

MIS0001  
2012  
ej. 1

1323839

**Universidad Simón Bolívar**  
**Maestría en Ingeniería de Sistemas y Computación**



**Trabajo de Grado de Maestría**

**Modelo Aplicable A La Gestión De Procesos Ágiles De Desarrollo De Software  
Basado En CMMI –DEV 1.3 Y SCRUM**

**Adriana María Iglesias Solano  
Acellys Messino Suárez**

**Director: Patty Pedroza Barrios**

**Barranquilla, Colombia. Octubre de 2012**

## **RESUMEN**

Las necesidades en las organizaciones dedicadas al desarrollo de software de adaptabilidad, desarrollo de productos de calidad acorde a las necesidades del cliente, cortos tiempos de desarrollo, evolución continua y la alineación de los procesos de desarrollo y gestión con los objetivos organizacionales, hacen pertinente la formulación de un modelo de gestión de procesos de desarrollo basado en los principios y prácticas del enfoque ágil, orientado hacia la aplicación de buenas prácticas y procesos de mejora continua. Debido a lo anterior, el presente documento plantea la construcción de un modelo orientado a la gestión de procesos ágiles de desarrollo de software enmarcados en las buenas prácticas de CMMI-DEV 1.3 y el marco de trabajo de SCRUM, este modelo se denomina SUMM “Modelo Unificado de Madurez de Scrum”.

SUMM consta de cinco niveles de madurez, de acuerdo a la representación por etapas de CMMI, estos niveles a su vez están compuestos de Metas, Objetivos, Prácticas e Indicadores que permitirán a la organización poder emplear un modelo de desarrollo ágil de calidad, enfocados en procesos de mejora continua.

## TABLA DE CONTENIDO

<b>1 CAPITULO 1. INTRODUCCIÓN Y OBJETIVOS</b>	<b>9</b>
<b>1.1. CONTEXTUALIZACIÓN INICIAL</b>	<b>9</b>
<b>1.2 OBJETIVOS</b>	<b>10</b>
1.2.1 OBJETIVO GENERAL	10
1.2.2 OBJETIVOS ESPECÍFICOS	10
<b>1.3 METODOLOGIA</b>	<b>11</b>
<b>1.4. PLANTEAMIENTO DEL PROBLEMA</b>	<b>12</b>
1.4.1. DESCRIPCIÓN DEL PROBLEMA	12
<b>1.4 FORMULACION DEL PROBLEMA</b>	<b>15</b>
<b>1.5. LÍNEA DE INVESTIGACIÓN</b>	<b>15</b>
<b>1.5 DESCRIPCIÓN BREVE DE LA ESTRUCTURA DEL DOCUMENTO</b>	<b>16</b>
<b>2CAPITULO 2. ESTADO DEL ARTE Y MARCO CONCEPTUAL</b>	<b>17</b>
<b>2.1 DEFINICIONES PRELIMINARES</b>	<b>17</b>
2.1.1 MODELO	17
2.1.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE	17
2.1.3. MARCO DE TRABAJO – FRAMEWORK	18
2.1.4. SOFTWARE	18
2.1.5. PROCESOS DE DESARROLLO DE SOFTWARE	18
2.1.6. CALIDAD DE SOFTWARE	19
2.1.6.1. CALIDAD DE SOFTWARE ORIENTADA AL PROCESO	19
2.1.6.2. CALIDAD DE SOFTWARE ORIENTADO AL PRODUCTO	19

<b>2.2. CMMI</b>	<b>20</b>
2.2.1.COMPOSICIÓN GENERAL DE CMMI	20
<b>2.3. MODELO ÁGIL DE DESARROLLO DE SOFTWARE</b>	<b>23</b>
2.4.1.VALORES, PRINCIPIOS Y PRÁCTICAS DEL MANIFIESTO ÁGIL	26
<b>2.5. MARCO DE TRABAJO DE SCRUM</b>	<b>27</b>
2.5.1.USER STORIES	28
2.5.2.ACTORES Y ROLES	28
<b>2.5.3. Proceso de desarrollo en Scrum</b>	<b>28</b>
<b>2.5.3.1 Ciclo de Vida de Scrum</b>	<b>28</b>
<b>2.6. ESTADO DEL ARTE</b>	<b>30</b>
2.6.1.MODELO DE MADUREZ Y CAPACIDAD INTEGRADO CMMI	30
2.6.2.METODOLOGÍAS AGILES	38
2.6.3.CMMI VS METODOLOGIAS AGILES	37
2.6.4.ESTADISTICAS DE GESTION DE PROYECTOS	40
2.6.4.1.SITUACIÓN ACTUAL COLOMBIA	40
2.6.4.2.SITUACION ACTUAL INTERNACIONAL	48
<b>3.CAPITULO 3: CARACTERIZACIÓN DE CMMI Y SCRUM</b>	<b>52</b>
<b>3.1.¿QUÉ MEJORAS APORTA CMMI A SCRUM?</b>	<b>55</b>
<b>3.2.CORRESPONDENCIA DE LAS PRÁCTICAS DE SCRUM CON LAS ÁREAS DE PROCESO DE CMMI</b>	<b>55</b>
3.2.1. ÁREAS DE PROCESOS DE CMMI NIVEL DOS (2) REPRESENTACIÓN ESCALONADA	55
3.2.1.1 GESTIÓN DE REQUERIMIENTOS - REQM	55
3.2.1.3 MONITOREO Y CONTROL DEL PROYECTO	58

3.2.1.4. GESTIÓN DE LA CONFIGURACIÓN	59
3.2.1.5. ASEGURAMIENTO DE LA CALIDAD DE PRODUCTO Y PROCESO - PPQA	59
3.2.1.5. MEDICIÓN Y ANÁLISIS – MA	59
3.2.2. PRÁCTICAS GENÉRICAS DE NIVEL 2, REPRESENTACIÓN ESCALONADA CMMI	60
3.2.3. CORRESPONDENCIA CMMI NIVEL 3, REPRESENTACIÓN ESCALONADA	61
<b>4. CAPÍTULO 4: MODELO DE GESTIÓN PROPUESTO</b>	<b>62</b>
<b>4.1. COMPONENTES PILARES DEL MODELO SUMM</b>	<b>62</b>
<b>4.2. CARACTERÍSTICAS DE SUMM</b>	<b>62</b>
<b>4.2.1 NIVELES DE MADUREZ MODELO SUMM</b>	<b>62</b>
4.2.1.1. Nivel Inicial	63
4.2.1.2. Nivel Gestionado	64
4.2.1.3. Nivel Definido	79
4.2.1.3.1. Gestión de la comunicación con los stakeholders	79
4.2.1.3.2. Validación y verificación SUMM	82
4.2.1.3.3. Gestión de Sprint	85
4.2.1.4. Nivel Mejorado	91
4.2.1.4.1. Gestión cuantitativa de proyectos	91
4.2.1.4.2. Rendimiento de los procesos organizacionales	94
4.2.1.5. Nivel Optimizado	96
4.2.1.5.1. Análisis de causas y resolución	97

**5CAPITULO 5.ANOTACIONES FINALES Y TRABAJOS FUTUROS 104****5.1. ANOTACIONES FINALES 104****5.2. TRABAJOS FUTUROS 106****6.REFERENCIAS 107**

## **INDICE DE TABLAS**

Tabla 1. Factores de éxito	13
Tabla 2. Factores de Fracaso	14
Tabla 3. Descripción niveles CMMI	23
Tabla 4. Empresas Colombianas Valoradas Oficialmente Con CMMI	34
Tabla 5. Comparacion De Metodologias Vs Criterios	52
Tabla 13: Niveles de Madurez Metas, Objetivos y Prácticas SUMM	64
Tabla 14. Prácticas y métricas Asignación de Roles. Gestión Proceso SUMM nivel 2 SUMM. Elaboración Propia.	69
Tabla 15. Prácticas y métricas Artefactos SUMM. Gestión Proceso SUMM nivel 2 SUMM Elaboración Propia.	71
Tabla 16. Prácticas y métricas Flujo de trabajo. Gestión Proceso SUMM nivel 2 SUMM. Elaboración Propia.	72
Tabla 17. Prácticas y métricas Reuniones SUMM. Gestión Proceso Scrum nivel 2 SUMM.	74
Tabla 18. Prácticas y métricas Definición clara de requisitos por el Product Owner. Gestión de requisitos nivel 2 SUMM .Elaboración Propia	76
Tabla 19. Prácticas y métricas Gestión del Product Backlog. Gestión de requisitos nivel 2 SUMM, Elaboración Propia	77
Tabla 20 Prácticas y métricas Reunión de planificación de Sprint exitosa. Gestión de requisitos nivel 2 SUMM .Elaboración Propia	78
Tabla 21. Prácticas y métricas Flujo de trabajo SUMM Product Owner. Gestión de la comunicación con los Stakeholders nivel 3 SUMM . Elaboración Propia	82

<b>Tabla 22. Prácticas y métricas Gestión del DoD. Validación y Verificación nivel 3 SUMM. Elaboración Propia</b>	<b>85</b>
<b>Tabla 23. Prácticas y métricas Gestión del Sprint Backlog. Gestión de Sprint SUMM nivel 3 SUMM .Elaboración Propia</b>	<b>88</b>
<b>Tabla 24. Prácticas y métricas Gestión del Sprint Backlog. SUMM Diario SUMM nivel 3 SUMM. Elaboración Propia</b>	<b>89</b>
<b>Tabla 25. Prácticas y métricas Gestión del Sprint Backlog. Gestión de Sprint Burndown chart SUMM nivel 3 SUMM.Elaboración Propia</b>	<b>90</b>
<b>Tabla 26. Prácticas y métricas Gestión cuantitativa de proyectos. Gestión Unificada de Proyectos nivel 4 SUMM. Elaboración Propia</b>	<b>94</b>
<b>Tabla 27. Prácticas y métricas Medición y Análisis SUMM. Rendimiento de los procesos organizacionales 4 SUMM. Elaboración Propia</b>	<b>96</b>
<b>Tabla 28. Prácticas y métricas Gestión Retrospectiva de Sprint. Análisis de causas y resolución SUMM nivel 5.Elaboración Propia</b>	<b>99</b>

## INDICE DE FIGURAS

FIGURA 1: HISTORIA DE CMMI	19
FIGURA 2. COMPOSICIÓN CMMI-DEV	20
FIGURA 3. DEFINICIÓN DEL PRODUCT BACKLOG Y SPRINT BACKLOG	28
FIGURA 4. CICLO DE UN SPRINT	28
FIGURA 5. RESULTADO DE UN SPRINT	28
FIGURA 6: PAÍSES REPORTADOS AL SEI SEGÚN SCAMPI.	30
FIGURA 7: SCAMPI POR CONTINENTES	30
FIGURA 8: PAÍSES REPORTADOS AL SEI SEGÚN SCAMPI.	31
FIGURA 9: COMPARATIVO ANUAL DEL NÚMERO DE EVALUACIONES SCAMPI	32
FIGURA 10: PAÍSES CON MAYOR NÚMERO DE EVALUACIONES	32
FIGURA 11: TIPO DE PROYECTOS DE TECNOLOGÍA DE INFORMACIÓN EN COLOMBIA	39
FIGURA 12: DISTRIBUCIÓN DE SECTORES 2011 VS 2010	40
FIGURA 13: PRESUPUESTO PLANEADO 2009-2011	41
FIGURA 14: PRESUPUESTO PLANEADO ÚLTIMOS 5 AÑOS	41
FIGURA 15: PRESUPUESTO PLANEADO VS PRESUPUESTO REAL	42
FIGURA 16: DURACIÓN PLANEADO ÚLTIMOS 3 AÑOS	43
FIGURA 17: DURACIÓN REAL ÚLTIMOS 5 AÑOS	44
FIGURA 18: DURACIÓN PLANEADA VS REAL	44
FIGURA 19: CARACTERÍSTICAS DE PROYECTOS EXITOSOS SEGÚN CLIENTES 2011 VS 2010	45
FIGURA 20: CARACTERÍSTICAS DE PROYECTOS EXITOSOS SEGÚN CLIENTES ÚLTIMOS 3 AÑOS	46
FIGURA 21: USO DE METODOLOGÍAS 2009-2011	46
FIGURA 22: USO DE HERRAMIENTAS Y TÉCNICAS	48
FIGURA 23: USO DE METODOLOGÍAS ÁGILES	48
FIGURA 24: NÚMERO DE PROYECTOS USANDO METODOLOGÍAS ÁGILES	48
FIGURA 25: PRÁCTICAS ÁGILES EMPLEADAS	49
FIGURA 26: RAZONES DE ADOPCIÓN DE ÁGIL	50

<b>FIGURA 27: SRUM : AREAS DE PROCESO DE DESARROLLO VS ETAPAS DE DESARROLLO</b>	<b>51</b>
<b>FIGURA 28 NIVELES DE MADUREZ REPRESENTACIÓN POR ETAPAS SUMM</b>	<b>63</b>
<b>FIGURA 29. METAS POR NIVELES DE MADUREZ SUMM</b>	<b>63</b>
<b>FIGURA 30. NIVELES DE MADUREZ SUMM. ELABORACIÓN PROPIA</b>	<b>63</b>
<b>FIGURA 31: ESTRUCTURA GENERAL GESTIÓN PROCESO SCRUM NIVEL 2 DE SUMM</b>	<b>65</b>
<b>FIGURA 32: GESTIÓN DE REQUISITOS DE NIVEL 2 DE SUMM</b>	<b>74</b>
<b>FIGURA 33. ESTRUCTURA GENERAL GESTIÓN DE LA COMUNICACIÓN CON LOS STAKEHOLDERS SUMM NIVEL 3</b>	<b>80</b>
<b>FIGURA 34 RELACIÓN OBJETIVO, PRÁCTICA Y MÉTRICA</b>	<b>81</b>
<b>FIGURA 35 ESTRUCTURA GENERAL VALIDACIÓN Y VERIFICACIÓN SUMM</b>	<b>83</b>
<b>FIGURA 36 RELACIÓN OBJETIVO, PRÁCTICA Y MÉTRICA</b>	<b>84</b>
<b>FIGURA 37 ESTRUCTURA GENERAL GESTIÓN DE SPRINT SUMM</b>	<b>86</b>
<b>FIGURA 38 RELACIÓN OBJETIVO, PRÁCTICA Y MÉTRICA</b>	<b>87</b>
<b>FIGURA 39 GESTIÓN CUANTITATIVA DE PROYECTOS NIVEL 4 DE SUMM</b>	<b>92</b>
<b>FIGURA 40 RELACIÓN OBJETIVO, PRÁCTICA Y MÉTRICA</b>	<b>93</b>
<b>FIGURA 41 RENDIMIENTO DE LOS PROCESOS ORGANIZACIONALES NIVEL 4 DE SUMM</b>	<b>95</b>
<b>FIGURA 42 RELACIÓN OBJETIVO, PRÁCTICA Y MÉTRICA MEDICIÓN Y ANÁLISIS SUMM</b>	<b>95</b>
<b>FIGURA 43. ANÁLISIS DE CAUSAS Y RESOLUCIÓN SUMM NIVEL 5</b>	<b>97</b>
<b>FIGURA 44. RELACIÓN OBJETIVO, PRÁCTICA Y MÉTRICA GESTIÓN RETROSPECTIVA</b>	<b>98</b>

## **CAPÍTULO 1: INTRODUCCIÓN**

Este capítulo Introdutorio provee una visión contextual de la temática abordada en este trabajo de grado, es así como se describe las contribuciones de esta investigación a la Maestría de Ingeniería de Sistemas y Computación y a la comunidad científica en general, seguido de los objetivos, la descripción y formulación del problema de investigación, de igual forma, se plantea la metodología y el enfoque de investigación adoptado

### **1.1. Contextualización Inicial**

Las empresas desarrolladoras de software reconocen cada vez más la necesidad de la adopción de herramientas de ingeniería de software, procesos, metodologías, marcos de trabajo y modelos, que permitan establecer pautas y procesos al momento de desarrollar software, sin embargo solo reconocerlo no ha sido suficiente. Es de esta forma cómo conseguir desarrollar software con éxito es un desafío cada vez más exigente no solo porque las organizaciones requieren adquirir la capacidad de gestionar la complejidad, sino también la innovación tecnológica y el cambio de requisitos. Encontrar el equilibrio entre los valores de las metodologías ágiles y la robustez de los modelos tradicionales brinda una luz para dar frente a esta situación dado que la Gestión de la complejidad requiere disciplina en los procesos, mientras que la gestión del cambio exige capacidad de adaptación. Es así como, CMMI proporciona disciplina y el marco de trabajo de Scrum mejora la adaptabilidad.

En este contexto es como se pretende formular un modelo para abordar las problemáticas asociadas al proceso de gestión de desarrollo de software enmarcadas dentro de las metodologías ágiles y la formalidad de las buenas prácticas de CMMI-DEV en la feroz competencia del mercado vigente, en la que los productos de software quedan obsoletos rápidamente, y en donde los requerimientos de cortos tiempos de desarrollo, calidad en los procesos y productos, reducción de costos son cada vez más

frecuentes, pero para asumir estos retos, se hace necesario tener agilidad, complejidad, adaptabilidad y flexibilidad.

## **1.2. OBJETIVOS**

### **1.2.1. OBJETIVO GENERAL**

Crear un modelo aplicable a la gestión de procesos ágiles de desarrollo de software basado en CMMI-DEV 1.3. y SCRUM, que permita la ejecución de proyectos de desarrollo de aplicaciones manteniendo la calidad en los procesos y en los productos.

### **1.2.3. OBJETIVOS ESPECIFICOS**

- Analizar la situación actual de las industrias desarrolladoras de software, sus procesos, técnicas y metodologías de gestión de proyectos, haciendo énfasis en los desarrollos ágiles y en la implementación de buenas prácticas.
- Definir los componentes estructurales que conformaran el modelo de gestión ágil de desarrollo de software a construir.
- Establecer los procesos que conformaran cada una de las etapas del marco de trabajo del modelo de gestión propuesto identificando las prácticas, metas, objetivos y métricas asociadas.

## **1.3. METODOLOGÍA**

Se centrará en un tipo de investigación: Descriptiva comparativa - Propositiva, enmarcado en un enfoque Cualitativo debido a que, se realizará en primera instancia una revisión documental sobre los antecedentes asociados a las temáticas abordadas, partiendo de las metodologías ágiles de desarrollo de software existentes hasta el modelo de buenas prácticas de calidad del software, a partir de los hallazgos encontrados en la revisión documental se planteará el modelo para la gestión de

procesos ágiles de desarrollo de software, para lograrlo se abarcarán las siguientes fases metodológicas:

- ❖ **Fase de Diagnóstico:** En esta fase se pretende realizar un diagnóstico del estado en que son gestionados los proyectos de desarrollo de software, teniendo en cuenta parámetros esenciales relacionados con los factores claves que apunten hacia la implementación o no de metodologías, herramientas y modelos.
- ❖ **Fase de Correspondencia:** En esta fase se realiza una caracterización del modelo de calidad escogido, CMMI – DEV 1.3, y el Marco de trabajo de Scrum. El objetivo de esta fase es indicar la correspondencia entre las áreas de procesos de CMMI y las prácticas de Scrum.
- ❖ **Fase de Definición:** En esta fase se realiza la definición de las etapas y/o fases que conformaran el modelo. Estableciendo los procesos inmersos e identificando los objetivos, metas, prácticas e indicadores.

Según Salinas y Pérez, 1993 entiéndase como Investigación Descriptiva como aquella que se refiere a la descripción de algún objeto, sujeto, fenómeno, etc en total o parte del mismo, tal como un aparato, técnica, método, procedimiento, proceso, es decir, que en este tipo de investigación se parte del supuesto que la descripción que se va a realizar no ha sido hecha anteriormente. Sin embargo, se acepta como perfectamente válida y original, la descripción de alguna variación o modificación de algo ya descrito, por ejemplo, en un aparato o técnica o proceso, se pueden modificar sus componentes y así obtener resultados diferentes y mejores a los anteriormente descritos [1].

## **1.4. PLANTEAMIENTO DEL PROBLEMA**

### **1.4.1. DESCRIPCIÓN DEL PROBLEMA**

En la actualidad la industria del software representa una importante actividad económica en muchos países del mundo, constituyéndose en una parte fundamental para el crecimiento de sus economías. En países como Estados Unidos, Brasil, Canadá, India, China, Finlandia, entre otros, las pequeñas compañías de software que cuentan con menos de 50 empleados representan hasta un 85 por ciento de las compañías [2]. Sin embargo, cuando comienzan a crecer, en muchos casos, la falta de un proceso de desarrollo visible termina creando un caos en toda la empresa [3]. En esta búsqueda constante de adaptarse a procesos de desarrollo, metodología y/o marcos de trabajo las organizaciones, según la encuesta internacional denominada State of Agile Survey [4], tienen en cuenta factores como minimizar costos, optimizar tiempo, optimizar recursos y mejorar la calidad de sus productos.

Pero infortunadamente, Según el CHAOS Report [5], “Solo el 35% de los proyectos de software tienen éxito”, entonces ¿En qué se está fallando?, este proyecto se centrará en tres factores que han sido incidentes; en primera instancia se tiene que algunas organizaciones tratan de realizar un buen trabajo de desarrollo, sin contar con procesos bien definidos, esto se ha convertido en uno de los principales escollos debido a que éste proceso debe ser contemplado como un proceso de negocio que tiene que ser gestionado, ser eficiente, evolutivo, rápido y altamente adaptable. Como segundo punto, muchas de éstas organizaciones de desarrollo de software aún no han implementado un modelo o metodología que gestione no sólo el proceso de desarrollo de software sino a toda la organización, esto se debe a factores como: desconocimiento del tema, temor a invertir en la implementación de una metodología de gestión del proceso de desarrollo de software al no ver la recuperación de la inversión a corto plazo o simplemente no han notado la importancia de valorizar sus proyectos de desarrollo en función de convertirse éstos en aporte a los logros de las metas del negocio. Y en tercera instancia la utilidad y eficacia de algunas prácticas, técnicas, metodologías y modelos varían entre unas organizaciones y otras, o unos proyectos u otros [6]. Lo anteriormente planteado en gran parte, se debe a que al mismo tiempo que unas prácticas son muy útiles para

algunas compañías, otras han sido consideradas muy negativamente por algunas de ellas.

Así, Turner y Jain [7] afirman que las organizaciones que usan métodos ágiles corren el riesgo de hacer demasiado énfasis en el conocimiento tácito y poco en la comunicación formal dentro de un equipo de desarrollo software, esto es en muchos casos dependiente de la experiencia de las personas que conforman el equipo, así como de su capacidad para compartir la información entre los diferentes miembros. Sin embargo, el proceso de desarrollo ágil de software no incluye únicamente comunicación informal, la comunicación formal: como código fuente, casos de pruebas, y una mínima cantidad de documentación es también utilizada en desarrollos ágiles, pero en la mayoría de los casos es deficiente, es en este punto en donde la aplicación de buenas prácticas como las de CMMI- DEV 1.3 aportan al manejo del proceso de desarrollo en los ámbitos de pre-proyecto, proyecto y post-proyecto y transforman las debilidades de la informalidad de metodologías ágiles como SCRUM en fortalezas y nuevas oportunidades.

En el estudio realizado por Chow sobre 109 proyectos ágiles se obtienen datos que evidencian que existen prácticas que apuntan hacia el éxito o fracaso de un proyecto, Estos factores son: la estrategia de entrega, las técnicas ágiles de ingeniería de software (uso de estándares de codificación, refactorización, diseño simple, pruebas de integración), la capacidad del equipo, los procesos de gestión de proyecto, el entorno del equipo, y el compromiso del stakeholder y su vinculación en el proceso. En las siguientes tablas se puede observar un resumen de lo anteriormente mencionado:

Organización	Personas	Proceso	Técnicas
<ul style="list-style-type: none"> <li>• Soporte ejecutivo firme</li> <li>• Financiación, respaldo, gestión comprometida</li> <li>• Cultura organizacional cooperativa en lugar de jerárquica</li> <li>• Comunicación cara a cara</li> <li>• Entorno de trabajo apropiado</li> </ul>	<ul style="list-style-type: none"> <li>• Equipo con altas competencias y experiencia</li> <li>• Equipo con gran motivación</li> <li>• Jefes, gestores con conocimientos en procesos ágiles</li> <li>• Estilo de gestión flexible, adaptable</li> <li>• Equipos de trabajo auto-organizativos, coherentes</li> <li>• Buena relación con el cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Ámbito de proyecto no definido</li> <li>• Requerimientos de proyecto no definidos</li> <li>• Plan de proyecto no definido</li> <li>• Falta de mecanismos de trazabilidad del progreso del proyecto</li> <li>• Falta de presencia del cliente</li> <li>• No manejo de riesgos</li> <li>• Mala relación con el cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Estándares de codificación bien definidos</li> <li>• Actividades de refactorización</li> <li>• Correcta cantidad de documentación</li> <li>• Entrega de software periódica</li> <li>• Entrega de los principales requerimientos primero</li> <li>• Pruebas de integración</li> </ul>

**Tabla No. 1 Factores de éxito Fuente: elaboración propia [8].**

Organización	Personas	Proceso	Técnicas
<ul style="list-style-type: none"> <li>• Falta de financiación o respaldo</li> <li>• Falta de gestión de los compromisos</li> <li>• Cultura organizacional demasiado tradicional</li> <li>• Cultura organizacional demasiado burocrática</li> <li>• Tamaño organizacional demasiado grande</li> </ul>	<ul style="list-style-type: none"> <li>• Falta de habilidades necesarias</li> <li>• Falta de capacidad de gestión de proyecto</li> <li>• Falta de trabajo en equipo</li> <li>• Resistencia de grupos o individuos</li> </ul>	<ul style="list-style-type: none"> <li>• Ámbito de proyecto no definido</li> <li>• Requerimientos de proyecto no definidos</li> <li>• Plan de proyecto no definido</li> <li>• Falta de mecanismos de trazabilidad del progreso del proyecto</li> <li>• Falta de presencia del cliente</li> <li>• No manejo de riesgos</li> <li>• Mala relación con el cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Falta de un conjunto de prácticas ágiles completo y correcto</li> <li>• Tecnologías y herramientas inapropiadas</li> </ul>

**Tabla No. 2 Factores de fracaso Fuente: elaboración propia [9].**

Precisamente, para evitar los efectos negativos enunciados anteriormente, las empresas deben dedicarle tiempo a la definición, adecuación y mejoramiento continuo de sus procesos organizacionales. Con el fin de incrementar la probabilidad de finalizar un proyecto con éxito, cumplir con mayor precisión los tiempos planeados y los recursos invertidos o gastados, el desarrollo de soluciones estandarizadas y amigables al usuario, asegurando una mayor calidad del software y trazabilidad, facilitar el mantenimiento de las aplicaciones y simplificar la documentación.

A nivel nacional, en Colombia, se han realizado esfuerzos para mejorar la competitividad del sector informático, a través de la colaboración con modelos de calidad como CMMI, la capacitación comercial y administrativa, entre otros. Específicamente, respecto a los modelos de calidad, el sector público en conjunto con las empresas privadas ha realizado esfuerzos para mejorar la posición de las empresas de software colombianas. El Gobierno Nacional, por medio de Proexport ha realizado un progreso sustancial en las empresas productoras de software con respecto a la evaluación de procesos, de acuerdo con el modelo CMMI; buscando así ampliar la base de dieciséis empresas Colombianas que están actualmente evaluadas según el SEI.

Sin embargo, la principal causa de la problemática realmente radica que en muchos casos estas empresas se mantienen al margen de las metodologías y los estándares internacionales debido a que tienen la percepción de que su implementación es difícil, costosa, y se consume mucho tiempo en el proceso.

Es así, como se pretende formular un modelo para abordar las problemáticas asociadas al proceso de gestión de desarrollo de software enmarcadas dentro de las metodologías ágiles y la formalidad de las buenas prácticas de CMMI-DEV en la feroz competencia del mercado vigente, en la que los productos de software quedan obsoletos rápidamente, y en donde los requerimientos de rapidez, calidad en los procesos y productos, reducción de costes son cada vez más frecuentes, pero para asumir estos retos, es necesario tener agilidad y flexibilidad.

#### **1.4.2. FORMULACIÓN DEL PROBLEMA**

La definición del problema plantea la siguiente pregunta:

¿Cómo crear un modelo aplicable a la gestión de procesos ágiles de desarrollo de software basado en CMMI-DEV 1.3. Y SCRUM, que permita la ejecución de proyectos de desarrollo de aplicaciones manteniendo la calidad en los procesos y en los productos?

#### **1.5. LÍNEA DE INVESTIGACIÓN**

El presente trabajo de Maestría está enmarcado dentro de la línea de investigación de Ingeniería de Software, teniendo en cuenta que esta disciplina se constituye de la suma de cuatro componentes: Procesos de desarrollo de software y aseguramiento de calidad; arquitecturas de software, metodologías de desarrollo, tecnologías y herramientas. De acuerdo a lo anterior, tres de los cuatro componentes son los ejes fundamentales entre los cuales se plantea la Investigación, la cual pretende crear un modelo de gestión de procesos ágiles basados en un marco de trabajo y en un modelo de calidad, generando de esta manera un modelo híbrido que permita el desarrollo de aplicaciones adaptables enmarcadas en procesos orientados a la aplicación de buenas prácticas.

## **1.5. DESCRIPCIÓN DE LA ESTRUCTURA DEL DOCUMENTO**

A continuación se describe la estructura del presente documento distribuido así:

Capítulo 2: Estado del arte y Marco conceptual: En este capítulo, se definen los términos relacionados al objeto de investigación y el estado actual de las industrias de desarrollo de software en cuanto a la implementación de metodologías de desarrollo de software y modelos orientados a la gestión de buenas prácticas

Capitulo 3: Caracterización de CMMI Y SCRUM: En este capítulo, se realiza una caracterización de cada uno de los niveles de CMMI, describiendo su correspondencia con los procesos de Scrum.

Capítulo 4: Modelo de Gestión propuesto: En este capítulo, se propone un modelo de gestión que combine la agilidad de Scrum y la madurez de CMMI.

## **CAPÍTULO 2: ESTADO DEL ARTE Y MARCO CONCEPTUAL**

### **2.1. DEFINICIONES PRELIMINARES.**

#### **2.1.1. MODELO**

Se denomina Modelo como la representación de un objeto, sistema o idea, de forma diferente al de la entidad misma, su principal propósito es explicar, entender o mejorar un sistema, por medio de su representación. Los modelos pueden ser una réplica exacta o una abstracción de las propiedades dominantes del objeto [10]. Los seres humanos siempre han tratado de representar, expresar ideas y objetos para tratar de entender y manipular el entorno.

Un modelo debe describir el sistema de forma detallada con la meta de hacer predicciones válidas sobre el comportamiento de este. Al realizar un análisis de sistemas, se crea un modelo del sistema que muestre las entidades, las interrelaciones, etc. La adecuada construcción de un modelo ayuda a organizar, evaluar y examinar la validez de pensamientos.

#### **2.1.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE**

Una metodología de desarrollo de software, se puede definir como un conjunto de pasos, procedimientos y actividades que deben seguirse para desarrollar software de una manera eficiente y efectiva, a través de cada una de las actividades asociadas que la conformen. Entre los componentes principales de una metodología están: la división o desglose de un proyecto en etapas, las tareas que se llevan a cabo en cada una de esas etapas, las restricciones y/o condiciones a que hubiere lugar, técnicas y herramientas que se emplean, el control y la gestión [11].

### **2.1.3. MARCO DE TRABAJO – FRAMEWORK**

Un marco de trabajo denominado también como Framework, es un conjunto de conceptos, prácticas y criterios estandarizados enfocados en dar solución a una problemática particular, sirviendo como referencia, para enfrentar y resolver nuevos problemas de índole similar [12]. En el contexto del desarrollo de software, un Framework básicamente define una estructura conceptual y tecnológica de soporte con artefactos o módulos, que funcionaran como base para que otro proyecto de software pueda ser organizado, estructurado y desarrollado más fácilmente.

### **2.1.4. SOFTWARE**

Un software básicamente es un programa de ordenador y la documentación asociada. El producto derivado del proceso de desarrollo de software puede construirse para un cliente específico o para un mercado general, de acuerdo, a un objeto de negocio o a las necesidades del cliente o del mercado [13].

### **2.1.5. PROCESOS DE DESARROLLO DE SOFTWARE**

Un proceso define un conjunto de pasos secuencialmente definidos requeridos para el desarrollo de un software, descritos en entradas (Insumo), métodos, herramientas, artefactos, personas y salidas (Producto), agrupándolos en teniendo en cuenta las consideraciones técnicas y administrativas [14].

### **2.1.6. CALIDAD DE SOFTWARE**

De acuerdo al estándar IEEE 610-1990, “La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”[15]. De acuerdo a lo anterior, la calidad de software se puede establecer como la correspondencia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo y con los requerimientos implícitos

no establecidos formalmente, durante la fase de ingeniería de requisitos por el cliente.

El tipo y número de actividades tendientes a garantizar la calidad dependerán del tamaño y complejidad de los productos software que se estén desarrollando. Sin embargo, también influyen otros factores, como pueden ser el tipo de proceso de desarrollo de software o los métodos y herramientas utilizados, la estructura organizativa de la organización, la motivación del personal, entre otros.

#### **2.1.6.1. CALIDAD DE SOFTWARE ORIENTADA AL PROCESO [16].**

La calidad del proceso es uno de los factores que influyen en la calidad del producto y se basa en:

- Medir número de defectos en los productos y relacionar estos defectos con el proceso.
- El proceso se mejora con el propósito de reducir el número de defectos en el producto.
- El proceso se mejora hasta que sea repetible.

#### **2.1.6.2. CALIDAD DE SOFTWARE ORIENTADO AL PRODUCTO.**

Según definición en ISO/IEC 9126 la calidad de software orientado al producto corresponde a la totalidad de la funcionalidad y prestaciones de un producto de software que están relacionadas con su capacidad de satisfacer las necesidades explícitas o implícitas [17].

## **2.2. CMMI [18] [16].**

Los modelos de capacidad y madurez (CMM) iniciaron en el año 1993 y han sido desarrollados para disciplinas como la Ingeniería de Sistemas, la Ingeniería de Software, la Adquisición de Software, la administración del Personal de trabajo y administrativo y el Desarrollo Integrado de Productos y Procesos.

El Modelo Integrado de Capacidad y Madurez (CMMI) fue creado con la finalidad de evitar problemas al usar varios modelos de capacidad y madurez. El propósito inicial de CMMI fue el de combinar tres modelos: el modelo de capacidad y madurez para software denominado SW-CMM, el modelo de capacidad de Ingeniería de Sistemas denominado SECM y el modelo de capacidad y madurez para el Desarrollo Integrado de Productos CMMI-DEV, cuyo propósito es ayudar a las organizaciones de software a mejorar sus procesos de desarrollo y mantenimiento tanto de productos como de servicios. En la versión oficial del modelo, se puede definir CMMI-DEV, como [19]:

- Un marco de mejores prácticas de administración, ingeniería y soporte, establecidos en 22 tópicos denominadas en el modelo como Áreas de Proceso.
- Un conjunto de requisitos de procesos y mejores prácticas, que ayudan como guía a la organización para definir e implementar los procesos relacionados con los 22 tópicos que lo conforman.

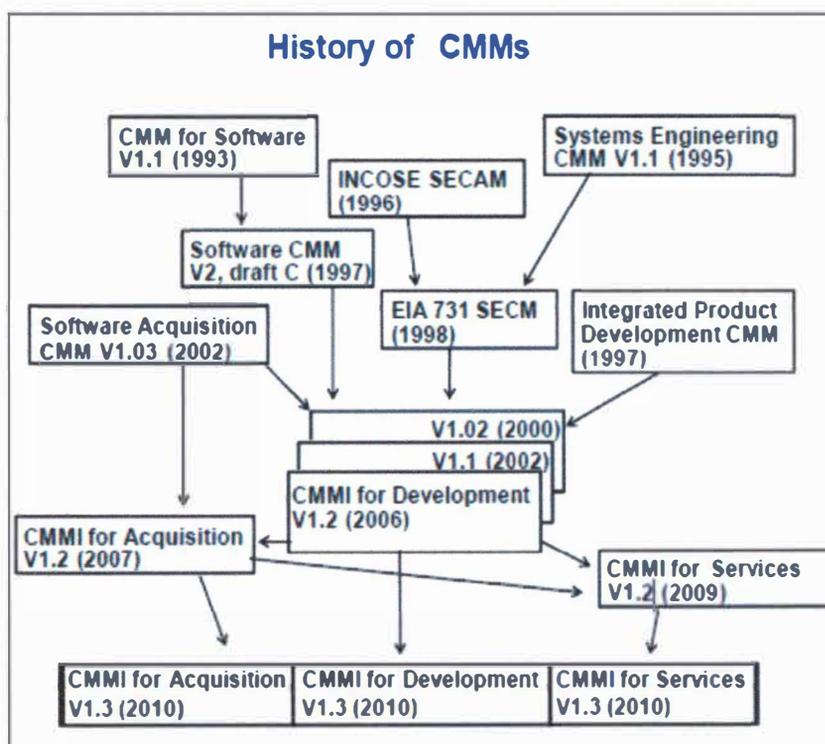
### **2.2.1. COMPOSICIÓN GENERAL DE CMMI [20]**

En la definición formal del modelo está conformado por:

- 22 áreas de proceso con componentes particulares para cada área de proceso.
- Componentes requeridos: Describen lo que una organización debe alcanzar para satisfacer en un área de proceso.
- Componentes esperados, que son los que una organización puede implementar para alcanzar un componente requerido.
- Componentes informativos, que pueden ayudar en la forma de abordar los componentes requeridos y esperados.

- El modelo utiliza los niveles de madurez y capacidad para describir las rutas recomendadas que deben seguir las organizaciones para mejorar sus procesos.

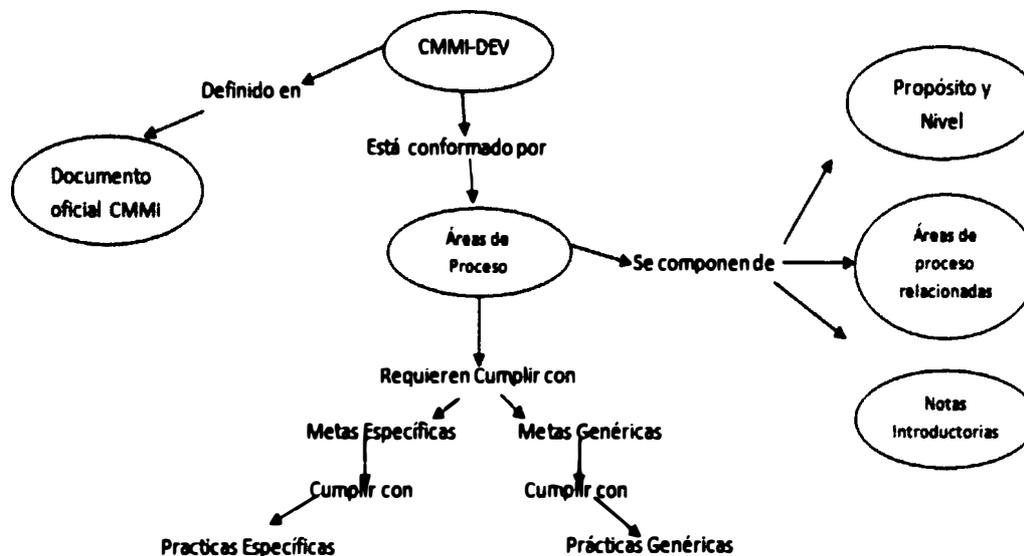
CMMI ayuda a integrar funciones organizativas, tradicionalmente separadas, establecer objetivos y prioridades en la mejora de procesos, proporciona una orientación para la calidad de procesos, y un punto de referencia para la evaluación de los procesos actuales [15]. En la siguiente grafica se muestra la evolución que ha tenido el modelo CMMI desde 1993 hasta la actualidad:



**Figura 1: Historia de CMMI [22]**

CMMI-DEV es el modelo de capacidad y madurez integrado para desarrollo cuyo objetivo es proponer una solución integrada y completa para las actividades de desarrollo y de mantenimiento aplicadas a los productos [23].

La siguiente figura muestra la composición general de CMMI-DEV:



**Figura 2. Composición CMMI-DEV. Elaboración Propia, Adaptada de [24]**

CMMI para el desarrollo según la especificación, es un modelo de referencia que abarca las actividades para el desarrollo de productos y servicios. Las organizaciones de muchas industrias, incluyendo la aeroespacial, la banca, hardware, software, la defensa, la fabricación de automóviles, y las telecomunicaciones [25]. CMMI para el Desarrollo contiene las prácticas que abarcan la gestión de proyectos, gestión de procesos, ingeniería de sistemas, ingeniería de hardware, ingeniería de software, y otros procesos de apoyo utilizados en el desarrollo y mantenimiento. Como se ha mencionado el uso de CMMI está organizado en 22 tópicos llamados áreas de Proceso y organizados por niveles. Las áreas de procesos tienen un propósito u objetivo además, se componen de unas metas y prácticas específicas que son actividades que permiten cumplir con el propósito de cada meta.

Cada área de proceso se ubica dentro de un nivel específico en CMMI los niveles se ubican en la categoría de 0 a 5 [26] y se definen en la siguiente tabla:

NIVEL	NOMBRE	DESCRIPCION
0	Incompleto	Hace referencia a un proceso que no es ejecutado o que es ejecutado parcialmente, es decir, algunas metas Específicas del proceso no se cumplen.
1	Ejecutado	Es un proceso que aunque cumple las metas específicas del Área de Proceso, no está institucionalizado, es decir, con el tiempo pueden desaparecer las mejoras realizadas.
2	Administrado	Es un proceso ejecutado en el cual existe una infraestructura y un personal capacitado que lo soporta, además es monitoreado, controlado, evaluado y revisado para cumplir con su definición.
3	Definido	Es un proceso administrado que utiliza un conjunto de procesos estandarizados en una organización y adaptarlos para un propósito específico. En este nivel los procesos son descritos más detalladamente que en el nivel administrado.
4	Administrado cuantitativamente	Es un proceso definido que es controlado utilizando técnicas estadísticas y cuantitativas. Se establecen objetivos cuantitativos que se utilizan como criterios en la administración del proceso.
5	En optimización	Es un proceso administrado cuantitativamente que es mejorado continuamente basándose en el entendimiento de las causas que ocasionan variaciones en los procesos.

**Tabla 3: Descripción niveles CMMI. Elaboración Propia, Adaptada de [27]**

### 2.3. Modelo Ágil De Desarrollo De Software [28]

Las metodologías de desarrollo ágil a pesar haber sido formuladas a mediados de los años 90, es en la actualidad en donde su uso se ha proliferado, estas metodologías nacieron con el afán de remplazar a las tradicionales o también denominadas metodologías pesadas de desarrollo de software. Fue entonces como en el año 2001

algunas de las personas afines con los principios de estas metodologías se agruparon formando lo que se conoce hoy como Alianza Ágil, que promueve el desarrollo ágil de aplicaciones. Es de esta forma como actualmente cada vez más empresas de desarrollo le apuestan al uso de estas metodologías para la construcción de sus productos de software y ha sido así como su popularidad ha ido cada vez más en aumento [29]. Pero, ¿Por qué utilizar metodologías ágiles?, las metodologías de desarrollo de software tradicionales o pesadas han demostrado presentar ciertos problemas, entre ellos se pueden citar los siguientes:

- La ingeniería de requisitos es una etapa inmersa en el análisis, la formulación de estos requisitos deben reflejar exactamente lo que el cliente desea que realice su programa de aplicación, tratando de evitar que se produzcan cambios de este listado de requisitos establecido, debido que a medida en como avanza el proyecto por cada una de sus etapas de desarrollo será más costoso la realización de dichos cambios o corrección errores [30]. Partiendo de lo anterior se puede afirmar que se hace necesario que la especificación de requisitos y las entrevistas con el cliente sean desarrolladas iterativamente y con cierta frecuencia, debido a que en gran parte de las veces el cliente no conoce sus necesidades con la profundidad suficiente como para definir las exactamente al inicio del proyecto.
- Otra dificultad de las metodologías tradicionales es lo lento que puede ser el proceso de desarrollo, debido a que es difícil la mayoría de las ocasiones que el equipo de desarrollo entienda en su totalidad un sistema complejo.
- El equipo debe realizar un enorme esfuerzo en la etapa de planeación, debido a que como se mencionó anteriormente la corrección de errores en las metodologías tradicionales son muy costosos y estos redundarán en el presupuesto del proyecto y deben ser asumidos por alguna de las partes o por ambos.
- En algunas ocasiones y debido en gran parte a los factores ya mencionados el resultado final no es exactamente lo que quiere el cliente o no tiene la calidad esperada.

- Los retrasos en la entrega de productos son el pan de cada día o lo mismo que la incorrecta ejecución de las últimas fases del ciclo de vida.

En contraposición a todo lo anterior, suscitaron las metodologías ágiles como alternativa de solución a las dificultades esbozadas del uso de metodologías tradicionales y se ajustan cuando los requisitos están en expuestos a constante cambio y en este contexto presenta las siguientes ventajas:

- Se realizan entregas continuas del producto en cortos lapsos de tiempo, lo que permite al cliente disfrutar de las funcionalidades del software a medida que se va construyendo [31].
- Los riesgos y dificultades se reparten a lo largo del proceso de desarrollo, lo que permite realizar retroalimentaciones constantes y establecer controles a los riesgos.
- La capacidad de respuesta a los cambios es mayor que con la metodología tradicional, ya que estos son percibidos como oportunidades de mejora del software en construcción.
- Mejora continua de los procesos y el equipo de desarrollo.
- El trabajo conjunto entre el cliente y el equipo de desarrollo es de vital importancia para el buen manejo del proceso y la aplicación de buenas prácticas, gracias a este fuerte vínculo se minimizan los malentendidos o inconsistencias a la hora de hacerla labor de licitación.

## **2.4. Valores, principios y prácticas del manifiesto ágil**

El desarrollo de software a través de la aplicación de metodologías ágiles no especifica unos procesos o métodos a seguir, por el contrario este establece cuatro (4) valores y doce (12) principios [32].

1. Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
2. Desarrollar software que funcione por encima de una completa documentación.
3. La colaboración con el cliente por encima de la negociación contractual.
4. Responder a los cambios más que seguir estrictamente un plan.

Para cumplir estos valores se siguen doce principios [33]:

1. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporten valor.
2. Dar la bienvenida a los cambios de requisitos. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
3. Liberar software que funcione frecuentemente, desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
4. Los miembros del negocio y los desarrolladores deben trabajar juntos diariamente a lo largo del proyecto.
5. Construir el proyecto en torno a individuos motivados. Darles el entorno y apoyo que necesiten y confiar en ellos para conseguir finalizar el trabajo.
6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
7. El software que funciona es la principal medida de progreso.
8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos que se auto organizan.

12. En intervalos regulares el equipo debe reflexionar sobre cómo ser más efectivo y según estas reflexiones ajustar su comportamiento.

Estos principios marcan el ciclo de vida de un desarrollo ágil, así como las prácticas y procesos a utilizar.

- Planificación
- Programación en pareja
- Integración continua
- Refactorización.
- Desarrollo dirigido por pruebas

## **2.5. Marco de trabajo de Scrum.**

SCRUM, es un marco de referencia utilizado para el desarrollo ágil de aplicaciones de software [34]. Scrum presenta las siguientes características:

- División de la organización, en pequeños equipos, interdisciplinarios y organizados.
- División del trabajo, esta se debe organizar en una lista de entregables pequeños y concretos, esta lista debe ser organizada según la prioridad y la estimación de cada elemento.
- División del tiempo, esto se realiza en iteraciones cortas de longitud fija, normalmente con una duración entre una (1) a cuatro (4) semanas, con código potencialmente entregable y funcional después de cada iteración.
- Actualiza Las Prioridades, de la mano con el cliente, basada en los conocimientos adquiridos mediante la verificación del entregable después de cada iteración.

### **2.5.1. User Stories**

Los User Stories son el elemento base que utiliza SCRUM para describir las características que el usuario espera que tenga el software que se va a desarrollar. Por lo tanto, pueden incorporar tanto cuestiones relacionadas con las funciones del sistema como con cualquier otro aspecto del mismo (restricciones, rendimiento, etc.).

Las historias de usuario se presentan desde la perspectiva del usuario. Así, no se describen utilizando una terminología técnica sino que se escriben utilizando un lenguaje cercano al dominio de la aplicación que se está desarrollando de forma que sea comprensible por los clientes y por los desarrolladores.

### **2.5.2. Actores y Roles**

Los actores que intervienen dentro del proceso son de forma general los siguientes:

- Propietario del producto ó Product Owner
- Maestro de scrum ó Scrum Master
- Equipo de desarrollo ó Scrum Team
- Cliente o usuario

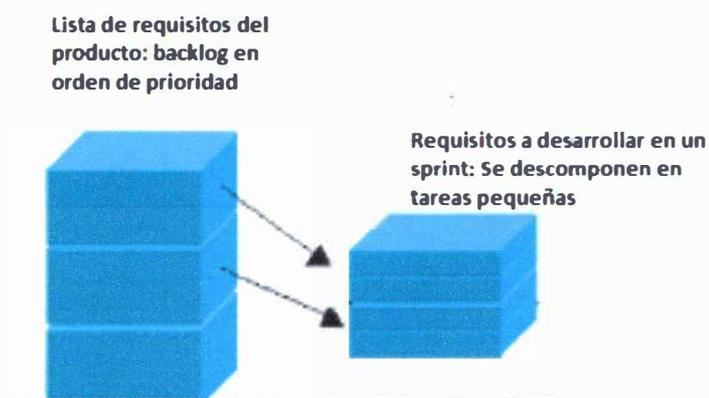
### **2.5.3. Proceso de desarrollo en Scrum**

Scrum como se definió anteriormente es un método iterativo e incremental. En Scrum al principio del proyecto se define el Product Backlog, que contiene todos los requerimientos funcionales y no funcionales que deberá satisfacer el sistema que se va a desarrollar. El Product Backlog será definido durante reuniones de planeamiento con los stakeholders. A partir de ahí se definirán las iteraciones, que se denominan Sprint, cada Sprint tendrá su propio Sprint Backlog que se define como un subconjunto del Product Backlog con los requerimientos a ser construidos en el Sprint correspondiente. El tiempo de duración sugerido del Sprint es de un mes.

### 2.5.3.1 Ciclo de Vida de Scrum

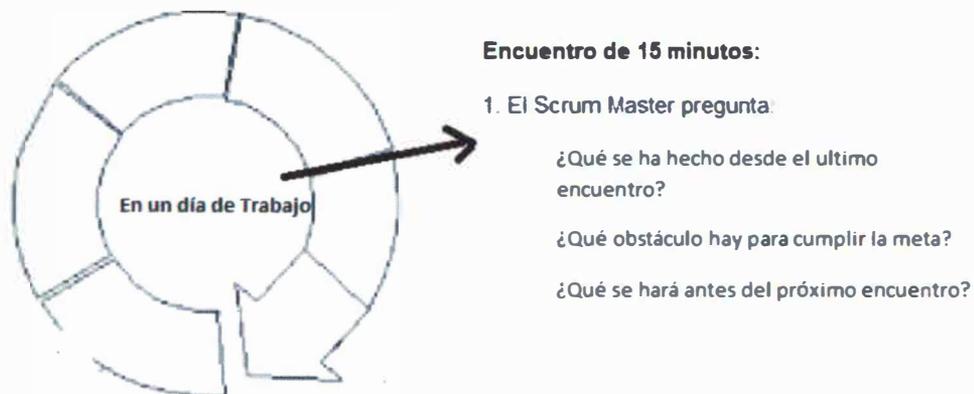
El ciclo de vida de Scrum se puede resumir en 3 fases denominadas pre-juego, juego y pos-juego.

- **Pre-Juego:** En esta primera fase se definen las expectativas y la financiación del desarrollo. Entre las actividades de esta fase se tiene la definición del presupuesto y del registro de acumulación denominado backlog, así como el diseño exploratorio, los prototipos y la planificación. Adicionalmente, en esta fase se identifican requerimientos adicionales y se priorizan las tareas para la primera iteración.



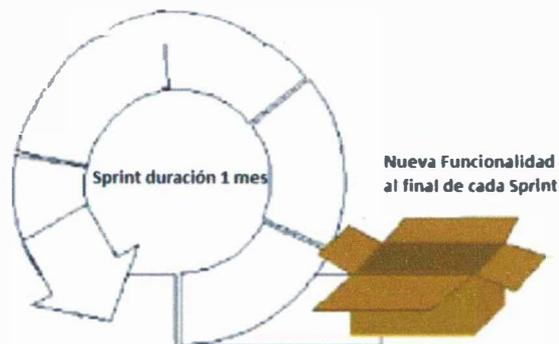
**Figura 3. Definición del Product Backlog y Sprint Backlog. Elaboración Propia Adaptada de [35].**

- **Juego:** En esta fase se implementa un sistema listo para entrega en una serie de iteraciones que son las denominadas sprints. Entre las actividades de esta fase se encuentra la de planeamiento de sprints en cada iteración, la definición del registro de acumulación de sprints y los estimados, y finalmente los encuentros diarios de Scrum:



**Figura 4. Ciclo de un Sprint. Elaboración Propia Adaptada de [36].**

- Pos-Juego: Esta fase incluye entrenamiento, preparación para el lanzamiento, incluyendo la documentación final, previa al lanzamiento por etapas, pruebas y puesta en marcha.



**Figura 5. Resultado de un Sprint. Elaboración Propia Adaptada de [37].**

## 2.6. Estado del arte

### 2.6.1. Modelo de Madurez y Capacidad integrado CMMI

En la actualidad grandes compañías han adoptado diferentes modelos y prácticas para mejorar la calidad de sus procesos o sus productos en beneficios para sus clientes, uno de los modelos que ha entrado en este desafío es el Modelo de Madurez y Capacidad integrado (Capability Maturity Model Integration - CMMI) (1993).

Desde la versión 1.1 se evidencia que este modelo de mejora puede ser aplicado en otras áreas de interés, es por esto que CMMI en su versión 1.2 y 1.3, maneja el concepto de constelación [38], que se define según la versión oficial como una colección de componentes de CMMI que incluyen un modelo, sus materiales de entrenamiento y los documentos sobre la evaluación para un área de interés.

El marco de modelo de la versión actual 1.3 está soportado por 3 constelaciones [39]: La de Desarrollo, la de Servicios y la de Adquisiciones. En adelante se denominará CMMI-DEV a la constelación para el Desarrollo.

Para evaluar el estado de los procesos de software de una organización basado en los modelos CMMI el SEI (Software Engineering Institute) desarrolló un método llamado SCAMPI, por sus siglas en inglés, Standard CMMI Appraisal Method for Process Improvement. Basándose en esta evaluación, a continuación, se muestra el último reporte de publicado por el SEI con corte a marzo de 2012 sobre el uso del modelo a lo largo de todo el mundo:

## Countries Where Appraisals Have Been Performed and Reported to the SEI



Argentina	Australia	Austria	Bahrain	Bangladesh	Belgium	Brazil	Brunei Darussalam
Bulgaria	Canada	Chile	China	Colombia	Costa Rica	Cuba	Cyprus
Czech Republic	Denmark	Egypt	Finland	France	Germany	Greece	Guatemala
Hong Kong	Hungary	India	Indonesia	Ireland	Israel	Italy	Japan
Jordan	Korea, Republic Of	Latvia	Lebanon	Lithuania	Luxembourg	Macedonia	Malaysia
Mauritius	Mexico	Morocco	Nepal	Netherlands	New Zealand	Norway	Pakistan
Panama	Paraguay	Peru	Philippines	Poland	Portugal	Qatar	Romania
Russia	Saudi Arabia	Singapore	Slovakia	South Africa	Spain	Sri Lanka	Sweden
Switzerland	Syrian Arab Republic	Taiwan	Thailand	Tunisia	Turkey	Ukraine	United Arab Emirates
United Kingdom	United States	Uruguay	Viet Nam				

Red country name: New additions with this reporting since September 2011



Software Engineering Institute

Carnegie Mellon

CMMI-SCAMPI v1.2/v1.3 Class A  
Appraisal Results  
March 2012

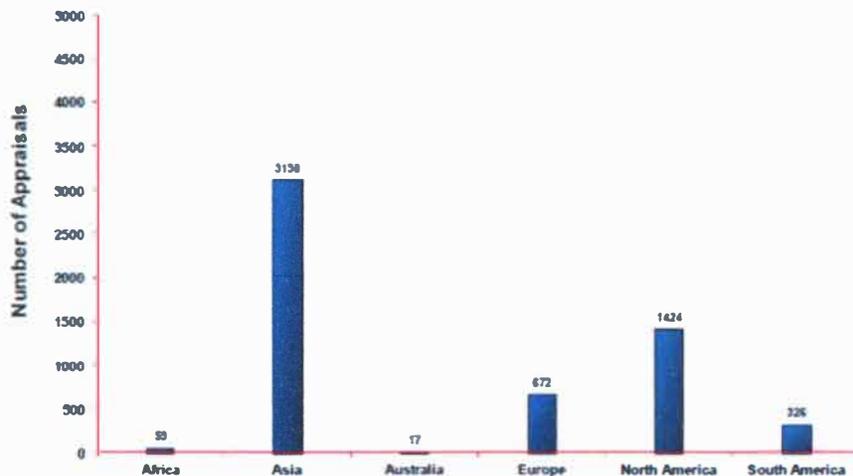
© 2012 Carnegie Mellon University

16

Figura 6: Países reportados al SEI según SCAMPI. [40]

A continuación se observa un reporte de comparativo del número de evaluaciones por continente:

## Number of Appraisals Reported to the SEI by Continent



Based on 5628 appraisals

North America includes Canada, the USA, and Mexico; South America includes Central America and the Caribbean; Australia includes New Zealand



Figura 7: SCAMPI por continentes [41]

Desde el año 2006, se han reportado al SEI 4846 evaluaciones SCAMPI versión 1.2 y 1.3. Según el informe de las últimas evaluaciones reportada (Marzo de 2012) China, España, Brasil, Argentina y la India han aumentado a un ritmo rápido. El porcentaje se incrementa constantemente. La mayoría de las evaluaciones reportadas fueron en el nivel de madurez 2 y China presentó mayor evaluaciones que EE.UU. Seguidamente, se muestra el número de evaluaciones y niveles de madurez que han obtenido los países según el SEI hasta Marzo de 2012:

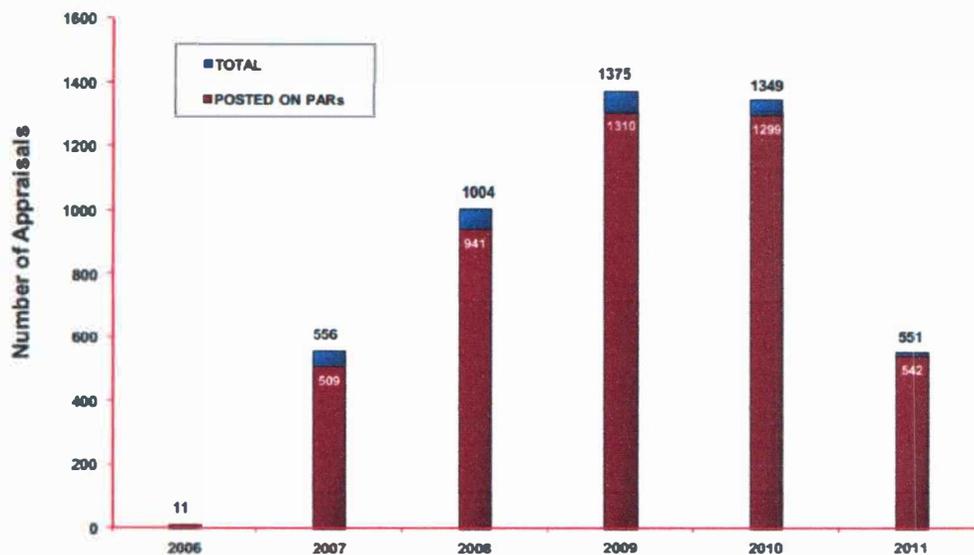
Country	Number of Appraisals	Maturity Level 1 Reported	Maturity Level 2 Reported	Maturity Level 3 Reported	Maturity Level 4 Reported	Maturity Level 5 Reported	Country	Number of Appraisals	Maturity Level 1 Reported	Maturity Level 2 Reported	Maturity Level 3 Reported	Maturity Level 4 Reported	Maturity Level 5 Reported
Argentina	10		40	23		5	Macedonia	10 or fewer					
Australia	15		2	6		7	Malaysia	28		9	38		2
Austria	10 or fewer						Mauritius	10 or fewer					
Bahrain	10 or fewer						Mexico	116		54	52	7	5
Bangladesh	10 or fewer						Morocco	10 or fewer					
Belgium	10 or fewer						Nepal	10 or fewer					
Brazil	143	1	16	56		8	Netherlands	12	1	6	8		
Brunei Darussalam	10 or fewer						New Zealand	10 or fewer					
Bulgaria	10 or fewer						Norway	10 or fewer					
Canada	30	1	9	12		3	Pakistan	24	1	21	1		1
Chile	31	2	22	12		5	Panama	10 or fewer					
China	1866	11	616	616	56	25	Paraguay	10 or fewer					
Colombia	33	1	14	10	3	2	Peru	18		10	11		
Costa Rica	10 or fewer						Philippines	12	2	2	1		4
Cuba	10 or fewer						Poland	10 or fewer					
Cyprus	10 or fewer						Portugal	20		8	10		2
Czech Republic	10 or fewer						Qatar	10 or fewer					
Denmark	10 or fewer						Romania	11	1	2	8		
Egypt	14	1	25	14		1	Russia	10 or fewer					
Finland	10 or fewer						Saudi Arabia	12	2	5	1		2
France	124	3	71	39		1	Singapore	17		5	12	1	
Germany	55	8	24	18			Sierra Leone	10 or fewer					
Greece	10 or fewer						South Africa	10 or fewer					
Guatemala	10 or fewer						Spain	122	1	125	14	2	8
Hong Kong	11		3	3		2	Sri Lanka	16	2	14			
Hungary	10 or fewer						Sweden	10 or fewer					
India	518	1	15	366	4	123	Switzerland	10 or fewer					
Indonesia	10 or fewer						Syrian Arab Republic	10 or fewer					
Ireland	10 or fewer						Taiwan	113	1	25	51	2	1
Israel	14			8		4	Thailand	51		16	33		2
Italy	56		27	24			Tunisia	10 or fewer					
Japan	187	1	53	97		3	Turkey	24			21		3
Jordan	10 or fewer						Ukraine	10 or fewer					
Korea, Republic Of	158		56	14	12	9	United Arab Emirates	10 or fewer					
Larva	10 or fewer						United Kingdom	85	2	25	29		8
Lebanon	10 or fewer						United States	1278	10	570	536	5	68
Lithuania	10 or fewer						Uruguay	10 or fewer					
Luxembourg	10 or fewer						Viet Nam	22			18	1	3

**Software Engineering Institute** | **Carnegie Mellon**

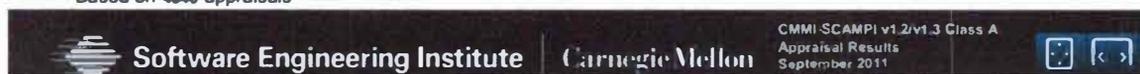
**CMMI-SCAMPI v1.2/v1.3 Class A**  
**Appraisal Results**  
**March 2012**  
© 2012 Carnegie Mellon University

Figura8: Países reportados al SEI según SCAMPI. [42]

A nivel mundial se muestra un comparativo anual del número de evaluaciones SCAMPI reportadas por el SEI en los últimos seis años con corte a la última publicación realizada en Marzo de 2012:



Based on 4846 appraisals



**Figura 9: Comparativo anual del número de evaluaciones SCAMPI [43]**

Entre los países que mayor número de evaluaciones han reportado se encuentra China, India, España y Estados Unidos. En México, se evidencia un incremento en el número de evaluación presentadas entre el año 2009 vs 2010 y una disminución notable en el año 2011, a continuación se muestra el resumen de los países que ocupan los 10 primeros lugares en cuanto a empresas evaluadas con corte Marzo de 2012:

Country	2006	2007	2008	2009	2010	2011	Total
China	1	103	273	482	499	508	1866
United States	3	149	238	311	307	270	1278
India	5	81	80	117	110	125	518
Spain		15	30	75	43	59	222
Japan		16	47	40	40	44	187
Korea, Republic Of		21	30	28	35	44	158
Brazil		16	27	38	37	25	143
France		13	29	31	28	23	124
Mexico		4	18	30	35	29	116
Taiwan		18	29	19	24	23	113



**Figura 10: Países con Mayor Número de Evaluaciones [44]**

Para las organizaciones que iniciaron su evaluación SCAMPI en CMMI versión 1.2 se utilizó un tiempo promedio para aprobar la evaluación según el nivel de madurez que se requiere de:

- 19 meses para nivel de madurez 2 y 3
- 21 meses para pasar de nivel de madurez 3 a 4
- 25,5 meses para pasar de nivel de madurez 3 a 5

Según la cifras presentadas por el SEI, existe evidencia estadística de que utilizar CMMI permite:

- Reducir Costos en un 20% en promedio
- Reducir Tiempos en un 37% en promedio
- Aumentar productividad en un 62% en promedio
- **Calidad** en un 50% en promedio
- Satisfacción del Cliente en un 14% en promedio

En la actualidad y a lo largo del mundo todas las organizaciones apuntan hacia la premisa de minimizar costos, optimizar tiempo, optimizar recursos y mejorar la calidad de sus productos. La industria del software no es la excepción ya que representa una importante actividad económica en muchos países del mundo, siendo parte fundamental en el crecimiento de sus económicas. En países como Estados Unidos, Brasil, Canadá, India, China, Finlandia, entre otros, las pequeñas compañías de software que cuentan con menos de 50 empleados representan hasta un 85 por ciento de las compañías [45]. Sin embargo, cuando comienzan a crecer, en muchos casos, la falta de un proceso de desarrollo visible termina creando un caos en toda la empresa [46].

A nivel nacional, en Colombia, se han realizado esfuerzos para mejorar la competitividad del sector informático, entre los cuales se destacan: la colaboración con modelos de calidad como CMMI, la capacitación comercial y administrativa, entre otros [47]. Según el informe del SEI de marzo de 2012, para Colombia fueron evaluadas en CMMI 43 empresas, en su mayoría aplicadas para nivel de madurez 2 y 3 entre ellas 15 empresas en nivel 2 y 8 nivel 3[48].

Algunas de las empresas Colombianas valoradas oficialmente con CMMI a la fecha de Marzo de 2012:

Empresa	Nivel 2	Nivel 3	Nivel 4	Nivel 5
Soft bolívar	X			
Heinsohn software house s.a.				X
Psl				X
Open systems			X	
Ilimitada s.a.		X		
intergrupo s.a.		X		
Asesoftware limitada		X		
Gestiontek		X		
Mvm ingeniería de software s.a.		X		
Itac s.a.		X		
Asesoftware ltd		X		
Avansoft s.a.	X			
Coomeva - unidad de tecnología v informática	X			
Fundación cardiovascular de colombia - fcv soft		X		
Gestiontek s.a.		X		
Trebol software s.a.		X		
Servinte s.a.	X			
Cidlis uis	X			

**Tabla 4: Empresas Colombianas valoradas oficialmente con CMMI. Elaboración Propia.**

En octubre de 2007 el Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnología “Francisco José de Caldas” – Colciencias presentó la convocatoria 432 - 2007 “Apoyo al Fortalecimiento de la Capacidad Nacional de Software” Inicialmente la Unión Temporal Red Colombiana de Calidad de Software denominada UT\_ RCCS, conformada por la Universidad Industrial de Santander UIS, la Universidad EAFIT, Procesix Colombia Ltda y la Asociación de Parques Tecnológicos Parquesoft Colombia presentaron una propuesta de proyecto pertinente en respuesta a la convocatoria, la cual fue seleccionada para su financiación[50].

El proyecto se denominó desarrollo del programa Apoyo al Fortalecimiento de la Capacidad Nacional de Software por parte de la UT\_ RCCS, incluyó varios objetivos de trabajo asociados a ampliar el conocimiento y aplicación del Modelo CMMI – DEV en Colombia. Este proyecto para la implementación del Modelo CMMI– DEV en 55 pymes del sector software colombiano, permitió identificar una versión inicial de un ciclo de proyecto para el mejoramiento de procesos en pymes del sector software teniendo como referencia un modelo de calidad, el cual puede utilizarse dada su flexibilidad en actividades, para realizarlo con otros modelos, normas o estándares de calidad como referencia, gracias al concepto de caso de mejora. Partiendo de la meta de impacto esperada para implementación del Modelo CMMI DEV en PYMES de software colombianas, las instituciones miembros de la UT\_ RCCS, generaron de forma integrada una metodología del proceso de asesoría y acompañamiento que permitiera lograr resultados acertados en la implementación del Modelo CMMI – DEV en un entorno PYME[51]. Se definió asesoría y acompañamiento de las PYMES de software para la implementación del Modelo CMMI – DEV v 1.2 y se ejecutó bajo un ciclo de vida de proyecto creado especialmente para atender los requerimientos del programa.

A manera de resultados el proyecto, según el informe final, se logró un despliegue amplio del Modelo CMMI – DEV y los conceptos de mejora de procesos en todas las regiones del país y en múltiples tamaños de pymes como son las micro, pequeñas micro, pequeñas y medianas empresas, consiguiendo que un 83% de las 55 pymes culminará un proceso de mejoramiento de principio a fin, y se perfilarán un 46% de las mismas para realizar una evaluación oficial [52]. Las áreas de proceso del Modelo CMMI - DEV preferidas por las pymes porque impactaron positivamente el desarrollo de sus productos y la organización interna de las empresas fueron: Planificación de Procesos

(PP), Seguimiento y control de proyectos (PMC), Gestión de Requisitos (REQM) y Gestión de Configuración (CM), las cuales obtuvieron las mayores frecuencias como áreas de proceso seleccionadas para la planeación de mejora y mostraron los más altos promedios de adherencia a nivel nacional durante el diagnóstico final [53].

Algunas de las dificultades más sobresalientes durante el proceso de implementación del Modelo – CMMI - DEV en las pymes fueron: la interpretación de las prácticas del modelo, la identificación y definición de indicadores gerenciales y de proceso, y la dedicación de mayor esfuerzo a la definición de procesos que a la institucionalización de los mismos [54].

### **2.6.2. Metodologías Ágiles [55]**

Las metodologías de desarrollo ágil a pesar de haber sido formuladas a mediados de los años 90, es, en la actualidad en donde su uso se ha proliferado. Estas metodologías nacieron con el afán de remplazar a las tradicionales o también denominadas metodologías *pesadas* de desarrollo de software. Fue entonces como en el año 2001 algunas de las personas afines con los principios de estas metodologías se agruparon formando lo que se conoce hoy como *Alianza Ágil*, que promueve el desarrollo ágil de aplicaciones. Es de esta forma como actualmente cada vez más empresas de desarrollo de software le apuestan al uso de estas metodologías para el desarrollo de sus productos y ha sido así como su popularidad ha ido cada vez más en aumento.

Los métodos ágiles fueron oficialmente iniciadas con el Manifiesto Agile en el 2001[56]. Algunos métodos ágiles realizan mayor énfasis en la descripción de prácticas y técnicas como Extreme Programming - XP (2003) (2004), otros en la gestión de las actividades como Scrum (2001) y otros en la integración continua (2007).

### **2.6.3. CMMI VS METODOLOGIAS AGILES [57] [58][59]**

A menudo se considera que las metodologías ágiles y las mejores prácticas expuestas en CMMI (Capability Maturity Model Integration) son contradictorias entre sí. Sin embargo, diferentes estudios demuestran que pueden ser compatibles y trabajar juntos mejorando drásticamente el rendimiento y el éxito del proyecto [60].

Según lo expuesto en el informe del SEI: “CMMI® or Agile: Why Not Embrace Both!”[61], existen 2 razones principales por las que se considera que son incompatibles las metodologías ágiles y CMMI:

1. Los primeros en usar CMMI son los desarrolladores de gran escala, riskaverse, sistemas de misión crítica, generalmente con altos niveles de supervisión de gestión de proyectos, mientras que los primeros en adoptar métodos ágiles en general, se centran en un solo equipo de proyectos de desarrollo u organizaciones pequeñas.
2. La información inexacta acerca de CMMI y Agile y el mal uso de ambos dio lugar a malas interpretaciones. Estas percepciones negativas que la posición de CMMI y ágil en desacuerdo entre sí surgió en gran parte de los siguientes factores:
  - Falta de información veraz y escasez de información precisa acerca de CMMI en la comunidad de métodos ágiles y viceversa.
  - Dificultades para el uso de terminología de la terminología y métodos Agile.
  - Uso indebido de CMMI y ágiles

CMMI-DEV en su versión 1.3 [62] ha sido ampliamente adoptado en todo el mundo para la disciplina de ingeniería de alta calidad. Esta versión tiene la capacidad de incluir modernos enfoques, así, como el uso de los métodos ágiles, Lean Six Sigma, y centrado en la arquitectura de desarrollo

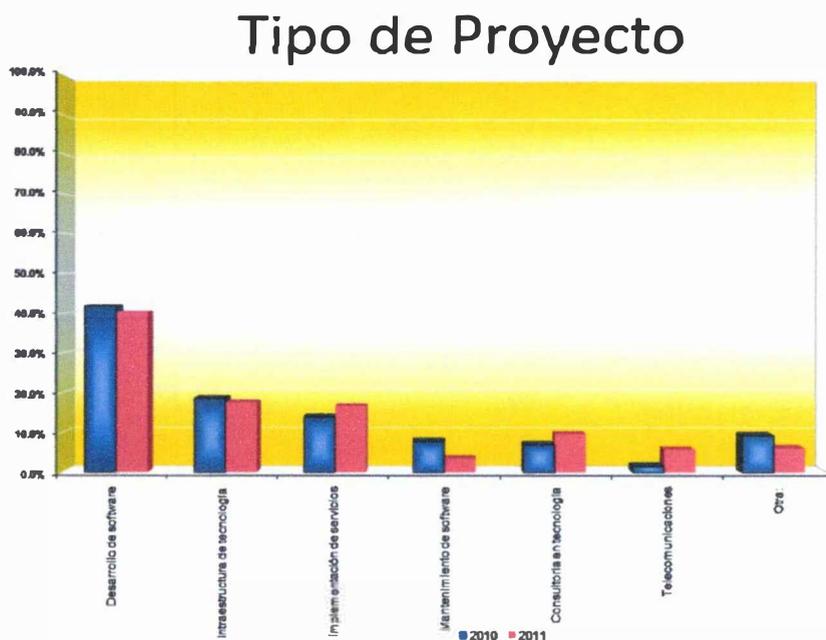
A lo largo del mundo han surgido nuevas situaciones, como por ejemplo, compañías certificadas en CMMI que necesitan introducir métodos de desarrollo ágiles para adaptarse a los cambios en el mercado, sin que ello conlleve a desechar totalmente las prácticas CMMI renunciando inclusive a su certificación CMMI que tan gran y exitosa inversión les ha llevado hasta el momento. También, podría ocurrir el caso de organizaciones que tienen ya implantados procesos de desarrollo ágiles de software bien consolidados, cuyos clientes les exigen cierto nivel de conformidad a CMMI. Existen casos de estudios sobre la correspondencia entre CMMI y métodos ágiles, que permiten

a pequeñas y medianas empresas implantar procesos de desarrollo software conformes a CMMI a través de métodos ágiles.

## 2.6.4. ESTADÍSTICAS DE GESTIÓN DE PROYECTOS

### 2.6.4.1. SITUACIÓN ACTUAL COLOMBIA

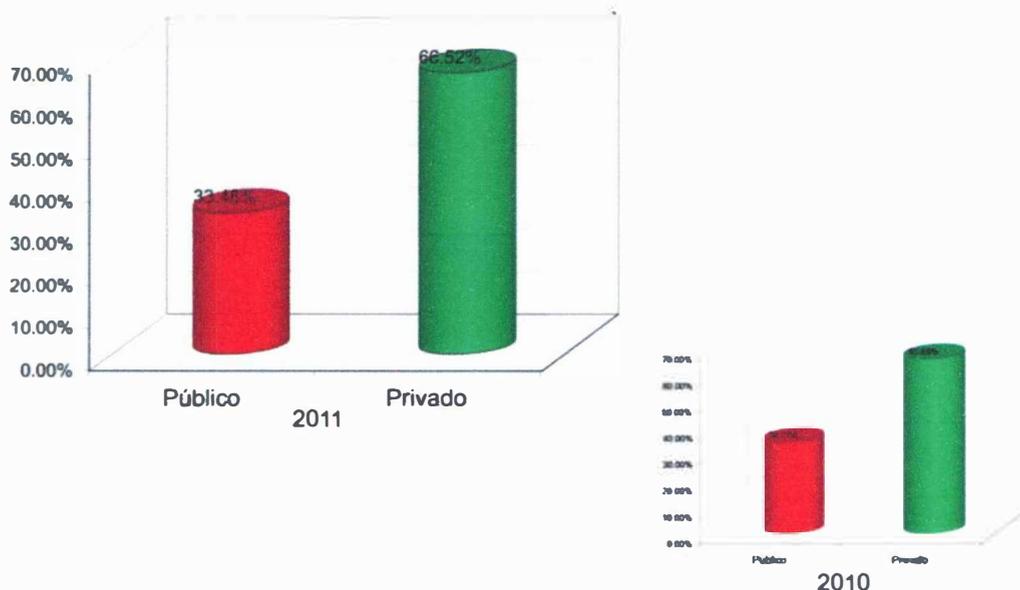
En la IX jornada de gerencias de TI se presentó por parte de la asociación Colombiana de Ingenieros de Sistemas: ACIS las tendencias actuales en cuanto al desarrollo de proyectos Colombianos relacionados con tecnología de informática. Este estudio fue realizado con una participación de 283 proyectos:



**Figura 11: Tipo de proyectos de tecnología de información en Colombia [63].**

Cómo se puede observar en la Figura 8 a nivel nacional los proyectos de desarrollo de software tanto en el año 2010 como en el año 2011 son los que en mayor porcentaje se llevan a cabo, con una diferencia altamente visible frente a los demás proyectos de tecnología de informática. En cuanto al sector donde se desempeñan estos proyectos se puede visualizar en la siguiente figura que el sector que mayor auge tiene en los proyectos es en el sector privado ya que duplica el porcentaje de proyectos realizados en el sector público con tan solo 33,48%. Esta tendencia se ha mantenido desde el año 2010 aunque disminuyó de forma mínima en un 0.63%.

## Sector del Proyecto



**Figura 12: Distribución de sectores 2011 vs 2010 [64].**

Otra variable importante que se analizó para los proyectos de tecnología de la información es el presupuesto planeado vs presupuesto real, en la figura 13 correspondiente a los años 2009 , 2010 y 2011 se mantiene la tendencia donde se evidencia que el mayor porcentaje de proyectos planean un presupuesto entre 100 y 499 millones de pesos y que los proyectos con presupuesto planeado entre 1000 y 4999 disminuyeron considerablemente en un 5% de 2009 a 2010 pero este mismo porcentaje incremento de 2010 al 2011.

## Presupuesto Planeado

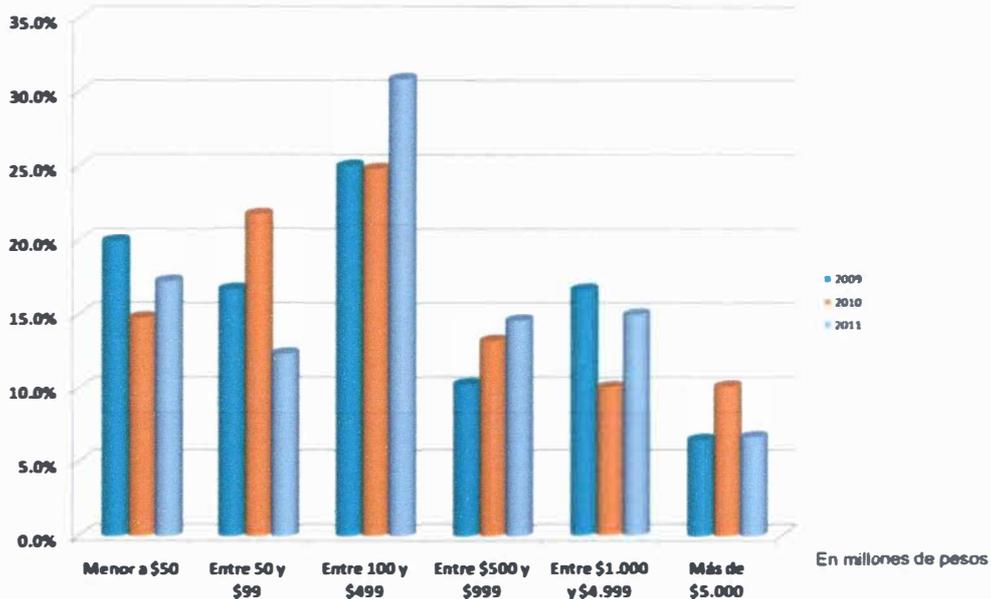


Figura 13: Presupuesto planeado 2009-2011 [65].

Sin embargo, en la realidad existe una diferencia de porcentajes de no más del 8% entre el presupuesto planeación y el presupuesto real. En la siguiente figura se evidencia esta tendencia a lo largo de los últimos 5 años.

## Presupuesto Real

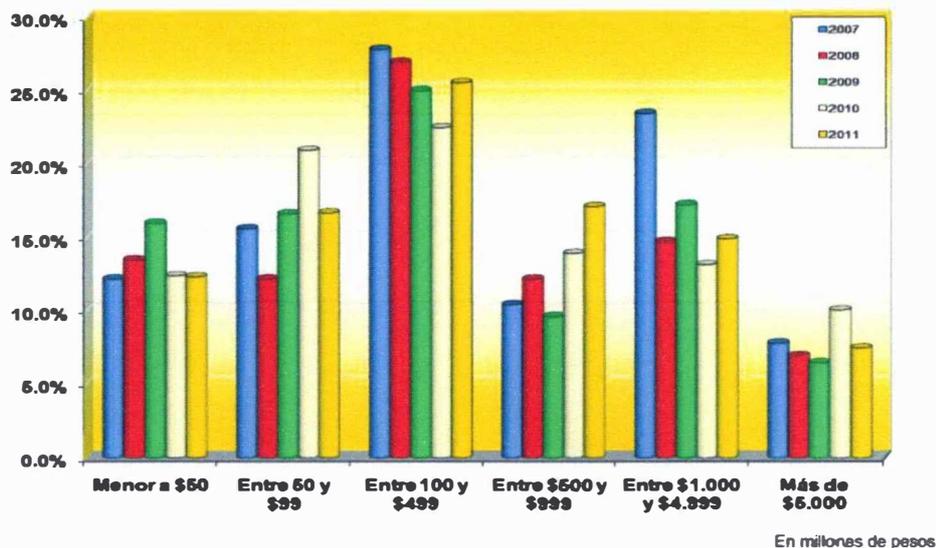


Figura 14: Presupuesto planeado últimos 5 años [66].

A manera de comparar el presupuesto planeado vs el presupuesto real se muestran las siguientes cifras donde se puede analizar que los presupuestos que más difieren tanto en el sector público como en el sector privado son los que están en el rango e 1000 y 4999 con una diferencia de más de 9%.

## Presupuesto planeado vs. real



Figura 15: Presupuesto planeado vs Presupuesto real [67].

En cuanto la variable duración, la tendencia se ha mantenido hacia una duración planeada de 4 a 6 meses alcanzando un porcentaje de hasta 30%, incluso del año 2009 al 2010 disminuyó el porcentaje en un 10% pero posteriormente para el año 2011 se incrementó superando en el 2009 en un porcentaje de 6%.

## Duración Planeada

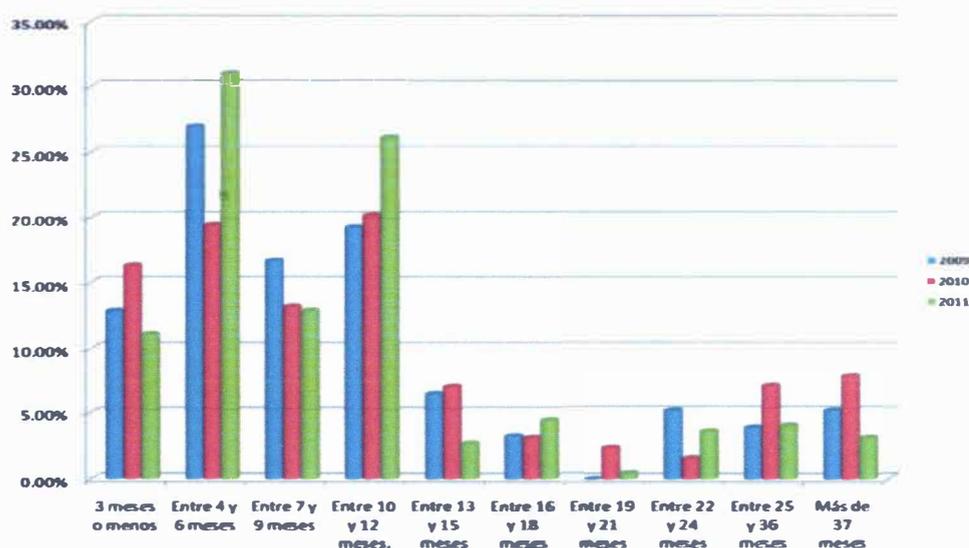


Figura 16: Duración planeado últimos 3 años [68].

En contraste con la duración planeada se tiene una duración real en los últimos 5 años distribuida por porcentajes de la siguiente forma:

## Duración Real

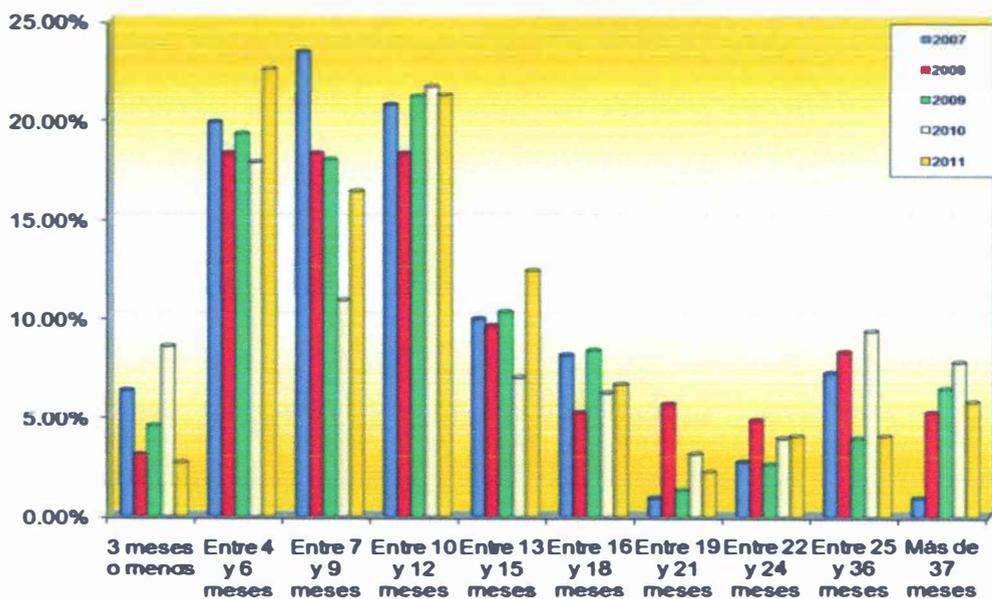


Figura 17: Duración real últimos 5 años [69].

En conclusión, se puede afirmar que para el año 2009 la diferencia entre la duración planeada vs la duración real era notoria dado que alcanzaba un porcentaje de hasta 5%, sin embargo, para los últimos 2 años la diferencia se ha ido disminuyendo y el porcentaje máximo de diferencia es de 3% como se muestra en la siguiente gráfica:

## Duración Planeada vs. Real

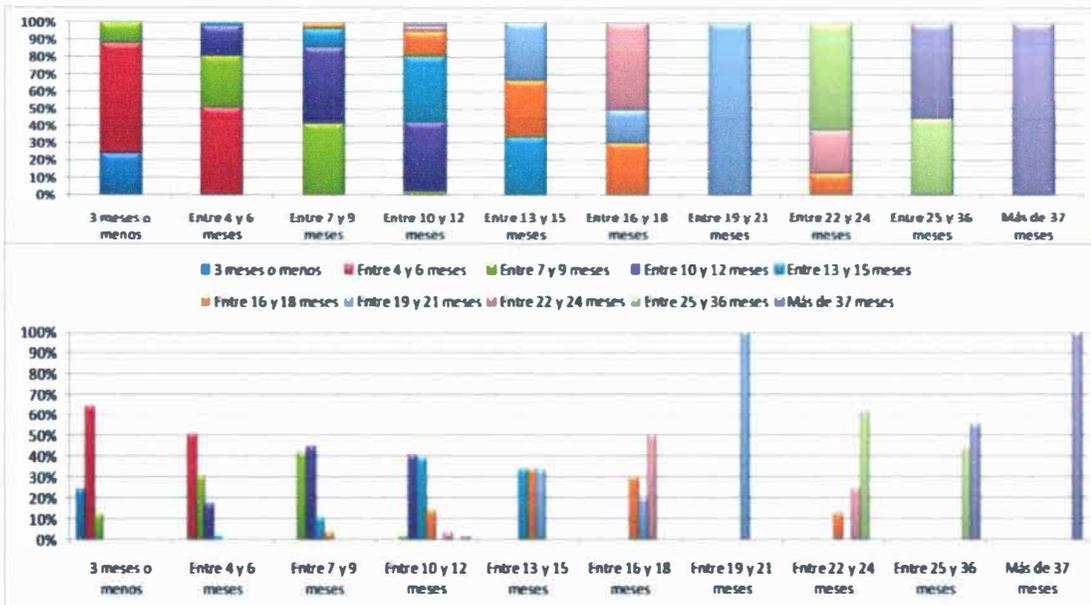
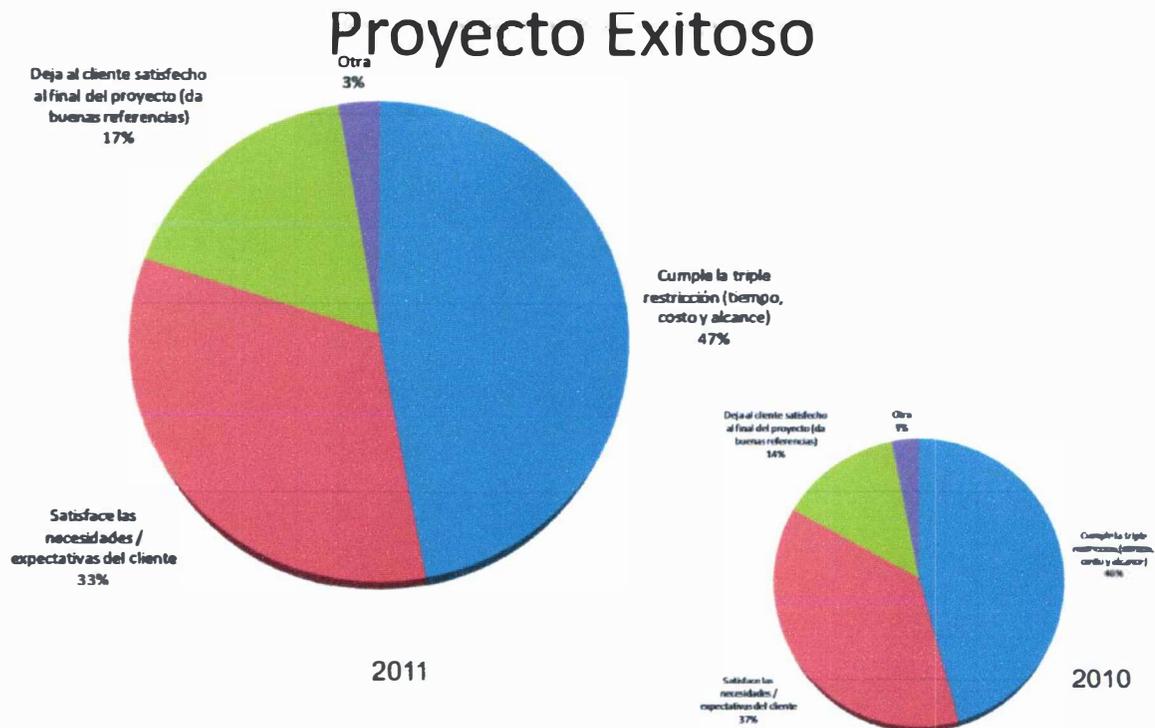


Figura 18: Duración planeada vs real [70].

Dentro de la encuesta se midió el porcentaje de las características que tiene en cuenta el cliente para calificar el proyecto como exitoso. Para este porcentaje se tuvo en cuenta las variables como tiempo, alcance y costo. Los resultados encontrados se muestran en la figura 16:



**Figura 19: Características de proyectos exitosos según clientes 2011 vs 2010 [71].**

Dentro de este resultado se observa que la tendencia de 2010 a 2011 se ha mantenido puesto que el porcentaje máximo de diferencia entre un resultado y otro es de 4%. También, se destaca que el mayor porcentaje de clientes (47%) coinciden en que el factor principal para que un proyecto sea exitoso es que cumpla con la triple restricción de tiempo costo y alcance. Esta tendencia se ha mantenido en los últimos 3 años y la variación ha sido mínima de un año al siguiente:

## Proyecto Exitoso

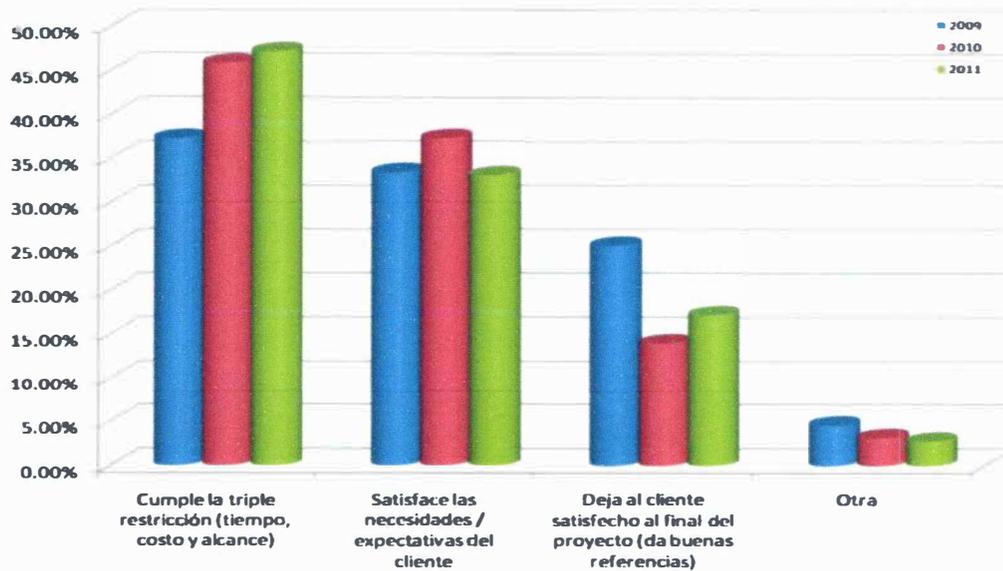


Figura 20: Características de proyectos exitosos según clientes últimos 3 años. [72].

En cuanto al uso de metodologías ágiles en los proyectos de tecnología en Colombia el estudio arrojó los siguientes resultados:

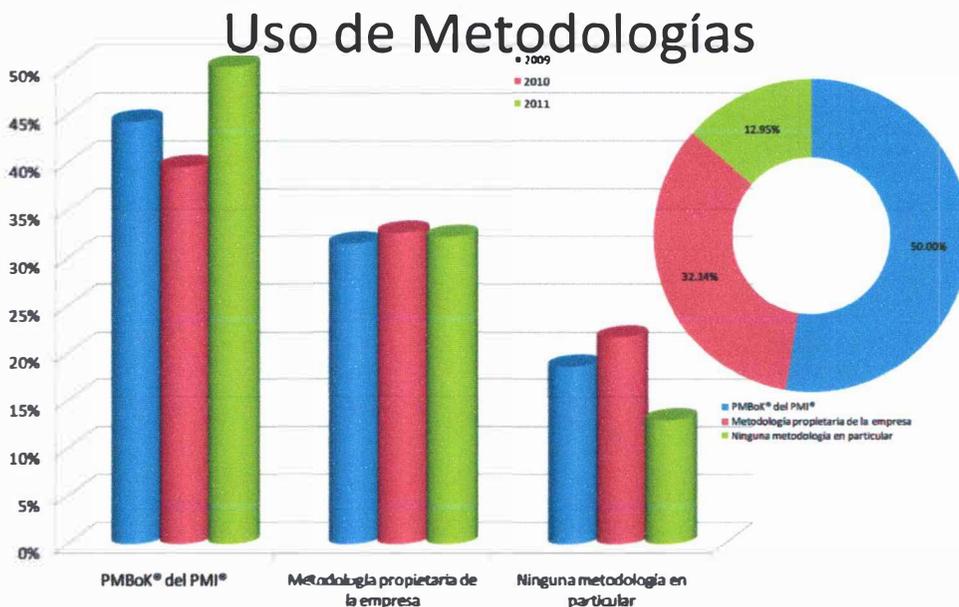


Figura 21: Uso de metodologías 2009-2011 [73].

En cuanto al uso de metodologías propietarias de las empresas el porcentaje se conservó de 2009 a 2011 y este último año aumentó el número de empresas que utilizan PMBOK como metodología. Para analizar el uso de herramientas y técnicas se muestra la siguiente figura donde se resume el % de utilización de herramientas y técnica para los proyectos en Colombia:

## Uso de Herramientas y Técnicas

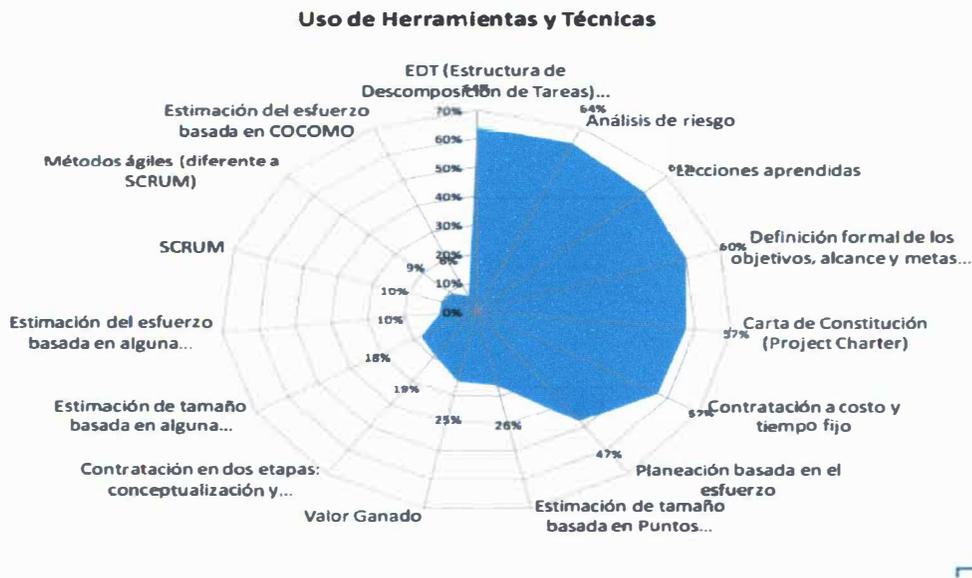


Figura 22: Uso de herramientas y técnicas [74].

### 2.6.4.2. SITUACION ACTUAL INTERNACIONAL

De la encuesta internacional denominada 2011 State of Agile Survey realizada entre los meses de junio y noviembre de 2011, con una muestra de 6.000 participantes y 500 empresas desarrolladoras de software. Se pudieron obtener los siguientes resultados:

A la pregunta de ¿Cuál metodología ágil ha usado o implementado? Se evidenciaron los siguientes resultados: Scrum y las variantes de Scrum cuentan con más de dos tercios de las metodologías que se utilizan en la actualidad, mientras que Kanban ha entrado en

la escena este año 2011. La categoría que experimentó un crecimiento fuerte fue la de los Híbridos personalizados que pasó a un 9% frente al 5% en la encuesta del año inmediatamente anterior. Como se puede evidenciar en la siguiente figura:

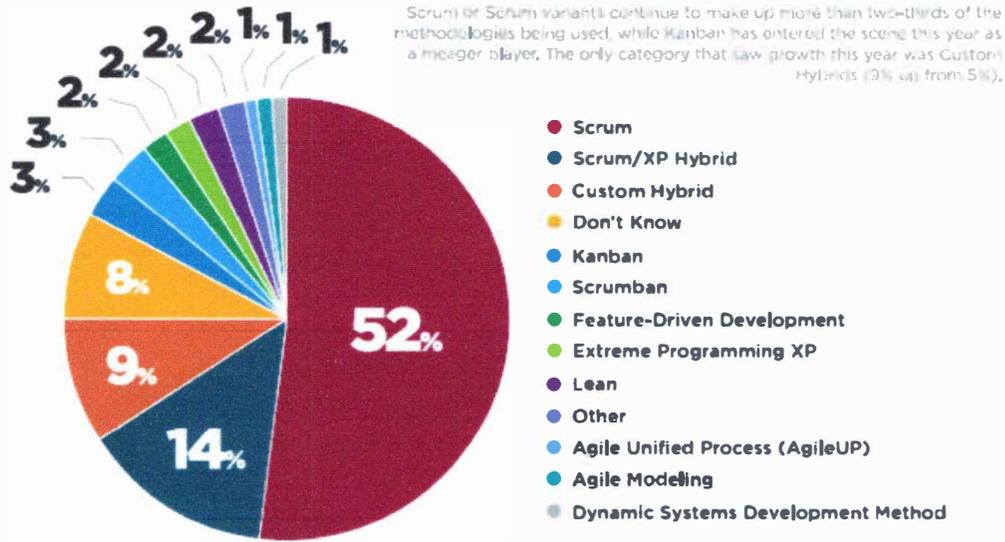


Figura 23: uso de metodologías ágiles [75].

En cuanto a la pregunta sobre el número de proyectos ejecutados empleando metodologías ágiles se obtuvieron los siguientes resultados:

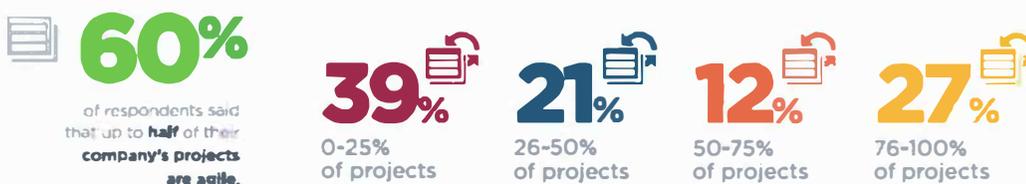


Figura 24: Número de proyectos usando metodologías ágiles [76].

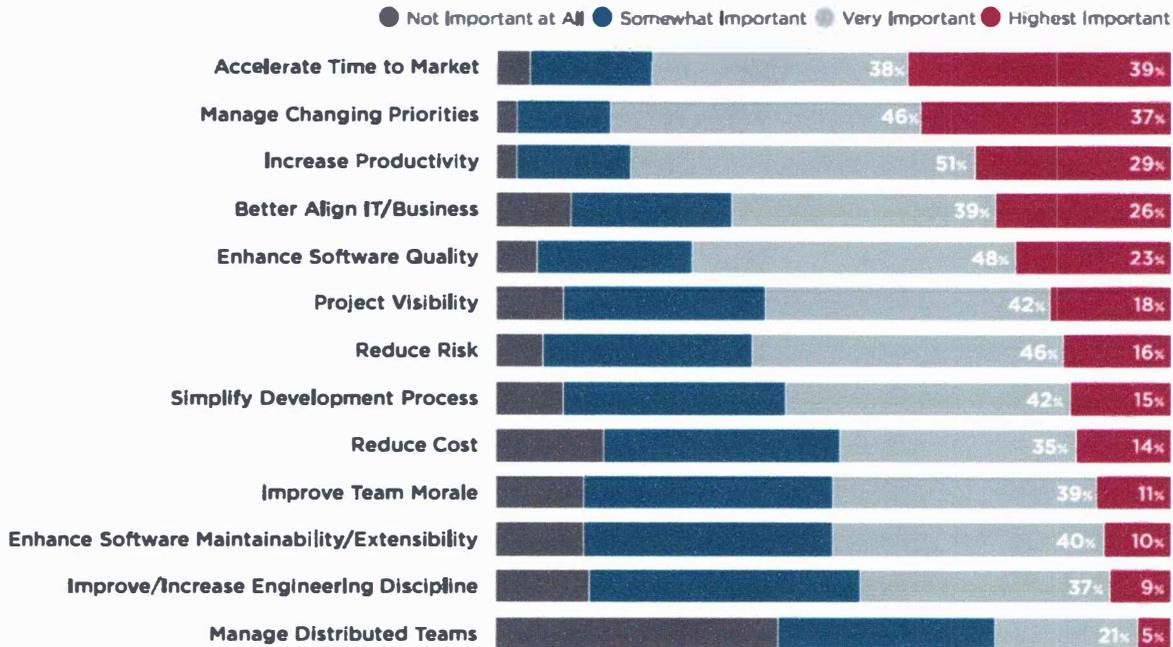
Analizando los resultados obtenidos en el estudio se puede afirmar que entre los participantes un 60 % de los proyectos ejecutados en su empresa o departamento de desarrollo son ágiles.

A la pregunta de cuáles prácticas ágiles enunciadas dentro del manifiesto ágil son mayormente empleadas dentro de los proyectos ejecutados, como se puede evidenciar en la siguiente gráfica las reuniones diarias tienen la delantera, mientras que kanban ha obtenido un crecimiento considerable de 24%. Respecto al 18% del año 2010.



**Figura 25: Prácticas ágiles empleadas [77].**

A pesar del repunte en el apoyo a la gestión ágil, resultados de la encuesta sugieren que el mayor obstáculo para una mayor adopción de ágiles no parece ser el conocimiento de la metodología, pero si la cultura internas de la compañía. Sólo el 13% de los encuestados en las organizaciones más grandes (más de 500 empleados), dijo que casi todos sus proyectos utilizan ágil [78]. En estas grandes empresas, los encuestados dijeron que la falta de apoyo a la gestión con un 27% y "resistencia general al cambio" un 26% fueron los principales obstáculos para la adopción ágil. Por el contrario, entre las empresas más pequeñas, la gran mayoría (casi el 75%) de los proyectos utilizados ágil y sólo el 10% de los encuestados citaron la falta de apoyo a la gestión o la "resistencia general al cambio" como un problema o una barrera a la adopción.



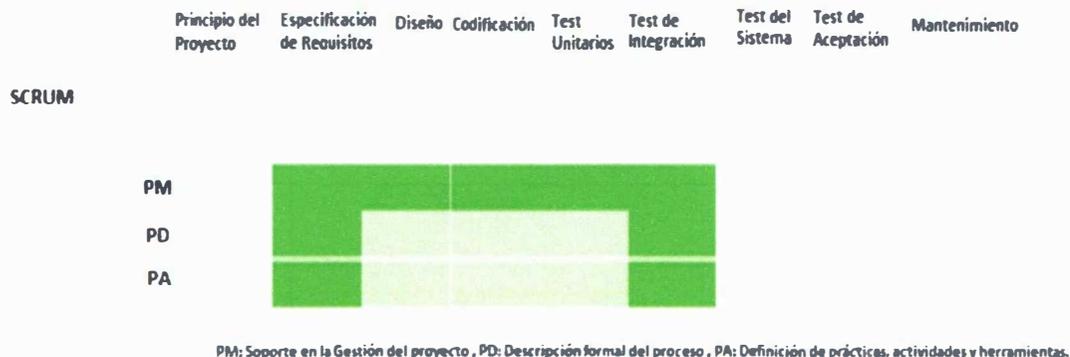
**Figura 26: Razones de adopción de ágil [79].**

Las tres razones principales por las cuales se han adoptado metodologías ágiles entre las cuales están: el acelerado crecimiento del mercado, incremento en la productividad, y la gestión de requerimientos cambiantes.

El 2011 survey también mostró un aumento en el número de empresas que actualmente no practican ágil, pero en el futuro lo harán (17% este año frente a un 13% el año pasado). De los que ya están practicando ágil, un tercio va a seguir haciéndolo, y sólo el 3% dijo que no va a continuar [80].

### Capítulo 3: Caracterización de CMMI Y SCRUM

Scrum es un Framework de gestión de proyectos, pero No presenta prácticas concretas que aseguren la calidad, su propósito no es ese, sino sacar el proyecto adelante de una forma colectiva y colaborativa. Tal y como se puede describir en la siguiente figura, en la cual se observa cómo se comporta cada una de las etapas de desarrollo de software en el marco de trabajo de Scrum.



**Figura 27. SCRUM: AREAS DE PROCESO DE DESARROLLO VS ETAPAS DE DESARROLLO. ADAPTADA DE [81].**

En la figura anterior se visualiza como actúan las áreas de proceso de desarrollo de software a lo largo de cada etapa en SCRUM. Se observa que este marco de trabajo en sus etapas de diseño, codificación y test de integración utiliza soporte en la gestión de proyecto (PM) y presentan deficiencia en los métodos que utilizan para describir sus prácticas, actividades y artefactos, adicionalmente no presenta una descripción formal del proceso al inicio del proyecto y en las etapas finales. Pero, ¿Por qué utilizar Scrum para el modelo planteado (SUMM)?, la siguiente presentación de criterios ayudará a entender por qué se hizo la selección de este marco de trabajo para el modelo propuesto. De igual forma, se analizará porque a partir de las falencias mostradas en la figura anterior se confirma que el marco de trabajo ideal para utilizar en el modelo propuesto dio como resultado Scrum.

Metodología	Agile Managemet (APM)	Crystal Methods	Dynamic Systems Development Methods (DSDM)	Scrum	Test Driven Development	Extreme Programmig (XP)
Define prácticas y actividades en cada etapa		X	X	X	X	
Integración con el cliente	X	X	X	X	X	X
Planificación de Requisitos	X	X	X	X	X	
Gestión de Calidad en el producto						
Integración de control de versiones				X		
Versiones integradas y planificación de iteraciones				X		
Acepta cambio constantes de requisitos				X		
Menos de 10 integrantes del equipo	X	X	X	X	X	X

**TABLA 5. COMPARACION DE METODOLOGIAS VS CRITERIOS.ELABORACIÓN PROPIA.**

De la comparación anterior se puede afirmar que aunque varias de las metodologías comparadas cumplen con los criterios analizados, Scrum presenta ventajas en varios de los criterios. Este marco de trabajo tiene características inmersas que lo han convertido en una de las metodologías ágiles más utilizadas a nivel mundial, según las estadísticas presentadas en el capítulo 2:

- Fácil usabilidad, flexibilidad y adaptación mediante aceptación de requerimientos cambiantes.
- Los clientes se convierten en parte del equipo de desarrollo.
- Se desarrollan planes de riesgos y mitigación frecuentes por parte del equipo de desarrollo, la mitigación de riesgos, la monitorización y la gestión de riesgos se lleva a cabo en todas las etapas.
- La gestión de las expectativas del cliente a lo largo de todas las etapas del proyecto mediante: lista de requisitos priorizada y demostración de los resultados en cada iteración.
- Resultados Anticipados por la utilización de listados priorizados de requisitos.

En cuanto a las desventajas de Scrum evidenciadas en la figura 26 se observa:

- Al principio del proyecto y en la especificación de requisitos no utiliza gestión del proyecto.
- En la etapa de finalización (test de aceptación, test del sistema y mantenimiento) no utiliza gestión del proyecto.
- No tiene definido prácticas y actividades en las etapas del principio del proyecto

Partiendo de estas falencias se puede entender que Scrum es el complemento adecuado combinada CMMI en el híbrido del modelo que se plantea (SUMM).

### **3.1. ¿Qué mejoras aporta CMMI a Scrum?**

De acuerdo a lo anterior, y recopilando un poco lo planteado en el capítulo anterior, se puede afirmar que:

- CMMI provee un mecanismo que permiten introducir mejoras continuas en los procesos de desarrollo, mediante el uso de la métrica en indicadores que permiten trazabilidad en los procesos.
- CMMI describe procesos orientados la vinculación de toda la organización y a los Stakeholders, tanto en el seguimiento de los proyectos como en el aprendizaje y los procesos de mejora continua
- Especificación formal de tareas, de tal forma, que pueda aprovecharse para establecer mecanismos de Lecciones aprendidas que permiten ser más eficiente y se evitar problemas relacionados con la calidad del proceso.
- El desarrollo más formal de requisitos de Stakeholders.

### **3.2. Correspondencia de las prácticas de Scrum con las áreas de proceso de CMMI [82]**

#### **3.2.1. Áreas de procesos de CMMI nivel dos (2) representación escalonada**

##### **3.2.1.1 Gestión de requerimientos - REQM**

En esta área de proceso el propósito es producir y analizar los requerimientos del cliente, de producto y de los componentes del producto [83].

<b>CODIGO</b>	<b>PRACTICA CMMI</b>	<b>SCRUM</b>
SP 1.1	Obtener las necesidades, las expectativas, las restricciones, y las interfaces de las partes interesadas para todas las fases del ciclo de vida del producto.	Revisar los requisitos con el propietario del producto y las partes interesadas.
SP 1.2	Obtener el compromiso de los participantes de proyecto sobre los requerimientos.	Dar a conocer las sesiones de planificación de las iteraciones para buscar compromiso por parte del equipo. 1. Agregar los cambios de requisitos en la lista de requisitos del producto.
SP 1.3	Gestionar los cambios a los requerimientos a medida que evolucionan durante el proyecto.	2. Gestionar los cambios en la reunión de planificación de la próxima iteración. 1. Establecer reuniones diarias para identificar los problemas.
SP 1.5	Identificar las inconsistencias entre los planes del proyecto, los productos de trabajo y los requerimientos.	2. Dar a conocer la sesiones de planificación para hacer frente a los problemas. 3. Tabla de seguimiento a las iteraciones para estimar el esfuerzo restante.

**Tabla 6: CORRESPONDENCIA PRÁCTICAS CMMI VS SCRUM EN REQM.ELABORACION PROPIA.**

### 3.2.1.2. Planificación de proyectos – PP

En términos básicos la finalidad de Planificación de proyecto (PP) es establecer y mantener planes que definan las actividades del proyecto [84].

<b>CODIGO</b>	<b>PRACTICA EN CMMI</b>	<b>SCRUM</b>
SP 1.1	Establecer una estructura de descomposición del trabajo (WBS) de alto nivel para estimar el alcance del proyecto.	Establecer la lista de objetivos/requisitos priorizada. La lista de objetivos/requisitos deben ser expresadas en forma de historia de usuarios. .
SP 1.2	Establecer y mantener las estimaciones de los atributos de los productos de trabajo y de las tareas.	Para cada objetivo/requisito se indica el valor que aporta al cliente y el coste estimado de completarlo.
SP 1.3	Definir las fases del ciclo de vida del proyecto en las que encuadrar el esfuerzo de la planificación.	Proceso definido por scrum.
SP 1.4	Estimar el esfuerzo y el coste del proyecto para los productos de trabajo y para las tareas, basándose en estimaciones razonadas.	Estimación de tiempo ideal. En la lista de objetivos/requisito se indican las posibles iteraciones y las entregas (releases) esperadas por el cliente .

**Tabla 7: CORRESPONDENCIA PRÁCTICAS CMMI VS SCRUM EN PP. ELABORACION PROPIA.**

CODIGO	PRACTICA EN CMMI	PRACTICAS SCRUM
SP 2.1	Establecer y mantener el presupuesto y el calendario del proyecto.	1. Estimación de l tiempo ideal y costo en cada iteración. 2. Cada objetivo hay que asociarle su condición de satisfacción, preferiblemente en forma de casos de prueba de aceptación, en el momento de crear la lista de requisito(Product Backlog)
SP 2.4	Planificar los recursos necesarios para ejecutar el proyecto.	Estimaciones de scrum en tiempo ideal , dar a conocer la planeación , lista de requisitos con asignación de responsables para cada iteración.
SP 2.6	Planificar la involucración de las partes interesadas identificadas.	Roles de proceso Scrum (incluyendo el equipo, Scrum Master, producto Propietario).
SP 2.7	Establecer y mantener el contenido del plan del proyecto global.	Publicación del Plan, Lista de requisitos en cada iteración.

**Tabla 8: CORRESPONDENCIA PRÁCTICAS CMMI VS SCRUM EN PP.ELABORACION PROPIA.**

CODIGO	PRACTICAS CMMI	PRACTICAS SCRUM
SP 3.1	Revisar todos los planes que afectan al proyecto para comprender los compromisos del proyecto.	•Reunión de planificación de las iteraciones • Reuniones diarias Scrum • Reunión de planificación de las iteraciones
SP 3.2	Reconciliar el plan del proyecto para reflejar los recursos disponibles y los estimados.	• Reuniones diarias Scrum • Reunión de planificación de las iteraciones
SP 3.3	Obtener el compromiso de las partes interesadas relevantes responsables de ejecutar y de dar soporte a la ejecución del plan.	• Reuniones diarias Scrum

**Tabla 9: CORRESPONDENCIA PRÁCTICAS CMMI VS SCRUM EN PP.ELABORACION PROPIA.**

### 3.2.1.3 Monitoreo y control del proyecto

El propósito es proporcionar una comprensión de los avances del proyecto de manera que las acciones correctivas apropiadas pueden ser tomadas cuando los resultados del proyecto se desvían significativamente del plan [85].

CODIGO	PRACTICA CMMI	PRACTICA SCRUM
SP 1.1	Monitorear los valores actuales del proyecto de acuerdo al plan	<ul style="list-style-type: none"> <li>• Sprint burndown chart .</li> <li>• Release burndown chart realiza el seguimiento de los story points</li> <li>• Project Task Board, es usado para realizar los seguimientos de los Stories</li> </ul>
SP 1.2	Seguimiento de los compromisos identificados en el plan del proyecto..	<ul style="list-style-type: none"> <li>• Reunión Daily Scrum</li> <li>• Reunión Sprint review</li> <li>• Sprint burndown chart</li> <li>• Release burndown chart Rastrea los Story points que han sido completados</li> </ul>
SP 1.5	Monitoreo de los involucrados según el plan del proyecto	<ul style="list-style-type: none"> <li>• Reunión Daily Scrum</li> <li>• Reunión Sprint review</li> </ul>
SP 1.6	Revisar periódicamente el progreso del proyecto, el rendimiento y Otros asuntos	<ul style="list-style-type: none"> <li>• Reunión Daily Scrum</li> <li>• Reunión Sprint review</li> </ul>

**Tabla 10: CORRESPONDENCIA PRÁCTICAS CMMI VS SCRUM EN MONITOREO Y CONTROL. ELABORACION PROPIA.**

<b>CODIGO</b>	<b>PRACTICA CMMI</b>	<b>PRÁCTICA SCRUM</b>
SP 1.7	Revisión de logros y resultados del proyecto	• Reunión Sprint review
SP 2.1	Recopilar y analizar los problemas y determinar las acciones correctivas necesarias para resolverlos	• Reunión Daily Scrum • Reunión Sprint review
SP 2.2	Ejecutar las acciones correctivas identificadas	• Reunión Daily Scrum • Reunión Sprint review
SP 2.3	Gestionar las acciones correctivas para el cierre	• Reunión Daily Scrum • Reunión Sprint review

**Tabla 11: CORRESPONDENCIA PRÁCTICAS CMMI VS SCRUM EN MONITOREO Y CONTROL ELABORACION PROPIA.**

#### **3.2.1.4. Gestión de la configuración**

No se especifica Ninguna práctica asociada en Scrum. Teniendo en cuenta que la Gestión de la configuración, requiere algo más que un control de versiones.

#### **3.2.1.5. Aseguramiento de la Calidad de Producto y Proceso - PPQA**

Scrum no implica automáticamente la aplicación PPQA. A pesar de que Algunas actividades básicas de la PPQA se están haciendo naturalmente cuando el Scrum Master comprueba que el Proceso de Scrum se está siguiendo:

- comentarear código,
- revisión de documento,
- pruebas,
- eliminación de procesos y obstáculos,
- ineficiencia

Sin embargo NO Formaliza específicamente

- Gestión de los niveles de Procesos Objetivos
- Estándares y procesos que deben ser definidos y usados

### 3.2.1.5. Medición y Análisis - MA

El propósito es desarrollar y mantener una capacidad de medición que se utiliza para apoyar la información de gestión necesidades. No hay prácticas en Scrum que establezcan un programa de medición similar al de MA. Sin embargo, las medidas en Scrum se pueden utilizar para implementar MA. [86]

### 3.2.2. Prácticas Genéricas de nivel 2, representación escalonada CMMI

CODIGO	PRACTICA GÉNERICA CMMI	PRÁCTICA SCRUM
GP 2.2	Establecer y mantener el plan de los procesos de REQM/PP/PMC/MA	Definición del ciclo de vida de Scrum y los milestone
GP 2.3	Proporcionar recursos adecuados para desarrollar los productos de trabajo y la prestación de los servicios en los procesos REQM / PP / PMC / MA	Los recursos y tiempo asignados para llevar a cabo la planificación de Scrum, seguimiento y requisitos.
GP 2.4	Asignar la responsabilidad y la autoridad para desarrollar los productos de trabajo y la prestación de los servicios en los procesos REQM / PP / PMC / MA	Las asignaciones de recursos asignados para llevar a cabo la planificación de Scrum, seguimiento y requisitos
GP 2.6	Lugar designado para trabajar los productos de los procesos de REQM/PP/PMC/MA, bajo los niveles adecuados de control	No especificado por las prácticas de Scrum
GP 2.8	Monitorear y controlar los procesos de REQM / PP / PMC / MA según el plan para llevar a cabo el proceso y tomar las medidas adecuadas y las acciones correctivas.	*El Scrum master monitorea que los pasos de Scrum sean seguidos

**Tabla 12: Prácticas Genéricas de nivel 2, representación escalonada CMMI VS SCRUM.ELABORACION PROPIA.**

### **3.2.3. Correspondencia CMMI nivel 3, representación escalonada**

Los siguientes componentes no son fácilmente implementados por Scrum:

- Enfoque del Proceso Organizacional
- Definición de procesos organizacionales
- Formación Organizacional
- Gestión Integrada de Proyectos
- Gestión de Riesgos
- Análisis de Decisiones y Resolución
- Verificación
- Validación

## Capítulo 4: Modelo de Gestión SUMM

El modelo que se propone es un modelo híbrido para desarrollo de proyectos de software ágiles teniendo en cuenta aspectos inherentes a la calidad del software, este modelo denominado SUMM combina las prácticas de SCRUM y la robustez de CMMI - DEV en su versión 1.3., lo cual lo transforma en un híbrido entre lo tradicional y lo ágil.

SUMM, es la sigla que describe el “Modelo de Madurez Unificado de Scrum”, este consta de cinco (5) niveles los cuales son: Inicial, Gestionado, Definido, Mejorado y Optimizado. A través de esta representación se busca que una organización se enfoque en mejorar un conjunto de procesos relacionados obteniendo así un nivel de madurez enmarcado en procesos de mejora continua, el diseño de SUMM por su misma naturaleza híbrida está basado en la representación por etapas de CMMI.

SUMM brinda un enfoque estructurado orientado a la mejora de los procesos de Scrum paso a paso, la consecución de una etapa sugiere que se ha dado un mejoramiento, de igual manera, se habrá establecido una base para dar inicio a la siguiente etapa. Entre las características de SUMM se encuentran que este modelo aplica a proyectos de desarrollo de software ya que está cimentada en CMMI – DEV, por estar basada en los principios y prácticas de Scrum los proyectos deben desarrollarse en un lapso no mayor a 6 meses, de igual forma los equipos de desarrollo deben estar conformados con un número máximo de 10 integrantes sin contar a los usuarios ni al cliente.

En la siguiente figura se observa la representación de SUMM propuesta, la cual consta de un nivel inicial uno (1) y el máximo nivel de madurez que es el cinco (5).



Figura 28 Niveles de Madurez representación por etapas SUMM Elaboración Propia

## 4.2. PRINCIPIOS DEL MODELO

### 4.2.1 NIVELES DE MADUREZ MODELO SUMM

Cómo se observó en la anterior figura SUMM contempla cinco (5) niveles de madurez, asociados a cada nivel se establecen metas basadas en las prácticas, principios y valores del marco de trabajo de Scrum. Tal y como se observa en la figura 27.

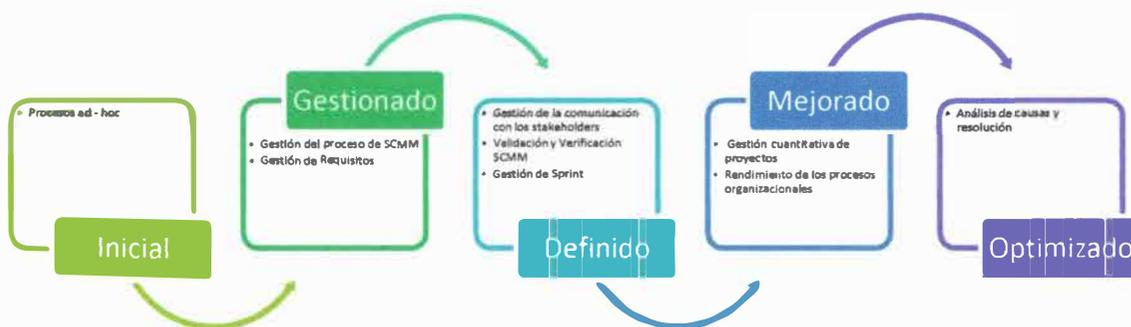


Figura 29. Niveles de madurez SUMM. Elaboración Propia

Como se mencionó anteriormente, SUMM está compuesto por una jerarquía de niveles de madurez, estos niveles lo componen Metas las cuales son un conjunto de prácticas que se definen con el fin de alcanzar determinados Objetivos. A continuación se presentan la relación de Metas, Objetivos y Prácticas de SUMM de manera general.

<b>Metas</b>	<b>Nivel de Madurez</b>	<b>Objetivos</b>	<b>Prácticas</b>
Gestión del proceso de SUMM	2	4	18
Gestión de Requisitos	2	3	12
Gestión de la comunicación con los Stakeholders	3	2	6
Validación y Verificación SUMM	3	1	5
Gestión de Sprint	3	3	13
Gestión cuantitativa de proyectos	4	1	2
Rendimiento de los procesos organizacionales	4	1	2
Análisis de causas y resolución SUMM	5	1	4
<b>Totales</b>		<b>16</b>	<b>62</b>

**Tabla 13: Niveles de Madurez Metas, Objetivos y Prácticas SUMM.**

De acuerdo a la anterior tabla, SUMM está compuesto de Ocho (8) Metas que componen el marco general de cada nivel de madurez, de igual forma, lo conforman un total de dieciséis (16) Objetivos y Sesenta y dos Prácticas (62) que darán cumplimiento a cada Meta. Cada nivel de madurez en SUMM se mide por el alcance de los Objetivos determinados por cada Meta o por un conjunto de ellas. Para tener una visión global de cada uno de los niveles de madurez anteriormente descritos respecto a su representación, en la siguiente figura se podrá visualizar de manera general cada nivel de madurez asociado con las Metas que lo conforman.

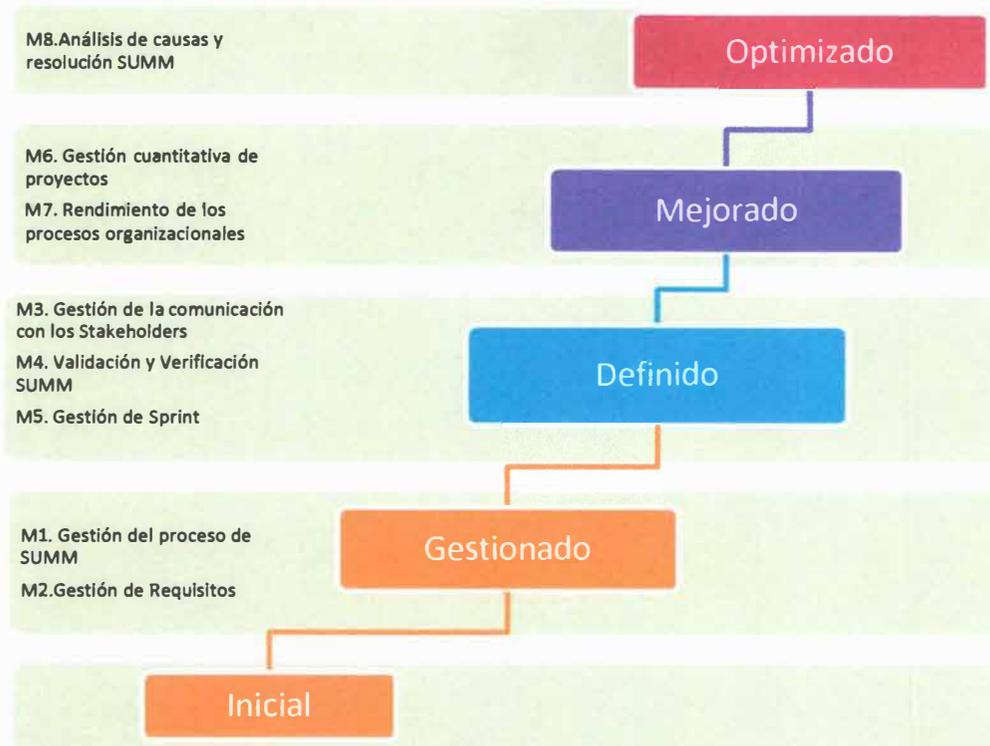


Figura 29. Metas por niveles de Madurez SUMM

A continuación se describe cada nivel de Madurez de SUMM

#### 4.2.1.1. Nivel Inicial

El nivel Inicial no consta de una estructuración de procesos, de igual manera, no hay objetivos que apunten hacia una mejora de procesos definidos, es el nivel más bajo de madurez dentro de SUMM. En este nivel, se encuentran las organizaciones que emplean algunas prácticas del marco de trabajo de Scrum, pero que aún no han definido con toda la formalidad y rigurosidad el cumplimiento del proceso de desarrollo bajo el marco de trabajo, ni mejoras de procesos. Principalmente, los problemas que se pueden asociar en este nivel son: incumplimiento en los tiempos proyectados, dificultades en la comunicación fluida entre el Team y los Stakeholders, calidad del software, costos de desarrollo.

El éxito en los proyectos de estas organizaciones generalmente depende de la capacidad y el nivel de preparación y competencia de su personal. Las organizaciones que no cumplan con las metas definidas para el nivel 2 están clasificadas en el nivel 1.

#### **4.2.1.2. Nivel Gestionado**

En este nivel los procesos asociados a las prácticas de desarrollo de software bajo el marco de trabajo de Scrum son más estructurados y completos que en el nivel anterior. Los objetivos de este nivel son principalmente la planificación del proyecto, Mejorar la gestión de requisitos, Mejorar las prácticas de valor, colaboración y planificación de procesos. Los problemas asociados a este nivel son aquellos relacionados con la comunicación de los stakeholders, establecimiento de estándares de codificación e integración y la satisfacción del cliente.

En este nivel la organización debe establecer las pautas de la planeación de proyectos a través del planning poker u otros métodos de planeación definidos por el marco de trabajo, se entiende planning poker como una técnica que ayuda a todo el equipo de Scrum a realizar estimaciones basados en el consenso, es empleado para estimar el esfuerzo o el tamaño relativo de las tareas del proceso de desarrollo de software descritas en el sprint backlog. Por otra parte, El objetivo de este nivel, es ayudar al Team y a los Stakeholders a identificar y mejorar los problemas relacionados con la planificación, ingeniería de requisitos en lugar de solo aprender de los éxitos y fracasos del proyecto anterior, estableciendo una bitácora de lecciones aprendidas. Para dar cumplimiento al objetivo de este nivel, se propone dar cumplimiento a dos (2) metas, las cuales son:

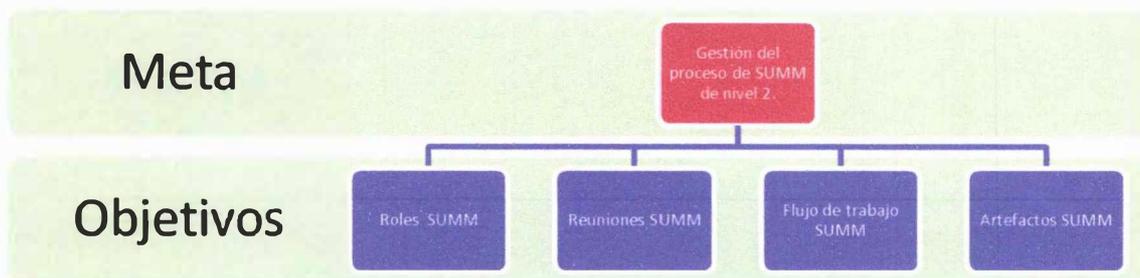
**4.2.1.2.1. Gestión del proceso de SUMM:** Esta meta tiene como propósito el cumplimiento del flujo básico del marco de trabajo de Scrum, estableciendo

- roles,
- división de la organización, en equipos pequeños, interdisciplinarios y auto-organizados,
- división del trabajo, en una lista de entregables pequeños concretos y priorizados,
- división del tiempo, en Sprint no superior a 4 semanas, generando código potencialmente entregable y testeado al finalizar cada sprint,

- plan de entregas, actualizando prioridades en colaboración con el stakeholder,

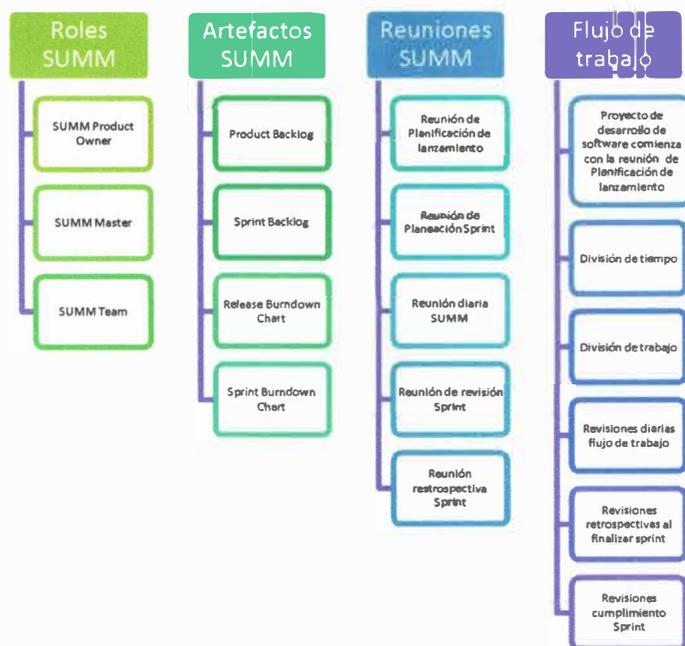
Sin embargo, en este nivel las organizaciones a pesar de que gestionan sus procesos de desarrollo a través del marco de trabajo de Scrum, de acuerdo a las prácticas mencionadas anteriormente, es posible que a dichas prácticas no se les de cumplimiento de manera eficiente, eficaz y óptima.

A continuación, se podrá visualizar la estructura general de la Gestión del proceso de Scrum de nivel 2.



**Figura 30: Estructura General Gestión Proceso Scrum Nivel 2 de SUMM.  
Elaboración propia.**

La estructura general de la meta de Gestión de Proceso de SUMM en el nivel 2 de SUMM describe cuatro (4) objetivos pilares los cuales son: Asignación de roles de SUMM, Gestión de reuniones de SUMM, Flujo de trabajo de SUMM y los artefactos de SUMM, como se visualiza en la anterior figura, sin embargo, en el siguiente esquema se describen cada uno de estos objetivos y las prácticas de la Gestión de proceso SUMM nivel 2



**Figura 19: Objetivos y prácticas Gestión de proceso de SUMM nivel 2 SUMM.Elaboración Propia.**

De acuerdo a lo anterior, a continuación se describen los objetivos, prácticas y las métricas de la Gestión de proceso de SUMM de acuerdo al nivel 2 de SUMM

- **Objetivo 1: Asignación Roles Scrum**

Este objetivo denota tres (3) prácticas asociadas las cuales son: Definición Team, Definición SUMM Master, Definición Product Owner.

- **Práctica 1. Definición SUMM Product Owner:** Este rol está conformado por máximo dos (2) representantes del cliente y representan a todos los stakeholders, lidera el desarrollo del producto, activo impulsor de una visión del producto alineada con los objetivos del negocio. Entre las características de su perfil se destacan las siguientes:

- Debe tener los conocimientos necesarios sobre el producto y/o proceso que va a recibir. Además, debe tener excelente visión y conocimiento de la razón de ser del producto.
- Tener capacidad y autorización para tomar decisiones en cualquier momento que se requiera.

- Antes de seleccionar el SUMM Product Owner se debe tener en cuenta que es él, quien va a ser un experto en los requisitos del producto, esta persona no puede ser temporal y debe garantizar su permanencia en la compañía durante el desarrollo del producto.
- El SUMM Product Owner debe ser una persona pro-activa y comprometida en el desarrollo del producto, con tiempo y dedicación para definir los requerimientos y con la premisa de sacarlo adelante.

Entre sus principales funciones están:

- Elaborar la lista de requisitos.
  - Definir los objetivos del producto o proyecto.
  - Priorizar la lista de requisitos.
  - Revisar y aceptar el producto de cada iteración.
  - Retroalimentar al Team de desarrollo sobre los errores o solicitudes requeridas encontradas en el resultado de cada iteración.
  - Vinculación al desarrollo del producto tiempo completo.
- **Práctica 2. Definición SUMM Master:** Es el Primer responsable de que el desarrollo del producto sea un éxito, debe ser el ente motivador en el equipo de trabajo. Entre las características de su perfil se destacan las siguientes:
    - Profesional con capacidad de Liderazgo.
    - Capacidad para tomar decisiones y ejecutor.
    - Líder en la implantación de mejoras de proceso, capaz de detectar y resolver oportunamente impedimentos que se presenten.

Entre sus principales Funciones están:

- Velar por que todos los participantes del proyecto sigan las reglas y proceso establecido en SUMM.
  - Asegurar que la lista de requisitos priorizada esté preparada para cada inicio de iteración.
  - Facilitar las reuniones establecidas en SUMM de manera que sean productivas y consigan sus objetivos.
  - Enseñar al equipo a auto gestionarse. Es decir, guiar al equipo con preguntas para que descubra por sí mismo una solución.
  - Solucionar los impedimentos que el equipo tiene en su camino para conseguir el objetivo de cada Sprint.
  - Proteger y aislar al equipo de interrupciones externas durante el desarrollo de cada Sprint.
- **Práctica 3. Definición SUMM Team:** es un grupo de personas que de manera conjunta desarrollan el producto del proyecto. Comparten la responsabilidad del trabajo que se realiza en cada Sprint. Debe estar conformado por máximo ocho (8) personas y mínimo cinco (5). Entre las características de su perfil se destacan las siguientes:
    - Los miembros del equipo deben dedicarse al proyecto a tiempo completo para evitar dañar su productividad por cambios de tareas en diferentes proyectos, para evitar interrupciones externas y así poder mantener el compromiso que adquieren en cada Sprint.
    - Idónea y con las habilidades y destrezas técnicas requeridas
    - Excelente actitud y entusiasmo.
    - Trabajo en equipos sinérgicos

Entre sus principales Funciones están:

- Seleccionar los requisitos que se compromete a completar en un Sprint, de forma que estén preparados para ser entregados al cliente.
- Estimar la complejidad de cada requisito en la lista de requisitos priorizada del producto o proyecto.
- Durante el Sprint, trabajar de manera conjunta para conseguir los objetivos del Sprint. Cada especialista lidera el trabajo en su área y el resto colaboran si es necesario para poder completar un requisito.

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas.

Meta	Objetivo	Practica	Métrica
Gestión Proceso SUMM	Asignación de Roles SUMM	Definición SUMM Product Owner	Número de cambios de rol del Product Owner durante el proyecto.
		Definición SUMM Master	Número de cambios de rol del SUMM Master durante el proyecto.
		Definición SUMM Team	Número de cambios de rol del Team durante el proyecto. Número de cambios de miembros en el Team durante el proyecto.

**Tabla 14.** Prácticas y métricas Asignación de Roles. Gestión Proceso SUMM nivel 2 SUMM. Elaboración Propia.

- **Objetivo 2: Artefactos SUMM**

Este objetivo lo conforman cuatro (4) prácticas asociadas las cuales son: Definición del product Backlog, Definición Sprint backlog, Release Burndown chart, Sprint Burndown chart.

- **Práctica 1. Definición Product Backlog:** también denominado definición de la pila de producto, se trata de establecer un documento que contenga la descripción de todos los requerimientos o user stories, funcionalidades organizadas de acuerdo al valor del negocio aportado por el SUMM Product Owner. Para construir el Product Backlog es importante tener en cuenta que los User Stories de mayor prioridad deben tener mayor nivel de detalle en su descripción.

Por otra parte, es importante resaltar que la definición Product Backlog no sólo incluye nuevos requisitos o User Stories para el producto, sino también mejoras en requisitos ya implementados o correcciones.

- **Práctica 2. Definición Sprint Backlog:** para definir el sprint Backlog es necesario elaborar un documento en donde el equipo describa como se implementaran los User stories definidos en el product backlog durante el sprint. Las tareas dentro del Sprint Backlog se dividirán en horas, sin embargo, ninguna tarea deberá tener una duración superior a 16 horas. Si una tarea es mayor de 16 horas, deberá ser dividida en otras menores. Las tareas en el sprint backlog nunca deben ser asignadas, por el contrario, deben ser tomadas por los miembros del SUMM Team del modo que les parezca oportuno.
- **Práctica 3. Definición Release Burndown Chart:** Al construir este artefacto se puede identificar cuánto falta por terminar a nivel global en el proceso de desarrollo, debido a que este facilita al SUMM Product Owner proyectar cuándo se terminará el proyecto, ya que se emplean los Sprints como métrica de tiempo. Si se observa que en el Release Burndown Chart no existe un avance, no necesariamente es porque el SUMM Team se esté retrasando, sino es porque quizás se está cumpliendo con funcionalidades adicionales.

- Práctica 4. Definición Sprint Burndown Chart:** Este artefacto se define de acuerdo al nivel de avance en el proyecto de desarrollo, se hace por medio de una gráfica de tiempo estimado Vs. tiempo devengado, compuesta de dos ejes: el horizontal que muestra el tiempo de duración del Sprint y el vertical que muestra el trabajo por terminar, siendo las unidades de trabajo definidas de manera arbitraria, de acuerdo a las preferencias del SUMM Team.

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas.

Meta	Objetivo	Práctica	Métrica
Gestión Proceso SUMM	Artefactos SUMM	Definición Product Backlog	Número de actualizaciones al producto Backlog
		Definición Sprint Backlog	Número de actualizaciones al Sprint Backlog
		Definición Release Burndown Chart	Número de actualizaciones a lRelease Burndown Chart
		Definición Sprint Burndown Chart	Número de actualizaciones al Release Burndown Chart

**Tabla 15.** Prácticas y métricas Artefactos SUMM. Gestión Proceso SUMM nivel 2  
SUMM Elaboración Propia.

- **Objetivo 3: Flujo de trabajo SUMM**

Este objetivo lo conforman seis (6) prácticas asociadas las cuales son: El Proyecto de desarrollo de software comienza con la reunión de Planificación de lanzamiento, División del tiempo, División del trabajo, Revisiones diarias Flujo de trabajo, Revisiones retrospectivas al finaliza Sprint, Revisiones de cumplimiento Sprint.

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

Meta	Objetivo	Práctica	Métrica
Gestión Proceso SUMM	Flujo de trabajo	Reunión de Planificación de lanzamiento	Duración media de Sprint
		División del tiempo	
		División del trabajo	Número de revisiones retrospectivas
		Revisiones diarias Flujo de trabajo	
		Revisiones retrospectivas al finalizar Sprint	Porcentaje de cumplimiento del Sprint
Revisiones de cumplimiento Sprint			

**Tabla 16.** Prácticas y métricas Flujo de trabajo. Gestión Proceso SUMM nivel 2 SUMM. Elaboración Propia.

- **Objetivo 4: Reuniones SUMM**

Este objetivo lo conforman cinco (5) prácticas asociadas las cuales son: Reunión de planificación de lanzamiento, Reunión diaria, Reunión de revisión de Sprint, Reunión Retrospectiva de Sprint, Reunión de planeación de Sprint.

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

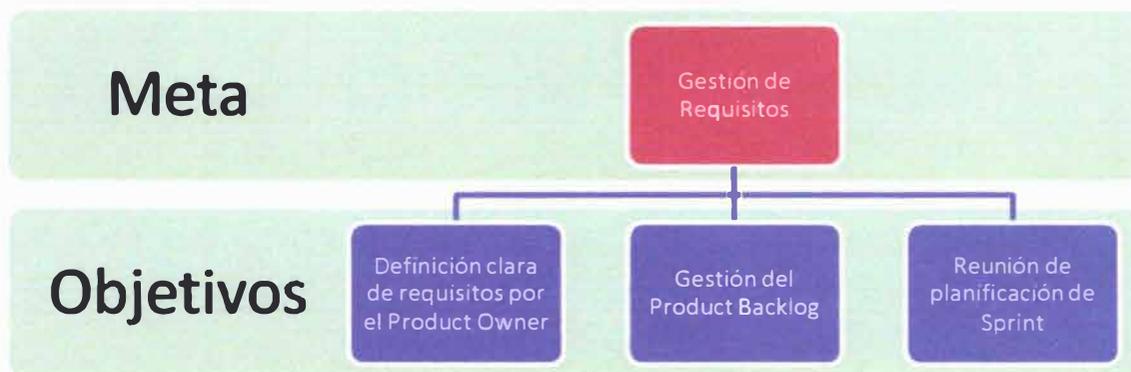
Meta	Objetivo	Práctica	Métrica
Gestión Proceso SUMM	Reuniones SUMM	Reunión de Planificación de lanzamiento	<ul style="list-style-type: none"> <li>• Porcentaje de las reuniones programadas de planificación</li> <li>• Porcentaje de asistentes reales contra los asistentes previstos de las reuniones de planificación de entregas</li> <li>• Duración media de las reuniones de planificación de entregas</li> </ul>
		Reunión Diaria	<ul style="list-style-type: none"> <li>• Porcentaje de reuniones diarias de Scrum</li> <li>• Porcentaje de asistentes reales contra los asistentes previstos de reuniones diarias de Scrum</li> </ul>
		Reunión de revisión de Sprint	<ul style="list-style-type: none"> <li>• Porcentaje de asistentes reales contra los asistentes previstos de las reuniones de revisión de Sprint</li> <li>• Media de duración de</li> </ul>

			la reunión de revisión de Sprint
		Reunión retrospectiva de Sprint	<ul style="list-style-type: none"> <li>• Porcentaje de las reuniones programadas de retrospectiva</li> <li>• Porcentaje de asistentes reales contra los asistentes de las reuniones planificadas retrospectivas</li> <li>• Media de duración de la reunión retrospectiva Sprint</li> </ul>
		Reunión de Planeación Sprint	<ul style="list-style-type: none"> <li>• Porcentaje de las reuniones de planeación de sprint programadas</li> <li>• Porcentaje de asistentes reales contra los asistentes de las reuniones planificadas de sprint</li> <li>• Media de duración de la reunión de planeación de Sprint</li> </ul>

**Tabla 17.** Prácticas y métricas Reuniones SUMM. Gestión Proceso Scrum nivel 2 SUMM. Elaboración Propia.

**4.2.1.2.2. Gestión de Requisitos:** Mientras que se realiza la planificación, se estiman la duración de los sprint y se organiza la pila de producto de acuerdo a las necesidades planteadas por el stakeholder a través del SUMM Product Owner del proyecto, se deben tener en cuenta los requisitos para poder establecer los compromisos de los participantes del proyecto, del mismo modo se debe comparar los requisitos de acuerdo al alcance y la definición misma del proyecto para poder identificar cualquier inconsistencia, incongruencia y evitar las ambigüedades. Según lo anterior de manera general, la Gestión de requisitos comprende un conjunto de prácticas que las organizaciones de nivel 2 de SUMM deben cumplir con el fin de lograr la satisfacción de la calidad del producto final desarrollado por la organización teniendo en cuenta la calidad del proceso para concebirlo.

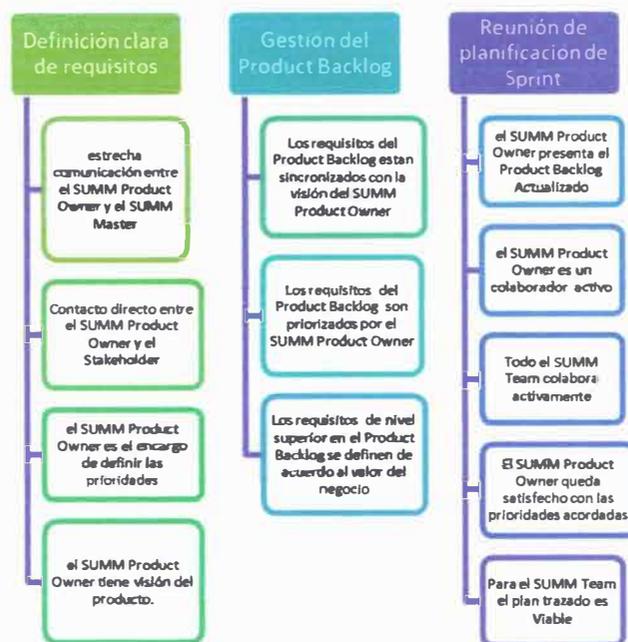
A continuación, se podrá visualizar la estructura general de la Gestión de requisitos de nivel 2 de SUMM.



**Figura 30.** Gestión de requisitos de nivel 2 de SUMM. Elaboración Propia.

Como se ilustra en la figura anterior, la gestión de requisitos de nivel 2 de SUMM, destaca la necesidad de una definición clara de requisitos por parte del SUMM Product Owner para dirigir el SUMM Team y la correcta aplicación de los conceptos de Scrum en relación a la gestión del Product Backlog y reuniones de planificación de Sprint exitosas.

En este orden de ideas, para alcanzar una meta se debe dar cumplimiento a los objetivos descritos para esto, adicionalmente a las prácticas asociadas y a las métricas de uso para medir la calidad de la implementación de estas.



**Figura 31. Relación objeto, práctica y métrica. Elaboración Propia**

A continuación se observará más en detalle esta relación Objetivo, práctica y métrica.

- **Objetivo 1: Definición clara de requisitos por el Product Owner**

Este objetivo lo conforman cuatro (4) prácticas asociadas las cuales son: estrecha comunicación entre el SUMM Product Owner y el SUMM Master, Contacto directo entre el SUMM Product Owner y el Stakeholder, el SUMM Product Owner es el encargado de definir las prioridades, el SUMM Product Owner tiene visión del producto.

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

Meta	Objetivo	Práctica	Métrica
Gestión de requisitos	Definición clara de requisitos por el Product Owner	Estrecha comunicación entre el SUMM Product Owner y el SUMM Master	Porcentaje de reuniones atendidas por el SUMM Product Owner
		Contacto directo entre el SUMM Product Owner y el Stakeholder	Porcentaje de reuniones canceladas por el SUMM Product Owner
		El SUMM Product Owner es el encargado de definir las prioridades	
		El SUMM Product Owner tiene visión del producto.	

**Tabla 18.** Prácticas y métricas Definición clara de requisitos por el Product Owner. Gestión de requisitos nivel 2 SUMM .Elaboración Propia

- **Objetivo 2: Gestión del Product Backlog**

Este objetivo lo conforman tres (3) prácticas asociadas las cuales son: Los requisitos del Product Backlog están sincronizados con la visión del SUMM Product Owner, los requisitos del Product Backlog son priorizados por el SUMM Product Owner, los requisitos de nivel superior en el Product Backlog se definen de acuerdo al valor del negocio.

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

Meta	Objetivo	Práctica	Métrica
Gestión de requisitos	Gestión del Product Backlog	Los requisitos del Product Backlog están sincronizados con la visión del SUMM Product Owner	Porcentaje de las prioridades definidas en el Product Backlog
		Los requisitos del Product Backlog son priorizados por el SUMM Product Owner	Número de Requisitos que conforman el Product Backlog Porcentaje de requisitos de nivel superior definidos en el Product Backlog.
		Los requisitos de nivel superior en el Product Backlog se definen de acuerdo al valor del negocio	

**Tabla 19.** Prácticas y métricas Gestión del Product Backlog. Gestión de requisitos nivel 2 SUMM .Elaboración Propia

- **Objetivo 3: Reunión de planificación de Sprint**

Este objetivo lo conforman cinco (5) prácticas asociadas las cuales son: el SUMM Product Owner presenta el Product Backlog actualizado, el SUMM Product Owner es un colaborador activo, Todo el SUMM Team colabora activamente, el SUMM Product Owner queda satisfecho con las prioridades acordadas, para el SUMM Team el plan trazado es Viable.

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

Meta	Objetivo	Práctica	Métrica
Gestión de requisitos	Reunión de planificación de Sprint exitosa	El SUMM Product Owner presenta el Product Backlog actualizado	Porcentaje de consenso en la definición de la planeación del Sprint
		El SUMM Product Owner es un colaborador activo	Porcentaje de consenso en la confianza de la viabilidad del plan trazado
		Todo el SUMM Team colabora activamente	Número de cambios en las prioridades del Sprint
		el SUMM Product Owner queda satisfecho con las prioridades acordadas	Backlog tras reunión de planificación de Sprint
		Para el SUMM Team el plan trazado es Viable	Nivel de satisfacción con las prioridades acordadas

**Tabla 20** Prácticas y métricas Reunión de planificación de Sprint exitosa. Gestión de requisitos nivel 2 SUMM .Elaboración Propia

#### 4.2.1.3. Nivel Definido

Definido, corresponde al nivel 3 de SUMM, en este la relación con los stakeholders, los procesos de validación y verificación, las entregas a tiempo según lo planeado son el pilar principal. De acuerdo a lo anteriormente planteado, las metas de este nivel son la Gestión de relaciones con el cliente, comunicación SUMM Team y la Gestión de sprint.

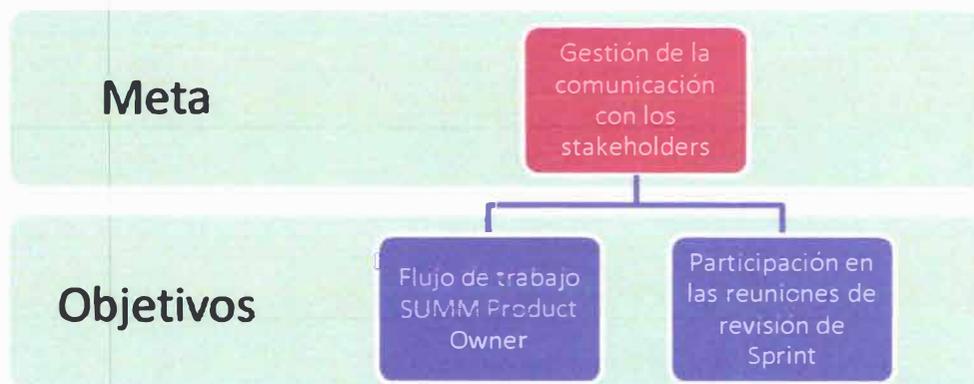
Las organizaciones en este nivel, tienen un mayor control sobre las prácticas relacionadas con el establecimiento de canales efectivos de comunicación con los stakeholders, la gestión de riesgos pasa a ser un ítem importante, sin embargo, no se realiza un panorama de riesgos de forma rigurosa ni exhaustivo.

Para dar cumplimiento al objetivo de este nivel, se propone dar cumplimiento a cuatro (4) metas, las cuales son:

**4.2.1.3.1. Gestión de la comunicación con los stakeholders:** Si se parte del hecho de que el propósito principal en los procesos de desarrollo de software es la satisfacción de las necesidades de los stakeholders, se puede afirmar que estos son el pilar fundamental que brinda apoyo a estos procesos. En las metodologías ágiles los stakeholders pasan de ser un actor pasivo a participar activamente en el proceso de desarrollo, inclusive en Scrum estos ocupan uno de los roles dentro del flujo de trabajo como se ha evidenciado en ítems anteriores.

Basados en lo anterior, se hace necesario incrementar los esfuerzos en estrechar la comunicación con los stakeholders, procurando establecer canales efectivos y eficaces entre los stakeholders, su representante, SUMM Product Owner, SUMM Master y el SUMM Team.

A continuación, se podrá visualizar la estructura general de la Gestión de la comunicación con los stakeholders.



**Figura 21. Estructura general Gestión de la comunicación con los Stakeholders SUMM nivel 3 .Elaboración Propia**

Como se ilustra en la figura anterior, la gestión de la comunicación con los stakeholders de SUMM, es una meta importante ya que se informa a los stakeholders de una forma clara, transparente y veraz, como va el proceso. En los proyectos propiciados en entornos distribuidos, el SUMM Product Owner debe con la ayuda del SUMM Master colocar a disposición de los stakeholders los mecanismos de comunicación suficientes para que éstos puedan involucrarse en el proyecto. De esta manera, el SUMM Team puede gestionar el flujo trabajo en los Sprint planeados de la forma más eficaz posible, sorteando los obstáculos que se encuentren durante su desarrollo y ejecución, de tal forma que, tanto las funcionalidades como los requisitos no funcionales que han demandado, sean fielmente plasmados en el producto desarrollado.

En este orden de ideas, para alcanzar una meta se debe dar cumplimiento a los objetivos descritos para esto, adicionalmente a las prácticas asociadas y a las métricas de uso para medir la calidad de la implementación de estas.



**Figura 32 relación Objetivo, práctica y métrica. Elaboración Propia**

A continuación se observará más en detalle esta relación Objetivo, práctica y métrica

- **Objetivo 1: Flujo de trabajo SUMM Product Owner**

Este objetivo lo conforman tres (3) prácticas asociadas las cuales son: SUMM Product Owner disponible para el SUMM Team SUMM, Product Owner

colabora en la eficiencia de las tareas asignadas y el SUMM Product Owner gestiona la toma de decisiones críticas con los Stakeholders.

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

Meta	Objetivo	Practica	Métrica
Gestión de la comunicación con los Stakeholders	Flujo de trabajo SUMM Product Owner	<ul style="list-style-type: none"> <li>SUMM Product Owner disponible para el SUMM Team</li> <li>SUMM Product Owner colabora en la eficiencia de las tareas asignadas</li> <li>SUMM Product Owner gestiona la toma de decisiones críticas con los Stakeholders</li> </ul>	<p>Número de asistencias del Product Owner a las reuniones planeadas</p> <p>Tiempo en tomar decisiones críticas por parte del Product Owner</p>
	Participación en las reuniones de revisión de Sprint	<ul style="list-style-type: none"> <li>Compartir las actividades de team building</li> <li>El SUMM Product Owner y los Stakeholders retroalimentan</li> </ul>	Porcentaje de funcionalidades aceptados

		<p>el proceso</p> <ul style="list-style-type: none"> <li>• El Product Owner de la mano con el SUMM Master dan por finalizado el Sprint</li> </ul>	
--	--	---	--

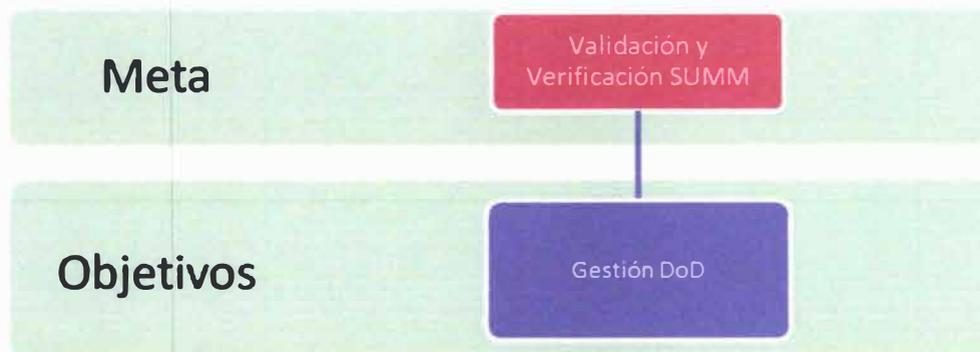
**Tabla 21.** Prácticas y métricas Flujo de trabajo SUMM Product Owner. Gestión de la comunicación con los Stakeholders nivel 3 SUMM .Elaboración Propia

**4.2.1.3.2. Validación y verificación SUMM:** En los procesos de desarrollo de software se evidencia la necesidad de que los Stakeholders y sus representantes validen si el producto intermedio y/o final, dependiendo de cómo se haya descrito en el sprint Backlog, se ajusta con las necesidades implícitas y explícitas planteadas por ellos al SUMM Team.

De acuerdo con lo anterior, estas validaciones se deben realizar en los procesos de revisión de los Sprint ejecutados una vez se finalice este sobre el producto entregable funcional resultado del sprint. Una vez concluido los procesos de validación debe existir la retroalimentación por parte del Stakeholder o en su defecto del SUMM Product Owner. Obtenida la retroalimentación en la reunión retrospectiva el SUMM Team en cabeza del SUMM Master estos deben analizar los resultados para tomar acciones correctivas, en caso de que se requiera.

Por otra parte, los procesos de verificación se orientan a comprobar que el software que se está desarrollando da cumplimiento a su especificación. Se comprueba que el sistema cumple los requerimientos funcionales y no funcionales que se le han especificado impuestos al inicio de cada Sprint, de acuerdo a lo descrito en el Product Backlog.

A continuación, se podrá visualizar la estructura general de la validación SUMM



**Figura 33 Estructura general Validación y Verificación SUMM. Elaboración Propia**

Como se ilustra en la figura anterior, la validación y verificación SUMM, proporciona procesos de evaluación de productos, para conseguirlo se hace necesario la “Definición de Done” (DoD), si el SUMM Team cumple con los criterios definidos se podrá asegurar que se está cumpliendo con las necesidades del Stakeholder, no solo en términos de funcionalidad, sino también de calidad. La DoD consiste en definir una lista de actividades entre las cuales se encuentra: Buenas prácticas de codificación (código limpio), pruebas unitarias, pruebas de integración, documentos de diseño que se agregan valor verificable al producto.

Sin embargo, se puede establecer DoD en varios niveles estos son: DoD para un Story establecido en el Product Backlog, DoD para Sprint y DoD para la liberación del producto, este último determina un estado de potencialmente entregable. Entre las características DoD se pueden destacar que este no es estático, tal y como el marco de trabajo tampoco lo es y que la DoD debe ser una lista de verificación auditable. En este orden de ideas, para alcanzar una meta se debe dar cumplimiento a los objetivos descritos para esto, adicionalmente a las prácticas asociadas y a las métricas de uso para medir la calidad de la implementación de estas.



**Figura 34. Relación Objetivo, práctica y métrica Elaboración Propia**

Ahora, se observará más en detalle esta relación Objetivo, práctica y métrica

- **Objetivo 1: Gestión del DoD**

Este objetivo lo conforman cinco (5) prácticas asociadas las cuales son: Listar DoD por parte del SUMM Team, Definición del DoD por el SUMM Team, Verificar DoD, Verificar DoD para actividades de valor agregado, Product Owner y Stakeholder aceptan los productos entregados intermedios o finales generando un incremento en el Sprint Backlog, una retroalimentación o ajuste al Product Backlog o el cierre satisfactorio o no del proyecto según sea el caso.

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

Meta	Objetivo	Práctica	Métrica
Validación y Verificación SUMM	Gestión del DoD	<ul style="list-style-type: none"> <li>Listar DoD por parte del SUMM Team.</li> <li>Definición del DoD por el SUMM Team.</li> <li>Verificar DoD.</li> <li>Verificar DoD para actividades de valor agregado</li> <li>Product Owner y Stakeholder aceptan los productos entregados</li> </ul>	<p>Porcentaje de pruebas de aceptación fallidos</p> <p>Porcentaje de Stories aceptados</p> <p>Nivel de satisfacción del cliente</p> <p>Porcentaje de consenso del SUMM Team en la definición del DoD</p>

**Tabla 22.** Prácticas y métricas Gestión del DoD. Validación y Verificación nivel 3 SUMM .Elaboración Propia

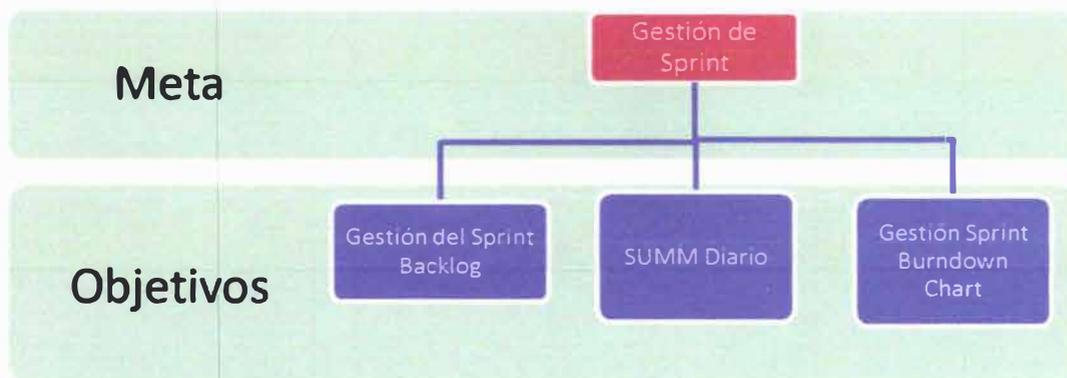
**4.2.1.3.3. Gestión de Sprint:** La gestión de Sprint contribuye a aumentar los niveles de satisfacción del cliente a través de un conjunto de prácticas que deben cumplirse y aplicarse con el fin de administrar los procesos de entrega sus los proyectos desarrollados a tiempo y acorde a su presupuesto.

De igual manera, la gestión de Sprint ayuda al SUMM Team en la monitorización de su progreso, estableciéndose como el indicador que informará tanto al SUMM Master como al SUMM Team sus posibilidades de cumplir los Stories al finalizar el Sprint, respecto a los tiempos proyectados. Por otro lado, la gestión de Sprint también guía al

SUMM Product Owner a analizar el nivel de avance del desarrollo respecto al producto.

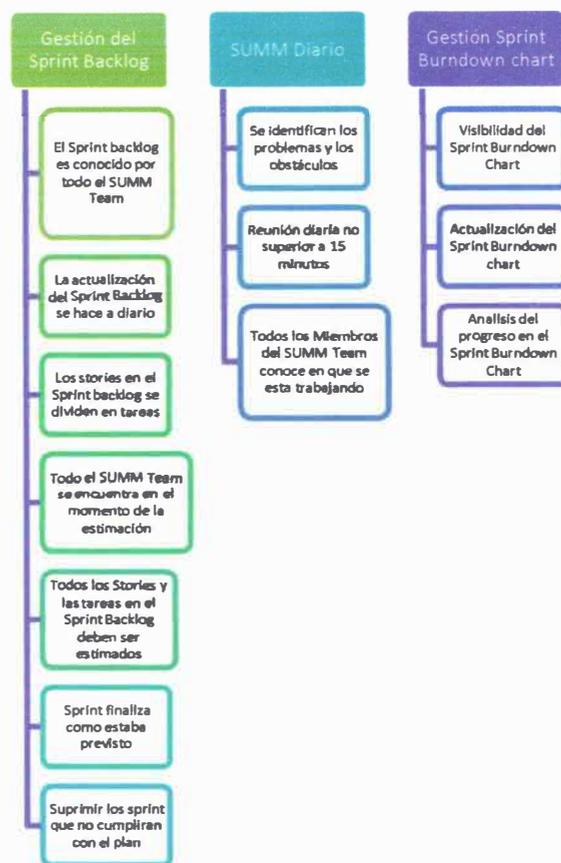
En la Gestión de sprint es importante tener en cuenta que: la estimación de nuevas tareas y la reestimación de tareas que se encuentran en progreso requiere de un esfuerzo considerable, el esfuerzo asignado a las tareas a menudo suele ser impreciso, debido al mismo proceso de estimación, es importante tener en cuenta no estimar el sprint en unidades de tiempo sería mejor hacerlo en esfuerzos, el SUMM Team debe tener presente que la terminación de una tarea o conjunto de ellas, no entrega valor alguno; sólo los Stories completos entregan valor de acuerdo al Product Backlog definido.

A continuación, se podrá visualizar la estructura general de la Gestión de Sprint SUMM



**Figura 35 Estructura general Gestión de Sprint SCMM. Elaboración propia**

Como se ilustra en la figura anterior, la Gestión de Sprint SUMM proporciona procesos para la administración de las iteraciones, para conseguirlo se hace necesaria la Gestión del Sprint Backlog, a través del flujo Diario SUMM y por último es importante establecer la medición de la velocidad del SUMM Team para dar cumplimiento a cada iteración por medio del Sprint burndown chart. En este orden de ideas, para alcanzar una meta se debe dar cumplimiento a los objetivos descritos para esto, adicionalmente a las prácticas asociadas y a las métricas de uso para medir la calidad de la implementación de estas.



**Figura 36 relación Objetivo, práctica y métrica. Elaboración propia**

Ahora, se observará más en detalle esta relación Objetivo, práctica y métrica

- **Objetivo 1: Gestión del Sprint Backlog**

Este objetivo lo conforman siete (7) prácticas asociadas las cuales son: el Sprint backlog es conocido por todo el SUMM Team, la actualización del Sprint Backlog se hace a diario, los stories en el Sprint backlog se dividen en tareas, todo el SUMM Team se encuentra en el momento de la estimación, Todos los Stories y las tareas en el Sprint Backlog deben ser estimados, Sprint finaliza como estaba previsto, Suprimir los sprint que no cumplirán con el plan.

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

Meta	Objetivo	Práctica	Métrica
Gestión de Sprint SUMM	Gestión del Sprint Backlog	<ul style="list-style-type: none"> <li>el Sprint backlog es conocido por todo el SUMM Team</li> <li>la actualización del Sprint Backlog se hace a diario</li> <li>los stories en el Sprint backlog se dividen en tareas</li> <li>todo el SUMM Team se encuentra en el momento de la estimación.</li> <li>Todos los Stories y las tareas en el Sprint Backlog deben ser estimados</li> </ul>	<p>Número de Actualizaciones del Sprint Backlog</p> <p>Tiempo medio estimado para cada Story</p> <p>Tiempo medio estimado por tarea</p> <p>Número de Sprint Finalizados</p> <p>Duración media cada de Sprint</p> <p>Número de participantes en reuniones de estimación</p> <p>Número de Stories estimados</p> <p>Número de tareas estimadas</p> <p>Porcentaje de Sprint suprimidos</p> <p>Tiempo medio de la duración real del Sprint Vs. Duración estimada.</p>

		<ul style="list-style-type: none"> <li>• Sprint finaliza como estaba previsto.</li> <li>• Suprimir sprint que no cumplirán con el plan.</li> </ul>	
--	--	--	--

**Tabla 23.** Prácticas y métricas Gestión del Sprint Backlog. Gestión de Sprint SUMM nivel 3 SUMM .Elaboración Propia

- **Objetivo 2: SUMM Diario**

Este objetivo lo conforman tres (3) prácticas asociadas las cuales son: identificar los problemas y los obstáculos que se presentan en el proceso diario de desarrollo, la reunión diaria no debe ser superior a 15 minutos, todos los miembros del SUMM Team deben conocer en que se está trabajando participando activamente en las reuniones y los planes de estimación

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

Meta	Objetivo	Práctica	Métrica
Gestión de Sprint SUMM	SUMM Diario	<ul style="list-style-type: none"> <li>• Se identifican los problemas y los obstáculos</li> <li>• Reunión diaria no superior a 15 minutos</li> </ul>	<p>Tiempo de duración media del SUMM diario</p> <p>Número de problemas y obstáculos identificados en el SUMM diario</p> <p>Porcentaje de Reuniones</p>

		<ul style="list-style-type: none"> <li>• Todos los Miembros del SUMM Team conoce en que se está trabajando</li> </ul>	<p>diarias realizadas</p> <p>Porcentaje de reuniones diarias canceladas.</p> <p>Número de Participantes en la reuniones diarias</p>
--	--	---	---

**Tabla 24.** Prácticas y métricas Gestión del Sprint Backlog. SUMM Diario SUMM nivel 3 SUMM. Elaboración Propia

- **Objetivo 3: Gestión Sprint Burndown Chart**

Este objetivo lo conforman tres (3) prácticas asociadas las cuales son: mantener visible el Sprint Burndown Chart a todo el SUMM Team Incluyendo el SUMM Product Owner, Actualizar del Sprint Burndown chart, analizar el progreso en el Sprint Burndown Chart esta labor debe ser realizada por el SUMM Master

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

Meta	Objetivo	Práctica	Métrica
Gestión de Sprint SUMM	Gestión del Sprint Burndownchart	<ul style="list-style-type: none"> <li>• Visibilidad del Sprint Burndown Chart</li> <li>• Actualización del Sprint Burndown chart</li> <li>• Análisis del progreso en el</li> </ul>	<ul style="list-style-type: none"> <li>• Media de velocidad de Sprint</li> <li>• Porcentaje de avance de Sprint</li> <li>• Porcentaje de actualización del Sprint Backlog</li> </ul>

		Sprint Burndown Chart	
--	--	--------------------------	--

**Tabla 25.** Prácticas y métricas Gestión del Sprint Backlog. Gestión de Sprint Burndown chart SUMM nivel 3 SUMM. Elaboración Propia

Con la implementación del nivel 3 de SUMM, una organización puede tener éxito en el desarrollo de varios proyectos de software, sin embargo, este éxito es sólo parcial debido a la falta de una gestión estandarizada que garantice la misma calidad y rendimiento en todos los procesos de desarrollo.

#### 4.2.1.4. Nivel Mejorado

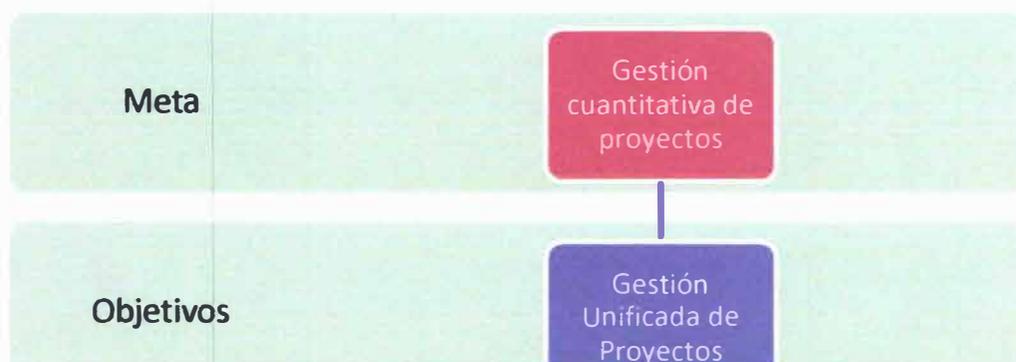
El Nivel Mejorado o también conocido como mejorado cuantitativamente, es el cuarto nivel en la representación de SUMM, en este las organizaciones poseen procesos de desarrollo de software y gestión de proyectos estandarizados apoyados en la gestión de rendimiento de los procesos a través de la medición y el análisis de las prácticas asociadas, esto se debe a que los procesos son entendidos y controlados cuantitativamente, es decir, mediante mediciones detalladas. Este nivel de madurez de SUMM, lo conforman dos (2) metas, las cuales son:

**4.2.1.4.1. Gestión cuantitativa de proyectos:** esta meta conduce a las organizaciones a emplear el mismo proceso de desarrollo en todos los proyectos que ejecute, es decir, se estandariza y se maneja un lenguaje común en la organización respecto a los procesos de desarrollo. Lograrlo no es una tarea sencilla, ya que la organización debe completar un proceso de normalización, si se parte del hecho de que existen elementos comunes a los procesos de desarrollo de proyectos, entre los cuales están, herramientas, metas, objetivos, prácticas y que todos los proyectos tienen como punto de partida la planeación inicial, se pueden realizar estimaciones en pro de evitar errores y optimizar el proceso mediante acciones de mejora continua. Dichas estimaciones se pueden calcular basadas en métricas establecidas por la organización.

La gestión cuantitativa de proyectos aporta a las organizaciones: garantía de calidad en el proceso de desarrollo, una definición y mejora en los proyectos ejecutados, al SUMM Team y al SUMM Master aporta conocimiento mejor control del esfuerzo estimados, el seguimiento y divulgación de problema por medio de las lecciones aprendidas.

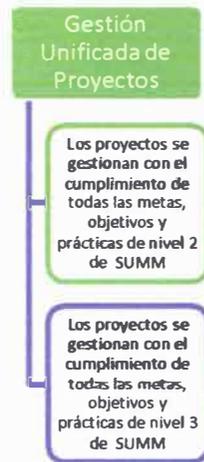
Teniendo en cuenta lo anteriormente mencionado, la gestión cuantitativa puede predecir el grado en el cual el proyecto podría satisfacer sus objetivos de calidad y funcionalidades explícitas e implícitas, teniendo en cuenta información estadística de proyectos anteriores, realizando una especie de “análisis post-mortem”. De igual forma, el desarrollo de proyectos de software abarca varios procesos de la organización, estos procesos deben ser identificados claramente teniendo en cuenta los subprocesos asociados, tener esto claro será de gran ayuda para lograr un control en cuanto a calidad y funcionamiento.

A continuación, se podrá visualizar la estructura general de la Gestión cuantitativa de proyectos nivel 4 de SUMM.



**Figura 37. Gestión cuantitativa de proyectos nivel 4 de SUMM .Elaboración Propia**

La Gestión Cuantitativa de Proyectos tiene asociado un objetivo que dará cumplimiento a las características anteriormente descritas, en este orden de ideas, para alcanzar una meta se debe dar cumplimiento a los objetivos descritos para esto, adicionalmente a las prácticas asociadas y a las métricas de uso para medir la calidad de la implementación de estas.



**Figura 3 relación Objetivo, práctica y métrica. Elaboración Propia**

A continuación se observará más en detalle esta relación Objetivo, práctica y métrica.

- **Objetivo 1: Gestión Unificada de Proyectos**

- Este objetivo lo conforman dos (2) prácticas asociadas, las cuales son: proyectos que se gestionan con el cumplimiento de todas las metas, objetivos y prácticas de nivel 2 de SUMM, de igual manera, sucede con el cumplimiento de todas las metas, objetivos y prácticas de nivel 3 de SUMM. Esto es debido a que se necesita contar con información histórica suficiente de los indicadores del proceso recolectada

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

Meta	Objetivo	Práctica	Métrica
Gestión cuantitativa de proyectos	Gestión Unificada de Proyectos	<ul style="list-style-type: none"> <li>Los proyectos se gestionan con el cumplimiento de todas las metas, objetivos y prácticas de nivel 2 de SUMM</li> <li>Los proyectos se gestionan con el cumplimiento de todas las metas, objetivos y prácticas de nivel 3 de SUMM</li> </ul>	Número de incidencias ocurridas en la Gestión Unificada de Proyectos

**Tabla 26.** Prácticas y métricas Gestión cuantitativa de proyectos. Gestión Unificada de Proyectos nivel 4 SUMM. Elaboración Propia

**4.2.1.4.2. Rendimiento de los procesos organizacionales:** La principal característica de esta meta es establecer la interpretación cuantitativa de los resultados obtenidos a partir de la estandarización de los procesos de la organización asociados a la gestión de proyectos de desarrollo de software con el objeto de obtener un mayor rendimiento de los procesos y proporcionar información para la Gestión Cuantitativa de Proyectos.

Este exige el seguimiento de todas las prácticas sugeridas hasta el nivel cuatro de madurez SUMM. Lo anterior permitirá, proporcionar información suficiente sobre el proceso real y gestionar el rendimiento, determinar si los procesos son predecibles, identificar si los procesos en el rendimiento están dentro de los límites y son coherentes, establecer criterios que permitan determinar las medidas y las técnicas de análisis que emplearan en procesos y subprocesos, identificar los procesos inusuales, identificar los procesos que pueden ser estandarizados.

A continuación, se podrá visualizar la estructura general del Rendimiento de los procesos organizacionales nivel 4 de SUMM.



**Figura 39. Rendimiento de los procesos organizacionales nivel 4 de SUMM. Elaboración Propia**

En este orden de ideas, para alcanzar una meta se debe dar cumplimiento a los objetivos descritos para esto, adicionalmente a las prácticas asociadas y a las métricas de uso para medir la calidad de la implementación de estas.



**Figura 4. Relación Objetivo, práctica y métrica Medición Y análisis SUMM. Elaboración Propia**

A continuación se observará más en detalle esta relación Objetivo, práctica y métrica.

- **Objetivo 1: Medición y Análisis SUMM**

Este objetivo lo conforman dos (2) prácticas asociadas la cual es: las métricas del nivel 2 al nivel 4 son monitoreados y administrados para todos los proyectos, de igual forma, las métricas son sistematizadas a través de Indicadores monitoreados y administrados. En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

Meta	Objetivo	Práctica	Métrica
Rendimiento de los procesos organizacionales	Medición y Análisis SUMM	Las métricas del nivel 2 al nivel 4 son monitoreados y administrados para todos los proyectos  Las métricas son sistematizadas a través de	Número de procesos optimizados  Número de subprocesos optimizados  Porcentaje de métricas sistematizadas  Porcentaje de indicadores

		Indicadores monitoreados y administrados	monitoreados
--	--	---	--------------

**Tabla 27.** Prácticas y métricas Medición y Análisis SUMM. Rendimiento de los procesos organizacionales 4 SUMM. Elaboración Propia

#### **4.2.1.5. Nivel Optimizado**

En el nivel 5 de SUMM se busca una superación continua de la competencia por medio del aumento en el nivel de satisfacción de los Stakeholders e interesados y el SUMM Team. En este nivel, las organizaciones están obligadas a aplicar procesos de mejora continua como eje fundamental de su cultura organizacional. De igual manera, es en este nivel en el cual las organizaciones se brindan espacios para la innovación generando ideas o tecnologías que otorguen ventajas comparativas y competitivas, Aunque en este último aspecto aún no se puede establecer correspondencia solo hay una meta asociada a este nivel

**4.2.1.5.1. Análisis de causas y resolución:** esta meta tiene como pilar principal la identificación de las causas de los resultados escogidos, buscando mejorar el desarrollo de los procesos optimizándolo tanto de nivel de proyectos como a nivel organizacional. Para dar cumplimiento a esta meta se hace necesario un entendimiento cuantitativo del proceso para poder ser efectivas, como la mayoría de metas de nivel 4 y 5.

A continuación, se podrá visualizar la estructura general del Análisis de causas y resolución SUMM nivel 5.

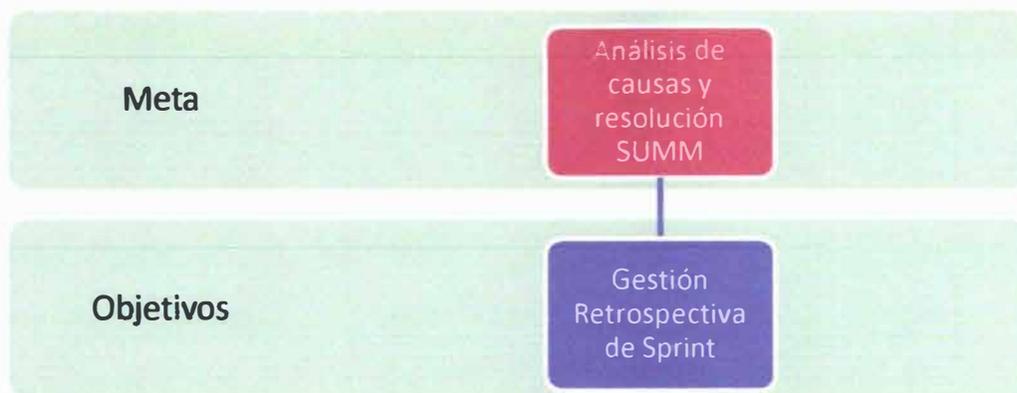


Figura 41. Análisis de causas y resolución SUMM nivel 5. Elaboración Propia

El Análisis de causas y resolución SUMM tiene asociado un objetivo, el cual, dará cumplimiento a las características anteriormente descritas. El objetivo tiene asociado cuatro (4) prácticas tal y como se observa en la siguiente figura.

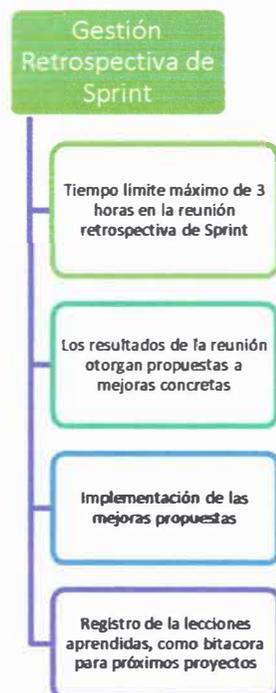


Figura 42. Relación Objetivo, práctica y métrica Gestión Retrospectiva. Elaboración Propia

A continuación se observará más en detalle esta relación Objetivo, práctica y métrica.

- **Objetivo 1: Gestión Retrospectiva de Sprint**

Este objetivo lo conforman cuatro (4) prácticas asociadas, las cuales son: Tiempo límite máximo de 3 horas en la reunión retrospectiva de Sprint, los resultados de la reunión otorgan propuestas a mejoras concretas, la implementación de las mejoras propuestas y registro de la lecciones aprendidas, como bitácora para próximos proyectos.

En la siguiente tabla se establece la relación entre cada práctica y las métricas asociadas:

Meta	Objetivo	Práctica	Métrica
Análisis de causas y resolución SUMM	Gestión Retrospectiva de Sprint	<ul style="list-style-type: none"> <li>• Tiempo límite máximo de 3 horas en la reunión retrospectiva de Sprint</li> <li>• Los resultados de la reunión otorgan propuestas a mejoras concretas</li> <li>• Implementación de las mejoras propuestas</li> <li>• Registro de la lecciones aprendidas, como bitácora para próximos proyectos</li> </ul>	<p>Número de acciones correctivas generadas</p> <p>Número de acciones correctivas implementadas</p> <p>Número de Lecciones aprendidas</p> <p>Duración media de las reuniones retrospectivas</p>

**Tabla 28.** Prácticas y métricas Gestión Retrospectiva de Sprint. Análisis de causas y resolución SUMM nivel 5. Elaboración Propia

## **CAPÍTULO 5: ANOTACIONES FINALES Y TRABAJOS FUTUROS**

### **5.1. Anotaciones Finales**

Este trabajo de investigación nació de la inquietud generada frente al nivel de proyectos de desarrollo de software fallidos, de acuerdo a lo que se observó en el capítulo 2 y 3. A pesar de los múltiples esfuerzos, tendencias, metodologías, modelos, estándares, estrategias las cifras correspondientes a los niveles de fracaso y cancelación de proyectos sigue siendo amplia frente a los proyectos que logran culminar de manera exitosa. El mercado de desarrollo de software cada vez es más extenso y los clientes cada vez más demandan mejores desarrollos de acuerdo a sus necesidades con tecnología de punta, cortos tiempos de desarrollo, costos reducidos, altamente confiables y con calidad.

De acuerdo a lo anterior, se centró la investigación en proponer un modelo que permitiera la ejecución de proyectos de desarrollo de software manteniendo la calidad en los procesos y en los productos, fue así como se inició un proceso de revisión del estado del arte a través del cual se pudo identificar cuales metodologías, modelos o marcos de trabajo de desarrollo de software cumplían los criterios de: definición de prácticas y actividades por etapas o niveles no de manera secuencial sino de forma iterativa e incremental, inmersión del cliente en el proceso de desarrollo, ingeniería de requisitos, desarrollos a mediano y largo plazo. Fue de esta manera como después de realizar una comparación de las metodologías, modelos y marcos de trabajo encontrados se resolvió que Scrum cumplía la mayoría de los criterios establecidos, de igual forma al revisar estadísticas de uso de metodología, Scrum resultó ser de las más usadas o al menos una de las más conocidas en la actualidad dentro del enfoque ágil.

De igual forma, gracias a la revisión del proceso de desarrollo de Scrum se vislumbró algunas falencias en cuanto al empleo de buenas prácticas orientadas a la calidad del proceso propias del marco de trabajo, fue de esta forma, como se decidió incorporar las buenas prácticas de CMMI – DEV en su más reciente versión en el modelo propuesto.

Una vez definidos los componentes base pilares del modelo, es decir, proceso de desarrollo ágil basado en Scrum y buenas prácticas de calidad basadas en CMMI – DEV 1.3. Se procedió a realizar una correspondencia entre las prácticas genéricas y específicas de las áreas de proceso de CMMI – DEV y las metas, artefactos y Objetivos del proceso de Scrum. Dicha correspondencia, sirvió de base para poder establecer los procesos en los cuales se debía hacer énfasis en el modelo a proponer.

El modelo propuesto se compone de cinco (5) niveles de madurez enfocados en la representación por etapas de CMMI, estos niveles en su orden son: Inicial, Gestionado, Definido, Mejorado, Optimizado; cada nivel lo componen metas, a su vez las metas se componen de Objetivos y asociados a ellos están las prácticas y los indicadores. La principal dificultad a la hora de proponer el modelo se relacionó al hecho de no cargar tanto el modelo propuesto, sacrificando la agilidad de Scrum, ni tampoco ir en contravía de los valores y principios que lo componen.

A pesar de que en la correspondencia se encontró que en los niveles superiores de CMMI había carencia de prácticas de Scrum asociadas, se logró definir metas, objetivos, prácticas e inclusive indicadores para estos niveles, procurando siempre no sacrificar la agilidad. Respecto a los procesos de Validación y Verificación en los cuales tampoco había aplicación aparente y clara de prácticas de Scrum se logró definir prácticas que modelan ambos procesos, el empleo del artefacto DoD se convirtió en pieza clave para esta labor.

## **5.2. Trabajos Futuros**

A pesar de que se realizó un esfuerzo para llevar el proceso de Scrum a los niveles 4 y 5 de CMMI, aún existen áreas de proceso que podrían emplearse y que de hecho son necesarias, sin sacrificar la agilidad, estas áreas serían la gestión de la configuración, la gestión de riesgos y la innovación e implementación organizacionales.

La implementación del modelo propuesto en al menos 5 organizaciones piloto que retroalimenten el modelo a partir del día a día de las empresas de desarrollo de software en todos los niveles.

Establecer un método de Valoración y evaluación (*Appraisal*) a las organizaciones que permita establecer el nivel de madurez en que se encuentran y como deben incorporar los procesos descritos en el modelo propuesto.

Construcción de mapa de procesos completo de SUMM, identificando cómo implementarla

Construcción de un documento guía donde se presenten elementos adicionales como formatos, cálculo de indicadores y procedimientos

## REFERENCIAS

- [1] Salinas, P. J., Pérez, M. 1993. *Iniciación práctica a la investigación científica*. 2ª ed. Consejo de Publicaciones de la Universidad de Los Andes. Mérida. Venezuela.
- [2][3][39][40] Why are small software organizations different? Richardson, Ita and Gresse, Christiane. IEEE Computer society, Febrero 2007, IEEE Software, Vol. 24. ISSN 0740-7459.
- [10] Dave ,W. *Water-Scrum-Fall Is The Reality Of Agile For MostOrganizations Today*. Julio 2011.[http://www.versionone.com/state\\_of\\_agile\\_development\\_survey/11/](http://www.versionone.com/state_of_agile_development_survey/11/)
- [5] Salo, O., Abrahamsson, P.2008. Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. *Software*, págs. 58–64.
- [6] [8] [9] Chow, T. a. B. 2009. A survey study of critical success factors in agile software projects. *J. Syst. Softw.* , 81 (6), 961-971.
- [7] Turner, R., Jain, A. 2002. Agile Meets CMMI: Culture Clash or Common Cause? En *Extreme Programming and Agile Methods — XP/Agile Universe 2002* págs. 153–165.
- [10] Pressman, Roger S. 2000. *Software Engineering*. [book auth.] Thayer Richard. *SoftwareEngineering Project Management*. Segunda edición. s.l. : IEEE Computer Society, , pp. 30-46.
- [11] Canós, J.H., Letelier, P., Penadés, M.C. 2003, "Metodología Ágiles en el Desarrollo Software". VIII Jornadas de Ingeniería de Software y Bases de Datos.
- [12] [13] Sommerville, Ian. *Ingeniería del software*. Séptima Edición. Madrid : Pearson Educación S.A., 2006. ISBN 13: 978-84-7829-074-1.
- [14] [19] [21] Campo, Luis F. *Marco Metodológico para la Administración de Proyectos de Software utilizando las Mejores Prácticas Propuestas por el Modelo CMMIDEV Versión 1.2*. Departamento de Sistemas Informáticos y Computación, Universidad del Norte. Barranquilla, 2008. Tesis Maestría.
- [15] IEEE Standard Glossary of Software Engineering Terminology," IEEE Std 610.12-1990 ,vol.1,no.22,pp.1,1990 doi: 10.1109/IEEESTD.1990.101064.
- [16] [17] Pedroza, Patty. *Modelo para la gestión de la calidad de servicios ti Aplicable a las pymes de la ciudad de Barranquilla*. Ingeniería del software, Universidad del Norte. Barranquilla. Tesis Maestría.
- [18] [20] [21] Software Engineering Institute. *CMMI for Development*. Pittsburg : s.n., 2006. CMU/SEI-2006-TR-008.
- [22] [26] [27] CMMI Guía para la integración de procesos y la mejora de productos. Segunda edición. Pearson Educación. Cátedra de Mejora de Procesos de Software en el Espacio Iberoamericano de la Universidad Politécnica de Madrid 2009. ISBN: 9788478290963.

- [23] <http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2010SepCMMI.pdf> - 2010-10-01.
- [28] [29] Beck, K. (2001), Manifesto for Agile Software Development, <http://agilemanifesto.org/>
- [30] Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., y Otros (2001). Manifesto for Agile Software Development
- [31] Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J. 2002. Agile Software Development Methods: Review and Analysis. VTT Publications 478.
- [32] Beck, K. 2000. Extreme Programming Explained: Embrace Change. Addison- Wesley.
- [33] Beck, K. 2002. Test Driven Development: By Example. Addison Wesley.
- [34] Schwaber, K. 2007. The Enterprise and SCRUM. Microsoft Press, Washington
- [35][36][37] Schwaber, K. SCRUM Development Process. Advanced Development Methods 131 Middlesex Turnpike Burlington, MA 01803.
- [38][39] CMMI for Development, Version 1.3. Improving processes for developing better products and services. Noviembre 2010. CMU/SEI-2010-TR-033 ESC-TR-2010-033. <http://www.sei.cmu.edu>
- [40][41][42][43][44] CMMI for SCAMPISM Class A Appraisal Results 2012 Mid-Year Update. Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213. Marzo 2012. <http://www.sei.cmu.edu/cmmi/results.html>
- [45][46] Why are small software organizations different? Richardson, Ita and Gresse, Christiane. IEEE Computer society, Febrero 2007, IEEE Software, Vol. 24. ISSN 0740-7459.
- [47] Gallardo, Sara M. El TLC bajo la lente de Fedesoft. 91 Sistemas. [Revista]. 2005. <http://www.acis.org.co/index.php?id=388>. ISSN 0120-5919.
- [48] EIA 731 SECM is the Electronic Industries Alliance standard 731, or the Systems Engineering Capability Model. INCOSE SECAM is International Council on Systems Engineering Systems Engineering Capability Assessment Model [EIA 2002a].
- [49] Paula, G., Parada, D. 2008. Agilidad y disciplina del proceso de desarrollo de software para las Pequeñas y Medianas Empresas (PYMES) y las Cooperativas en Latinoamérica. Caso: Venezuela. VII Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento.
- [50][51][52][53][54] Aprendizaje y aplicación del CMMI-DEV en PYMES de software Colombianas. La experiencia RCCS. ESTRADA Y, LILIA. Gerenc. Tecnol. Inform, Vol. 9 , N° 24 , Abril 2010. pp 57 – 76.
- [55] Beck, K. 2001, Manifesto for Agile Software Development, <http://agilemanifesto.org/>
- [56] Berrocal, J., García-Alonso, J., Murillo, J. 2010 Agilizando las herramientas de gestión de proyectos. Actas de 1ª Conferencia Agile-Spain CAS 2010.
- [57] Mendes Calo, K., Estevez, E., Fillotrani, P. 2008. "Un Framework para Evaluación de Metodologías Ágiles"

- [58] Casallas, R., Arboleda, H. 2010. QualDev Process: Procesos adaptables de desarrollo de software para proyectos ágiles.
- [59] Salo, O., & Abrahamsson, P. 2008. Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. *Software*, págs. 58–64.
- [60] Turner, R., & Jain, A. (2002). Agile Meets CMMI: Culture Clash or Common Cause? En *Extreme Programming and Agile Methods — XP/Agile Universe 2002* (págs. 153–165).
- [61] Shelton, C. CMMI or Agile: Why Not Embrace Both!. Julio 09-2008 pp 1.
- [62] CMMI for Development, Version 1.3. Improving processes for developing better products and services. Noviembre 2010. CMU/SEI-2010-TR-033 ESC-TR-2010-033. <http://www.sei.cmu.edu>
- [63] [64] [65] [66] Resultados de la I Encuesta de GP de TI en Colombia. V Jornada de Gerencia de Proyectos de TI. [http://www.acis.org.co/fileadmin/Base\\_de\\_Conocimiento/V\\_Jornada\\_Gerencia/V\\_Encuesta\\_Gerencia\\_Proyectos\\_TI.pdf](http://www.acis.org.co/fileadmin/Base_de_Conocimiento/V_Jornada_Gerencia/V_Encuesta_Gerencia_Proyectos_TI.pdf)
- [67] [68] [69] [70] [71] [72] Resultados de la I Encuesta de GP de TI en Colombia. VI Jornada de Gerencia de Proyectos de TI. [http://www.acis.org.co/fileadmin/Base\\_de\\_Conocimiento/VI\\_Jornada\\_Gerencia/VJGPTIAcuto.pdf](http://www.acis.org.co/fileadmin/Base_de_Conocimiento/VI_Jornada_Gerencia/VJGPTIAcuto.pdf)
- [73] [74] Chow, T. a.-B. 2009. A survey study of critical success factors in agile software projects. *J. Syst. Softw.*, 81 (6), 961-971.
- [75] [76] [77] [78] [79] Dave, W. Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today. Julio 2011. [http://www.versionone.com/state\\_of\\_agile\\_development\\_survey/11/](http://www.versionone.com/state_of_agile_development_survey/11/)
- [81] Turner, R., Jain, A. 2002. Agile Meets CMMI: Culture Clash or Common Cause? En *Extreme Programming and Agile Methods — XP/Agile Universe 2002* (págs. 153–165).
- [82] Díaz, Y. 2009. Estudio sobre la correspondencia entre las prácticas CMMI y prácticas ágiles y su aplicación en Pymes.
- [83] [84] [85] [86] CMMI Guía para la integración de procesos y la mejora de productos. Segunda edición. Pearson Educación. <http://www.sei.cmu.edu/cmmi>