, disminuye la moral del equipo y generalmente es síntoma de mayores y profundos problemas en el proyecto, en cambio se promueve la claridad mental y física mediante un descanso justo entre cada jornada de trabajo.

XP como un proceso:

XP es iterativo y promueve la realización de entregas pequeñas que entreguen valor al cliente, para asegurar esto es un requisito que el cliente sea parte del equipo, este se dedicará de tiempo completo a crear las historias sobre las cuales deben trabajar los programadores, resolver dudas y diseñar las pruebas. Si el cliente no está completamente dedicado y cerca al equipo de programadores, la velocidad del desarrollo será más lenta.

Las pruebas diseñadas por el cliente se convierten en pruebas unitarias que los programadores automatizan y con las que pueden garantizar que cada funcionalidad puede ser probada y responde a las necesidades del cliente.

El juego de planeación permite al cliente y a los programadores negociar y comprometerse a realizar entregas periódicas según las necesidades del cliente. En general el cliente escribe historias y los programadores estiman el costo de realizarlas, finalmente el cliente puede decidir qué historias quiere que se desarrollen primero.

2.3. MODELOS DE REFERENCIA

A continuación se explica en forma general el Modelo de Capacidad y Madurez CMMI, este modelo servirá de base de evaluación y comparación de los diferentes procesos y metodologías que se tratarán en este proyecto.

2.3.1. CMMI (Capability Maturity Model Integration)

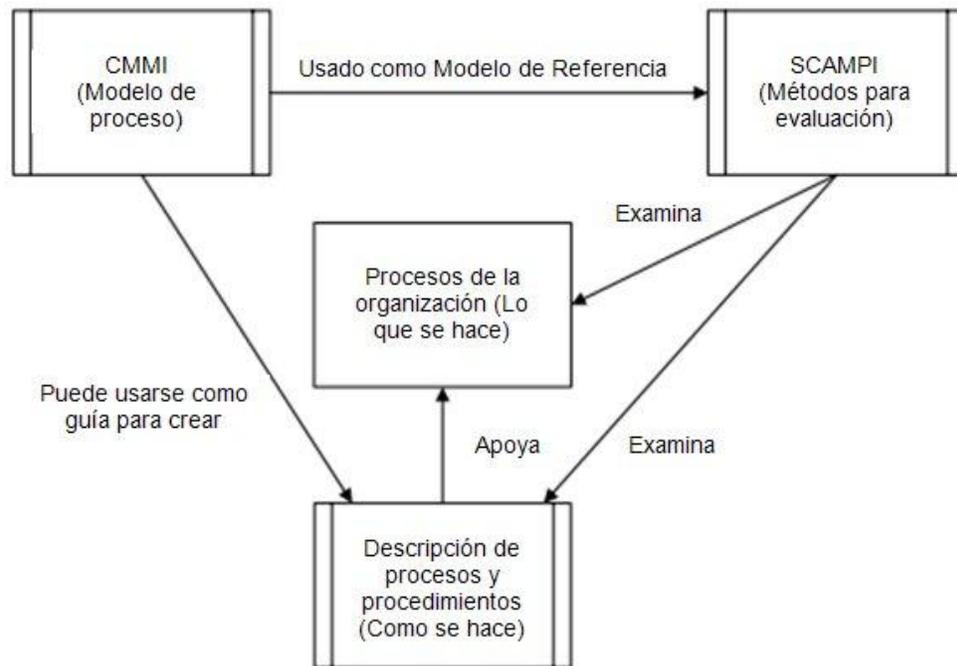
Según (Kulpa & Johnson, 2008, pág. 3) CMMI es la unión de modelos de proceso de mejoramiento para Ingeniería de sistemas, software, hardware y equipos integrados.

CMMI es una colección de mejores prácticas que ayudan a optimizar los procesos de la organización, describiendo cuales actividades deberían realizarse a su interior.

También existen un conjunto de métodos para evaluar los procesos y procedimientos de la organización llamada SCAMPI (Estándar CMMI Appraisal Method for Process Improvement), la idea es evaluar cómo fueron implementados estos procesos y procedimientos, qué tanto se usan en la organización y qué tanto se acercan a las prácticas de CMMI.

CMMI es usado como guía para crear procesos y procedimientos y como modelo de referencia para los metodos de evaluacion SCAMPI. (Kulpa & Johnson, 2008, pág. 7) Ver Figura 4.

Figura 4. CMMI, SCAMPI y la organización



Fuente: Del original en inglés (Kulpa & Johnson, 2008, pág. 8)

CMMI está compuesto por un marco de referencia, unos modelos, métodos de evaluación y cursos de entrenamiento.

A partir de la versión 1.2 de CMMI se incluyó el concepto de constelación, las constelaciones son diferentes tipos de CMMI que se enfocan en diferentes temas y cada una incluye además del modelo otros componentes como son el material de entrenamiento y de evaluación. (Kulpa & Johnson, 2008, pág. 39)

Existen actualmente 3 constelaciones en la familia de CMMI:

1. CMMI para Desarrollo
2. CMMI para Adquisición
3. CMMI para Servicios

Figura 5. Constelaciones CMMI



De acuerdo con Phillips y Shrum (2011) parte de cada uno de estos modelos son idénticos con los otros modelos debido a que son prácticas que aplican a cualquier negocio, sin embargo cada modelo también tiene prácticas únicas debido a sus diferentes enfoques:

CMMI para Desarrollo está diseñado para negocios que se enfocan en desarrollar productos y servicios, este modelo explora de forma detallada la conversión de requerimientos del cliente en requerimientos que son usados por desarrolladores, integrando efectivamente componentes de productos en un producto o servicio final, realizando trabajo de análisis y desarrollo para diseñar un producto o servicio y asegurando que el trabajo de desarrollo cumpla con las necesidades del usuario final y de la especificación formulada durante el diseño. El presente proyecto se centrará en el modelo definido en esta constelación CMMI-DEV (CMMI Product Team, 2010).

CMMI para Adquisición está diseñado para negocios que se enfocan en trabajar con proveedores para ensamblar productos o entregar servicios.

CMMI para Servicios está diseñado para negocios enfocados en establecer administrar y entregar servicios.

Existen 2 representaciones de CMMI la continua y la escalonada; en la primera la organización mejora en uno o varias áreas de proceso que tienen relación entre si y van alcanzando un nivel de capacidad a la vez. En la representación escalonada toda la organización realiza la mejora en todas las áreas de proceso hasta alcanzar cierto nivel de madurez. En ambos casos CMMI cuenta con áreas de proceso así como metas y prácticas específicas.

Seguidamente se describe SPEM el cual se usará con el fin de modelar de forma precisa el proceso resultante de este proyecto

2.3.2. SPEM (Software Process Engineering Meta-model)

El Object Management Group (OMG) ha creado un meta-modelo llamado SPEM, el cual permite modelar, documentar, presentar y gestionar métodos y procesos de desarrollo.

Mediante una herramienta que se ajuste a la especificación SPEM, es posible definir un modelo de proceso mediante un lenguaje de modelamiento no ambiguo usando generalmente tres elementos básicos: rol, producto de trabajo y tarea.

La idea central de SPEM para representar procesos está basada en tres elementos básicos: rol, producto de trabajo y tarea. Las tareas representan el esfuerzo a hacer, los roles representan quien lo hace y los productos de trabajo representan las entradas que se utilizan en las tareas y las salidas que se producen. La idea central subyacente es que un modelo de proceso consiste, básicamente, en decir quién (rol) realiza qué (tarea) para, a partir de unas entradas (productos de trabajo) obtener unas salidas (productos de trabajo).

SPEM define 4 elementos básicos usados para estructurar y administrar información (Sibbald, 2006):

- Librería de Métodos: Es una colección de *Plug-ins* de Métodos y de Configuraciones de Métodos.
- Plug-in de Métodos: Es un contenedor para contenido que puede ser importado o exportado de manera independiente. Puede componerse de Paquetes de métodos y Paquetes de proceso.
- Configuraciones de Métodos: Define una sub-categoría de la librería de métodos que pueden ser publicados o exportados. Puede entenderse como un “filtro” que se aplica a la librería con el fin de limitar los elementos necesarios para un propósito específico.
- Proceso de entrega (*Delivery Process*): Es un proceso que se crea para un tipo de proyecto en particular para reflejar una forma específica de planeación y gestión.

SPEM divide claramente entre Contenido del Método y Proceso.

El Contenido del Método: Está compuesto de material altamente reutilizable, incluye los “que”, “quien”, “como” y “porque”. Define los roles, tareas, productos de trabajo y sus relaciones, incluye además guías y categorías.

El proceso: Usa el contenido reutilizable para crear procesos completos o componentes de procesos que a su vez pueden ser reutilizados. Incluye secuencia de fases, iteraciones, actividades e hitos que definen el ciclo de vida de desarrollo. Define cuanto se desarrollan

las tareas mediante diagramas de actividad o estructuras de descomposición del trabajo (EDT) (OMG, 2008).

Figura 6. Diferencia Método y Proceso SPEM.



Fuente: Del original en inglés (Sibbald, 2006, pág. 63)

2.4. COMPATIBILIDAD ENTRE CMMI Y ÁGIL

Antes de comenzar con el análisis de las posibles similitudes y diferencias y la posibilidad de integrar CMMI y algunas prácticas ágiles, debe contemplarse la pregunta si estas

visiones con compatibles o en cambio entran en conflicto impidiendo su coexistencia, este tema ha sido abordado en varias ocasiones por diferentes autores, a continuación se presenta un resumen de dichos puntos de vista.

En 2001 Mark C. Paulk explica su visión de XP y el CMM¹¹ en (Paulk, 2001) después de analizar las características primordiales tanto de XP como CMM Paulk presenta una tabla (ver Fig. 7) de cumplimiento de cada área de proceso encontrando un cubrimiento aproximado del 61%¹².

¹¹ En el 2006 se desarrolló CMMI con el fin de integrar y estandarizar los modelos que se encontraban separados en el Modelo de Capacidad y Madurez (CMM).

¹² Este porcentaje es inferido del número de áreas de proceso evaluadas en contraste con el número de áreas de proceso que muestran un nivel de satisfacción.

Figura 7. Nivel de satisfacción de XP a las áreas de proceso de CMM.

Satisfacción de XP a cada área de proceso, dado el ambiente apropiado

Nivel	Área de proceso clave	Satisfacción
2	Administración de requerimientos	++
2	Planeación del proyecto de software	++
2	Seguimiento y supervisión del proyecto	++
2	Administración de subcontratistas de software	—
2	Aseguramiento de la calidad del software	+
2	Administración de la configuración del software	+
3	Enfoque del proceso de la organización	+
3	Definición del proceso de la organización	+
3	Programa de capacitación	—
3	Administración de la integración del software	—
3	Ingeniería del producto de software	++
3	Coordinación intergrupala	++
3	Revisión de pares	++
4	Administración cuantitativa de los procesos	—
4	Administración de la calidad del software	—
5	Prevención de defectos	+
5	Administración de cambios de tecnología	—
5	Administración de cambio de proceso	—

* Abordado parcialmente en XP
 ++ Abordado ampliamente en XP (posiblemente por inferencia)
 — No Abordado en XP

Fuente: Del original en inglés (Paulk, 2001, pág. 6)

Aunque cuestiona el nivel de institucionalización de XP se puede observar un cubrimiento del 67% de satisfacción de las prácticas relacionadas con este tema (ver Figura 8). Paulk finalmente concluye en su artículo que considera XP y CMM complementarios. Aclarando que para proyectos donde dependa la vida o sistemas de alto desempeño tal vez no se deba usar XP debido al riesgo que represente la falta de diseño, documentación y falta de énfasis en la arquitectura.

Figura 8. XP y las prácticas de institucionalización

XP y las prácticas de institucionalización

Práctica común (en cada KPA)	Práctica	Satisfacción
Compromisos	Políticas	—
	Liderazgo y acompañamiento	—
Habilidades necesarias	Estructura organizacional	+
	Recursos y financiación	+
	Entrenamiento	+
Medición v análisis	Medición	+
Verificaciones de implementación	Supervisión gerencial	—
	Supervisión de la gestión de proyectos	++
	Aseguramiento de la calidad del software	+

+ Abordado parcialmente en XP
 ++ Abordado ampliamente en XP (posiblemente por inferencia)
 — No Abordado en XP

Fuente: Del original en inglés (Paulk, 2001, pág. 7)

En 2007 otro estudio (Fritzsche & Keil, 2007) realizó un análisis similar, esta vez teniendo en cuenta además Scrum y utilizando un sistema de calificación similar, agregando un nivel intermedio llamado “soportado” y agregando el ítem “en conflicto”.

Tabla 1. Comparación de calificaciones usadas por Paulk y Fritzsche & Keil

(Paulk, 2001)	(Fritzsche & Keil, 2007)
(-) No abordado	(-) En conflicto
	(0) No abordado
(+) Parcialmente soportado	(+) Parcialmente soportado
(++) Muy soportado	(++) Soportado
	(+++) Muy soportado

El análisis se realiza de manera independiente para XP y Scrum, este se realiza a nivel de la meta específica asociada a cada área de proceso. Las conclusiones son muy similares al

estudio anterior, el cubrimiento obtenido ahora es de un 64%. Los autores coinciden en que en niveles 2 y 3 puede lograrse un alcance de CMMI satisfactorio pero en niveles superiores el cubrimiento es muy pobre.

Un año más tarde en el libro *Interpreting CMMI* (Kulpa & Johnson, 2008) describen la experiencia en empresas que aplican Scrum y CMMI dando como resultado un incremento en el doble de la productividad y una reducción de defectos del 40%, en una empresa con certificación CMMI nivel 5 (Kulpa & Johnson, 2008, pág. 343). Los autores consideran que en empresas que usan prácticas ágiles CMMI puede ayudar a institucionalizar dichas prácticas (Kulpa & Johnson, 2008, pág. 347). Para este análisis usan 12 prácticas genéricas de los niveles 2 y 3, a diferencia de los estudios revisados anteriormente que se apoyaban en las prácticas específicas o las metas específicas.

Cabe resaltar que los autores consideran que la evaluación SCAMPI no puede ser la mejor opción cuando se trata de evaluar una implementación de tipo ágil, debido a la ausencia de algunas evidencias o actividades obligatorias (Kulpa & Johnson, 2008, pág. 351).

Finalmente en (Gandomani & Zulzalil, 2013) basado en trabajos de compatibilidad desde el 2002 hasta el 2010 realizan un estudio estadístico basado en encuestas a expertos con experiencia tanto en CMMI y Ágil. Con respecto a las prácticas específicas este estudio encontró que los expertos consideraban conflictivas las áreas de proceso DAR y OPF de nivel 3, y OPP de nivel 4. Esta vista reciente de la concepción de los métodos ágiles y CMMI encuentra una mayor compatibilidad entre ambos dando como resultado solo tres áreas de proceso en conflicto que representan un 14%, de esta manera se podría asumir un nivel de compatibilidad del 84%, una visión que ha mejorado con el tiempo en comparación del 61% propuesto inicialmente por (Paulk, 2001).

Estos estudios demuestran que en niveles 2 y 3 CMMI y los enfoques ágiles pueden ser compatibles y complementarios, aprovechando las prácticas ágiles para mejorar la comunicación y la cercanía con el cliente entre otros aspectos y en niveles mayores CMMI ayuda a institucionalizar dichas prácticas y a mantener estadísticas que soportan los procesos de mejora.

2.5. MATRICES DE COMPARACIÓN

En Brasil se realizó un mapeo de CMMI y Scrum (Marcal, de Freitas, Furtado Soares, & Belchior, 2007), fue realizada sobre la versión 1.2 y se encontraba enfocado en áreas relacionadas con la gestión del proceso que incluyen 3 áreas de nivel 2, dos de nivel 3 y QPM en el nivel 4.

La evaluación en este proyecto se realiza a nivel de cada práctica específica, revisando en forma general el propósito de la práctica y asignándole una de tres posibles calificaciones: insatisfecha (U), Parcialmente satisfecha (PS) y Satisfecha (S).

En este estudio encontraron prácticas ágiles que cumplían en algún porcentaje con todas las áreas analizadas menos una QPM en nivel 4 y en un bajo porcentaje (~14%) RSKM, las demás áreas estudiadas tienen un nivel de satisfacción superior al ~57%.

En una revisión detallada algunas de estas prácticas no satisfechas podrían ser motivo de controversia debido a las diferentes interpretaciones que pueden existir acerca de las prácticas en Scrum.

En el 2009 Diaz y Garbajosa (Díaz Fernandez, 2009) (Diaz, Garbajosa, & Calvo-Manzano, 2009) mapearon las prácticas de XP y Scrum con las tres primeras áreas del nivel 2 de CMMI: Administración de Requerimientos (REQM), Planificación de Proyectos (PP) y Monitoreo y Control de Proyectos (PMC), el estudio no es de gran cobertura en las áreas de CMMI pero realiza un análisis detallado de cada subpráctica con el fin de evaluar cada práctica específica con tres calificaciones posibles: alternativa, soportada y plenamente soportada.

Tabla 2. Calificaciones usadas por Diaz y Garbajosa

Alternativa	La SP no es soportada o conducida a través de prácticas ágiles, y una práctica alternativa es requerida. Todas las subprácticas, o la mayor parte, son insatisfactorias (no soportadas a través de prácticas ágiles).
Soportada	La SP es soportada o conducida a través de prácticas ágiles. Sin embargo, una o más subprácticas requieren de mayor interpretación, o una o más subprácticas son insatisfactorias (no

soportadas a través de prácticas ágiles).	
Plenamente soportada	La SP es plenamente soportada o conducida a través de prácticas ágiles. Todas las subprácticas son satisfechas

Fuente: (Díaz Fernandez, 2009, pág. 55)

Cabe comentar que aunque se contempla la posibilidad de una calificación “Alternativa” para el caso de que la SP no esté soportada por prácticas ágiles, dentro del alcance del estudio no se encontró ninguna práctica con esta calificación, lo que sugiere que para estas primeras 3 prácticas seleccionadas se considera que pueden cubrirse en su totalidad mediante prácticas ágiles. Esto contrasta un poco con el estudio anterior de (Marcal, de Freitas, Furtado Soares, & Belchior, 2007) en el cual se encontraron al menos algunas prácticas no soportadas, esto podría deberse a lo estricto del estudio del 2007 donde se buscaba una mención explícita de cómo cumplir la práctica en contraste con el enfoque de reinterpretar o utilizar artefactos sugeridos en Scrum mencionado en el estudio de (Díaz Fernandez, 2009)

En 2012 (Lukasiewicz & Miler) se realizó el estudio sobre un cubrimiento mayor, solo excluyendo los niveles 4, 5 y las áreas de nivel 3 Enfoque a Procesos Organizacionales (OPF), Definición de Procesos Organizacionales (OPD) y Entrenamiento Organizacional (OT).

Al momento de este estudio no se pudo obtener información más detallada sobre el trabajo de comparación de las prácticas o su método de evaluación. Según los autores en su artículo se usó un modelo llamado “modelo C-S” donde solo resume la relación entre las prácticas de CMMI, Scrum y las prácticas adicionales necesarias, pero no contempla un método para poder extender este modelo a las demás áreas o relacionarlo con otras prácticas ágiles diferentes de Scrum.

(Jan & Javed, 2013) realizan un mapeo entre las prácticas específicas de CMMI (8 áreas de proceso, seis de ellas de nivel 2 y otras dos de nivel 3) agregando a los estudios anteriores de Diaz y Garbajosas 3 áreas de nivel 2 Medición y Análisis (MA), Aseguramiento de la Calidad del Proceso y el Producto (PPQA) y Administración de Configuraciones (CM) al igual que Verificación (VER) y Administración de Riesgos (RSKM) de nivel 3, pero excluyen áreas ya evaluadas en el estudio previo de (Lukasiewicz & Miler, 2012) como REQD, TS, PI, IPM y DAR de nivel 3 sin embargo no encuentran prácticas ágiles ya existentes para estas nuevas áreas incluidas por lo que deben definir nuevas prácticas que

cubran los vacíos encontrados. Lo que contrasta con Lukasiewicz & Miler que si encontraron prácticas en Scrum al menos en las áreas CM, VER y RSKM, resaltando la diferencia encontrada entre los autores en el área de Verificación (VER) en el que mientras Lukasiewicz & Miler encuentra un cubrimiento completo de las prácticas de Scrum, un año más tarde Jan & Javed en su estudio, no encuentran ninguna práctica ágil por lo que incluyen nuevos artefactos, sin embargo esta diferencia puede deberse a diferentes interpretaciones de Scrum, en (Deemer, Benefield, Larman, & Vodde, 2009, pág. 14) se observa como en la Revisión del Sprint se busca inspeccionar y adaptar el producto, esta revisión se hace con el dueño del producto de manera que es una práctica importante para verificar que el producto cumple con los requerimientos del cliente. De igual manera ninguno de estos dos últimos estudios encuentran prácticas ágiles para medición y análisis (MA) es posible suponer al menos un cubrimiento parcial gracias a la pila de Entrega y la gráfica de trabajo restante explicada en (Deemer, Benefield, Larman, & Vodde, 2009, págs. 15-16)

En la Tabla 3 se pueden observar en paralelo los estudios mencionados anteriormente en lo referente al porcentaje de cubrimiento que puede deducirse a partir de los hallazgos en cada una de las áreas de proceso.

Tabla 3. Comparación del porcentaje de cubrimiento de estudios previos

Metodología		XP/SCRUM	SCRUM	XP/SCRUM	SCRUM
Año		2013	2012	2009	2007
Investigación		SCXTERME	LUKA	Díaz	Marcal
Nivel	Área de Proc.	% Cubrimiento	% Cubrimiento	% Cubrimiento	% Cubrimiento
2	1. REQM	80%	100%	100%	
2	2. PP	57%	86%	100%	79%
2	3. PMC	80%	90%	100%	90%
2	4. SAM		0%		57%
2	5. MA	0%	0%		
2	6. PPQA	0%	0%		

2	7. CM	0%	86%	
3	8. REQD		100%	
3	9. TS		63%	
3	10. PI		100%	
3	11. VER	0%	100%	
3	12. VAL		100%	
3	13. OPF			
3	14. OPD			
3	15. OT			
3	16. IPM		57%	57%
3	17. RSKM	0%	29%	14%
3	18. DAR		0%	
4	19. OPP			
4	20. QPM			0%
5	21. OPM			
5	22. CAR			

Fuente: Elaboración propia.

2.6. FRAMEWORKS MIXTOS

(Lukasiewicz & Miler, 2012) presentaron un método para combinar SCRUM y CMMI para los niveles 2 y 3, Para esto usaron un modelo de referencia CMMI-Scrum donde se

relacionan las prácticas de ambos, usaron un cuestionario para identificar las necesidades de mejoramiento del proceso y diseñaron un algoritmo para sugerir las prácticas que podrían aplicarse en el proceso de mejora. Finalmente basado en este algoritmo desarrollaron el software MatureScrum que implementa el cuestionario y algoritmo de selección.

Dado que Scrum no cubre completamente todas las prácticas de CMMI propuestas, algunas prácticas nuevas deben introducirse con el fin de cubrir estos vacíos, no se aclara como se obtuvieron o desarrollaron estas prácticas. El estudio de casos aquí se vio limitado a realizar el cuestionario y las sugerencias pero las prácticas no pudieron ser implementadas en las empresas evaluadas.

(Jan & Javed, 2013) Proponen un marco de trabajo llamado SCXTREME, este busca mejorar las deficiencias encontradas en el uso de prácticas ágiles en Pakistán. Los autores construyen SCXTREME partiendo como base de CMMI, mapeando las prácticas específicas con las posibles prácticas ya sea de Scrum o XP, incluso algunas prácticas y artefactos propuestos no son naturales de estas metodologías ágiles y parecen provenir simplemente de las mejores prácticas conocidas en la industria, esto puede observarse por ejemplo en las áreas de proceso Administración de Configuraciones (CM) y Administración de Riesgos (RSKM).

3. APORTE INVESTIGATIVO

El desarrollo de este proyecto espera servir como ejercicio práctico y ejemplo a la comunidad educativa sobre de la elaboración de un estudio de la situación actual de un proceso de desarrollo de software como etapa preliminar de un proceso de mejoramiento.

Supone una base sobre la cual se puedan desarrollar futuros proyectos para el mejoramiento de los procesos de software, que lleven a la definición formal y extensible a otras empresas de un análisis que contribuya a la selección de buenas prácticas para la construcción de procesos estructurados de software.

El instrumento de comparación utilizado para relacionar las prácticas de los diferentes métodos y del proceso CMMI servirá a la comunidad para realizar sus propios análisis con estos y otros métodos y procesos de desarrollo de software.

Los resultados de este proyecto servirán:

- Como Guía de estudio para la implementación de procesos de desarrollo ágil mediante procesos estructurados.
- Como base para nuevos proyectos que definan instrumentos para la comparación y ponderación de prácticas para la definición de procesos híbridos.
- Para que Comfamiliar implemente y ponga en marcha un proceso de mejoramiento usando la evaluación y el proceso planteado como fase inicial, y continúe mejorando y ajustando el proceso estructurado de desarrollo de software propuesto.
- A la comunidad educativa como ejemplo del ajuste de metodologías ágiles en empresas de la región.

4. METODOLOGÍA PROPUESTA

4.1. ENFOQUE

Con el fin de desarrollar los objetivos el proyecto se aplicará una metodología empírica¹³ basada en el método científico donde se propone una mejora y se evalúan los resultados, esta se desarrollará en 5 fases básicas así:

1. Descripción y evaluación del proceso de desarrollo de software actualmente utilizado en el área de sistemas de Comfamiliar Risaralda.
2. Análisis comparativo de las prácticas de SCRUM, XP e ICONIX frente a las prácticas específicas de CMMI-DEV y el proceso actual de desarrollo del área de sistemas de Comfamiliar Risaralda.
3. Elaboración del proceso ajustado para el área de desarrollo de software de Comfamiliar Risaralda.
4. Evaluación del proceso estructurado de software propuesto.
5. Conclusión

Cada fase se conformará de las siguientes actividades, como se detalla a continuación:

1. Descripción y evaluación del proceso de desarrollo de software actualmente utilizado en el área de sistemas de Comfamiliar Risaralda.
 - 1.1. Reunir y organizar la documentación actual del proceso de desarrollo software utilizado en Comfamiliar Risaralda.

¹³ Se usa esta metodología debido a que el estudio será aplicado y se tomará como base una empresa concreta, acercándose más a un estudio de caso que a una investigación aplicada a la comunidad en general.

- 1.2. Realizar la evaluación basado en CMMI con el instrumento de la UAM adaptado a la versión 1.3 al proceso actual de desarrollo de software del ADSCR.
 - 1.3. Identificar fortalezas y debilidades del proceso actual del área de desarrollo de Comfamiliar Risaralda a partir de la evaluación realizada en el paso 1.2.
2. Análisis comparativo de las prácticas de SCRUM, XP e ICONIX frente a las prácticas específicas de CMMI-DEV y el proceso actual de desarrollo de del área de sistemas de Comfamiliar Risaralda.
 - 2.1. Construir un instrumento que permita la comparación entre los modelos y métodos seleccionados y la ponderación de las prácticas identificadas en el paso 2.1 basado en las prácticas sugeridas por CMMI-DEV.
 - 2.2. Incluir en el instrumento de comparación anterior puntos débiles del proceso actual de desarrollo del área de sistemas de Comfamiliar Risaralda.
 - 2.3. Realizar la evaluación total priorizando las prácticas seleccionadas de los modelos.
3. Elaboración del proceso ajustado para el área de desarrollo de software de Comfamiliar Risaralda.
 - 3.1. Basado en la priorización del punto anterior seleccionar las prácticas que debería incluirse en el nuevo proceso con el fin de superar las debilidades.
 - 3.2. Diseñar el nuevo proceso de desarrollo de software aplicando las mejores prácticas encontradas en las metodologías evaluadas incluyendo la definición de roles, artefactos, actividades, guías, su secuencia y relaciones.
 - 3.3. Documentar el nuevo proceso usando la herramienta Eclipse Process Framework (EPF) mediante el lenguaje SPEM.
4. Evaluación del proceso estructurado de software propuesto.
 - 4.1. Aplicar el proceso estructurado de software propuesto en el proyecto piloto “Módulos Legalización y Administración del sistema de Crédito”.
 - 4.2. Realizar la evaluación del PES mediante el instrumento elaborado en el 1 paso.

- 4.3. Analizar el resultado de la evaluación anterior identificando debilidades y fortalezas del nuevo proceso.

5. Conclusión

- 5.1. Elaboración del resumen gerencial dirigido a la coordinación del área de sistemas de Comfamiliar Risaralda
- 5.2. Elaborar las conclusiones y un informe sobre futuros trabajos que resuelvan los puntos no incluidos en el alcance y los elementos que aun puedan ser mejorados en el nuevo PES.
- 5.3. Elaboración de un artículo científico resumiendo el proceso y los resultados obtenidos de este trabajo, así como la experiencia al aplicar los instrumentos seleccionados.
- 5.4. Elaboración del informe final.

4.2. UNIDAD DE TRABAJO

Con el fin de realizar la evaluación al proceso se analizará el uso del proceso actual de desarrollo de software de Comfamiliar Risaralda según su nivel de aplicación en los proyectos: Sistema de información integral de Salud, Sistema de Gestión Humana y Nómina, Modulo único de Facturación. Proyectos que por su envergadura y complejidad, presentaron retos en su gestión, desarrollo, implementación y soporte, y que por lo tanto demuestran las características de un proyecto de importancia y riesgo significativo para la institución.

4.3. INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN

Para la Fase 1 se usó el instrumento de la UAM denominado “CARACTERIZACION DEL NIVEL DE DESARROLLO DE SOFTWARE DE LAS EMPRESAS DE MANIZALES” desarrollado por la estudiante L. Isabel Peralta M. para la evaluación del proceso actual del ADSCR, actualizado por el autor para corresponder con la versión 1.3 de CMMI-DEV.

Para la Fase 2 se diseñó un instrumento propio pero basado en las experiencias de comparación de prácticas ágiles frente CMMI-DEV a nivel de subprácticas propuesto por Díaz Fernández (2009) en su tesis de máster.

En la Fase 3 se modeló formalmente el PES resultante mediante lenguaje SPEM usando la herramienta Eclipse Process Framework.

En la Fase 4 se usó el mismo instrumento de la Fase 1 para evaluar el nuevo PES de manera que se comparan el estado anterior del proceso y el nuevo estado luego de la definición del PES. También se usó una encuesta con el fin de obtener información cualitativa sobre la aceptación y uso del nuevo PES por parte del equipo de desarrollo involucrado en el piloto con el fin de obtener retroalimentación y así poder ajustar la ponderación usada para la selección de las prácticas y de esta manera acercar más el proceso según la cultura de la empresa.

5. RESULTADOS ESPERADOS

Gracias a este proyecto se obtuvieron los siguientes productos:

Un **Informe gerencial** para la coordinación del área de sistemas de la empresa estudiada que incluye:

- Estado actual del proceso de desarrollo de software.
- Áreas para mejorar encontradas en el proceso de desarrollo actual.
- Descripción del proceso estructurado de software propuesto para el área de desarrollo de Comfamiliar Risaralda.
- Resultado de la prueba piloto de aplicación del nuevo proceso estructurado de desarrollo de software propuesto.

Un proceso estructurado de desarrollo (con CMMI como referente y basado en las buenas prácticas de SCRUM, XP e ICONIX) sugerido de acuerdo a los resultados de la investigación debidamente documentado en la herramienta Eclipse Process Framework (EPF).

También se presentó un **artículo científico** a una publicación sobre la experiencia de la evaluación del proceso y comparación de metodologías y un sumario de los instrumentos utilizados.

6. IMPACTOS ESPERADOS A PARTIR DEL USO DE LOS RESULTADOS

Se espera dotar al área de Informática de Comfamiliar Risaralda con información analizada sobre la experiencia, fortalezas y debilidades de las prácticas aplicadas en el desarrollo de proyectos de software en esta institución durante los últimos años, con el fin de permitir la retroalimentación al proceso y brindando herramientas para realizar ajustes en las prácticas aplicadas en futuros proyectos.

Un proceso estructurado de software adecuado para el área de desarrollo de Comfamiliar Risaralda, junto con el resultado de la prueba piloto que valide su aplicabilidad y puntos de posible mejora futura.

Se espera suministrar a la comunidad educativa de material de estudio contextualizado en la región cafetera del país que sirva como base para futuros estudios, validaciones e investigaciones sobre los procesos de desarrollo y mejoramiento de estos procesos de desarrollo de software en empresas de la región mencionada.

Se dotará a la comunidad educativa con un instrumento para la evaluación y comparación de metodologías a la luz de CMMI-DEV.

Se dotará a la comunidad educativa de los resultados de la comparación de SCRUM, XP, ICONIX a la luz de todas las áreas de proceso definidas por CMMI-DEV.

Se dotará a la comunidad educativa de una experiencia documentada de la selección de prácticas para la mejora de procesos de desarrollo de software que podrían ser fácilmente implementadas en las PyMES de la región del eje cafetero.

7. DESARROLLO

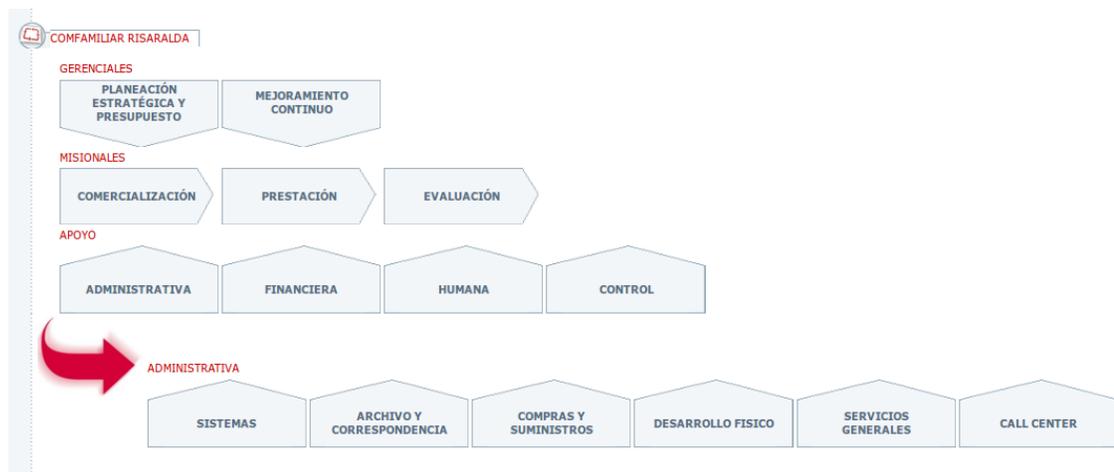
7.1. DESCRIPCIÓN Y EVALUACIÓN DEL PROCESO DE DESARROLLO ACTUAL DEL ADSCR

7.1.1. Descripción del proceso de desarrollo actual del ADSCR

Introducción

Comfamiliar Risaralda cuenta con una estructura jerárquica, en la cual área de sistemas se encuentra ubicada dentro de la estructura organizacional como un proceso de apoyo dentro del área administrativa (Ver Figura 9). Este proceso es el encargado de desarrollar y/o adquirir las soluciones tecnológicas requeridas por otros procesos de apoyo y misionales.

Figura 9. Ubicación del proceso de sistemas en el organigrama de Comfamiliar Risaralda



En el año 2006 la organización logra la certificación ICONTEC bajo la norma ISO 9001. La organización cuenta con un proceso de Auditoría Interna bajo el área de control y de un proceso de mejoramiento continuo a nivel de los procesos gerenciales.

El proceso de sistemas cuenta con un instructivo para el desarrollo de software de la institución, este se encuentra documentado en el sistema de gestión de calidad bajo el nombre “Estándares de Análisis, Diseño y Desarrollo de Sistemas de Información” e identificado con el código “1-IN-030”. (Ver [Anexo A](#)). El objetivo de este documento es recopilar los elementos necesarios para realizar una documentación de los proyectos informáticos durante todas sus etapas. Incluye descripciones para realizar un manual técnico, comprendido por los resultados obtenidos en las fases de análisis, diseño y desarrollo.

Inicia con una serie de definiciones de términos técnicos, seguida de una descripción de los cargos que intervienen en los procesos del área de sistemas de la institución.

Describe luego la metodología aplicada para la documentación de proyectos informáticos que incluye el desarrollo de:

- Documento de análisis
- Documento de diseño.
- Documento de desarrollo o manual técnico.
- Documento para el registro de modificaciones a los sistemas de información.

Se describen los estándares técnicos para la codificación en diferentes lenguajes y plataformas como son:

- Estándar para el nombramiento de objetos, campos y tablas para el sistema de base de datos DB2 sobre As/400.
- Estándar para el nombramiento de objetos del proyecto para desarrollo en Visual Basic.
- Estándar para el nombramiento de objetos dentro de un proyecto desarrollado en JAVA.

Documento de análisis

Este documento describe el procedimiento para recopilar información sobre las necesidades que originan el desarrollo del sistema de información, documentando el proceso que requiere la solución y un cronograma tentativo para el desarrollo del análisis. Se documentan también los procedimientos que actualmente se desarrollan en el proceso generalmente de forma manual y que deben incluirse en la solución propuesta. También se registran las limitaciones que impone el negocio así como los beneficios que se podrían

obtener al implementar la solución. Con ayuda del coordinador de desarrollo también se determina la complejidad del proyecto y sus objetivos.

Como resultado final se determina un diagnóstico el cual describe desde el punto de vista del equipo de desarrollo si la solución es viable y si debe ser un sistema nuevo o una adecuación a uno ya existente.

Documento de Diseño

Este documento define el alcance del proyecto a desarrollar, el cronograma para cumplir con los objetivos. También se enuncian las restricciones que enmarcarán el desarrollo. Es también en parte un documento técnico donde se realiza un diagrama de casos de uso, un diagrama de transición de estados, un modelo entidad relación, un diccionario de datos y un diagrama de funciones y navegación.

Documento de desarrollo o manual técnico

Este documento permite describir los elementos del sistema con el fin de transmitir este conocimiento a otros ingenieros y de esta manera dar soporte o realizar modificaciones futuras al software. Los elementos principales de este documento son:

- La versión final del modelo entidad relación.
- Detalle de cada función incluida en el sistema de información incluyendo tipo, una descripción de la funcionalidad, precondiciones, post condiciones y parámetros de entrada y salida.

Documento para el registro de modificaciones a los sistemas de información

Dado que la mayoría de esta documentación es impresa y reposa físicamente en el archivo de la dependencia se incluye este formato de modificaciones con el fin de mantener los documentos descritos anteriormente actualizados a medida que al sistema se le realizan actualizaciones, sin la necesidad de imprimir nuevamente todos los documentos.

Este documento describe quien realiza la modificación y el tipo de la misma junto con los soportes para justificar la realización del cambio descrito. Tiene una descripción y un cronograma para la realización del cambio.

Como documentación técnica incluye:

- Un diccionario de datos donde se describen las tablas y los campos afectados en el cambio.
- Un listado de clases y/o funciones con sus precondiciones, post-condiciones y parámetros.
- Un listado de términos y convenciones.

7.1.2. Evaluación del proceso de desarrollo actual del ADSCR

Se realizó la evaluación del proceso de desarrollo de software de Comfamiliar Risaralda mediante 2 instrumentos basados en CMMI v1.2 disponibles en la UAM. El primero un instrumento donde se evalúa a nivel de Áreas de Proceso siendo una evaluación ágil y corta, el segundo instrumento usado fue desarrollado en la tesis de grado de Isabel Peralta Mendieta (Peralta, 2007) esta evaluación se realiza a nivel de prácticas específicas permitiendo una evaluación más precisa, pero su realización toma más tiempo.

Ambas evaluaciones fueron realizadas con el coordinador de desarrollo del proceso quién calificó cada punto usando la tabla de valores sugerido por cada instrumento, basado en su experiencia y conocimiento de los procesos.

El resumen del resultado obtenido con el instrumento basado en las áreas de proceso se puede observar en la Tabla 4. El resultado de la evaluación detallada puede encontrarse en el [Anexo B](#).

Tabla 4. Resumen evaluación del proceso actual por áreas de proceso

Área de Proceso	Calificación
1. Administración de Requerimientos (REQM)	Frecuentemente
2. Planificación de Proyectos (PP)	Frecuentemente
3. Monitoreo y Control de Proyectos (PMC)	Algunas veces
4. Administración de Contratos con Proveedores (SAM)	Siempre
5. Medición y Análisis (MA)	Algunas veces
6. Aseguramiento de la Calidad del Proceso y el Producto (PPQA)	Siempre
7. Administración de Configuraciones (CM)	Algunas veces
8. Desarrollo de Requerimientos (REQD)	Siempre
9. Solución Técnica (TS)	Siempre
10. Integración de Productos (PI)	Frecuentemente
11. Verificación (VER)	Siempre
12. Validación (VAL)	Siempre
13. Enfoque a Procesos Organizacionales (OPF)	Siempre
14. Definición de Procesos Organizacionales (OPD)	Frecuentemente
15. Entrenamiento Organizacional (OT)	Siempre
16. Administración Integrada de Proyectos (IPM)	Frecuentemente
17. Administración de Riesgos (RSKM)	Casi nunca
18. Análisis de Decisiones y Resolución (DAR)	Frecuentemente
19. Desempeño de Procesos Organizacionales (OPP)	Frecuentemente
20. Administración Cuantitativa de Proyectos (QPM)	Casi nunca
21. Gestión del rendimiento de la organización (OPM)	Algunas veces
22. Análisis de Causas y Resolución (CAR)	Algunas veces

Una vez realizada la evaluación detallada calificando cada práctica mediante la tabla de valores sugeridos, se usó un método de ponderación porcentual para medir el grado de cumplimiento de cada meta específica y luego de cada área de proceso¹⁴.

¹⁴ El método para realizar esta ponderación esta descrito la tesis de grado de Isabel Peralta.

El resumen del resultado obtenido con el instrumento basado en prácticas específicas se puede observar en la Tabla 5. El resultado de la evaluación detallada puede encontrarse en el [Anexo C](#).

Tabla 5. Resumen resultado del proceso actual evaluación por prácticas específicas

	Área de Proceso	Grado de cumplimiento
1.	Administración de Requerimientos (REQM)	75%
2.	Planificación de Proyectos (PP)	89%
3.	Monitoreo y Control de Proyectos (PMC)	60%
4.	Administración de Contratos con Proveedores (SAM)	100%
5.	Medición y Análisis (MA)	44%
6.	Aseguramiento de la Calidad del Proceso y el Producto (PPQA)	100%
7.	Administración de Configuraciones (CM)	46%
8.	Desarrollo de Requerimientos (REQD)	98%
9.	Solución Técnica (TS)	75%
10.	Integración de Productos (PI)	53%
11.	Verificación (VER)	88%
12.	Validación (VAL)	90%
13.	Enfoque a Procesos Organizacionales (OPF)	83%
14.	Definición de Procesos Organizacionales (OPD)	57%
15.	Entrenamiento Organizacional (OT)	100%
16.	Administración Integrada de Proyectos (IPM)	58%
17.	Administración de Riesgos (RSKM)	0%
18.	Análisis de Decisiones y Resolución (DAR)	79%
19.	Desempeño de Procesos Organizacionales (OPP)	80%
20.	Administración Cuantitativa de Proyectos (QPM)	25%
21.	Gestión del rendimiento de la organización (OPM)	38%
22.	Análisis de Causas y Resolución (CAR)	60%

Con el fin de verificar la concordancia de ambas evaluaciones se homologaron ambos sistemas de calificación (ver Tabla 6) y se compararon los resultados para cada área de proceso (ver Tabla 7).

Tabla 6. Homologación de calificaciones para los 2 instrumentos.

Calificación cualitativa		Homologación	
Instrumento 1	Instrumento 2	No.	%
NA/NS	NS/NR	5	
Siempre	Siempre	4	100%
Frecuentemente	Muchas veces	3	75%
La mitad de las veces	Algunas veces	2	50%
Algunas veces	Algunas veces	2	50%
Casi nunca	Casi nunca	1	25%
	Nunca	0	0%

Tabla 7. Comparación resultados evaluación.

Área de Proceso	Instrumento		Variación ¹⁵
	1	2	
1. Administración de Requerimientos (REQM)	75%	75%	0%
2. Planificación de Proyectos (PP)	75%	89%	-19%
3. Monitoreo y Control de Proyectos	50%	60%	-20%
4. Administración de Contratos con Proveedores (SAM)	100%	100%	0%
5. Medición y Análisis (MA)	50%	34%	32%
6. Aseguramiento de la Calidad del	100%	100%	0%
7. Administración de Configuraciones	50%	46%	8%
8. Desarrollo de Requerimientos	100%	98%	2%
9. Solución Técnica (TS)	100%	75%	25%
10. Integración de Productos (PI)	75%	53%	29%
11. Verificación (VER)	100%	88%	12%
12. Validación (VAL)	100%	90%	10%
13. Enfoque a Procesos	100%	83%	17%

¹⁵ La variación se calcula mediante la formula $1 - \frac{\text{Instrumento 2}}{\text{Instrumento 1}}$

14. Definición de Procesos	75%	57%	24%
15. Entrenamiento Organizacional (OT)	100%	100%	0%
16. Administración Integrada de	75%	58%	23%
17. Administración de Riesgos (RSKM)	25%	0%	100%
18. Análisis de Decisiones y Resolución	75%	79%	-5%
19. Desempeño de Procesos Organizacionales (OPP)	75%	80%	-7%
20. Administración Cuantitativa de	25%	25%	0%
21. Gestión del rendimiento de la	50%	38%	24%
22. Análisis de Causas y Resolución (CAR)	50%	60%	-20%

Los resultados que tuvieron una variación alta entre los 2 instrumentos fueron revisados y reevaluados hasta que el porcentaje de variación fue inferior al 30% entre las cuales se encontró Medición y Análisis (MA), Solución Técnica (TS) e Integración de Productos (PI) , en el caso del área de proceso de Administración de Riesgos (RSKM) no fue posible disminuir el porcentaje de variación ya que en el instrumento 1 la calificación menor era casi nunca y en el proceso no se encontraron elementos que contribuyeran a la relación de las prácticas relacionadas con esta área de proceso.

7.1.3. Fortalezas y debilidades del proceso de desarrollo actual

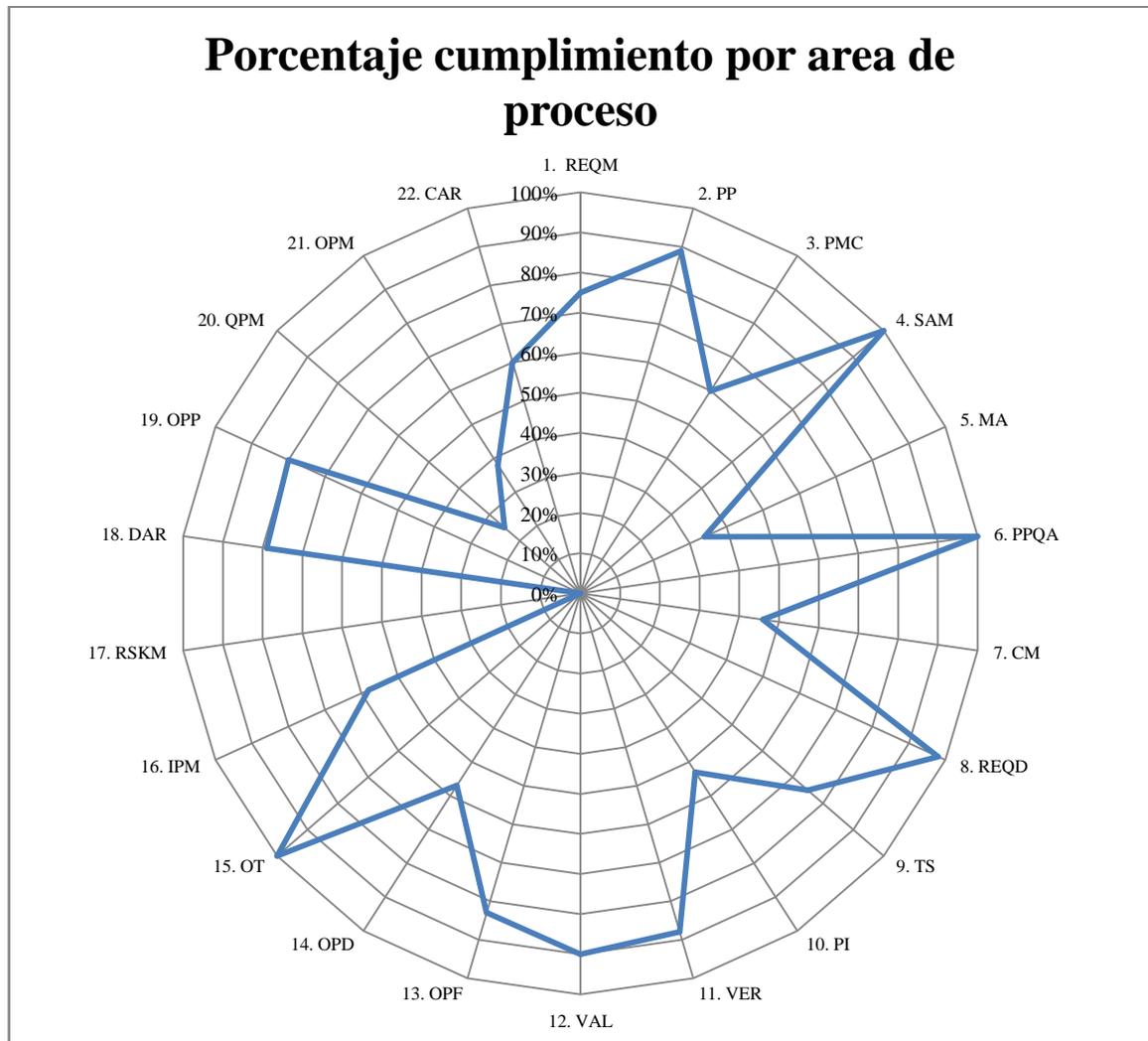
Dado que el resultado del instrumento 2 fue más preciso se usó este como base para la identificación de las fortalezas y debilidades del proceso de desarrollo de la empresa. Aquellas áreas de proceso con una evaluación por debajo de 60% (identificadas en color amarillo en la Tabla 8) fueron consideradas como puntos débiles y aquellas por encima del 90% como puntos fuertes (que pueden observarse en color verde).

Tabla 8. Fortalezas y debilidades del proceso actual identificadas mediante la evaluación realizada.

Área de Proceso	Grado de cumplimiento
1. Administración de Requerimientos (REQM)	75%
2. Planificación de Proyectos (PP)	89%
3. Monitoreo y Control de Proyectos (PMC)	60%

4. Administración de Contratos con Proveedores (SAM)	100%
5. Medición y Análisis (MA)	34%
6. Aseguramiento de la Calidad del Proceso y el Producto (PPQA)	100%
7. Administración de Configuraciones (CM)	46%
8. Desarrollo de Requerimientos (REQD)	98%
9. Solución Técnica (TS)	75%
10. Integración de Productos (PI)	53%
11. Verificación (VER)	88%
12. Validación (VAL)	90%
13. Enfoque a Procesos Organizacionales (OPF)	83%
14. Definición de Procesos Organizacionales (OPD)	57%
15. Entrenamiento Organizacional (OT)	100%
16. Administración Integrada de Proyectos (IPM)	58%
17. Administración de Riesgos (RSKM)	0%
18. Análisis de Decisiones y Resolución (DAR)	79%
19. Desempeño de Procesos Organizacionales (OPP)	80%
20. Administración Cuantitativa de Proyectos (QPM)	25%
21. Gestión del rendimiento de la organización (OPM)	38%
22. Análisis de Causas y Resolución (CAR)	60%

Figura 10. Resultado porcentual cumplimiento por área de proceso evaluación inicial del área de desarrollo de SW



Cinco áreas de proceso fueron identificadas como de un mayor grado de cumplimiento (mayor a 90%):

- Administración de contratos de con proveedores (SAM)
- Aseguramiento de la calidad del proceso y del producto (PPQA)
- Desarrollo de Requerimientos (REQD)
- Validación (VAL)
- Entrenamiento Organizacional (OT)

Diez áreas de proceso fueron identificadas con bajo grado de cumplimiento (menor a 60%):

- Integración de Productos (PI)
- Monitoreo y Control de Proyectos (PMC)
- Definición de Procesos Organizacionales (OPD)
- Administración Integrada de Proyectos (IPM)
- Análisis de Causas y Resolución (CAR)

Cinco de estas obtuvieron las calificaciones más bajas¹⁶:

- Medición y Análisis (MA)
- Administración de Configuraciones (CM)
- Administración de Riesgos (RSKM)
- Administración Cuantitativa de Proyectos (QPM)
- Gestión del rendimiento de la organización (OPM)

Las mayoría de fortalezas encontradas son derivadas de la aplicación del sistema de calidad implementado en la empresa y de la gestión del área administrativa tanto del proceso como de la organización, áreas de proceso como la administración de proveedores y entrenamiento organizacional tienen procesos bien establecidos y controlados desde la dirección; por otro lado las áreas de aseguramiento de la calidad del proceso y del producto y de validación, son fuertemente apoyadas por el área de auditoría y control interno de la institución. Solo el área de desarrollo de requerimientos resalta como una fortaleza del área técnica encargada del mantenimiento y desarrollo de proyectos de software.

Las debilidades en cambio se encuentran en su mayoría en áreas relacionadas con la gestión y la medición (Monitoreo y Control de Proyectos, Definición de Procesos Organizacionales, Administración Integrada de Proyectos, Medición y Análisis, Administración de Riesgos, Administración Cuantitativa de Proyectos, Gestión del rendimiento de la organización.) y algunas relacionadas con la correcta documentación o registro (Análisis de Causas y Resolución, Administración de Configuraciones) , y solo evidencia un área relacionada con elementos técnicos (Integración de Productos).

¹⁶ Estas áreas se encuentran identificadas en la Tabla 8 cuyo porcentaje se encuentra en color rojo.

7.2. ANÁLISIS COMPARATIVO DE LAS PRÁCTICAS ÁGILES SELECCIONADAS

7.2.1. Construcción del instrumento de comparación

Durante esta fase se tomó como base algunos trabajos previos para realizar la comparación entre las prácticas definidas en CMMI y las posibles correspondencias en Scrum, XP e ICONIX. La relación de Scrum y CMMI en las áreas de PP, PMC, SAM, IPM, RSKM, QPM fue analizada en el trabajo de (Marcal, de Freitas, Furtado Soares, & Belchior, 2007). Luego este estudio fue profundizado analizando la correspondencia de Scrum y XP en las áreas de proceso PP, PMC y REQM realizando una evaluación a nivel de subprácticas por (Díaz Fernandez, 2009). Estos estudios coinciden en el uso de 3 criterios para calificar el cubrimiento de cada práctica de CMMI (ver Tabla 9) unos criterios similares serán usados en el presente trabajo. Se tomó la decisión de realizar la comparación a nivel de subprácticas por su alto nivel de detalle lo que permite mejores comparaciones y análisis.

Tabla 9. Criterios de cubrimiento entre prácticas CMMI y Metodologías ágiles

Calificación	Criterio
No soportada (NS)	La práctica no es soportada por las prácticas de la metodología
Parcialmente Soportada (PS)	Algunos elementos son abordados por prácticas de la metodología pero se requiere mayor interpretación o elementos adicionales para cumplir completamente la practica seleccionada.
Soportada (S)	La práctica es soportada completamente.

Aunque XP busca también cumplir algunos de los objetivos de CMMI, no lo hace usando el mismo enfoque lo que hace difícil compararlos, XP puede resolver un problema con

prácticas específicas completamente distintas a las descritas en CMMI y aun así cumplir con el objetivo principal de la práctica¹⁷.

Scrum, XP e ICONIX fueron comparados con cada subpráctica asignando la calificación “OK” en caso de considerarse que se cumple con la subpráctica o “X” en caso negativo, esta calificación definirá con mayor precisión la calificación para la práctica específica de CMMI correspondiente.

A continuación, en la Tabla 10 se puede observar un ejemplo de esta comparación con SCRUM en el área de proceso Administrar Los Requerimientos (REQM):

Tabla 10. Ejemplo de comparación de Scrum frente a CMMI.

Práctica específica	1) SP1.1 Comprenden los requerimientos (requisitos) de los proveedores sobre el significado de dichos requerimientos.
Calificación de subprácticas	OK 1. Establecer criterios para distinguir a los proveedores apropiados de requisitos. X 2. Establecer criterios objetivos para la evaluación y aceptación de los requisitos. X 3. Analizar los requisitos para asegurar que se cumplen los criterios establecidos. OK 4. Alcanzar una comprensión de los requisitos con los proveedores de requisitos para que los participantes del proyecto puedan comprometerse con ellos.
Calificación práctica específica	PS (Parcialmente soportada)
Observación	Durante el pre-game ¹⁸ se definen los roles incluyendo al propietario del producto y los clientes representativos, estos proveen los requerimientos y apoyan al equipo

¹⁷ El uso de prácticas alternas para cumplir con los mismos objetivos propuestos para las prácticas en CMMI ya ha sido sugerido en el pasado por (Paulk, 2001) (Fritzsche & Keil, 2007) por ejemplo en su interpretación sobre la gestión del riesgo en XP.

¹⁸ El pre-game es la primera fase de Scrum donde se realiza la planeación y se decide la arquitectura que tendrá el sistema (Schwaber, 1997).

proporcionando detalles sobre los mismos.

Requiere de interpretación, el cliente debe involucrarse durante todo el ciclo de vida y no solo al principio y puede agregar nuevas funcionalidades a medida que descubre nuevos requerimientos.

En general se analizaron de esta forma el 60% de todas las prácticas específicas de todas las áreas de proceso, algunas que claramente no eran soportadas o no eran del alcance de la metodología no fueron analizadas a nivel de subpráctica dada su irrelevancia.

El sistema de calificación presentado en la Tabla 9 fue afinado luego de observar que un análisis más detallado de calificación basado en las subprácticas permitía definir mejor las diferencias y fortalezas entre las diferentes metodologías, quedando finalmente con las calificaciones que muestra la Tabla 11.

La evaluación detallada de las Scrum, XP e ICONIX puede ser observada en el [Anexo D](#).

Tabla 11. Calificaciones ajustadas a 5 niveles de detalle.

Calificación	Criterio
No soportada (NS)	La práctica no es soportada por las prácticas de la metodología
Parcialmente Soportada en menor medida (PS-)	Solo uno o muy pocos elementos son abordados por prácticas de la metodología pero se requiere mayor interpretación o elementos adicionales para cumplir completamente la práctica seleccionada.
Parcialmente Soportada (PS)	Cerca de la mitad de los elementos son abordados por prácticas de la metodología pero se requiere mayor interpretación o elementos adicionales para cumplir completamente la práctica seleccionada.

Parcialmente medida (PS+)	Soportada en gran medida	La mayoría de los elementos son abordados por prácticas de la metodología pero se requiere mayor interpretación o elementos adicionales para cumplir completamente la practica seleccionada.
	Soportada (S)	La práctica es soportada completamente.

7.2.2. Comparación prácticas ágiles y el proceso del ADSCR

Con base en la evaluación por prácticas específicas realizadas se generó la homologación correspondiente con el fin de poder comparar la evaluación mediante el instrumento UAM y las matrices de comparación frente a CMMI de cada metodología de (ver Tabla 12) por ejemplo, el puntaje 4 de esta evaluación correspondiente a que “Siempre” se cumplía con esta práctica dará como resultado una práctica “Soportada”.

Tabla 12. Homologación de calificaciones entre el instrumento de caracterización y la matriz de cumplimiento

Calificación		
Instrumento de caracterización		Matriz de cumplimiento
0	Nunca	No Soportada
1	Casi nunca	No Soportada
2	Algunas veces	Parcialmente Soportada
3	Muchas veces	Parcialmente Soportada
4	Siempre	Soportada

Esta homologación fue finalmente mejorada con el fin de afinar el instrumento de comparación y hacerlo más preciso quedando finalmente como lo muestra la Tabla 13.

Tabla 13. Homologación final para la calificación de la matriz de cumplimiento.

Calificación		
Instrumento de caracterización		Matriz de cumplimiento
0	Nunca	No Soportada
1	Casi nunca	Parcialmente Soportada -

2	Algunas veces	Parcialmente Soportada
3	Muchas veces	Parcialmente Soportada +
4	Siempre	Soportada

Las calificaciones de cada metodología frente a CMMI realizadas en el numeral anterior fueron transcritas al instrumento de la UAM con el fin de lograr una calificación numérica que pudiera ser comparada con la evaluación realizada al proceso de desarrollo de software de la empresa evaluada.

La Tabla 14 muestra la conversión de las calificaciones alfanuméricas a numéricas según la homologación propuesta para el área de proceso PMC con respecto a XP. En la tercera columna se puede observar la evaluación obtenida por ejemplo para la práctica específica 1.1 de la meta específica SG1 como Parcialmente soportada, lo que se traduce en una calificación numérica de 2. Luego se puede calcular el porcentaje de cumplimiento de las metas específicas y del área de proceso teniendo en cuenta el número de prácticas específicas totales y el puntaje total obtenido, aplicando la siguiente fórmula¹⁹:

$$\text{Porcentaje Cubrimiento} = \frac{\sum \text{Puntaje obtenido}}{\text{Puntaje Máximo} \times \text{Número Total Prácticas}}$$

Tabla 14. Homologación de calificaciones y cálculo del porcentaje de cubrimiento de la metodología frente a CMMI para el área PMC con respecto a la metodología XP.

MONITOREO Y CONTROL DE PROYECTOS 55%		
(PMC)		
SG1 MONITOREAR EL PROYECTO CONTRA EL 43%		
PLAN (PMC)		
SP 1,1	2	PS

¹⁹ El puntaje máximo corresponde al valor 4, que es la puntuación máxima posible permitida por el instrumento.

SP 1,2	4	S
SP 1,3	0	NS
SP 1,4	0	NS
SP 1,5	0	NS
SP 1,6	2	PS
SP 1,7	4	S
7	12	
SG2 ADMINISTRAN ACCION CORRECTIVA 83% HASTA EL CIERRE (PMC)		
SP 2,1	4	S
SP 2,2	4	S
SP 2,3	2	PS
3	10	
10	22	

En este punto una nueva evaluación fue realizada al proceso de desarrollo de Comfamiliar Risaralda, tomando como base la homologación anterior, esta evaluación consistió en la revisión del cubrimiento de las prácticas identificadas comparándolas con cada subpráctica con el fin de tener una visión más objetiva, dando como resultado un ajuste a la puntuación total obtenida en el numeral 1.3. Este resultado se encuentra en el [Anexo E](#) y el resumen de la calificación se puede observar en la Tabla 15, en tonos verdes valores superiores a 70%, tonos rojos para valores inferiores a 50% y tonos amarillos en medio.

Esta evaluación se basa en la descripción de los procesos actuales incluyendo alguna evidencia de soporte para la evaluación a diferencia del instrumento de la UAM que se basa solo en la percepción del entrevistado, este resultado tiende a ser más objetivo y más detallado gracias al análisis a nivel de subpráctica mencionado anteriormente y debido a esto, esta última calificación se utilizó como base para este estudio.

Tabla 15. Reevaluación del proceso actual.

Área de Proceso	Evaluación	
	Instrumento UAM	A nivel de subprácticas

1. Administración de Requerimientos (REQM)	75%	70%
2. Planificación de Proyectos (PP)	89%	66%
3. Monitoreo y Control de Proyectos (PMC)	60%	55%
4. Administración de Contratos con Proveedores (SAM)	100%	100%
5. Medición y Análisis (MA)	34%	38%
6. Aseguramiento de la Calidad del Proceso y el Producto (PPQA)	100%	100%
7. Administración de Configuraciones (CM)	46%	43%
8. Desarrollo de Requerimientos (REQD)	98%	63%
9. Solución Técnica (TS)	50%	47%
10. Integración de Productos (PI)	33%	19%
11. Verificación (VER)	88%	22%
12. Validación (VAL)	90%	60%
13. Enfoque a Procesos Organizacionales (OPF)	83%	67%
14. Definición de Procesos Organizacionales (OPD)	57%	57%
15. Entrenamiento Organizacional (OT)	100%	100%
16. Administración Integrada de Proyectos (IPM)	58%	45%
17. Administración de Riesgos (RSKM)	0%	0%
18. Análisis de Decisiones y Resolución (DAR)	79%	75%
19. Desempeño de Procesos Organizacionales (OPP)	80%	60%
20. Administración Cuantitativa de Proyectos (QPM)	25%	7%
21. Gestión del rendimiento de la organización (OPM)	38%	35%
22. Análisis de Causas y Resolución (CAR)	60%	40%

Luego de esto el análisis del cubrimiento de las prácticas ágiles frente a CMMI puede ser directamente comparable con el desempeño de las prácticas aplicadas en la empresa en cuestión en la Tabla 16 se muestra el resumen de esta comparación.

Tabla 16. Resumen cubrimiento porcentual por área de proceso de cada metodología

Nivel	Área de Proceso	SCRUM	XP	ICONIX	Comfamiliar
--------------	------------------------	--------------	-----------	---------------	--------------------

2	1. REQM	80%	80%	90%	70%
2	2. PP	68%	43%	14%	66%
2	3. PMC	88%	55%	0%	55%
2	4. SAM	0%	0%	0%	100%
2	5. MA	38%	38%	0%	38%
2	6. PPQA	13%	50%	13%	100%
2	7. CM	0%	100%	0%	43%
3	8. REQD	93%	93%	83%	63%
3	9. TS	0%	31%	44%	47%
3	10. PI	0%	61%	53%	19%
3	11. VER	0%	72%	69%	22%
3	12. VAL	80%	90%	55%	60%
3	13. OPF	0%	0%	0%	67%
3	14. OPD	0%	0%	0%	57%
3	15. OT	0%	14%	0%	100%
3	16. IPM	58%	33%	0%	45%
3	17. RSKM	7%	36%	0%	0%
3	18. DAR	0%	0%	0%	75%
4	19. OPP	0%	0%	0%	60%
4	20. QPM	0%	0%	0%	7%
5	21. OPM	0%	0%	0%	35%
5	22. CAR	0%	0%	0%	40%

Se puede observar un cubrimiento nulo en los niveles 4 y 5 por las prácticas ágiles frente a CMMI, lo que confirma lo que estudios previos muestran (Paulk, 2001) (Fritzsche & Keil, 2007) (Gandomani & Zulzalil, 2013).

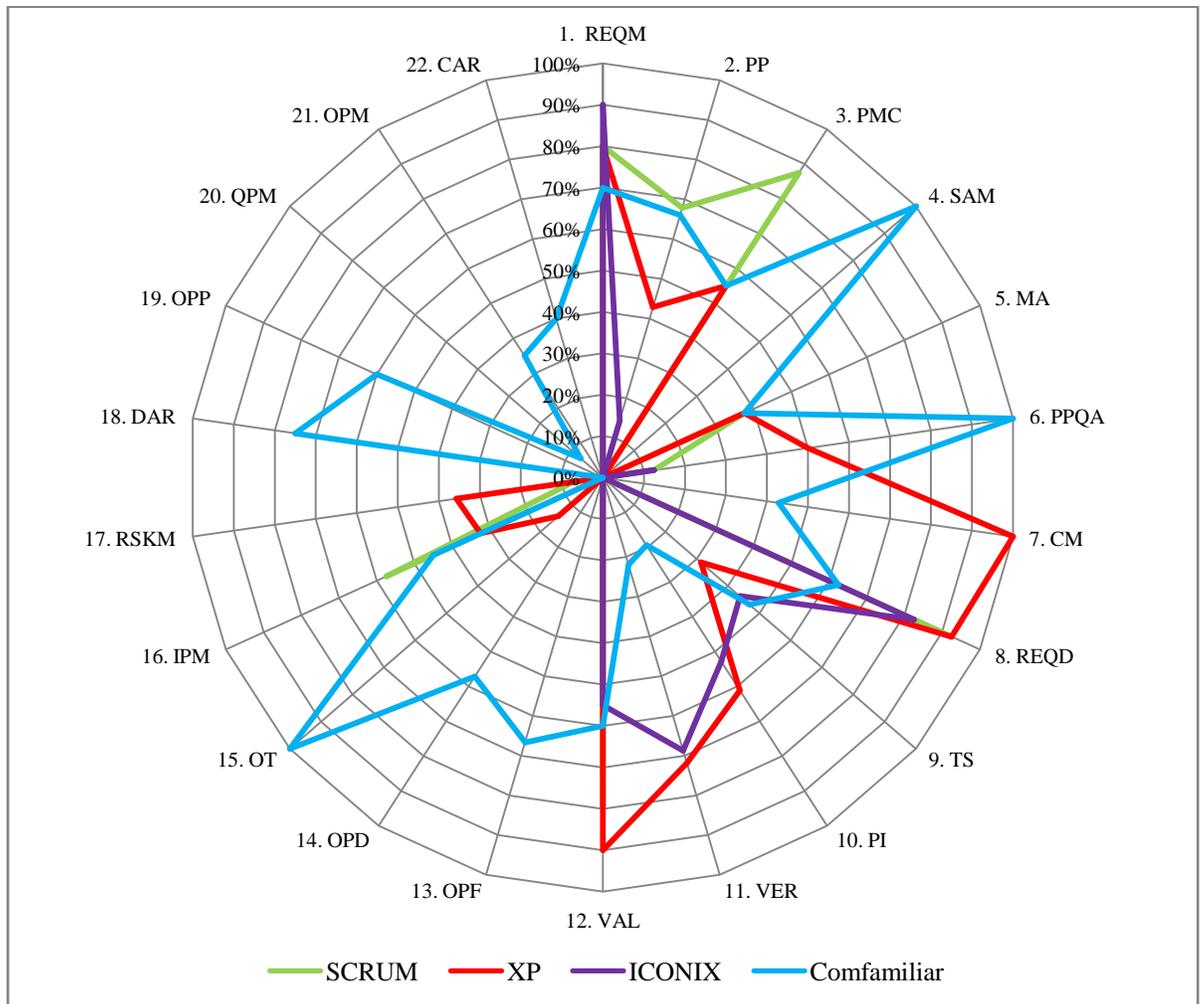
Scrum demuestra su fortaleza principalmente en las áreas REQD, PMC, REQM, VAL. También se observa que XP tiene un cubrimiento mayor de CMMI en los niveles 2²⁰ y 3. Los únicos procesos de nivel 3 que XP no cubre son OPF, OPD y DAR proceso que tienen un enfoque organizacional y quedan fuera del alcance de XP.

Finalmente si solo analizamos este indicador de cubrimiento parece demostrar que ICONIX no agrega nuevos elementos a la mezcla de Scrum y XP, mostrando solo un mayor cubrimiento en las áreas de proceso REQM y TS.

Esto era de esperarse, Scrum e ICONIX parecen ser más complementarios que una mezcla de este último frente a XP, y a que Scrum está más enfocado a gestión mientras que tanto XP como ICONIX se concentran en la construcción y desarrollo del producto.

²⁰ Ninguno de los métodos analizados en este estudio tienen un cubrimiento del área de proceso SAM debido al enfoque de estos métodos ágiles al desarrollo en casa en contraste a la práctica de contratación de proveedores.

Figura 11. Comparación radial del cubrimiento de CMMI por las metodologías y el proceso actual del ADSCR



El próximo paso fue resumir la capacidad de cubrimiento de las prácticas ágiles, tomando como posible mejora el máximo porcentaje encontrado en todos los métodos ágiles frente a cada área de proceso ver Tabla 17.

Tabla 17. Máximo potencial de cubrimiento de las prácticas ágiles seleccionadas frente a CMMI

Área de Proceso	Ágiles	Metodología Líder
1. Administración de Requerimientos (REQM)	90%	ICONIX

2. Planificación de Proyectos (PP)	68%	SCRUM
3. Monitoreo y Control de Proyectos (PMC)	88%	SCRUM
4. Administración de Contratos con Proveedores (SAM)	0%	
5. Medición y Análisis (MA)	38%	XP/SCRUM
6. Aseguramiento de la Calidad del Proceso y el Producto (PPQA)	50%	XP
7. Administración de Configuraciones (CM)	100%	XP
8. Desarrollo de Requerimientos (REQD)	93%	XP/SCRUM
9. Solución Técnica (TS)	44%	ICONIX
10. Integración de Productos (PI)	61%	XP
11. Verificación (VER)	72%	XP
12. Validación (VAL)	90%	XP
13. Enfoque a Procesos Organizacionales (OPF)	0%	
14. Definición de Procesos Organizacionales (OPD)	0%	
15. Entrenamiento Organizacional (OT)	14%	XP
16. Administración Integrada de Proyectos (IPM)	58%	SCRUM
17. Administración de Riesgos (RSKM)	36%	XP
18. Análisis de Decisiones y Resolución (DAR)	0%	
19. Desempeño de Procesos Organizacionales (OPP)	0%	
20. Administración Cuantitativa de Proyectos (QPM)	0%	
21. Gestión del rendimiento de la organización (OPM)	0%	
22. Análisis de Causas y Resolución (CAR)	0%	

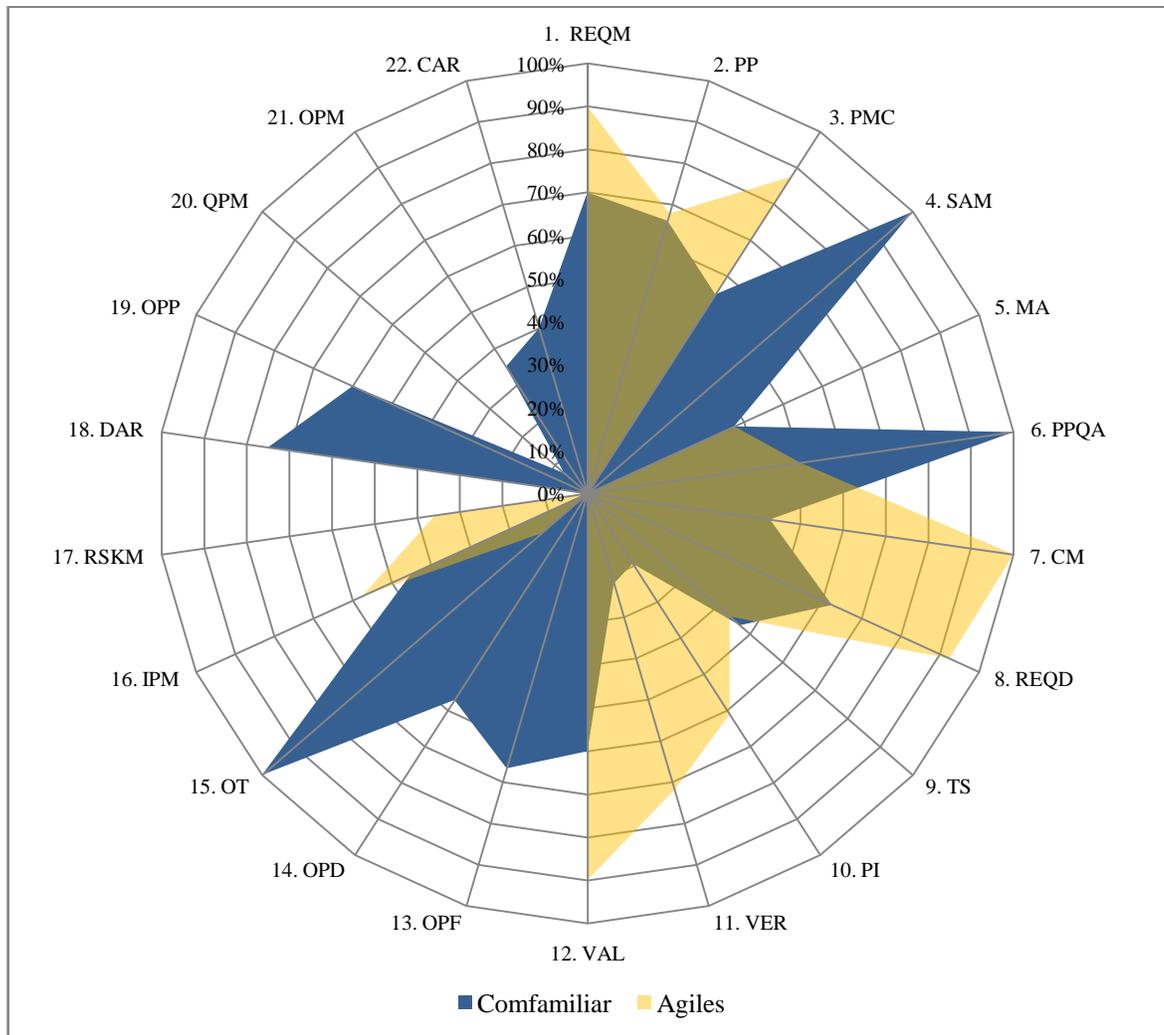
En esta visión general del cubrimiento de prácticas ágiles se deduce nuevamente que aparte de los niveles 4 y 5 mencionados anteriormente, las únicas áreas en las que estas prácticas ágiles no tienen elementos que aporten al cumplimiento de CMMI son en SAM, OPF, OPD y DAR a las cuales podríamos agregar OT por su baja calificación, áreas con un enfoque claramente organizacionales que se alejan en general del objetivo orientado a proyectos de las metodologías ágiles.

En este punto es posible realizar un análisis para encontrar los mejores nichos de mejora, comparando la capacidad actual del proceso con el potencial de cubrimiento de las prácticas ágiles seleccionadas ver Tabla 18.

Tabla 18. Potencial de mejora de cada área de proceso

Área de Proceso	Comfamiliar	Ágiles	Posible mejora
1. Administración de Requerimientos (REQM)	70%	90%	20%
2. Planificación de Proyectos (PP)	66%	68%	2%
3. Monitoreo y Control de Proyectos (PMC)	55%	88%	33%
4. Administración de Contratos con Proveedores (SAM)	100%	0%	-100%
5. Medición y Análisis (MA)	38%	38%	0%
6. Aseguramiento de la Calidad del Proceso y el Producto (PPQA)	100%	50%	-50%
7. Administración de Configuraciones (CM)	43%	100%	57%
8. Desarrollo de Requerimientos (REQD)	63%	93%	30%
9. Solución Técnica (TS)	47%	44%	-3%
10. Integración de Productos (PI)	19%	61%	42%
11. Verificación (VER)	22%	72%	50%
12. Validación (VAL)	60%	90%	30%
13. Enfoque a Procesos Organizacionales (OPF)	67%	0%	-67%
14. Definición de Procesos Organizacionales (OPD)	57%	0%	-57%
15. Entrenamiento Organizacional (OT)	100%	14%	-86%
16. Administración Integrada de Proyectos (IPM)	45%	58%	13%
17. Administración de Riesgos (RSKM)	0%	36%	36%
18. Análisis de Decisiones y Resolución (DAR)	75%	0%	-75%
19. Desempeño de Procesos Organizacionales (OPP)	60%	0%	-60%
20. Administración Cuantitativa de Proyectos (QPM)	7%	0%	-7%
21. Gestión del rendimiento de la organización (OPM)	35%	0%	-35%
22. Análisis de Causas y Resolución (CAR)	40%	0%	-40%

Figura 12. Comparación del cubrimiento frente a CMMI del conjunto de prácticas ágiles y el proceso actual del ADSCR



En la Figura 12 se observa en azul el cubrimiento del proceso actual del ADSCR y en amarillo el máximo potencial de cubrimiento en conjunto de todas las metodologías ágiles seleccionadas, aquí visualmente se hace evidente el potencial de mejoramiento en las áreas de proceso REQM, PMC, CM, REQD, PI, VER, VAL, IPM y RSKM.

7.2.3. Priorización de prácticas

Ordenando por la mayor capacidad de mejora obtenemos que es posible mejorar el proceso actual de desarrollo de software en al menos 9 áreas de proceso de nivel 2 y 3 con porcentajes que van del 13% al 57% de mejora posible (ver la Tabla 19).

También podemos definir gracias a este análisis que las prácticas ágiles de las demás áreas de proceso no representan una mejora sustancial, ya sea debido a la madurez del proceso actual del ADSCR o a la deficiencia de cubrimiento de las prácticas ágiles en dichas áreas de proceso. En color azul se puede observar las áreas donde el ADSCR es mucho más maduro y por lo tanto aunque la aplicación de las prácticas ágiles aporta algún porcentaje a su cubrimiento, las prácticas ya presentes en la institución han logrado un mayor porcentaje de cubrimiento. En gris están las áreas de proceso que no son abordadas por las prácticas ágiles seleccionadas en este proyecto y por lo tanto no ningún porcentaje de cubrimiento. Estos porcentajes negativos representan que “no hay mejora” y es la diferencia entre el estado actual del proceso y el propuesto usando prácticas ágiles.

Tabla 19. Áreas con mayor posibilidad de mejora

Nivel	Área de Proceso	Posible mejora
2	7. Administración de Configuraciones (CM)	57%
3	11. Verificación (VER)	50%
3	10. Integración de Productos (PI)	42%
3	17. Administración de Riesgos (RSKM)	36%
2	3. Monitoreo y Control de Proyectos (PMC)	33%
3	8. Desarrollo de Requerimientos (REQD)	30%
3	12. Validación (VAL)	30%
2	1. Administración de Requerimientos (REQM)	20%
3	16. Administración Integrada de Proyectos (IPM)	13%
2	2. Planificación de Proyectos (PP)	2%
2	5. Medición y Análisis (MA)	0%
3	9. Solución Técnica (TS)	-3%
4	20. Administración Cuantitativa de Proyectos (QPM)	-7%
5	21. Gestión del rendimiento de la organización (OPM)	-35%
5	22. Análisis de Causas y Resolución (CAR)	-40%
2	6. Aseguramiento de la Calidad del Proceso y el Producto (PPQA)	-50%

3	14. Definición de Procesos Organizacionales (OPD)	-57%
4	19. Desempeño de Procesos Organizacionales (OPP)	-60%
3	13. Enfoque a Procesos Organizacionales (OPF)	-67%
3	18. Análisis de Decisiones y Resolución (DAR)	-75%
3	15. Entrenamiento Organizacional (OT)	-86%
2	4. Administración de Contratos con Proveedores (SAM)	-100%

Por esta razón el siguiente proceso de ponderación se realizara solamente sobre las áreas con mayor posibilidad de mejora:

Tabla 20. 9 Áreas de proceso seleccionadas

Área de Proceso	Posible mejora
7. Administración de Configuraciones (CM)	57%
11. Verificación (VER)	50%
10. Integración de Productos (PI)	42%
17. Administración de Riesgos (RSKM)	36%
3. Monitoreo y Control de Proyectos (PMC)	33%
8. Desarrollo de Requerimientos (REQD)	30%
12. Validación (VAL)	30%
1. Administración de Requerimientos (REQM)	20%
16. Administración Integrada de Proyectos (IPM)	13%

Para realizar la ponderación, todas las prácticas de las 3 metodologías seleccionadas fueron analizadas en paralelo junto al área de proceso de CMMI y se le agregó una calificación de peso a cada práctica específica²¹. Esta ponderación es subjetiva y tiene en cuenta los objetivos de mejora de la empresa, la cultura actual y los recursos disponibles para el proceso de mejora.

²¹ La calificación va de 1 a 3 siendo 3 la más relevante o compatible para el proceso y 1 la menos atractiva para la mejora.

Tabla 21. Ejemplo de ponderación para las prácticas específicas de REQM en cada uno de los métodos a evaluar

	SCRUM			XP			ICONIX			
	COBERTURA	#	PONDERACION	COBERTURA	#	PONDERACION	COBERTURA	#	PONDERACION	CONCLUSION
NIVEL II:										
SG1 (REQM)										
1) SP1 .1	PS	2	3	PS	2	1	S	4	2	Se prefiere el enfoque de SCRUM para esta área ya que permite identificar fácilmente los proveedores (dueño del producto) y las reuniones donde se descubren los cambios y se asignan los compromisos están claramente definidas. Algunos elementos de ICONIX puede mejorar el cubrimiento en la sub-practica 1.3 ya que facilita la creación de una matriz de trazabilidad
2) SP1 .2	S	4	3	S	4	1	S	4	2	
3) SP 1.3	S	4	3	PS+	3	1	PS	2	2	
4) SP 1.4	PS	2	3	PS+	3	1	S	4	3	
5) SP 1.5	S	4	3	S	4	1	S	4	2	
		16	15		16	5		18	11	
		240			80			198		Prácticas seleccionadas de SCRUM

Las prácticas del método con mejor puntuación total serán usadas como base para el desarrollo del nuevo proceso híbrido, otras prácticas de otros métodos que tuvieron puntuación total menor pero superan en puntuación individual por alguna práctica

específica, será tenida en cuenta para complementar el área de proceso y mejorar el cubrimiento total²².

En la Tabla 22 se resumen los resultados del proceso de ponderación el cual puede consultarse en forma detallada en el [Anexo F](#).

Tabla 22. Resumen de los resultados de la ponderación para las áreas de proceso seleccionadas.

Ponderación				
Área proceso	SCRUM	XP	ICONIX	Metodología seleccionada base
7. CM	0	588	0	XP
11. VER	0	368	484	ICONIX
10. PI	0	396	266	XP
17. RSKM	6	20	0	XP
3. PMC	1050	374	0	SCRUM
8. REQD	555	370	990	ICONIX
12. VAL	240	90	55	SCRUM
1. REQM	240	80	198	SCRUM
16. IPM	690	260	0	SCRUM

²² Este es el caso de ICONIX en el ejemplo de la Tabla 21 aunque en total no supera el cubrimiento de Scrum, si es superior en su cubrimiento y ponderación para la SP1.4

7.3. ELABORACIÓN DEL NUEVO PROCESO AJUSTADO PARA EL ADSCR

7.3.1. Selección de prácticas

Gestión de la configuración (CM)

En el caso de esta área de proceso no se encontraron prácticas en Scrum ni en ICONIX que aportaran al cumplimiento de las prácticas de CMMI, en este caso se usan solo elementos de XP sin embargo si se usan prácticas que aunque no están relacionadas directamente con CM finalmente si representan un soporte o agregan elementos de definición²³.

Los elementos de configuración son principalmente código y pruebas se agregan otros elementos como el análisis y diseño, con el fin de mantener la simplicidad fundamento de XP se usa una herramienta automática como CVS o GIT y se controlan con los elementos básicos obligatorios que la herramienta defina como son fecha, observación y usuario.

Se usa la práctica de integración continua con el fin de mantener los ítems de configuración en línea y sincronizados con todos los integrantes del equipo.

La práctica de propiedad colectiva se traducirá en permisos completos del equipo de desarrollo sobre el repositorio. Una línea base deberá crearse al finalizar cada iteración.

Las solicitudes de cambio son analizadas y priorizadas al inicio de la iteración, su impacto es evaluado y verificado gracias a las pruebas automáticas.

Una vez se realizan las actividades de integración continua constantemente las herramientas automáticas utilizadas ayudan a gestionar el control de cambios en los ítems de configuración.

²³ Este es el caso de la definición de manejar los elementos de configuración como modelos y diseños que aunque no están especificados en XP si son definidos en ICONIX lo que decanta finalmente en la necesidad de su control como ítem de configuración.

Gracias a la programación en parejas²⁴, la rápida integración, las pruebas automáticas y la entrega a producción continua se verifican la integridad y se confirma lo completo y correcto de los elementos de configuración y de las líneas base.

Verificación (VER)

ICONIX propone verificar los requerimientos, el diseño preliminar, el diseño y cada unidad de código, para esto se planteara una etapa de revisión de los artefactos de análisis y diseño.

ICONIX también define procedimientos y criterios para realizar cada una de las fases de verificación, se usarán las listas de verificación definidas en ICONIX para cada etapa de análisis y diseño.

En ICONIX una vez se alcanza un hito de revisión, los productos se encuentran listos para esta revisión también define los roles y los participantes de cada revisión, estos últimos se ajustarán según los roles definidos basados en Scrum. En cada hito de revisión se identifican y documentan los problemas encontrados.

ICONIX no contempla mediciones con respecto a la verificación sin embargo es posible medir la cobertura de la verificación comparando el número de casos de uso planteados en la iteración y el número de casos de uso incluidos en el hito de revisión.

Todos los aspectos identificados son tratados inmediatamente y se realizan las correcciones necesarias. Esta tarea puede ser combinada con el ciclo de gestión de Scrum para definir la discusión de estos temas en las reuniones periódicas definidas por el marco de trabajo.

Integración de productos (PI)

Para cubrir en alguna medida esta área de proceso se tendrá en cuenta la práctica de XP de Integración Continua, la cual permite tener un producto prácticamente listo para su uso al final de cada iteración. También se aprovecha el enfoque en su mayor medida del ADSCR a desarrollar software con poca relación con interfaces físicas o sistemas especiales, en estos casos se puede aprovechar XP con su rol de Personal de pruebas y la cercanía del cliente en sitio para garantizar que no solo se depende de las pruebas automáticas sino de

²⁴ Esta práctica se ha presentado espontáneamente dentro del ADSCR sin embargo puede llegar a ser agotador si se realiza constantemente durante largo tiempo, inicialmente se motivará su uso pero se implementará a discreción de los desarrolladores.

otro tipo de pruebas de usuario y de intercomunicación de sistemas o interfaces de hardware con el fin de garantizar el ensamblaje del producto al final de la iteración.

Con el fin de garantizar que cada componente del producto está listo para el ensamblaje se apoya en la práctica de escribir pruebas automáticas²⁵.

La empresa actualmente usa en su mayoría *frameworks* de desarrollo, lo que facilita el ensamblaje automático del producto en la mayoría de los casos.

También se aplica en este caso la práctica de Entregas cortas, así al final de cada iteración el cliente obtiene el producto ensamblado e instalado en el sitio de trabajo, el cual podría usarse en un entorno producción inmediatamente.

Administración de riesgos (RSKM)

Aunque en las prácticas ágiles analizadas no existe una gestión de riesgos formalmente definida, algunos autores (Paulk, 2001)(Fritzsche & Keil, 2007) consideran que el enfoque en general del método es la disminución del riesgo y de la incertidumbre, aplicando prácticas como las entregas cortas, la programación en parejas²⁶ y la integración continua dado que de esta manera que se disminuye la brecha entre los requerimientos del cliente y el producto construido. Además al incluir el marco de trabajo de Scrum se involucra al equipo en discusiones periódicas que permiten la identificación de los impedimentos y se dan espacios para la resolución de los mismos.

Aunque la implementación de esta área de proceso parece simple el ADSCR se ve beneficiado en cualquier forma debido a la actual ausencia de prácticas sobre gestión de riesgos identificada en la evaluación del proceso.

Monitoreo y control de proyectos (PMC)

En esta área de proceso Scrum es el más indicado debido a su enfoque en gestión de proceso.

²⁵ Esta práctica se deriva del desarrollo conducido por pruebas (TDD), este no se incluye en su totalidad en este punto debido a su complejidad y en la necesidad de la madurez del equipo.

²⁶ Aunque esta práctica es controversial, en el ADSCR se ha encontrado que esta se realiza de manera natural aunque no en la totalidad de las ocasiones. Los desarrolladores tienden a compartir jornadas de trabajo con el fin de realizar transmisión de conocimiento, resolver dudas o corregir bugs.

La reunión de sincronización permitirá cumplir con varias de las prácticas estipuladas en esta área, allí se puede realizar el monitoreo y la revisión rápida del progreso mediante el gráfico *burndown* del Sprint²⁷.

Aquí se toma nota también de los impedimentos para que el Scrum Master pueda gestionarlos inmediatamente o más tarde según la severidad del caso.

Aunque no hay un momento específico que indique la revisión de los datos, Scrum define claramente que esta tarea es del Scrum Master, para claridad se agregará específicamente esta actividad.

En las reuniones retrospectivas se tomarán medidas para los casos más significativos que hayan presentado desviaciones importantes del plan de la iteración, una vez solucionadas serán removidas de la lista de impedimentos.

Los hitos se revisan al final de cada sprint, en la reunión de revisión, allí se muestran los avances al propietario del producto y este evalúa el progreso del proyecto.

Desarrollo de requerimientos (REQD)

El ADSCR desea un poco de formalismo, pero no se utilizará en su totalidad las prácticas de ICONIX debido a que es muy estricto.

De ICONIX se tomarán las guías para recolectar requerimientos, las prácticas de revisiones y el uso de prototipos y diseños de pantalla.

ICONIX usa los casos de uso para ilustrar los requerimientos así que realmente no parte de los requerimientos funcionales y por lo tanto no trata el tema de su priorización, quedando fuera de su alcance. Esta priorización se realizará a nivel de iteraciones usando prácticas derivadas de Scrum pero a nivel de casos de uso, no de requerimientos.

Los requisitos se obtienen mediante otras actividades fuera del alcance de ICONIX y basado en estos se construyen los casos de uso, estos a medida que avanza el análisis son especificados con el fin de detallar sus elementos.

Los requisitos y casos de uso se validan en un hito de revisión de requerimientos según la lista de chequeo propuesta por ICONIX y en una presentación final del plan de iteración con los interesados.

²⁷ Inicialmente se usará un formato en Excel para registrar la información del esfuerzo y así generar esta medida de velocidad.

Validación (VAL)

Desde el principio se incluye al cliente en forma de dueño de producto con el fin de validar los requerimientos, al terminar el Sprint se presentan todos los elementos incluidos en este y se recibe retroalimentación del dueño del producto quien indica si considera hecho cada elemento o no.

Gracias a que se incluye al cliente desde el inicio se identifica fácilmente los requisitos del entorno pero no se cuenta con una planeación específica sobre la disponibilidad de los recursos, Scrum asume que esta infraestructura necesaria ya existe.

Las iteraciones continuas permiten mantener los cambios en el entorno y los procedimientos pero Scrum no define como documentar estos entornos.

Gracias a la participación del usuario en el sitio de desarrollo este participa activamente en el diseño y realización de pruebas.

Al finalizar el sprint el incremento del producto es presentado al cliente y los interesados determinando si el incremento es aceptado o si deben realizarse ajustes. Los elementos aceptados y no aceptados son actualizados en la pila del producto para su seguimiento.

Administrar los requerimientos (REQM)

Durante el *pre-game* se definen los roles incluyendo al propietario del producto y los clientes representativos, estos proveen los requerimientos y apoyan al equipo proporcionando detalles sobre los mismos.

Los compromisos se adquieren durante las reuniones de planificación, allí se refinan y priorizan los requerimientos del Backlog del producto y se obtiene el Backlog del Sprint que no se modifica durante su duración.

Los cambios se gestionan iterativamente en cada Sprint. Durante las reuniones de planificación de cada Sprint se actualizan los cambios en los Backlogs y se mantiene la información del cambio guardando los requisitos, cambios y ajustes. Todo el equipo tiene acceso al Backlog²⁸.

²⁸ Inicialmente se usará un formato en Excel con el fin de realizar este seguimiento.

A través de los Backlogs de producto y de Sprint es posible la trazabilidad bidireccional entre los requerimientos y los entregables.

Durante la reunión de planificación del Sprint el equipo junto con el cliente revisan la consistencia de los planes y los Backlogs. Allí mismo si hay inconsistencias se toman acciones correctivas inmediatamente.

Administración integrada de proyectos (IPM)

Todos los temas externos y de autorización, incluso temas sobre la coordinación de varios equipos, son tratados y resueltos mediante las reuniones, seguimiento y tratamiento de inconvenientes. El equipo trata internamente temas como integración, pruebas, etc.

Aunque SCRUM no es un proceso derivado de los procesos organizacionales si define los activos para realizar la gestión del proyecto incluyendo mediciones²⁹, revisión de las necesidades e impedimentos y cambios en los requisitos.

Scrum define claramente la construcción del equipo y sus funciones, comparte la visión inicial del proyecto. La evaluación periódica se cumple tratándolo como impedimentos a su debido momento.

En las reuniones retrospectivas se trata el tema de mejora del proceso y de aprendizaje de lecciones y se retroalimenta el proceso para la siguiente iteración. Las mediciones históricas de estimación y ejecución se utilizan para ajustar y mejorar la estimación del equipo y ayuda a medir la velocidad de los equipos

Scrum involucra a los interesados desde el inicio del proyecto y es monitoreado por el Scrum Master

Durante las reuniones de planificación de Sprint se profundiza con el cliente el entendimiento de las funcionalidades, cualquier dependencia es analizada en este punto y negociada con él antes de iniciar el sprint. Durante la revisión del sprint se verifica el cumplimiento de las funcionalidades.

Las situaciones se identifican en las reuniones diarias y de revisión del sprint. Y es el Scrum Master el encargado de ayudar a resolverlas. Solo salen del tablero las situaciones resueltas.

²⁹ Para el piloto se planteará el uso la medición de entrega de objetivos y velocidad.

Una vez se defina e institucionalice es posible que se logre mayor cobertura en esta área de proceso debido al nuevo cumplimiento de las prácticas específicas 1.1, 1.2 y 1.3.

Resumen

En la Tabla 23 Se puede observar los principales elementos que cada metodología aporta al área de proceso correspondiente, ICONIX ayuda a fomentar el formalismo mediante artefactos, guías y listas de chequeo tanto en REQD y VER. Scrum aporta los espacios para realizar las revisiones mediante sus reuniones periódicas en PMC y a incluir al cliente como dueño del producto y quien acepta los resultados para aportar a VAL. Por su lado XP define las prácticas técnicas base del equipo de desarrollo y del manejo del producto apoyando a PI, ayuda también a definir simple y concretamente los ítems de configuración y su manejo en CM, en general el uso de estas prácticas ayuda a disminuir algunos de los riesgos más importantes del desarrollo de software.

Tabla 23. Visión general del aporte de las prácticas ágiles a cada área de proceso

ICONIX	SCRUM	XP
REQD Artefactos y guías para Análisis y Diseño.	PMC Revisión del avance, Revisión de los logros, monitorio acciones correctivas.	CM Prácticas de codificación y pruebas del producto. Relación con el equipo de desarrollo.
VER Listas de chequeo y guías de Revisión en general.	VAL Involucrar al cliente en sitio, Realización de pruebas y demostraciones con el cliente.	PI Guías de compilación, presentación y puesta en producción del producto.
	REQM Definición de fuentes de requerimientos, negociación y gestión de los alcances de cada Iteración.	RSKM Integración con el cliente. Apoyo en la disminución general del riesgo.
	IPM Gestión de inconvenientes,	

involucrar y definir a los interesados, negociación de los alcances.

Apoya al área RSKM suministrando las reuniones diaria y retrospectiva que apoyan las prácticas.

7.3.2. Diseño del nuevo proceso de desarrollo

Las metodologías escogidas para este proyecto abarcan áreas que son complementarias, Scrum tienen un énfasis en el proceso y en la gestión del equipo, XP se concentra en las habilidades y tareas de los desarrolladores y de la comunicación entre estos y el cliente mientras que ICONIX tiene un énfasis en los artefactos de análisis y diseño y como construir el producto correcto. Para la construcción de los siguientes roles, artefactos, actividades y demás se tomaron como base las librerías de prácticas EPF referentes a Scrum y XP (EPF Team)

SCRUM

Scrum ha sido seleccionado como base para definir las prácticas de PMC, VAL, REQM, e IPM las cuales no solo aportan a estas áreas sino que también sirven de apoyo para otras áreas indirectamente como RSKM, CM, PI.

Scrum aporta la definición básica de roles (Dueño del producto, Scrum Master y el Equipo) con el fin de generalizarlos a las otras prácticas, los nuevos roles serán: Dueño del Producto que tendrá la misma definición que en Scrum, Desarrollador que representará a cada integrante del equipo destinado a la construcción del producto, el facilitador que es análogo al Scrum Master.

A estos roles suministrados por Scrum se agregarán otros tres derivados de la cultura organizacional de la empresa, coordinador de desarrollo que tradicionalmente es el encargado de supervisar todos los equipos y proyectos, personal de pruebas y personal de apoyo. Estos últimos representan al personal interdisciplinario del rol original de equipo de trabajo debido a que los desarrolladores se encuentran limitados en la empresa al personal

técnico dedicado principalmente al diseño y codificación del producto y por lo tanto excluye otras especialidades como diseño visual, redes, administración de base de datos, etc.

Dueño del Producto:

El dueño del producto representa a los interesados, define las características deseadas del producto y puede priorizar la lista de casos de uso del plan de iteración.

Responsabilidades:

- Definir las características del producto.
- Determinar las fechas de entrega requeridas y restricciones de tiempo.
- Priorizar las características del producto de acuerdo a las necesidades de los interesados.
- Ajustar las prioridades y características para cada iteración.
- Aceptar o rechazar las funcionalidades entregadas.
- Guía de selección:
- Una sola persona dedicada al proyecto, disponible para responder a las preguntas del equipo.
- Es un miembro adicional para el equipo y participa de la planeación de la iteración y de la reunión de revisión.
- Generalmente alguien perteneciente al departamento o empresa que requiere la solución.
- Puede ser un representante del cliente o intermediario con el cliente.

Desarrollador:

Uno o más desarrolladores son los encargados de construir el producto que necesita el cliente. Los desarrolladores son autónomos y auto-organizados y componen el núcleo del equipo de desarrollo. Especifican el trabajo que debe realizarse para cumplir con el objetivo de cada iteración y realizan las estimaciones de tiempo para completar cada una de estas tareas.

Uno o varios desarrolladores estarán encargados de realizar la demostración al cliente del producto y capacitar al cliente o al personal de soporte.

Cada desarrollador debe estar en capacidad todo el ciclo de vida del software incluyendo análisis, diseño, desarrollo y pruebas.

Facilitador:

Es el responsable de asegurarse que el equipo aplique los valores y prácticas del proceso. Encargado de proteger al equipo de no comprometerse más allá de su capacidad con el fin de cumplir el objetivo de la iteración.

Ayuda a que se realice la reunión de actualización y a remover los obstáculos que se detecten durante estas reuniones.

Responsabilidades:

- Representa la administración ante el proyecto.
- Es responsable de promover los valores y prácticas del proceso.
- Dirige las reuniones de actualización y seguimiento.
- Quita los impedimentos
- Protege al equipo de interferencias externas.
- Se asegura que el equipo esté completamente funcional y productos.
- Permite que haya cooperación entre todos los roles y funciones
- Realiza la revisión periódica de los artefactos producidos por el equipo.

Guía de Selección:

El facilitador es generalmente un gerente de proyecto o un líder técnico del equipo, pero puede ser cualquier individuo con conocimiento y convicción en el proceso de desarrollo.

Consideraciones:

- No es un gerente de proyecto, es más un líder del equipo o un administrador del proceso.
- Revisa las tareas de la iteración con el fin de asegurar su éxito, puede ayudar a crear y asignar las tareas.
- Debe permanecer físicamente junto al equipo y reunirse con ellos periódicamente.

Coordinador de Desarrollo:

El coordinador de desarrollo es el representante administrativo de la empresa o área de desarrollo, es quien tiene poder de decisión administrativo con el fin de asignar o remover recursos, definir restricciones presupuestales o de cronograma y es quien representa al equipo frente a entes administrativos y de control.

Responsabilidades:

- Definir el equipo de trabajo.
- Ayudar a resolver los impedimentos detectados por el facilitador cuando esto sobrepasen su alcance.
- Obtener recursos para capacitaciones, equipos, asesorías cuando el equipo y el proyecto lo requieran.
- Dar por terminado el proyecto cuando así lo considere necesario.

Guía de Selección:

El coordinador de desarrollo es generalmente un gerente de proyecto o un líder técnico con experiencia y recorrido administrativo, generalmente es un asistente del líder del área de desarrollo. Debe tener habilidades de negociación con el fin de obtener y gestionar los recursos necesarios para el proyecto.

Consideraciones:

- Debe tener empoderamiento con el fin de poder tomar decisiones administrativas y presupuestales o al menos estar en capacidad de gestionarlos con la dirección correspondiente.
- Este rol es generalmente desempeñado por un individuo o pequeño grupo de personas que dan apoyo a varios equipos.

Personal de pruebas:

Este rol ayuda al dueño del producto a escribir y ejecutar pruebas de aceptación.

Este se encarga de recibir la capacitación correspondiente con el fin de realizar pruebas en conjunto con los usuarios finales del producto, se encarga de registrar las pruebas y el resultado obtenido.

Responsabilidades:

- Llevar a cabo las pruebas en acompañamiento de los usuarios finales del sistema.
- Documentar las pruebas realizadas y sus resultados.
- Retroalimentar al equipo para realizar las correcciones o ajustes necesarios al producto.

Personal de apoyo:

Brinda ayuda en cualquier habilidad que es necesaria para completar el producto y con la que el personal de desarrollo no cuenta.

El personal de apoyo incluye a todo el personal necesario para el desarrollo del producto que no incurre en labores de codificación como diseñadores gráficos, administradores de sistemas de información, expertos en temas específicos entre otros.

Responsabilidades:

- Apoyar al equipo en las labores especializadas de su dominio que escapa al alcance de los Desarrollos.

Para definir el flujo de actividades se usará básicamente el ciclo de vida de Scrum. El producto se construye con iteraciones de corta duración, realizando entregas incrementales al cliente, se usan la reunión de planeación, la reunión de revisión, la reunión diaria y la reunión de retrospectiva.

Reunión de Planeación de iteración:

En esta reunión el equipo y el dueño del producto determinan las características deseadas para el producto.

A esta reunión asiste el Dueño del producto, el equipo de desarrollo, el facilitador y cualquier otro interesado en el proyecto. Durante esta reunión el dueño del producto describe las características de mayor prioridad al equipo.

El equipo puede realizar todas las preguntas que considere necesarias para realizar una primera estimación del trabajo necesario y de las relaciones entre las funcionalidades deseadas.

Después de la reunión el equipo se reúne solo, para decidir qué tanto puede comprometer en la iteración actual.

Esta reunión debe dar como resultado una lista de los componentes del producto que el equipo se compromete a realizar durante la iteración además elementos de trabajo de alto nivel que se requieran (como una migración de datos, instalación de un servidor, etc), asignando una estimación de tiempo y los integrantes del equipo que trabajaran en el desarrollo de cada característica.

Consideraciones:

No es necesario discutir cada detalle de cada elemento de la lista de características deseadas puede ser suficiente con mencionar las de mayor prioridad. El equipo recibirá información detallada durante la iteración en acompañamiento con el dueño del producto.

Esta reunión se divide en 2 fases la primera con el dueño del producto y otra solo del equipo, cada una de estas fases puede tener una duración máxima de 4 horas.

Reunión diaria de actualización:

Es una reunión rápida de 15 a 30 minutos máximo, donde se reúnen los integrantes del equipo y el facilitador.

Cada día de la iteración, el equipo se reúne a una hora y en un lugar específico, generalmente en la mañana con el fin de entrar en contexto del trabajo que se realizará durante el día.

En esta reunión no se resuelven problemas, estos se tienen en cuenta pero se resuelven aparte, generalmente luego de la reunión. Cada miembro del equipo debe responder a las preguntas: Qué se hizo ayer, qué se piensa realizar el día de hoy y qué impedimentos tiene.

Con esto se obtiene una visión clara del estado de cada actividad y característica del sistema, de lo que ya está hecho y lo que está pendiente por realizar.

Esta reunión ayuda a comunicar a cada miembro del equipo el estado de sus compromisos.

El facilitador debe recolectar y documentar la lista de impedimentos con el fin de gestionarla, ya sea resolviendo los impedimentos o escalándolos a niveles superiores si estos están fuera de su alcance.

El facilitador puede llevar a esta reunión una lista de los compromisos actuales y futuros del equipo y su supuesto estado actual con el fin de ayudar a los integrantes del equipo a presentar su estado.

Reunión de Revisión:

Al final de cada iteración se realiza la reunión de revisión en la que el equipo de desarrollo muestra el avance del producto al cliente y/o interesados.

Al final de la iteración el equipo muestra lo que se completó durante esta etapa generalmente es una demostración del producto presentando las nuevas funcionalidades. En esta reunión participan el dueño del producto, el equipo de desarrollo, el facilitador, el cliente y cualquier otro interesado en el proyecto.

Durante esta reunión se evalúa el producto frente al plan inicial de la iteración, se espera que todos los elementos del plan se cumplan pero es posible que algunos ítems queden por fuera del alcance y tenga que reevaluarse en una siguiente iteración.

Reunión retrospectiva:

Luego de la reunión de revisión, el equipo se reúne solo con el fin de analizar el proceso y la forma de trabajo durante la iteración, con el fin de proponer mejoras o cambios.

Esta reunión es una oportunidad para que el equipo se enfoque en discutir qué está funcionando y qué no lo hace. Una forma de iniciar esta reunión es que cada miembro del equipo comparta con el resto qué cosas nuevas deberían probar, cuales no están funcionando y quieren dejar de hacerlas y cuáles están bien como están y desean continuar realizando.

Consideraciones:

Es recomendable informar de los resultados más importantes de esta reunión a un observador externo al equipo, una persona muy interesada en esta reunión es el coordinador de desarrollo con el fin actualizarse y comprender como están funcionando los equipos y qué podría mejorarse al proceso en general.

También se necesitan algunos artefactos de Scrum con el fin de lograr el objetivo de estas reuniones, el más importante y alrededor del cual gira Scrum es la Pila del producto y del cual se deriva la lista de tareas.

Pila del producto:

Es la lista maestra de todas las funcionalidades deseadas en el producto.

Una vez analizados los requerimientos, estos se convierten en casos de uso que agrupan las funcionalidades requeridas por el cliente y están ordenados de una forma lógica de manera que agreguen valor al cliente de forma incremental a medida que avanzan las iteraciones.

Esta lista puede crecer y ser modificada a medida que avanza el proyecto durante las reuniones de planificación de iteración.

La lista es priorizada colocando primero los ítems de mayor valor para el cliente o que simplemente son necesarios antes que otros que dependen de ellos, cada ítem es estimado en horas y es asignado a un miembro del equipo a medida que se avanza en el proyecto. Una vez terminado un ítem es necesario registrar el tiempo real invertido³⁰ en la finalización del ítem.

En la reunión de planificación de iteración se actualizarán las estimaciones con la explicación detallada que suministra el cliente y se define el número de ítems que el equipo estima puede completar durante la duración de la iteración.

Lista de Tareas:

La lista de tareas contiene actividades adicionales que son necesarias para el cumplimiento del objetivo de la iteración como instalar un servicio, investigar una nueva tecnología o realizar una migración de datos. Estas tareas se detallan durante la iteración y son asignadas y estimadas al igual que los elementos de la pila del producto.

ICONIX

En el caso de ICONIX debido a que se usará el proceso base definido por Scrum, se tomarán solo algunos elementos, estos cubrirán las áreas de VER y REQD pero dan apoyo indirecto para áreas como PMC e IPM. Se tomará como fuente inicial para ICONIX la lista de requerimientos funcionales del sistema a partir de los cuales se construirá la vista

³⁰ Para el caso del ADSCR se desarrollará una herramienta para el control del proyecto, el cual incluye el registro del tiempo real invertido.

dinámica del sistema incluyendo casos de uso con sus respectivos prototipos y especificación, el diagrama de clases de diseño y el diagrama de secuencia. Con respecto al modelo estático solo se tomará el modelo de clases de diseño. Estas prácticas y artefactos apoyan directamente al área de proceso de Desarrollo de requerimientos REQD³¹.

La especificación de requerimientos está un poco fuera del alcance propuesto por ICONIX, así que es necesario agregar unas actividades adicionales para obtener esta información que es fundamental para el resto del proceso.

Entrevista con el cliente:

Al inicio del proyecto luego de presentar el equipo y su alcance, uno o varios miembros del equipo se reunirán con el dueño del producto o algunos de los usuarios finales del producto con el fin de recolectar la mayor cantidad posible de información, es recomendable grabar esta entrevista.

Se recomienda preparar esta entrevista con antelación, esta será el punto de inicio del proyecto y el material con el cual se definirán los requerimientos del producto.

Visita en sitio:

La visita en sitio en el lugar donde se usará el producto por los usuarios finales, da una visión general del personal y forma de trabajo y uso del producto esperado.

Una vez este en el lugar observe la forma de trabajo actual con el fin de descubrir:

- Forma y orden como se hacen los procedimientos actuales y su tiempo de duración.
- Hardware e infraestructura disponible.
- Personal disponible.
- Perfil o capacitación de los usuarios finales en sus diferentes roles.

Este paso es opcional, pero muy recomendado.

³¹ Para el caso del ADSCR estos artefactos serán creados en la herramienta Enterprise Architect con el fin de facilitar su tratamiento y gestión de la configuración.

Realizar la especificación de requerimientos:

Uno o varios desarrolladores deben revisar las entrevistas y reuniones con el cliente o el dueño del producto y redactar una lista de frases que describan cada uno de las necesidades de estos frente al producto que se construirá. Cada frase es generalmente de la forma “El sistema debe permitir realizar la asignación de tareas a un individuo”, cada una de estas características debe ser fácilmente verificable.

Lista de requerimientos:

El resultado de la especificación de requerimientos es una lista ordenada de todas las necesidades mencionadas por el cliente en la entrevista o reuniones adicionales. Se recomienda el uso de una herramienta CASE con el fin de registrar estos requerimiento de manera que se puedan vincular gráficamente “Arrastrar y soltar” a los casos de uso.

Diagrama de casos de uso:

El diagrama de casos de uso es una representación del comportamiento de los requerimientos, los casos de uso describen la forma en que el usuario interactuará con el sistema y cómo responderá este al usuario.

Cada caso de uso cumple con la realización de uno o más requerimientos, es recomendable usar una herramienta CASE que permita relacionar requerimientos y casos de uso.

Es común que cada caso de uso se relacione con un prototipo de pantalla que aclare cómo se comunicará el usuario con el sistema y mediante qué mecanismos este le responderá al usuario. Usar un prototipo facilita la aceptación de la propuesta de diseño de alto nivel por parte del dueño del producto y mejora la comunicación de la visión del equipo de desarrollo.

Es recomendable agrupar los casos de uso según su funcionalidad e interacción en módulos, el grado de dependencia de estos ayudará a determinar el alcance de cada iteración.

Diagrama de clases de análisis:

El diagrama de clases de análisis relaciona los casos de uso con objetos, para este fin se utilizan 3 estereotipos, la frontera, la entidad y el controlador, estos serán análogos a las pantallas que presentará el producto, los objetos de almacenamiento, ya sea en memoria o

en almacenamiento persistente y los objetos que contendrán el comportamiento o lógica de negocio del producto.

Objetos de entidad o control generalmente se comparten entre diferentes casos de uso.

Las entidades se compondrán principalmente de atributos identificados en los casos de uso estos crecerán y se complementarán a medida que avance el análisis sobre otros casos de uso que tenga relación con la entidad común.

Los controladores generalmente se componen de métodos que responden o interactúan con el cliente u otras áreas del sistema.

Diagrama de secuencia:

El diagrama de secuencia permite explorar el diseño en detalle analizando principalmente el escenario de flujo normal para los casos de uso principales o considerados “núcleo del negocio” solo se considerarán otros escenarios cuando el caso de uso contenga flujos alternos o excepciones realmente relevantes. Este diagrama permite ubicar los métodos en el mejor lugar para el sistema y ayuda a tomar decisiones sobre el diseño de los algoritmos.

Diagrama de clases de diseño:

El diagrama de clases de diseño se deriva directamente de las clases de análisis, aquí se depura y definen las clases que serán implementadas. Estas se encuentran más cerca de la implementación final y su agrupación o división dependerá del lenguaje o *framework* de desarrollo seleccionado, estas representaran clases reales a ser codificadas con sus atributos y sus respectivos tipos de dato, al igual que sus métodos con la definición de sus entradas y salidas correspondientes. Se recomienda solo realizar este diagrama solo para las clases relacionadas con casos de uso principales (núcleo del negocio).

Modelo entidad relación:

El modelo entidad relación especifica las entidades que deben ser almacenadas en el medio permanente. En algunos casos este será generado directamente por el *framework* de desarrollo, en caso contrario este debe construirse basado en las entidades realizando las definición de tipos de datos y relaciones según el motor seleccionado.

ICONIX define 3 hitos: La revisión de los requerimientos, la revisión de diseño preliminar y la revisión de diseño crítico. Para el nuevo proceso se definirán 2 hitos, la revisión de requerimientos y la revisión de diseño que reúne los 2 últimos hitos de ICONIX. Estos puntos de revisión apoyan directamente el área de proceso de Verificación (VER).

Revisión de requerimientos:

Una vez terminada la especificación de los requerimientos, se revisan los documentos de requerimientos, y casos de uso con sus respectivas especificaciones y prototipos de pantalla con el fin de garantizar su coherencia y completitud. Finalmente se realiza una reunión con el dueño del producto, el equipo de desarrollo y cualquier otro interesado en el proyecto. Usando los casos de uso y los prototipos de pantalla se presentan las funcionalidades que se esperan implementar y como el usuario interactuará con el sistema.

Se debe tener en cuenta los siguientes ítems:

- Asegúrese que los modelos están escritos en su mayoría en lenguaje no técnico que el cliente pueda entender.
- Asegúrese que todos los requerimientos están relacionados con al menos un caso de uso.
- Asegúrese de que cada caso de uso realiza al menos un requerimiento.
- Organice los casos de uso en paquetes o módulos de acuerdo con su funcionalidad.
- Verifique que los casos de uso se encuentran representados en algún prototipo de pantalla.
- Recoja todas las sugerencias y observaciones del cliente y repita este paso de ser necesario.
- Asegúrese que en los casos de uso se describe la interacción en ambos sentidos entre el sistema y el usuario.

Revisión de diseño:

Una vez terminado el diseño y justo antes de iniciar la codificación de cada caso de uso de la iteración se revisan los artefactos relacionados con el análisis detallado y el diseño del producto.

Esta revisión puede ser realizada por el facilitador o por otro integrante del equipo con el fin de obtener una segunda visión del diseño y de su nivel de calidad.

Para esta revisión se debe tener en cuenta:

- Asegurarse que el texto del caso de uso corresponde al diagrama de clases de análisis.
- Verifique que se escriben los casos alternos y cómo responderá el sistema en caso de errores o excepciones.
- En lo posible verifique con el dueño del producto que todas las entidades y atributos necesarios se encuentran descritos en el diagrama de clases de análisis.
- Asegúrese que los diagramas de clases de análisis corresponden con los prototipos de pantalla (a nivel de datos).
- Asegúrese que los diagramas de secuencia concuerdan con la especificación de los casos de uso.
- Revise nuevamente que los métodos están en las clases correctas.
- Revise los diagramas de secuencia verificando que son viables y cumple de la mejor manera con el funcionamiento del sistema.
- Verifique los tipos de datos, que estos concuerdan con los esperados y sus longitudes son coherentes.
- Asegúrese que cada módulo (o clase) tiene pocas responsabilidades altamente relacionadas.
- Revise que las responsabilidades de las clases correspondan a la arquitectura definida para el sistema (o en su defecto que cumplan con los estándares definidos).

XP

En el caso de XP, las prácticas de seleccionadas apoyan directamente las áreas de CM, PI y RSKM.

Para XP el código es el elemento más importante y es por esto que se convierte para el nuevo proceso en el ítem de configuración principal. A este se le agregan los ítems de configuración provenientes de ICONIX en relación con el análisis y diseño para ser gestionados de la misma forma y mediante una herramienta automática de versionado³². Finalmente Scrum también aporta ciertos ítems de configuración como son los resúmenes de las reuniones³³, que ayudan a mejorar la trazabilidad. Para este fin se define una

³² En el caso del ADSCR se usa actualmente CVS, pero igualmente podría usarse una herramienta más actual como GIT.

³³ En el caso de las reuniones aunque no son susceptibles a cambios y solo es requerida generalmente la primera versión, se incluyen por facilidad para ser gestionadas junto con los artefactos de análisis y diseño

actividad relacionada a la gestión de la configuración por parte del equipo y se agrega una tarea de revisión con el fin de garantizar el seguimiento de estas prácticas.

Crear estructura base:

Esta actividad realizada por el coordinador de desarrollo o un representante asignado por el mismo, es el punto de partida para la gestión de la configuración, aquí se asigna un nombre clave para el proyecto (preferiblemente una sola palabra), con este nombre se creará un repositorio en la herramienta elegida para la gestión de esta tarea, este repositorio deberá ser alimentando preferiblemente con una plantilla guía con el fin de ayudar al equipo a la gestión de los ítems diferentes al código que se han definido.

Revisión diaria:

Diariamente el facilitador o un integrante encargado del equipo, mediante la herramienta de gestión de la configuración, verifica que cada miembro del equipo este siguiendo las prácticas correspondientes. El encargado se asegura que diariamente se integre ya sea elementos de análisis diseño o código fuente a los repositorios correspondientes y que estos cumplan con los estándares esperados. Esta revisión facilita la revisión final tanto de análisis como de diseño. El equipo será retroalimentado inmediatamente de las desviaciones con el fin de ajustarse a los compromisos lo más pronto posible.

Con el fin de que el proyecto avance a la velocidad deseada sería recomendable que el dueño del producto se encontrara disponible en todo momento en el mismo espacio físico que el equipo como lo proponen las prácticas ágiles, sin embargo esto muchas veces no es posible en el ADSCR, por este motivo es necesario incluir una actividad llamada reunión adicional.

mediante la herramienta Enterprise Architect. Además esto ayuda a tenerlas accesibles y centralizadas para su posterior consulta.

Reunión Adicional:

Esta puede realizarse en cualquier momento durante la iteración con el fin de resolver dudas o tomar decisiones respecto a impedimentos que involucran decisión del cliente. Debe incluir al menos un representante del cliente y uno o más integrantes del equipo.

Con respecto a la codificación del producto se seguirá las prácticas propuestas por XP de propiedad colectiva del código, la programación en parejas, la refactorización y la integración continua³⁴.

Propiedad Colectiva:

La práctica de propiedad colectiva declara que cualquier integrante del equipo puede cambiar cualquier parte del código en el sistema en cualquier momento.

Esta práctica en conjunto con la integración frecuente y las pruebas aseguran que los problemas se identifiquen a tiempo y se arreglen rápidamente. En conjunto con la programación en parejas, la propiedad colectiva es una forma efectiva de diseminar el conocimiento del sistema a todo el equipo.

Beneficios:

- Conocimiento compartido del código: Permite a los programadores familiarizarse con más código y beneficiarse de la experiencia de otros.
- Código sencillo: Ayuda a que se encuentre el código complejo y sea refactorizado más rápido debido a que más personas leen el mismo código.
- Resolver las cosas más rápido: Elimina los obstáculos permitiendo que los cambios se hagan por aquellos que lo necesitan y cuando lo necesitan.

³⁴ No se incluyen más prácticas con el fin de facilitar la implementación del proceso y generar menos presión sobre los desarrolladores facilitando al mismo tiempo la adopción de las nuevas actividades. Con el tiempo podrían añadirse nuevas prácticas a medida que el equipo madure y haya interiorizado las primeras.

Programación en parejas:

El producto es generalmente codificado o revisado por 2 personas, sentados lado a lado trabajando sobre la misma máquina. Esta práctica permite que el código de producción sea revisado por al menos otra persona aparte del programador, lo que resulta en un mejor diseño, mejores pruebas y mejor código.

La investigación sobre programación en parejas muestra que las parejas producen mejor código en el mismo tiempo que los programadores trabajando solos.

Trabajar en parejas también ayuda a la comunicación del conocimiento a través del equipo. A medida que las parejas cambian, todos se ven beneficiados del conocimiento especializado de los demás. Los programadores aprenden, mejoran y se vuelven más valiosos para el equipo y la empresa.

Beneficios:

- Mejora el diseño, el código y las pruebas.
- Se comparte el conocimiento y habilidades en el equipo

Refactorización:

Refactorización es la práctica de mejorar el diseño del programa sin cambiar su comportamiento. Esta práctica es crítica en el desarrollo iterativo. El programador generalmente está agregando una nueva funcionalidad o refactorizando. Algunas refactorizaciones pueden ser triviales como renombrar o mover cosas, pero otras pueden ser más complejas.

A medida que se resuelven errores y se agregan funcionalidades adicionales el código tiende a perder calidad, en estos casos la refactorización permite a los programadores mantener un buen diseño. La refactorización también puede mejorar el diseño de un sistema ya existente.

Si un producto no es refactorizado a medida que es modificado, el diseño generalmente se deteriora o los métodos se vuelven grandes, las clases toman más responsabilidades y más código es copiado y pegado a través del sistema dificultando el mantenimiento del código.

Las pruebas automáticas proveen una red de seguridad cuando se realizan cambios, estas pueden reportar cuando la funcionalidad del sistema cambia y garantizar que luego de realizar un cambio estructural las pruebas continúan resolviéndose con satisfacción.

Beneficios:

- Evita que se deteriore el diseño.

- Reduce el costo de los cambios.

Integración Continua:

A medida que se construye incrementalmente el producto, las nuevas funcionalidades se integran continuamente y son demostradas al cliente.

La integración puede realizarse varias veces al día, pero en general debería realizarse al menos una cada día, generalmente al terminar la jornada. A medida que los desarrolladores terminan algún trabajo, integran lo que han realizado.

Como se menciona en el apartado de Scrum, se usará un ciclo iterativo, sin embargo el ADSCR requiere un estimado inicial del total del proyecto con el fin de realizar compromisos y asignar presupuestos. Para cumplir este objetivo se usará una iteración 0 llamada “Análisis Preliminar”³⁵.

Análisis Preliminar:

Todo proyecto inicia con una iteración llamada análisis preliminar, el objetivo de esta iteración no es producir un incremento del producto sino realizar el plan de iteraciones. Esta iteración se compone al igual que las demás de la reunión de planificación de iteración, la reunión diaria, la revisión diaria y la reunión retrospectiva. Adicionalmente se agregan solo para esta iteración la reunión inicial, la entrevista inicial, la reunión en sitio y el hito de revisión de requerimientos, mencionados en el apartado de ICONIX.

Los elementos que servirán como suministro para realizar la planeación de iteraciones serán la lista de requerimientos y el diagrama de casos uso (incluyendo sus prototipos y especificación)

Planeación de iteraciones:

Durante la planeación de iteraciones se toma la pila del producto y realiza una estimación general del esfuerzo necesario por el equipo actual para terminarlo en su totalidad, incluyendo la asignación de cada caso de uso a una iteración.

³⁵ Esta estrategia es mencionada por varios autores entre ellos (Deemer, Benefield, Larman, & Vodde, 2009)

Al inicio del proyecto el equipo creará un plan de alto nivel, como el equipo no lo puede saber todo en esta etapa inicial, no se requiere un plan muy detallado.

Este plan puede cambiar en el tiempo según cambien las prioridades de cada característica, incluso el proyecto puede terminar antes o luego de lo esperado.

Cada iteración siguiente tomará como base este plan pero lo ajusta según las nuevas necesidades del cliente durante la reunión de planificación de iteración.

Consideraciones:

Para la construcción de este plan se debe tener en cuenta:

- El número y duración de las iteraciones
- El número de entregas.
- Los casos de uso que se entregarán luego de cada iteración.
- Las fechas aproximadas de entrega de cada iteración.

Plan de iteraciones

Este plan consiste en una lista de los casos de uso con su correspondiente estimación, agrupados por cada una de las iteraciones necesarias para completar el proyecto.

Este plan permite informar a todos los interesados del tamaño esperado del producto según las características deseadas que se implementen y el número de iteraciones que tomará para terminarlas en su totalidad.

Presentación del plan de iteraciones:

En esta reunión a la que asiste el dueño del producto, el coordinador de desarrollo y el equipo, se presenta el plan de iteraciones describiendo los incrementos al producto que se esperan completar luego de cada iteración y la duración propuesta para cada iteración³⁶. Debe existir muy poca discusión sobre este plan, ya que es una idea global del alcance más que un plan detallado, sin embargo si el cliente o la administración requieren una fecha de

³⁶ El equipo puede ajustar la duración de las iteraciones en una o dos semanas con el fin de ajustarlas mejor al tamaño de los casos de uso de manera que cada entrega sea lógica y funcional. Sin embargo una vez inicia la iteración esta duración se convierte en un compromiso y no debería modificarse.

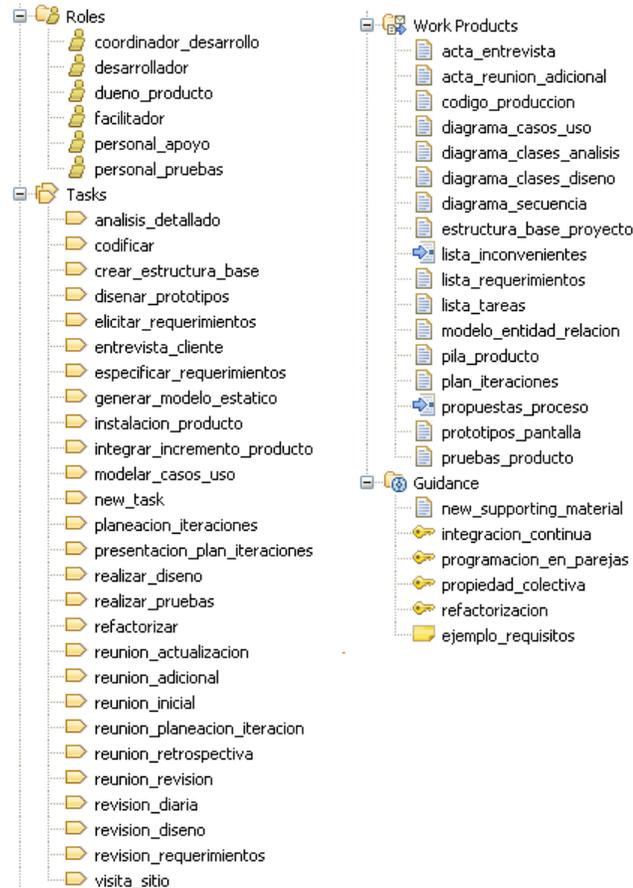
entrega que está fuera del alcance del plan este es el momento ideal para realizar la negociación con el fin de remover o agregar ciertas funcionalidades y así tener un producto que se apegue a los requerimientos más críticos pero que al mismo tiempo pueda ser entregado en la ventana de tiempo que se requiere.

7.3.3. Documentación del proceso en EPF

Este nuevo proceso propuesto para el ADSCR se ha documentado siguiendo el estándar SPEM allí se documenta claramente los roles, artefactos, actividades, guías y en general la implementación del proceso a seguir.

Inicialmente se definieron todos los elementos que componen el proceso, iniciando por los roles y cada una de las actividades que desarrollan o en que participan cada uno de estos, luego definiendo los resultados de las actividades en forma de productos de trabajo, finalmente se proveen algunas guías para facilitar la realización de las actividades (ver Figura 13).

Figura 13. Role, tareas, productos de trabajo y guías documentadas en EPF.



EPF genera una vista HTML para cada uno de los elementos anteriormente mencionados como lo muestra la Figura 14, permitiendo su navegabilidad y posterior consulta en una presentación agradable para los usuarios del proceso.

Figura 14. Ejemplo vista generada del rol Dueño del producto

Role: Dueño del producto

 El dueño del producto representa a los interesados, define las características deseadas del producto y puede priorizar la lista de casos de uso del plan de iteración.

[Expand All Sections](#) [Collapse All Sections](#)

Relationships



Additionally Performs	<ul style="list-style-type: none"> ● Presentación del plan de iteraciones ● Reunión Adicional ● Reunión de Revisión
Modifies	<ul style="list-style-type: none"> ● Acta de Entrevista ● Pila del Producto
Process Usage	<ul style="list-style-type: none"> ● MXP > Análisis Preliminar > Dueño del producto ● MXP > Desarrollo > Iteración Desarrollo 1..n > Dueño del producto

[Back to top](#)

Main Description

Responsabilidades:

- Definir las características del producto.
- Determinar las fechas requeridas de entrega y restricciones de tiempo.
- Priorizar las características del producto de acuerdo a las necesidades de los interesados.
- Ajustar las prioridades y características para cada iteración.
- Aceptar o rechazar las funcionalidades entregadas.

[Back to top](#)

Staffing

Assignment Approaches	<ul style="list-style-type: none"> ● Una sola persona dedicada al proyecto, disponible para responder a las preguntas del equipo. ● Es un miembro adicional para el equipo y participa de la planeación de la iteración y de la reunión de revisión. ● Generalmente alguien perteneciente al departamento o empresa que requiere la solución. ● Puede ser un representante del cliente o intermediario con el cliente.
------------------------------	--

Finalmente con cada uno de los elementos definidos, se construyen las fases e iteraciones del proceso definiendo sus dependencias, obligatoriedad, repetición entre otros. Las actividades ubicadas dentro del proceso traen consigo los roles, productos de trabajo y guías con los que se relacionan (ver Figura 15).

Figura 15. Proceso modelado en EPF

Presentation Name	In...	Predecessors	Model Info	Type	Planned	Repea...	Multipl...	Ongoing	Event...	Optional
MIXP	0			Delivery Pro...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis Preliminar	1			Phase	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Crear estructura base	2			Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reunión Inicial	3	2		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Entrevista con el Cliente	4	3		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Visita en sitio	5	4		Task Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Reunión de planeación de iteración	6	5		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reunión de actualización	7			Task Descrip...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Revisión Diaria	8	9,10,11		Task Descrip...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Especificar Requerimientos	9	6,12		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñar Prototipos	10	12		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelar los casos de uso	11	12		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reunión Adicional	12			Task Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Revisión de Requerimientos	13			Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Planeación de Iteraciones	14	13		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Presentación del plan de iteraciones	15	14		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reunión Retrospectiva	16	15		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Desarrollo	17	1		Phase	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Iteración Desarrollo 1...n	18			Iteration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reunión de planeación de iteración	19			Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reunión de actualización	20			Task Descrip...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar Análisis Detallado	21	19,28		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar Diseño	22	28		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Generar el Modelo Estático	23			Task Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Revisión de Diseño	24	23,28		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Desarrollar	25	28		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reunión Adicional	28			Task Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Revisión Diaria	29			Task Descrip...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar pruebas	30			Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrar un incremento del producto	31	30		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reunión de Revisión	32	31		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Instalación del Producto	33	32		Task Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Reunión Retrospectiva	34	33,32		Task Descrip...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Este proceso una vez publicado genera un paquete de archivos tipo wiki donde iniciando a partir de las actividades principales se puede consultar y navegar a través de todos los artefactos. Con el fin de obtener más detalles remítase al [Anexo G](#).

7.4. EVALUACIÓN DEL PROCESO ESTRUCTURADO PROPUESTO

7.4.1. Prueba Piloto

El proceso propuesto fue probado en el ADSCR en los módulos de legalización y administración del sistema de crédito desarrollado entre los meses de Mayo de 2013 y Enero de 2014.

Este proyecto fue elegido debido a que su tamaño aseguraba un equipo con suficientes integrantes para cumplir con la asignación de roles y con un tiempo para su desarrollo que permitiera la adopción del proceso y varias iteraciones para lograr realizar las dos fases del mismo.

El equipo inició el módulo de Legalización con dos desarrolladores, un asistente de proceso, el dueño del producto y el facilitador. Luego en Septiembre de 2013 se agregó al equipo dos nuevos desarrolladores.

Al iniciar el piloto ninguno de los desarrolladores contaba con experiencia real en ningún tipo de método o proceso.

El equipo fue reunido para dar inicio al proyecto y se definieron los objetivos y las nuevas reglas de juego. La instrucción se dio paulatinamente a medida que se avanzaba en el análisis incorporando a los elementos de gestión, los artefactos y actividades más técnicas.

Luego en Julio de 2013 el equipo, unido a otros integrantes del ADSCR participó en una capacitación con el fin de reforzar su conocimiento en ingeniería de software y el nuevo proceso de desarrollo.

7.4.2. Instrumento de evaluación

Con el fin de realizar una evaluación comparable con la realizada en el estado anterior del proceso, se usará el mismo instrumento usado en la fase de evaluación de las prácticas ágiles con respecto a CMMI, se usará la calificación detallada de 5 niveles como se muestra en la Tabla 11.

7.4.3. Resultado de la evaluación

El nuevo proceso usado en el piloto fue evaluado mediante el instrumento de comparación, cada práctica específica fue analizada a la luz de cada una de sus subprácticas de la misma forma en la que se realizó la evaluación de cada una de las prácticas ágiles, en la Tabla 24 se puede observar un ejemplo de la evaluación realizada³⁷.

³⁷ El método usado en este instrumento y sus convenciones se explican en el numeral 7.2.1

Tabla 24. Ejemplo evaluación realizada del nuevo proceso para la meta específica 3 de CM.

	COBERTURA	SUBPRACTICAS
SG3 MANTENER LA INTEGRIDAD DE LINEAS BASE		
53) SP 3.1 Establecen y mantienen registros que describen los elementos de las configuraciones.	PS-	<p>X1. Registrar las acciones de gestión de configuración con suficiente detalle para que se conozca el contenido y el estado de cada elemento de configuración, y para que se puedan recuperar versiones anteriores.</p> <p>OK2. Asegurar que las partes interesadas relevantes tengan acceso y conocimiento del estado de configuración de los elementos de configuración.</p> <p>X3. Especificar la última versión de las líneas base.</p> <p>X4. Identificar la versión de los elementos de configuración que constituyen una línea base particular.</p> <p>X5. Describir las diferencias entre líneas base sucesivas.</p> <p>OK6. Modificar, si procede, el estado y la historia (es decir, cambios y otras acciones) de cada elemento de configuración.</p>
OBSERVACION		
La falta de líneas base afecta el cumplimiento de esta práctica, también se evidencia en la práctica que las descripciones de los elementos en algunos casos no son suficientes para determinar su estado o para recuperar fácilmente versiones anteriores.		
54) SP 3.2 Realizan auditorías de configuración y mantienen la integridad de las líneas de base de las configuraciones.	S	<p>OK1. Evaluar la integridad de las líneas base.</p> <p>OK2. Confirmar que los registros de gestión de configuración identifican correctamente los elementos de configuración.</p> <p>OK3. Revisar la estructura y la integridad de los elementos en el sistema de gestión de configuración.</p> <p>OK4. Confirmar que los elementos en el sistema de gestión de configuración son completos, correctos y consistentes.</p> <p>OK5. Confirmar el cumplimiento con los estándares y procedimientos aplicables de gestión de configuración.</p> <p>OK6. Seguir los elementos de acción desde la auditoría hasta su cierre.</p>
OBSERVACION		
Todos los ítems están bajo escrutinio constante garantizando su integridad y el cumplimiento de los estándares aplicables.		

En general el resultado se muestra satisfactorio, mejorando en cada área de proceso enfocada a mejora. El resumen de este resultado se puede observar en la Tabla 25 y su detalle se puede encontrar en el [Anexo H](#).

Tabla 25. Resultado evaluación porcentaje cumplimiento frente a CMMI del nuevo proceso de software.

Area de Proceso	Comfamiliar	Nuevo proceso
1. Administración de Requerimientos (REQM)	70%	100%
2. Planificación de Proyectos (PP)	66%	66%
3. Monitoreo y Control de Proyectos (PMC)	55%	88%
4. Administración de Contratos con Proveedores (SAM)	100%	100%
5. Medición y Análisis (MA)	38%	38%
6. Aseguramiento de la Calidad del Proceso y el Producto (PPQA)	100%	100%
7. Administración de Configuraciones (CM)	43%	75%
8. Desarrollo de Requerimientos (REQD)	63%	80%
9. Solución Técnica (TS)	47%	47%
10. Integración de Productos (PI)	19%	44%
11. Verificación (VER)	22%	59%
12. Validación (VAL)	60%	75%
13. Enfoque a Procesos Organizacionales (OPF)	67%	67%
14. Definición de Procesos Organizacionales (OPD)	57%	57%
15. Entrenamiento Organizacional (OT)	100%	100%
16. Administración Integrada de Proyectos (IPM)	45%	55%
17. Administración de Riesgos (RSKM)	0%	21%
18. Análisis de Decisiones y Resolución (DAR)	75%	75%
19. Desempeño de Procesos Organizacionales (OPP)	60%	60%
20. Administración Cuantitativa de Proyectos (QPM)	7%	7%
21. Gestión del rendimiento de la organización (OPM)	35%	35%
22. Análisis de Causas y Resolución (CAR)	40%	40%

7.4.4. Análisis de resultados de la prueba

En las 9 áreas susceptibles a mejora se encontró un mejor cumplimiento que el proceso original, en el caso del área REQM se encontró una mejora mayor a la esperada debido a mejoras colaterales por el uso de otras prácticas en otras áreas que aportaron mejora a algunas sub prácticas originales. Este es el caso de la SP1.4 “Mantienen la trazabilidad

bidireccional entre los requerimientos y los planes del proyecto al igual que los productos del trabajo” que aunque Scrum no provee una forma de garantizar la trazabilidad de los requerimientos, gracias a los artefactos de ICONIX y el uso en el proyecto de la herramienta Enterprise Architect, se logra un mejor cubrimiento de la práctica.

En el resto de las áreas se identificó una mejora menor a la máxima calculada según el cumplimiento de las prácticas ágiles.

En el caso de PI se detectó que no incluir el equipo dedicado para integración fue una falencia importante en la práctica de integración continua, al igual que afectó el poco cubrimiento que se obtuvo con las pruebas automáticas. Aunque el equipo de desarrollo tuvo tiempo para aprender sobre esta tarea, la poca experiencia llevo a la construcción de métodos demasiado extensos y con muchas dependencias lo que dificultó la codificación de las pruebas unitarias.

En el área de REQD se encontraron deficiencias en la definición formal de interfaces debido a la ausencia de estándares y artefactos específicos para las mismas.

En el caso de VAL al momento de realizar la priorización se escogió usar las prácticas de Scrum aunque XP demostraba un mejor cubrimiento de las prácticas frente a CMMI debido a que no se podía garantizar la presencia del cliente en el sitio de desarrollo³⁸ y las actividades de Scrum al respecto se adaptaban más a la idiosincrasia de la empresa.

Por su parte en el área VER no se logró el cubrimiento esperado debido a la falta de registros principalmente en los resultados de la verificación por pares y en el resumen final de todas las actividades de verificación.

³⁸ Finalmente se logro gestionar la presencia a tiempo parcial de un representante del cliente en el sitio de desarrollo lo que demostró ser un gran apoyo en las labores de validación.

Algo similar suceden en IPM donde hace falta los documentos que evidencien los aportes del proyecto a los procesos institucionales impiden alcanzar el potencial deseado para esta área.

Debido a que el riesgo en proyectos ágiles tiene un enfoque diferente al planteado en CMMI, mas reactivo a los inconvenientes por un lado y con un cubrimiento natural de los riesgos más relevantes mediante el uso de las propias prácticas del método, el área de proceso de RSKM muestra un alcance menor al esperado, también se evidencia una deficiencia significativa al no usar XP en su totalidad sobre todo en la meta específica SG3 MITIGAR RIESGO que podría darse por cumplida mediante interpretación ágil, pero que a la luz de las subprácticas no puede ser evidenciada.

En el caso de CM se encontró el menor cubrimiento frente al presupuestado ya que no fue posible incluir la práctica de creación de líneas base debido a la falta de experiencia del equipo en su creación y mantenimiento lo que afecto el cubrimiento de 2 de las 7 prácticas específicas de esta área de proceso.

En general la implementación de la toma de medidas en el proyecto piloto se vio afectada por la falta de cultura organizacional en el tema, hasta el momento no se encontraron estrategias adecuadas para la toma de mediciones de manera que estas sean lo más reales posibles y no afecten el rendimiento general del equipo. Esta falta de mediciones afectó el cubrimiento en las áreas de VAL, REQD, VER e IPM.

El porcentaje de variación total de cubrimiento de cada área según el esperado inicialmente se puede observar en la Tabla 26, solo se consideran las áreas de proceso seleccionadas que a su vez son las únicas que presentan mejora.

Tabla 26. Comparación entre la mejora esperada y la real obtenida luego de la prueba piloto.

Area de Proceso	Nuevo proceso	Posible mejora	Mejora obtenida
1. REQM	100%	20%	30%
2. PP	66%	2%	-
3. PMC	88%	33%	33%
4. SAM	100%	-	-
5. MA	38%	-	-
6. PPQA	100%	-	-
7. CM	75%	57%	32%
8. REQD	80%	30%	18%
9. TS	47%	-	-
10. PI	44%	42%	25%
11. VER	59%	50%	38%
12. VAL	75%	30%	15%
13. OPF	67%	-	-
14. OPD	57%	-	-
15. OT	100%	-	-
16. IPM	55%	13%	10%
17. RSKM	21%	36%	21%
18. DAR	75%	-	-
19. OPP	60%	-	-
20. QPM	7%	-	-
21. OPM	35%	-	-
22. CAR	40%	-	-

7.5. RESUMEN GERENCIAL PARA COMFAMILIAR

Con el fin de informar a los interesados dentro de la institución de los resultados de este proyecto, se presentó un informe resumido de las actividades y resultados obtenidos, este informe presentado al líder de tecnología informática como cierre del estudio realizado se encuentra disponible en el [Anexo I](#).

7.6. ANÁLISIS DE RESULTADOS

A continuación se revisan los resultados del proyecto a la luz de los resultados e impactos esperados.

Con respecto a los resultados esperados

Todos los resultados esperados se cumplieron a cabalidad, iniciando en la sección 7.5 donde se comenta el entregable denominado inicialmente como Informe Gerencial, el cual se entregó a satisfacción al Líder del proceso del ADSCR y fue compartido con la dirección de la organización recibiendo comentarios muy positivos. Con respecto al proceso estructurado este se encuentra en el [Anexo G](#) y su contenido se encuentra detallado en las secciones 7.3.2 y 7.3.3. Finalmente el artículo científico fue terminado y enviado exitosamente para su evaluación y posterior publicación a la revista Scientia et Technica de la ciudad de Pereira bajo el título “Comparación de metodologías ágiles y procesos de desarrollo de software mediante un instrumento basado en CMMI”.

Con respecto a los impactos esperados

El ADSCR recibió toda la información sobre sus debilidades, fortalezas detalladas en el informe Gerencial mencionado anteriormente, recibiendo además el proceso estructurado

diseñado en el numeral 7.3 y retroalimentación gracias al piloto que puede encontrarse en el numeral 8.4. Toda esta experiencia ha permitido implementar el nuevo proceso estructurado de desarrollo de software dentro de la compañía.

En este informe y sus anexos se encuentra toda la información de la experiencia lograda durante su estudio brindando a la comunidad información detallada sobre este proceso de mejora incluyendo comparación de metodologías, evaluación del proceso actual, diseño de la mejora y validación de la solución propuesta. También se entrega a la comunidad los Instrumentos usados para la evaluación y comparación descritos en los numerales 7.1.2 y 7.2.1 mediante los anexos [Anexo B](#), [Anexo C](#) y [Anexo D](#) del presente documento. Adicionalmente se logra entregar a la comunidad resultados específicos de la comparación de Scrum, ICONIX y XP frente a CMMI tal como se describe en el numeral 7.2.

Finalmente se aporta a la comunidad toda la experiencia lograda sobre la aplicación del método para la selección de prácticas mediante ponderación descrito en el numeral 7.2.3.

Todo este informe sirve como guía para el desarrollo de la metodología propuesta en el numeral 4 donde se describe detalladamente la experiencia desde la evaluación de un proceso de desarrollo de una empresa del eje cafetero hasta el diseño y prueba de un nuevo proceso estructurado, el cual puede ser aplicado a cualquier empresa de desarrollo de software.

8. CONCLUSIONES

Como finalización de este proyecto se presentan las conclusiones derivadas del estudio del mismo, esta sección se encuentra dividida en cuatro áreas correspondientes a la evaluación del proceso inicial del ADSCR, luego la comparación de las prácticas estipuladas en CMMI y las prácticas ágiles seleccionadas, seguidamente de lo referente al diseño del nuevo proceso basado en el cubrimiento de las prácticas ágiles y para finalizar se comenta sobre la validación de este proceso de desarrollo de software mediante una prueba piloto.

8.1. EVALUACIÓN

Durante la evaluación del proceso inicial del ADSCR se colocaron a prueba dos instrumentos, uno con profundidad a nivel de áreas de proceso y otro a nivel de prácticas específicas.

Este primer instrumento puede ser útil con el fin de caracterizar un grupo de empresas de forma rápida, pero es difícil establecer un punto de partida a partir solo de esta información. El segundo instrumento permite tomar una mejor medida con respecto a cada práctica específica, en solo una tarde es posible obtener una caracterización del proceso en gran detalle. Sin embargo con el fin de conocer realmente el nivel de cumplimiento con respecto a CMMI es necesario contemplar cada sub práctica.

8.2. COMPARACIÓN

Este es uno de los estudios de mayor cubrimiento hasta la fecha, usa los 5 niveles de CMMI e incluye la comparación total con 3 prácticas ágiles. Generalmente los estudios de este tipo comparan entre 1 y 2 prácticas y solo en un pequeño número de áreas de proceso.

El método elegido para realizar el mapeo de CMMI frente a prácticas ágiles puede extenderse fácilmente a cualquier método o proceso, y da una visión realista del cubrimiento según lo propuesto en CMMI, esto permite fácilmente reconocer las áreas de fortaleza del método o prácticas que se toman como candidatos.

Al realizar un análisis cruzado con mapeos realizados por diferentes autores de las mismas prácticas puede observarse diferencias entre los puntos de vista, lo que muestra que es tal vez imposible no tener cierto nivel de subjetividad al realizar análisis de estas características.

Esta subjetividad se marca con mayor intensidad si se tiene en cuenta que el propio manual técnico de CMMI se dan pautas para la aplicación de prácticas ágiles que indican que las prácticas específicas podrían verse cubiertas sin la necesidad de seguir las al pie de la letra las indicaciones propuestas por las prácticas o por las sub-prácticas respectivas. Este caso se acentúa más en el caso de prácticas con enfoques menos tradicionalistas como es el caso de la gestión de riesgos en XP.

Al usar una base de comparación tan completa como CMMI, es posible obtener una matriz de mapeo donde todas las prácticas estudiadas demuestran sus fortalezas y debilidades frente al primero, lo que permite tener una visión completa de cómo se pueden complementar unas a otras según las áreas de proceso.

8.3. DISEÑO PROCESO ESTRUCTURADO DE SOFTWARE

Aunque luego de la comparación se tengan resultados teóricos sobre las mejores elecciones para cada práctica específica de CMMI es evidente la necesidad de realizar la selección de las prácticas que mejor se acomoden a la cultura y necesidades de la organización. Esta etapa debe realizarse con el conocimiento tanto de la organización como de las prácticas ágiles a implementar dentro del proceso.

El diseño de un proceso híbrido de esta naturaleza tiende a ser una tarea más artesanal que industrializada, requiriendo de varias iteraciones de prueba y error con el fin de encontrar el lugar apropiado para cada una de las prácticas que tienen su origen de fuentes distintas y de esta manera el proceso tenga un flujo de tareas lógicas y de actividades bien definidas y asignadas a los roles necesarios.

8.4. VALIDACIÓN DEL PROCESO

Durante el proceso de validación se pudo comprobar las afirmaciones de varias investigaciones anteriores con respecto al alto nivel de importancia que obtiene el tener en cuenta a las personas y su cultura. Aunque el proceso diseñado cumpla con todas las especificaciones y requerimientos de CMMI si las nuevas prácticas no se acomodan a las personas el éxito del proyecto se puede ver comprometido.

En el caso de ADSCR algunas de las prácticas más estrictas relacionadas con la documentación y medición sobre todo en el rol de desarrollador tuvieron que ser aplazadas a siguientes iteraciones del proceso de mejora debido a que el volumen del cambio era demasiado alto para el corto tiempo de la prueba.

Los tiempos de capacitación y asimilación de las nuevas prácticas deben tenerse en cuenta con el fin el equipo asimile el proceso de la forma más natural posible.

A esto se agrega la inclusión del equipo en los ciclos de mejora. Durante el piloto en el ADSCR y mediante las reuniones retrospectivas se pudieron identificar artefactos para mejorar la documentación de la revisión de pares así como la del cliente y estrategias ágiles con el fin de tomar algunas medidas que faciliten la construcción de indicadores de desempeño sin que impliquen una excesiva carga de trabajo adicional y a que los desarrolladores están dispuestos a documentar.

9. RECOMENDACIONES

Se puede extender este trabajo con nuevas prácticas ágiles agregándolas a la matriz de comparación con el fin de encontrar nuevas compatibilidades, posiblemente buscando una mejor cobertura sobre todo para los niveles 4 y 5.

Este estudio y otros mencionados en la bibliografía pueden ser usados para comparar los diferentes puntos de vista de los autores con el fin de realizar un análisis cruzado y así identificar las prácticas más controversiales o que generan conclusiones más variados entre las diferentes opiniones.

Es posible construir una herramienta automática para registrar los datos de comparación de las prácticas ágiles y que permita realizar la evaluación inicial del proceso. Esta herramienta podría también realizar sugerencias para la construcción del nuevo proceso basado en la posible mejora calculada. Finalmente se podría realizar la evaluación final al nuevo proceso con el fin de calcular los porcentajes de mejoramiento con respecto a la evaluación inicial.

REFERENCIAS

- Ahern, D. M., Clouse, A., & Turner, R. (2004). *CMMI distilled: a practical introduction to integrated process improvement*. Addison-Wesley Professional.
- Ambler, S., & Lines, M. (2012). *Disciplined Agile Delivery*. Indiana: Pearson plc.
- Beck, K., & Andres, C. (2004). *Extreme programming explained: embrace change*. Addison-Wesley Professional.
- Booch, Jacobson, & Rumbaugh. (1998). *The Unified Modeling Language User Guide*. : Addison Wesley.
- CMMI Product Team. (2010). *CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033)*. Recuperado el 17 de 02 de 2014, de Software Engineering Institute, Carnegie Mellon University: <http://cmmiinstitute.com/resource/cmmi-for-development-version-1-3/>
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2009). *Información Básica de Scrum*. California: Scrum Training Institute.
- Díaz Fernandez, Y. (2009). Estudio sobre la correspondencia entre prácticas CMMI y prácticas ágiles y su aplicación en PYMES.
- Diaz, J., Garbajosa, J., & Calvo-Manzano, J. A. (2009). Mapping CMMI level 2 to Scrum practices: An experience report. *In Software Process Improvement pringer Berlin Heidelberg*. , 93-104.
- EPF Team. (s.f.). *Eclipse Process Framework*. Recuperado el 26 de 02 de 2014, de Libraries Download:
http://www.eclipse.org/epf/downloads/praclib/praclib_downloads.php
- Fayad, M. E., Laitinen, M., & Ward, R. P. (2000). Thinking objectively: software engineering in the small. *Communications of the ACM* , 43 (3), 115-118.

- Fritzsche, M., & Keil, P. (2007). Agile Methods and CMMI: Compatibility or Conflict? *e-Infomatica Software Engineering Journal* .
- Gandomani, T. J., & Zulzalil, H. (2013). Compatibility of Agile Software Development Methods and CMMI. *Indian Journal of Science & Technology*, 6(8).
- Highsmith, J. (2002). What is agil software development. *CrossTalk The Journal of Defense Software Engineering* , 4-9.
- Jacobson, I., & Booch Grady, R. J. (2000). *El proceso unificado de desarrollo de software*. Madrid: Addison Wesley.
- Jan, S., & Javed, A. (2013). SCXTREME Framework: A Customized Approach of Process Improvements in Agile Blend with CMMI Practices in Pakistan. *International Journal of Information Technology and Computer Science (IJITCS)*, 5(3) , 69.
- Kulpa, M. K., & Johnson, K. A. (2008). *Interpreting the CMMI (R): A Process Improvement Approach*. Florida: Auerbach Publications.
- Leticia, B., Astorga Vargas, M. A., Olgúin Espinoza, J. M., & Andrade Peralta, M. d. (2008). *Experiences on the Implementation of MoProSoft and Assessment of Processes under the NMX-I-059/02-NYCE-2005 Standard in a Small Software Development Enterprise*. Baja California: Mexican International Conference on Computer Science (ENC '08).
- Lukasiewicz, K., & Miler, J. (2012). Improving agility and discipline of software development with the Scrum and CMMI. *Software, IET*, 6(5) , 416-422.
- Marcial, A., de Freitas, B., Furtado Soares, F., & Belchior, A. (2007). Mapping CMMI Project Management Process Areas to SCRUM Practices. *Software Engineering Workshop* , 13-22.
- OMG. (2008). *Software and Systems Process Engineering Meta-Model Specification*. Recuperado el 17 de 02 de 2014, de <http://www.omg.org/spec/SPEM/2.0/>
- Paulk, M. C. (2001). Extreme programming from a CMM perspective. . *Software, IEEE* 18.6 , 19-26.
- Peralta, I. (2007). Caracterización del nivel de desarrollo de software de las empresas de manizales (Tesis no publicada). *Universidad de Manizales, Colombia* .

- Phillips, M., & Shrum, S. (Agosto de 2011). *Library: Software Engineering Institute*. Recuperado el 20 de Noviembre de 2012, de <http://www.sei.cmu.edu/library/abstracts/whitepapers/Which-CMMI-Model.cfm>
- Pino, F., Vidal, J., Garcia, F., & Piattini, M. (2007). *Modelo para la Implementación de Mejora de Procesos en Pequeñas Organizaciones Software*. Zaragoza, España: XII Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2007).
- Pressman, R. (2010). *Ingenieria del Software*. España: McgrawHill.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2000). *El lenguaje unificado de modelado. Manual de Referencia*. Madrid: Addison Wesley.
- Schwaber, K. (1997). Scrum development process. In *Business Object Design and Implementation* , 117-137.
- Scott, L., Jeffery, D. R., Carvalho, L., D'Ambra, P., & Rutherford, P. (2001). *Practical Software Process Improvement-The IMPACT Project*. Canberra, ACT: Australian Software Engineering Conference 2001: 182-189.
- Sibbald, C. (10 de Octubre de 2006). *The Eclipse Foundation open source community*. Recuperado el 21 de 11 de 2012, de [epf/general/getting_started](http://www.eclipse.org/epf/general/getting_started): http://www.eclipse.org/epf/general/An_Introduction_to_EPF.zip
- Stephens, M., & Rosenberg, D. (2007). *Use Case Driven Object Modeling with UML. Theory and Practice*. New York: Apress.
- Team, C. P. (2010). *CMMI for Development version 1.3—Improving processes for developing better products and services*. SEI Report CMU/SEI-2010-TR-033.
- Wake, W. C. (2002). *Extreme programming explored (Vol. 1)*. Addison-Wesley.