

EXPERIENCIA DE MEJORA DEL PROCESO DE REQUISITOS APLICANDO
INVESTIGACIÓN ACCIÓN Y CASOS DE MEJORA

FRANK STIVEN CORREA AGUDELO

Trabajo de grado presentado como requisito para optar el título de Maestría

Asesor: RAQUEL ANAYA

MEDELLIN
UNIVERSIDAD EAFIT
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS
ESCUELA DE INGENIERÍA
2013

Agradecimientos

Agradezco a Dios, por su gran sabiduría y por acompañarme en este hermoso camino de la vida.

Agradezco a mis padres, por su amor transparente e incondicional.

Agradezco a mis hermanos, por el apoyo que me han brindado en todas las decisiones que he tomado.

Agradezco a mi esposa, por la paciencia y comprensión durante estos años.

Agradezco a mi asesora Raquel Anaya, por haber emprendido este camino conmigo, por convertirse en un gran apoyo para el éxito de mi carrera profesional, y sobre todo, por haber compartido su conocimiento sin restricciones.

Agradezco a la Universidad EAFIT y sus profesores, por haberme dado la oportunidad de haber estudiado en esa Universidad tan prestigiosa, por haber contribuido en mi formación académica y personal.

Agradezco a la Organización en estudio y a todo su equipo de trabajo, por haberme dado la oportunidad de implementar un programa de mejora que contribuye positivamente en mi formación académica.

Índice

1. INTRODUCCIÓN.....	15
1.1. RESUMEN	15
1.2. MOTIVACIÓN	16
1.3. OBJETIVOS	18
1.3.1. OBJETIVO GENERAL	18
1.3.2. OBJETIVOS ESPECÍFICOS	19
1.4. PREGUNTAS DE INVESTIGACIÓN	19
1.5. ORGANIZACIÓN	19
2. MARCO TEÓRICO.....	21
2.1. PROGRAMA DE MEJORA	21
2.2. MODELOS DE PROCESOS.....	21
2.2.1. CMMI-DEV	22
2.2.2. COMPETISOFT	23
2.2.3. MÉTRICA V3	24
2.2.4. MPS.BR	25
2.3. MODELOS DE EVALUACIÓN DE PROCESOS.....	26
2.3.1. SCAMPI	26
2.3.2. ISO/IEC 15504	27
2.4. MODELO DE MEJORA DE PROCESOS	29
2.4.1. MODELO IDEAL.....	29
2.4.2. ÁGIL SPI	30
2.4.3. PMCOMPETISOFT.....	31
2.5. FACTORES DE ÉXITO	34
2.6. CASOS DE MEJORA	36
2.7. EXPERIMENTACIÓN EN INGENIERÍA DE SOFTWARE.....	37
2.8. INVESTIGACIÓN – ACCIÓN	38
2.9. OBSERVACIÓN	40
3. CONTEXTUALIZACIÓN	42
3.1. CONTEXTO DE LA ORGANIZACIÓN	42
3.2. PROCESO SOFTWARE ACTUAL	44
3.3. PROBLEMÁTICAS.....	47
3.4. RESTRICCIONES ORGANIZACIONALES.....	49
3.4.1. RESTRICCIONES DE TIEMPO.....	49

3.4.2. RESTRICCIONES DE HORARIO	49
3.4.3. RESTRICCIONES DE PERSONAL.....	49
3.4.4. RESTRICCIONES DE SEGURIDAD LOCAL.....	50
3.4.5. RESTRICCIONES DE RECURSOS	50
3.5. LIMITACIONES	50
<u>4. DEFINICIÓN DEL MARCO DE TRABAJO PARA LA MEJORA.....</u>	<u>52</u>
4.1. MODELO DE REFERENCIA	52
4.2. MODELO DE EVALUACIÓN.....	54
4.3. MODELO DE MEJORA.....	58
4.3.1. GESTIÓN DE LA CONFIGURACIÓN.....	63
4.3.1.1. Gestión de cambios.....	64
4.3.2. PLAN DE COMUNICACIÓN	66
4.3.3. PLAN DE CALIDAD.....	67
<u>5. EXPERIENCIA DE MEJORA DEL PROCESO DE REQUISITOS.....</u>	<u>68</u>
5.1. FASE DE INICIO.....	68
5.2. FASE DE DIAGNÓSTICO.....	70
5.2.1. EVALUACIÓN DESDE LA PERSPECTIVA DEL USUARIO	71
5.2.2. EVALUACIÓN DESDE LA PERSPECTIVA DE LOS ANALISTAS DE DESARROLLO	76
5.2.3. RECOMENDACIONES.....	80
5.2.3.1. Recomendaciones de conocimiento	80
5.2.3.2. Recomendaciones de prácticas	81
5.3. FASE DE ESTABLECIMIENTO.....	82
5.4. FASE DE ACCIÓN.....	87
5.5. FASE DE APRENDIZAJE.....	91
<u>6. ANÁLISIS DE LOS RESULTADOS DEL PROGRAMA DE MEJORA</u>	<u>92</u>
6.1. ANÁLISIS DE LOS RESULTADOS DE MEJORA	92
6.2. ANÁLISIS DE LAS METAS DEL PROGRAMA DE MEJORA.....	94
6.3. ANÁLISIS FINANCIERO	95
6.4. INCONVENIENTES PRESENTADOS DURANTE EL PROGRAMA DE MEJORA	97
<u>7. CONCLUSIONES, LECCIONES APRENDIDAS Y TRABAJOS FUTUROS</u>	<u>100</u>
7.1. CONCLUSIONES.....	100
7.2. LECCIONES APRENDIDAS.....	104

7.3. TRABAJO FUTURO	105
<u>8. BIBLIOGRAFÍA.....</u>	<u>107</u>
<u>9. ANEXOS</u>	<u>114</u>
9.1. ANEXO 1.....	114
9.2. ANEXO 2.....	117

Índice de Figuras

Figura 1. Modelos de un programa de mejora (Construcción propia)	21
Figura 2. Estructura de las representaciones continua y por etapas [28].	22
Figura 3. Modelo Competisoft [29].	24
Figura 4. Modelo Métrica V 3 [31].	25
Figura 5. Modelo MPS.BR [32].	26
Figura 6. Composición ISO/IEC 15504 (Adaptación de [29])	28
Figura 7. Investigación – Acción (Adaptación de [47]).....	39
Figura 8. Modelo del Proceso Software (Construcción propia).	45
Figura 9. Componentes del programa de mejora (Construcción propia).....	52
Figura 10. Ciclo de vida de mejora (Construcción propia).....	59
Figura 11. Esquema de trabajo (Adaptación de [10]).	61
Figura 12. Significado de un requisito y su objetivo (Construcción propia).	71
Figura 13. Conocimiento de información (Construcción propia).....	72
Figura 14. Conocimiento de los resultados (Construcción propia).	72
Figura 15. Involucran a otras áreas de negocio (Construcción propia).	72
Figura 16. Conocimiento de Procesos (Construcción propia).	73
Figura 17. Conocimiento de Políticas (Construcción propia).....	73
Figura 18. Impacto de un mal requisito (Construcción propia).	73
Figura 19. Conocimiento del impacto y recursos (Construcción propia).....	73
Figura 20. Establecen validaciones (Construcción propia).....	74
Figura 21. Priorizan los requisitos (Construcción propia).	74
Figura 22. Responsable de los de requisitos (Construcción propia).	74
Figura 23. Importancia del Analista de Desarrollo (Construcción propia).....	75
Figura 24. Importancia del Analista de Procesos (Construcción propia).	75
Figura 25. Mejorar el proceso de requisitos (Construcción propia).	75
Figura 26. Cumplimientos de las prácticas de RD (Construcción propia).....	77
Figura 27. Cumplimiento de metas específicas de RD (Construcción propia).....	77
Figura 28. Nivel de adopción de prácticas de RD (Construcción propia).	78
Figura 29. Cumplimientos de las prácticas de REQM (Construcción propia).....	78
Figura 30. Cumplimiento de metas específicas de REQM (Construcción propia). ..	79
Figura 31. Saber vs Hacer [36].....	80
Figura 32. Analogía Casos de mejora y SCRUM (Construcción propia).	83
Figura 33. Ciclo de vida de Investigación – Acción (Adaptación de [47]).	89
Figura 34. Ejecución del plan de mejora (Construcción propia).	90
Figura 35. Nivel de adopción de metas específicas RD (Construcción propia).	92
Figura 36. Nivel de adopción de metas específicas REQM (Construcción propia) 93	

Figura 37. Estado actual de RD y REQM (Construcción propia).....	93
Figura 38. Índice de satisfacción del usuario (Construcción propia).	95

Índice de Tablas

Tabla 1. Tipo de representación de CMMI [28].	23
Tabla 2. Tipos de factores (Adaptación de [43]).	34
Tabla 3. Factores identificados en programas de mejora según [43].	36
Tabla 4. Actividades del Proceso Software (Construcción Propia).	47
Tabla 5. Áreas de proceso e instrumento de evaluación (Construcción propia). ...	58
Tabla 6. Estructura de un caso de mejora (Construcción propia).	60
Tabla 7. Estrategia de intervención del caso de estudio (Construcción propia). ...	63
Tabla 8. Ítems de configuración (Construcción propia).	64
Tabla 9. Plan de comunicación (Construcción propia).	67
Tabla 10. Asignación de responsabilidades (Construcción propia).	70
Tabla 11. Información de los encuestados (Construcción propia).	76
Tabla 12. Prácticas específicas menos adoptadas (Construcción propia).	79
Tabla 13. Relación casos de mejora y recomendaciones (Construcción propia). .	84
Tabla 14. Criterios de priorización (Construcción propia).	84
Tabla 15. Comparación casos de mejora por Importancia (Construcción propia). 85	
Tabla 16. Prioridad de casos de mejora por Importancia (Construcción propia). ..	85
Tabla 17. Comparación de casos de mejora por esfuerzo (Construcción propia). 85	
Tabla 18. Prioridad de casos de mejora por esfuerzo (Construcción propia).	86
Tabla 19. Comparación de casos de mejora por riesgo (Construcción propia).	86
Tabla 20. Prioridad de casos de mejora por riesgo (Construcción propia).	86
Tabla 21. Prioridad relativa por caso de mejora (Construcción propia).	87
Tabla 22. Redefiniciones por caso de mejora (Construcción propia).	88
Tabla 23. Distribución de horas por actividades (Construcción propia).	94
Tabla 24. Costos económicos por mes (Construcción propia).	96
Tabla 25. Beneficios económicos por mes (Construcción propia).	96
Tabla 26. Nivel de incidencia de los factores de éxito (Construcción propia).	104
Tabla 27. Prácticas específicas menos adaptadas (Construcción propia).	105

Lista de Abreviaturas

CASE	Computer Aided Software Engineering.
CM	Caso de Mejora.
CMMI	Capability Maturity Model Integration.
CMMI-DEV	Capability Maturity Model Integration for Development.
IAPM	Ideal Agile Process Model.
IDEAL	Initiating, Diagnosing, Establishing, Acting & Learning.
IEEE	Institute of Electrical and Electronics Engineers.
IR	Ingeniería de Requisitos.
ISO	International Organization for Standardization.
GQM	Goal/Question/Metric.
MPS.BR	Mejora del Proceso de Software Brasileiro.
PMBOK	Project Management Body of Knowledge.
PSP	Personal Software Process.
PyMEs	Pequeña y mediana empresa.
REGPG	Requirements Engineering Good Practice Guide.
SCAMPI	Standard CMMI Appraisal Method for Process Improvement.
SEI	Software Engineering Institute.
SG	Specific Goal.
SP	Specific Practice.
SPI	Software Process Improvement.
SPICE	Software Process Improvement and Capability Determination.
SPIMS	Software Process Improvement Model for Small Scale Industries.

Glosario

Acuerdo: Decisión tomada en común por varias personas.

Ambigüedad: Expresión que tiene diferentes interpretaciones debido a su condición confusa o imprecisa.

Análisis de requisitos: La determinación del rendimiento y de las características funcionales específicas del producto, basándose en el análisis de las necesidades, expectativas y restricciones del cliente; en el concepto operativo; en los entornos de uso proyectados para las personas, los productos y los procesos; y en las medidas de eficacia.

Análisis de riesgos: La evaluación, clasificación y priorización de los riesgos.

Análisis funcional: Examen de una función definida para identificar todas las subfunciones necesarias para realizarla; identificación de las relaciones funcionales e interfaces (internas y externas) y captura de éstas en una arquitectura funcional; y transferir los requerimientos de rendimiento de mayor nivel y asignación de estos requerimientos a subfunciones de menor nivel.

Arquitectura del proceso: Las relaciones de orden, las interfaces, las interdependencias y otras relaciones entre los elementos de proceso de un proceso estándar. La arquitectura de proceso también describe las interfaces, las interdependencias y otras relaciones entre los elementos de proceso y los procesos externos (p. ej., la gestión del contrato).

Arquitectura funcional: La organización jerárquica de las funciones, de sus interfaces funcionales internos y externos (externos a la propia agregación) e interfaces físicos externos, de sus respectivos requerimientos funcionales y de rendimiento, y de sus restricciones de diseño.

Cambio: Modificación de una cosa para convertirla en algo distinto o alteración de un estado.

Calidad: La capacidad de un conjunto de características inherentes de un producto, componente de producto o proceso para satisfacer los requerimientos de los clientes.

Ciclo de vida del producto: El periodo de tiempo, consistente en fases, que empieza cuando se concibe un producto y termina cuando el producto ya no está disponible para su uso. Dado que una organización puede estar produciendo múltiples productos para múltiples clientes, una descripción de un ciclo de vida del

producto puede no ser adecuada. Por tanto, la organización puede definir un conjunto de modelos aprobados de ciclo de vida del producto. Estos modelos se encuentran normalmente en la literatura publicada y es probable que se adapten para uso en una organización. Un ciclo de vida del producto podría constar de las siguientes fases: (1) concepto/visión (2) viabilidad (3) diseño/desarrollo (4) producción y (5) retirada.

Componente de producto: En el Conjunto de productos CMMI, un producto de trabajo que es un componente de bajo nivel del producto. Los componentes de producto se integran para producir el producto. Pueden existir varios niveles de componentes de producto.

Conjunto Operativo: Agrupaciones que funcionan o son válidos para algo, que permite obtener los resultados.

Diagrama de Actividades: Representan los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema.

Especificar: Determinación de los datos o detalles necesarios sobre una persona o una cosa para diferenciarlas con claridad de otra.

Elemento de configuración: Una agregación de productos de trabajo que se establece para la gestión de configuración y se trata como una entidad única en el proceso de gestión de configuración.

Elemento de proceso: La unidad fundamental de un proceso. Se puede definir un proceso en términos de subprocesos o elementos de proceso. Un subproceso puede descomponerse adicionalmente en subprocesos o elementos de proceso; un elemento de proceso no puede dividirse.

Cada elemento de proceso cubre un conjunto de actividades estrechamente relacionadas (p. ej., elemento de estimación o elemento de revisión entre pares). Se pueden describir los elementos de proceso usando plantillas que deben completarse, abstracciones que deben refinarse o descripciones que deben modificarse o usarse. Un elemento de proceso puede ser una actividad o una tarea.

Escenario operativo: Una descripción de una secuencia ficticia de eventos que incluye la interacción del producto con su entorno y usuarios, así como la interacción entre sus componentes de producto. Los escenarios operacionales se usan para evaluar los requerimientos y el diseño del sistema, y para verificar y validar el sistema.

Especificación de Requisitos: La especificación de requisitos de software es la actividad en la cual se genera el documento, con el mismo nombre, que contiene

una descripción completa de las necesidades y funcionalidades del sistema que será desarrollado; describe el alcance del sistema y la forma en como hará sus funciones, definiendo los requisitos funcionales y los no funcionales.

Estándar: Cuando vea la palabra estándar usada como un nombre en un modelo CMMI, se refiere a los requerimientos formales obligatorios, desarrollados y usados para prescribir aproximaciones coherentes al desarrollo (p. ej., los estándares ISO/IEC, los estándares IEEE y los estándares de la organización). En lugar de usar estándar en su sentido común diario, se usa otro término que significa la misma cosa (p. ej., típico, tradicional, usual o rutinario).

Evaluación: En el Conjunto de productos CMMI, un examen de uno o más procesos por un equipo de profesionales formados, usando un modelo de evaluación de referencia como base para determinar, como mínimo, fortalezas y debilidades .

Impacto: Conjunto de consecuencias provocadas por un hecho o actuación que afecta a un entorno o los resultados esperados.

Identificación de la configuración: Un elemento de la gestión de configuración que consiste en seleccionar los elementos de configuración para un producto, asignándole identificadores únicos y registrando sus características funcionales y físicas en la documentación técnica.

Identificación de riesgos: Una aproximación organizada y rigurosa para buscar riesgos probables o realistas en la consecución de los objetivos.

Informe del estado de configuración: Un elemento de la gestión de configuración que consiste en el registro y publicación de la información necesaria para gestionar eficazmente una configuración. Esta información incluye una lista de la identificación de la configuración aprobada, el estado de los cambios propuestos a la configuración y el estado de implementación de los cambios aprobados.

Integración: Incorporación y adaptación para lograr la funcionalidad general.

Interacción: Acción, relación o influencia recíproca entre dos o más personas o cosas.

Línea base: Un conjunto de productos de trabajo o especificaciones que se ha revisado y acordado formalmente, el cual sirve más tarde como la base para desarrollo posterior, y que solamente puede cambiarse mediante procedimientos de control de cambios.

Línea base de configuración: La información de configuración formalmente establecida en un momento dado durante la vida de un producto o de un componente de producto. Las líneas base de configuración, más los cambios aprobados a esas líneas base, constituyen la información de configuración actual (Véase también “ciclo de vida del producto”).

Línea base de producto: En gestión de configuración, el paquete inicial de datos técnicos aprobado (incluyendo, para el software, el listado de código fuente) que define un elemento de configuración durante la producción, operación, mantenimiento y soporte logístico de su ciclo de vida.

Línea base de rendimiento de proceso: Una caracterización documentada de los resultados reales alcanzados al seguir un proceso, que se usa como punto de referencia para comparar el rendimiento real del proceso frente al rendimiento esperado del mismo.

Línea de producto: Un grupo de productos que comparten un conjunto de características comunes y gestionadas que satisfacen las necesidades específicas de un mercado o de una misión seleccionada.

Plan de acción: Establece el conjunto de actividades, entregables, responsables, plazos, hitos de control, métodos de seguimiento que se debe hacer para cumplir con los objetivos del caso de mejora.

Plan de mejora: Establece el conjunto de actividades, entregables, responsables, plazos, hitos de control, métodos de seguimiento que se debe hacer para cumplir con los objetivos del programa de mejora. El plan de mejora está conformado por planes de acción.

Proceso: En el Conjunto de productos CMMI, las actividades que pueden reconocerse como implementaciones de las prácticas en un modelo CMMI. Estas actividades pueden corresponderse con una o más prácticas en las áreas de proceso de CMMI para permitir que un modelo sea útil para la mejora de procesos y la evaluación de procesos.

Hay un uso especial de la frase “el proceso” en las declaraciones y descripciones de las metas genéricas y las prácticas genéricas. “El proceso”, tal y como se usa en la Parte Dos, es el proceso o procesos que implementan el área de proceso.

Producto: En el Conjunto de productos CMMI, un producto de trabajo que está previsto entregar a un cliente o usuario final. La forma de un producto puede variar según el contexto. (Véase también “cliente”, “componente de producto”, “servicio” y “producto de trabajo”).

Reclamación: Protesta u oposición que se realiza sobre algo que se considera injusto o insatisfactorio. Reglas de Negocio describe las políticas, normas, operaciones, definiciones y restricciones presentes en una organización y que son de vital importancia para alcanzar los objetivos misionales.

Representantes de usuarios: Dueños de proceso, analista líder, subgerente de desarrollo, jefe de área y/o administrador del sistema.

Requisito de Software: (1) Propiedad o restricción, determinada con precisión, que un producto software debe satisfacer. (2) Expresa las necesidades y las limitaciones impuestas a un producto de software que contribuyen a la solución de algún problema en el mundo real.

Requisito: (1) Una condición o capacidad necesitada por un usuario para solucionar un problema o lograr un objetivo. (2) Una condición o capacidad que debe cumplir o poseer un producto o componente de producto para satisfacer un contrato, un estándar, una especificación u otros documentos impuestos formalmente. (3) Una representación documentada de una condición o capacidad como en (1) o en (2).

Servicio: En el conjunto de productos CMMI, un servicio es un producto que es intangible y no almacenable.

Solicitud: (1) Pedir algo, haciendo uso de un procedimiento formal. (2) Es una diligencia cuidadosa o un pedido.

Trazabilidad: Una asociación discernible entre dos o más entidades lógicas, tales como requerimientos, elementos de sistema, verificaciones o tareas.

Trazabilidad bidireccional: Una asociación entre dos o más entidades lógicas que es discernible en ambos sentidos (es decir, hacia y desde una entidad).

Validación: Confirmación de que el producto, tal y como se ha proporcionado (o será proporcionado), satisfará su uso previsto. En otras palabras, la validación asegura que “se ha construido el producto correcto”.

Verificación: Confirmación de que los productos de trabajo reflejan apropiadamente los requisitos que se han especificado para ellos. En otras palabras, la verificación asegura que “se construyó correctamente el producto”.

1. Introducción

En este capítulo se presentan el resumen de la experiencia de mejora del proceso de requisitos y las ideas que motivaron a compartirla, además, se indican los objetivos trazados y las preguntas que se desean resolver por medio de la experiencia obtenida durante la ejecución del programa de mejora.

1.1. Resumen

Este trabajo de maestría consiste en una experiencia de mejora del proceso software enfocado en el proceso de requisitos dentro de una Organización de servicios. Este documento contiene la estructura del marco de trabajo del programa de mejora, junto con la experiencia obtenida en la ejecución del programa de mejora, lo cual facilita la comprensión de los objetivos, preguntas de investigación y resultados obtenidos.

El objetivo principal fue mejorar el proceso de requisitos haciendo uso de un modelo de referencia, de evaluación y de mejora; donde el modelo de evaluación fue propuesto por el autor del presente trabajo, el cual evalúa el proceso de requisitos establecido por la Organización desde dos perspectivas: primero, desde la perspectiva de los usuarios, se evalúa y analiza el nivel de conocimiento y conciencia que se tiene sobre el proceso requisitos, y segundo, desde la perspectiva del Equipo de desarrollo, donde se evalúa y analiza el nivel de adopción de buenas prácticas recomendadas por el CMMI-DEV 1.3 para el desarrollo y gestión de requisitos. Para recopilar la información se diseñó y aplicó un instrumento de evaluación para cada perspectiva.

Además, para alcanzar los objetivos del programa de mejora del proceso de requisitos, fue necesario integrar el modelo IDEAL con casos de mejora e Investigación - Acción, con el propósito de generar resultados tangibles y a corto plazo; y lograr que el modelo de mejora este alineado a las necesidades de la Organización.

Asimismo, se comparte el análisis de resultados, financiero, de impacto y de adopción una vez finalizado el programa de mejora, donde se establece el ROI, los factores de éxito, los nuevos niveles de adopción de las prácticas recomendadas por CMMI y los índices de satisfacción de los usuarios.

Dentro de los resultados obtenidos se destacan los siguientes:

- La definición de un modelo de mejora ágil mediante la integración de modelo IDEAL, con casos de mejora para obtener mejoras en periodos de tiempo corto e Investigación – Acción y así establecer un esquema de trabajo colaborativo.
- Un Proceso de requisitos mejorado y en ejecución, donde las evidencias y la información recopilada justifican y respaldan el incremento del índice de satisfacción del usuario y el nivel de adopción de las áreas de proceso RD y REQM.
- Elaboración de un nuevo instrumento de evaluación del proceso de requisitos tomando como referencia CMMI-DEV.
- Elaboración de un artículo llamado “Complementar el modelo IDEAL con los principios ágiles por medio de casos de mejora e Investigación - Acción”, donde se justifica el uso de estos modelos en un programa de mejora.

1.2. Motivación

Actualmente, las empresas que desarrollan software ejecutan procesos altamente dependientes, en los que el resultado de un proceso es insumo para otros procesos; esto significa que garantizar resultados de calidad en un proceso influye directamente en resultados exitosos de los demás; como lo indica [6] “la calidad de un producto de software está altamente determinada por la calidad del proceso utilizado para su desarrollo y mantenimiento”, llevando a una mayor satisfacción de las necesidades o requisitos del cliente. Esta premisa ha llevado a las empresas productoras de software, o con áreas internas dedicadas específicamente a esta función, a mejorar su proceso software con el propósito de desarrollar soluciones de calidad y para alcanzar niveles de competitividad similares a los de los referentes en la industria. Algunas organizaciones realizan el esfuerzo de certificar o valorar sus procesos utilizando referentes internacionales, debido a que les permite obtener visibilidad y proyectarlas a un mercado nacional e internacional.

Las Organizaciones de servicios con áreas dedicadas al desarrollo de software, aunque no tienen como objetivo de negocio la producción de software, se han visto influenciadas por las mismas tendencias, pues sus objetivos de negocio en algunas ocasiones se ven directamente relacionados con el desarrollo o uso de software confiable y seguro, y el avance tecnológico puede representar para éstas una ventaja frente a sus competidores y un valor agregado para sus usuarios o clientes potenciales.

Algunos estudios han presentado los beneficios obtenidos por las empresas tras implementar mejoras en las diferentes fases del proceso software, entre los que se pueden evidenciar:

- Mejora en la capacidad para cumplir con los cronogramas y presupuestos establecidos [7].
- Aumento de la moral y la productividad del equipo de trabajo [7].
- Aumento de la calidad del producto desarrollado [7].
- Aumento de la satisfacción de los clientes [7].
- Disminución en los tiempos de ejecución de los proyectos [14].
- Mejoras en la documentación asociada al proyecto y a los procesos [14].
- Reducción de los defectos entregados a clientes [13].
- Identificación de defectos y su solución rápida en forma prioritaria [13].
- Manejo de líneas base, versiones evolutivas y auditorías a los cambios [13].
- Produce evidencia objetiva del nivel de calidad del producto [13].
- Permanentemente se identifican maneras para mejorar el proceso software y los productos desarrollados [13].

Como lo señalan en [11], los beneficios que se pueden obtener al implementar un programa que permita mejorar el proceso de requisitos varían entre:

- Obtener referencias precisas a requisitos específicos del proyecto.
- Guiar a todos los involucrados en las actividades concernientes a la gestión de los requisitos.
- Mantener y almacenar información consistente y rastreable.
- Gestionar fácilmente grandes cantidades de requisitos en un mismo proyecto.
- Contar con un framework para gestionar sistemáticamente el cambio en los requisitos.
- Identificar el impacto económico de un cambio en los requisitos.
- Simplificar la gestión de los requisitos.
- Evitar reprocesos cuando requisitos descartados son propuestos nuevamente.

Igualmente, se ha identificado que al mejorar el proceso de requisitos en las organizaciones que fabrican software ha representado un aumento en la calidad de sus productos y en la satisfacción de sus clientes, así como internamente se presenta disminución en los reprocesos generados, aumento en la productividad de los analistas y desarrolladores involucrados, y mayor satisfacción y confianza interna frente a los productos generados.

De otro lado, la importancia que han adquirido los programas de mejora del proceso software, ha permitido que algunos autores identifiquen y analicen el ROI obtenido tras la implementación de estos programas en diferentes tipos de

empresas, obteniendo para los casos analizados un rango para el ROI de 1.5 a 19, que en promedio implica un ROI de 7 para los procesos de mejora analizados en [14], lo que ha significado para las organizaciones implicadas un ahorro de varios millones de dólares en la ejecución de sus procesos, como lo demuestran los siguientes casos de éxito en la mejora del proceso de requisitos:

- Unisys australiano [8].
- Motorola [62]
- Hughes [61]
- Model of SPI for Requirement Engineering [27].

Aunque estos casos de estudio comparten los beneficios obtenidos después de mejorar el proceso de requisitos, también, se evidenció la escasez de literatura y experiencias compartidas en materia de programas de mejoramiento del proceso de requisitos, en particular. Las experiencias vividas por las organizaciones no trascienden al exterior de éstas; se percibe que la razón principal es que todo el conocimiento que se obtiene en dichos programas es considerado como una ventaja competitiva de la compañía. Esta situación afecta negativamente el crecimiento o fortalecimiento de la ciencia y la industria, pues la academia y las organizaciones no cuentan con bases suficientes para demostrar y aprender del éxito de estos programas de mejoramiento.

Se observa también que las organizaciones que desean efectuar mejoras a sus procesos conducentes a certificación, en su mayoría lo realizan con el acompañamiento de consultores externos, lo cual evidentemente acrecienta los costos de la inversión en los programas de mejoramiento del proceso de software. Es así como toman importancia iniciativas, como las que se presentan en este trabajo, que se efectúan desde dentro de los equipos de desarrollo por parte de sus líderes, con el fin de que las estrategias de mejora surjan de los mismos actores del proceso, creando una sinergia que mantiene la motivación del equipo y provee un espacio de aprendizaje permanente.

Con esta experiencia en una Organización de Servicio, no sólo se compartirán los modelos e instrumentos aplicados, los beneficios y dificultades encontradas después de mejorar el proceso de requisitos, sino que también se aproximará a los costos y los recursos a los cuales una organización deberá incurrir al momento de encarar la mejora a su proceso, en el área de la Ingeniería de Requisitos.

1.3. Objetivos

1.3.1. Objetivo General

Proponer y llevar a cabo mejoras al proceso de requisitos en una Organización de Servicios, con el fin de aumentar la calidad de los productos y servicios de software construidos.

1.3.2. Objetivos Específicos

- Definir un modelo de mejora ágil que se adapte a las necesidades de la Organización.
- Valorar el proceso de requisitos utilizado actualmente en la Organización frente a un referente reconocido en la Industria del Software.
- Establecer estrategias y planes de acción de mejora sobre el proceso de requisitos, las cuales estén acorde al contexto de la Organización.
- Ejecutar los planes de acción definidos para la mejora del proceso de requisitos utilizado por la Organización.
- Evaluar la aplicabilidad de los cambios de mejora y los resultados de la institucionalización.

1.4. Preguntas de Investigación

A través del programa de mejora se desea dar respuesta a las siguientes preguntas de investigación:

- ¿De qué manera integrar el Modelo IDEAL con Casos de Mejora y la Investigación – Acción para guiar la mejora del proceso de requisitos en la organización?
- ¿De qué manera evaluar logros tempranos en el proceso de mejora de requisitos?
- ¿Cuáles son las herramientas y prácticas de apoyo que facilitan un proceso de adopción liviana por parte del equipo de desarrollo?
- ¿Cuál es el papel que pueden jugar los clientes del producto en un proyecto de mejora de procesos?
- ¿Cuáles son los factores de éxito del proyecto de mejora?

1.5. Organización

Aparte de éste capítulo introductorio, esta experiencia contiene otros seis (6) capítulos y un (1) Anexo, los cuales están organizadas de la siguiente manera:

Capítulo 2. Marco Teórico: Presenta los diferentes modelos de proceso, de evaluación y de mejora utilizados por la industria en iniciativas SPI. También presenta una visión del estado del arte de los programas de mejora del proceso software con los factores involucrados en este tipo de iniciativas. Por último se conceptualiza sobre la experimentación en la Ingeniería del Software.

Capítulo 3. Contextualización: Presenta la estructura de la Organización en estudio con el proceso software utilizado actualmente para el desarrollo de software, donde se indica las limitaciones que posee. Además, se enumeran las restricciones establecidas por la Organización para el programa de mejora.

Capítulo 4. Definición del Programa de Mejora: Presenta los modelos de mejora, evaluación y de procesos seleccionados en esta experiencia de mejora y se indica la metodología de trabajo definida para el programa de mejora.

Capítulo 5. Experiencia de mejora del proceso de requisitos: Presenta en detalle la experiencia de mejora a través del ciclo de vida mejora.

Capítulo 6. Análisis de los resultados del programa de mejora: Presenta en detalle el análisis que se llevó a cabo sobre los resultados obtenidos del programa de mejora, con el propósito de responder las preguntas de investigación. Además, se muestra un análisis financiero consolidado.

Capítulo 7. Conclusiones: Presenta las contribuciones del trabajo y sus conclusiones

Anexo: Presenta los instrumentos construidos para la evaluación del proceso de requisitos desde la perspectiva de los usuarios y analistas.

2. Marco Teórico

En este capítulo se describen los componentes que conforman un programa de mejora y se presentan los modelos más relevantes e utilizados en cada componente, e igualmente, se referencian los factores de éxito que deben estar presentes en la definición de un plan de mejora, asimismo, se indica el significado de casos de mejora, la importancia de la experimentación en la Ingeniería del Software y como la Investigación – Acción ha contribuido a la ciencia.

2.1. Programa de Mejora

Un programa de mejoramiento de procesos de software o una iniciativa de SPI (Software Process Improvement) debe estar conformada por un modelo de procesos que provea el marco de referencia a seguir, un modelo de evaluación de procesos que permita establecer el estado actual del proceso utilizado por la Organización y un modelo de mejora de procesos que sirva de guía para realizar la mejora al proceso actual, como se muestra en la siguiente figura 1.

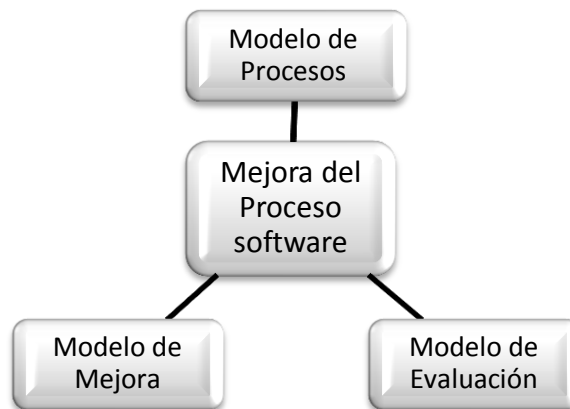


Figura 1. Modelos de un programa de mejora (Construcción propia)

A continuación se mostrarán los modelos más utilizados en programas de mejoramiento de procesos, los cuales serán de ayuda para establecer las mejoras prácticas en esta experiencia de mejora:

2.2. Modelos de Procesos

Los modelos de procesos de software son utilizados como marco de referencia, puesto que proveen las mejores prácticas que deben ser implementadas en los procesos software. Estos modelos son descritos genéricamente, lo que permite adaptarlos o modificarlos según las necesidades y recursos de las

Organizaciones. Entre los modelos de procesos más mencionados en los últimos años se encuentran:

2.2.1. CMMI-DEV

CMMI (Capability Maturity Model Integration) es un modelo de madurez de mejora de los procesos para el desarrollo de productos y de servicios. Este modelo es desarrollado por equipos de producto con miembros procedentes de la industria, del gobierno y del Software Engineering Institute (SEI) [28].

Este modelo, suministra un conjunto completo e integrado de guías para desarrollar productos y servicios y tiene como propósito proporcionar una orientación para aplicar las buenas prácticas CMMI en una organización de desarrollo. Las buenas prácticas del modelo se centran en las actividades para desarrollar productos y servicios de calidad con el fin de cumplir las necesidades de clientes y usuarios finales [28].

CMMI – DEV 1.3 se compone de 22 Áreas de Procesos, las cuales están estructuradas con un conjunto de metas específicas o genéricas, que a su vez tienen prácticas específicas y genéricas respectivamente, como se muestra en la figura 2.

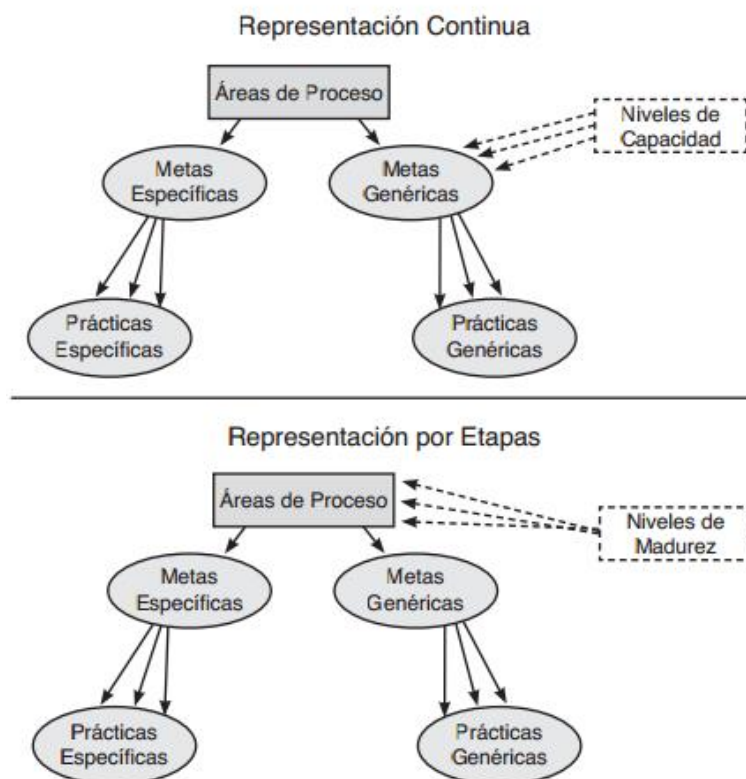


Figura 2. Estructura de las representaciones continua y por etapas [28].

Además, CMMI dispone de dos representaciones:

- **Continua:** Permite a la organización elegir el enfoque de sus esfuerzos de mejora de procesos, eligiendo aquellas áreas de proceso, o conjuntos de áreas de proceso interrelacionados, que más benefician a la organización y a sus objetivos de negocio. Aunque existen algunos límites sobre lo que una organización puede elegir debido a las dependencias entre áreas de proceso, la organización tiene una libertad considerable en su selección.
- **Por etapas:** Permite elegir un grupo de Áreas de Proceso, predefinidas por el modelo, sobre las cuales la organización centrará sus esfuerzos y una vez que se cubran todas las áreas de proceso de un nivel “n” se podrá afirmar que la organización ha alcanzado el nivel de madurez “n”.

Y de acuerdo a la representación, se establecen los niveles de capacidad o madurez, como se detalla en la tabla 1:

Nivel	Representación Continua / Niveles de Capacidad	Representación por Etapas / Niveles de Madurez
Nivel 0	Incompleto	
Nivel 1	Realizado	Inicial
Nivel 2	Gestionado	Gestionado
Nivel 3	Definido	Definido
Nivel 4		Gestionado Cuantitativamente
Nivel 5		En optimización

Tabla 1. Tipo de representación de CMMI [28].

2.2.2. Competisoft

El Modelo de Procesos de Competisoft está basado principalmente en el modelo MoProSoft [30], pero también adopta conceptos de Métrica V3, MPS.BR, CMMI, ISO/IEC 12207 e ISO/IEC 15504.

Esta iniciativa fue apoyada por el organismo Iberoamericano CYTED (Ciencia y Tecnología para el Desarrollo) con el objetivo de *“Incrementar el nivel de competitividad de las PyMES Iberoamericanas productoras de software, mediante la creación y difusión de un marco metodológico común que, ajustado a sus necesidades específicas, pueda llegar a ser la base para establecer un mecanismo de evaluación y certificación de la industria de software”* [29]. Y como resultado el proceso en tres grandes categorías:

- **Alta Dirección:** Contiene el proceso de Gestión de Negocios.
- **Gerencia:** Está integrada por los procesos de Gestión de Procesos, Gestión de Proyectos y Gestión de Recursos.
- **Operación:** Está integrada por los procesos de Administración de Proyectos Específicos, Desarrollo y Mantenimiento de Software.

Además, establece las relaciones entre las tres categorías, a través de sus procesos y subprocesos; y la retroalimentación se genera desde las categorías más genéricas hacia las más específicas y viceversa, como se ve en la figura 3.

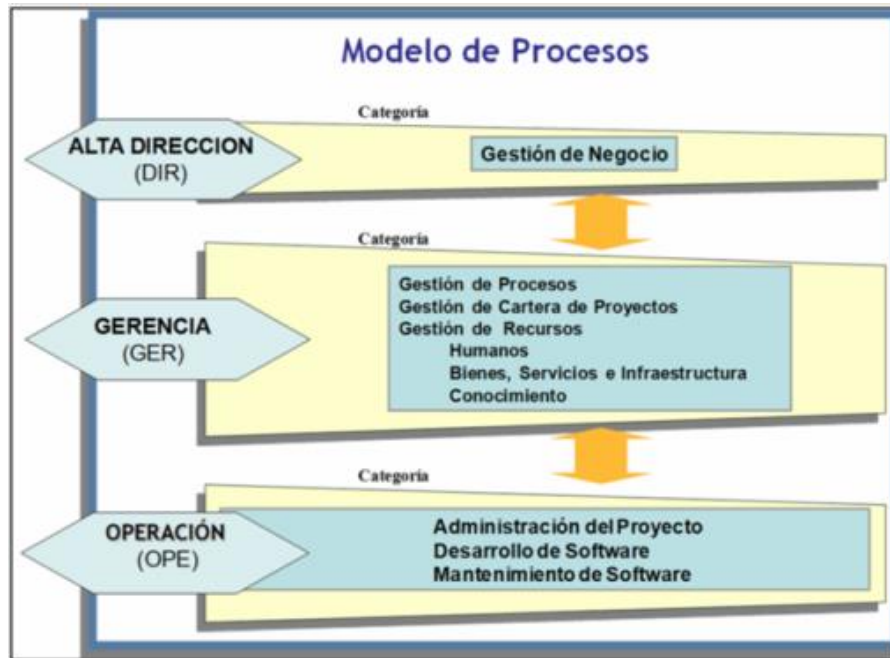


Figura 3. Modelo Competisoft [29].

Por último, este modelo puede ser utilizado por Organizaciones que tengan definido un Proceso Software o este en construcción, en caso la Organización que no disponga con procesos establecidos ni documentados, pueden usar el modelo ajustándolo de acuerdo a sus necesidades, pero si la Organización cuenta con procesos establecidos, se debe identificar las coincidencias y discrepancias entre ellos, para luego establecer las prácticas que se deben adoptar según las necesidades de la Organización [29].

2.2.3. Métrica V3

Métrica V3 es una metodología desarrollada y promovida por el Ministerio de Administraciones Públicas del Gobierno de España para la planificación, desarrollo y mantenimiento de sistemas informáticos para la gestión de actividades del ciclo de vida de los proyectos software dentro de las administraciones públicas, basada en Modelos de ciclo de vida de procesos de desarrollo "ISO/IEC 12207 e ISO/IEC 15504" [31]. Esta metodología tiene como objetivo:

- Proporcionar y definir Sistemas de Información que ayuden a alcanzar los fines de la organización.
- Mejorar la productividad de TIC.

- Facilitar la comunicación entre los participantes.
- Facilitar la operación, mantenimiento y uso de los productos de software.

En la figura 4, se muestra los elementos que componen la metodología de trabajo, entre los que se destacan: Procesos, actividades, tareas, interfaces, roles y técnicas.

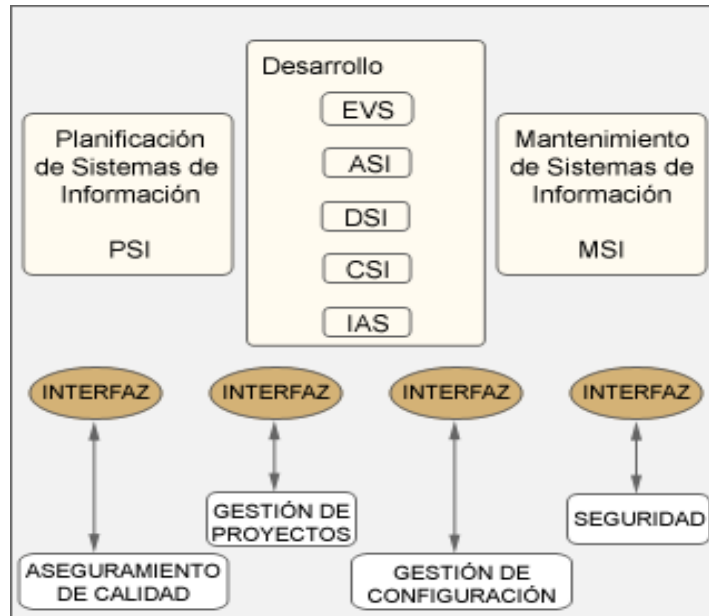


Figura 4. Modelo Métrica V 3 [31].

2.2.4. MPS.BR

MPS.BR es un modelo de mejora y evaluación de proceso de software para las micro, pequeñas y medianas empresas relacionadas a la industria del software del Brasil, el cual cuenta con el Apoyo del Ministerio de Ciencia y Tecnología, de la Financiera de Estudios y Proyectos y del Banco Interamericano de Desarrollo.

Este modelo es basado en las normas internacionales ISO/IEC 12207, ISO/IEC 15504 y CMMI-DEV y está dividido en tres componentes, como se puede ver en la figura 5 [32]:

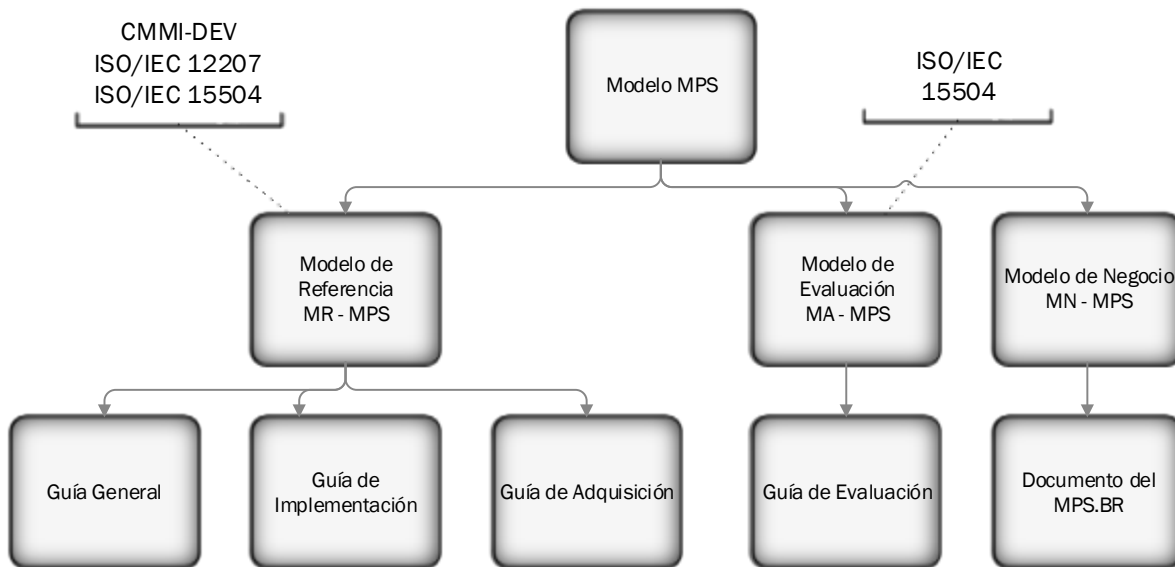


Figura 5. Modelo MPS.BR [32].

2.3. Modelos de Evaluación de Procesos

Los modelos de evaluación son muy importantes, puesto que permiten determinar el grado de madurez y capacidad del proceso actual con respecto a un marco de referencia y de esta manera identificar las fortalezas y debilidades del proceso. La gran mayoría de los modelos de evaluación están articulados con los Modelos de Procesos, por ejemplo: Para CMMI, SCAMPI es el modelo de evaluación, para Competisoft es pmCompetisoft Evaluación, mientras que para MoProSoft es EvalProSoft, entre otros. A continuación se describen dos de los más representativos.

2.3.1. SCAMPI

SCAMPI proviene de las siglas en inglés de Método Estándar de Evaluación CMMI para mejora de procesos (Standard CMMI Appraisal Method for Process Improvement) y determina el nivel de madurez o capacidad, que ha alcanzado una organización que aplica CMMI en sus procesos, es importante mencionar que el SEI (Software Engineering Institute) sólo reconoce las evaluaciones de procesos realizadas con SCAMPI como válidas para determinar el nivel de madurez de una organización [33].

El principal objetivo de esta evaluación es establecer el estado actual de las prácticas de la organización para identificar fortalezas y oportunidades de mejora, así como las prioridades para las acciones de mejora. De cierta manera se determina el grado de adherencia de la organización con respecto a las prácticas

sugeridas en el modelo de referencia, según la rigurosidad con que se haga dicho proceso de evaluación existen tres clases de evaluaciones:

- SCAMPI Clase A: El método más amplio, con mayor cobertura del modelo y es el único que puede proporcionar un nivel de madurez o perfil de capacidad. Es liderado por un SCAMPI Lead Appraiser autorizado por el SEI.
- SCAMPI Clase B: Es menos amplio y detallado que el clase "A" y eventualmente más económico. Se utiliza como evaluación inicial o parcial, enfocado en las áreas que requieren atención. En este caso no requiere de un Lead Appraiser para ser realizado.
- SCAMPI Clase C: Es el más sencillo, económico y requiere una capacitación menor. Se enfoca en áreas de interés o de mayor riesgo en la organización.

2.3.2. ISO/IEC 15504

ISO/IEC 15504, también conocido como Software Process Improvement Capability Determination, abreviado SPICE es un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas de información y productos de software, que permite establecer y mejorar la capacidad y madurez de los procesos de las organizaciones independiente si tienen relación con el proceso software.

Además, ISO/IEC 15504 provee un marco de referencia que tiene la capacidad de asegurar la coherencia y consistencia de los resultados de las evaluaciones; es decir, distintos evaluadores podrían otorgar las mismas calificaciones si estuvieran evaluando los mismos procesos de una organización [36], [57]. Esta norma puede emplearse de tres formas, según las necesidades de la organización:

- I. Si una organización quiere mejorar sus procesos y evaluar los procesos de mejora.
- II. Si una organización sub-contrata el desarrollo de software y desea evaluar la capacidad de su proveedor.
- III. Si una organización quiere certificar la adecuación de sus procesos, por ejemplo a la Norma NTP-ISO/IEC 9000.

La norma en el transcurso del tiempo ha sufrido modificaciones estructurales. Actualmente está conformada por siete partes, donde las partes 1, 2, 7 son normativas, puesto que en ellas se definen los requisitos mínimos para realizar una mejora de procesos de desarrollo y para medir el nivel de madurez de la organización en cuanto al desarrollo de software, y por otro lado, las partes 3, 4, 5 y 6 son no normativas, dado que es en donde se dan las guías de interpretación de los requisitos mínimos y en sí sobre la norma [26]. Figura 6.

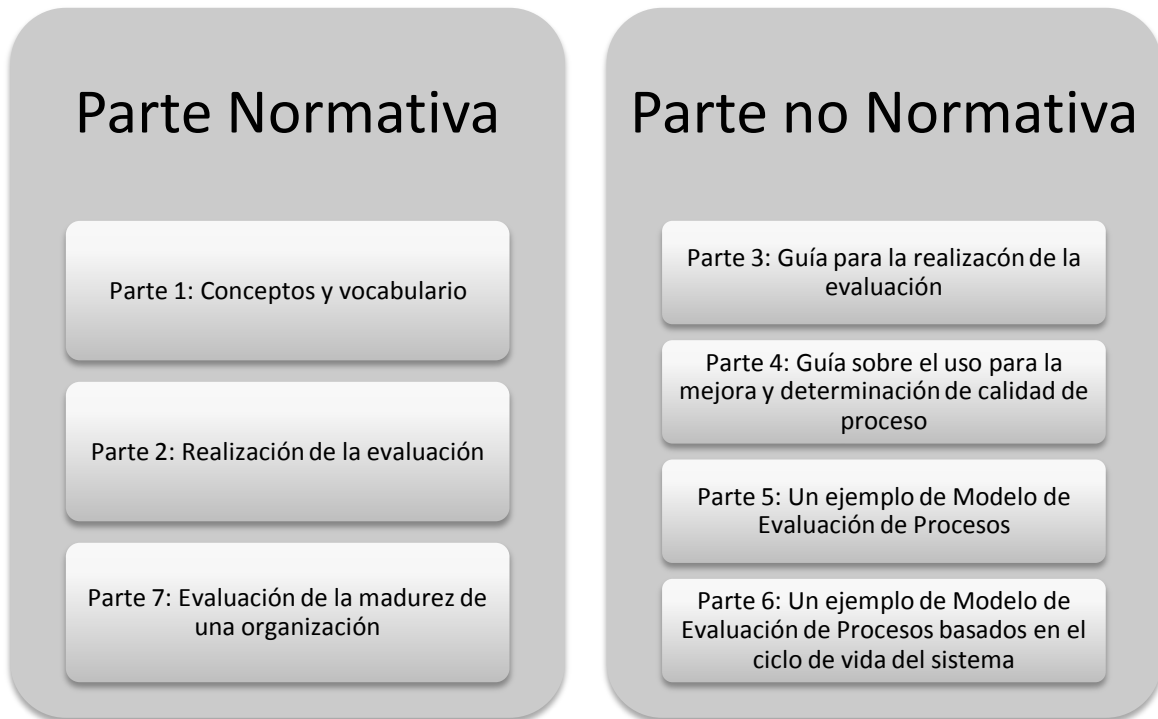


Figura 6. Composición ISO/IEC 15504 (Adaptación de [29])

En [34] indican las limitaciones y beneficios que ofrece la norma tomando como referencia el artículo de IEEE [54]:

Limitaciones:

- La dimensión de la capacidad ha alcanzado un alto grado de dificultad, Presentándose en algunos casos solapamientos con la dimensión de los procesos.
- La complejidad de la evaluación y su falta de practicidad, conllevan al incremento del costo, haciéndola significativamente menos factible de implementar comparándola con otros modelos.

Beneficios:

- ISO/IEC 15504 se constituye como el primer modelo de procesos que evalúa las capacidades en dos dimensiones independientes (Procesos Vs. Capacidades).
- Define criterios de conformidad para permitir la comparación de modelos y procesos externos, con la posibilidad de encontrar requisitos comunes.
- Es específico para la industria del software, incluye los procesos relacionados al desarrollo y mantenimiento de software.
- Por la antigüedad de su aparición y los continuos refinamientos, dispone de más pruebas y del consenso de la industria del software.

2.4. Modelo de Mejora de Procesos

Un modelo de mejora de procesos proporciona un marco de trabajo con un conjunto de actividades y guías que ayudan a mejorar las capacidades de un proceso inicial mediante un ciclo de vida mejora, el cual puede variar dependiendo del modelo a seguir.

Adicionalmente, para una Organización es importante implementar un proceso de estrategia de mejoramiento para producir un proceso de software bien definido [24] y que ayude alcanzar los objetivos propuestos.

Hoy en día existen diferentes modelos de mejora de procesos, de los cuales en esta sección se describirán: Modelo IDEAL, Agile SPI y PmCOMPETISOFT.

2.4.1. Modelo IDEAL

IDEAL es un modelo de mejora estructurado que sirve como guía para iniciar, planificar e implementar acciones de mejora eficazmente, además, le permite a las Organizaciones hacer uso un método integral para la gestión y la mejora de la capacidad [35].

Adicionalmente, este modelo de mejora ofrece un conjunto de actividades que permite mejorar o adoptar las prácticas recomendadas por el CMMI para el Proceso Software, estas prácticas podrán ser enfocadas de diferente manera dependiendo de las características de la organización. Dichas actividades están distribuidas en cinco fases que conforman el ciclo de vida de mejora, es importante indicar que las iniciales de estas fases conforma el nombre del modelo, a continuación se describen: [36]

- **Iniciación:** Esta fase del modelo es el punto inicial de mejora, puesto que en ella son establecidos la infraestructura de mejoramiento, además de los roles, las responsabilidades y los recursos. Igualmente se definen las metas principales del programa de mejora, las cuales son establecidas de acuerdo a las necesidades del negocio de la organización.
- **Diagnóstico:** Se caracteriza el estado inicial de la organización mediante evaluaciones a las prácticas actuales, que permitan identificar las debilidades y fortalezas, con el objetivo de establecer las recomendaciones pertinentes.
- **Establecimiento:** En la fase de establecimiento, los problemas que la organización que identifico en la fase previa, son priorizados y clasificados. Por otro lado, en esta fase la meta global establecida en la fase de

iniciación, es descompuesta en varias metas para construir los planes de acción.

- **Actuación:** Es en esta fase que se desarrollan las mejoras y soluciones a los problemas identificados anteriormente, por medio de pilotos de prueba y de evaluación, además, se desarrolla una estrategia de despliegue de la solución.
- **Aprendizaje:** En esta fase se miden los logros alcanzados y se detallan las ocurrencias y percances a manera de Lecciones Aprendidas para que puedan servir como guía base en los próximos ciclos de mejora.

2.4.2. Ágil SPI

El modelo de mejora Agile SPI es el resultado del proyecto SIMEP-SW, el cual consistía en crear, aplicar y probar un proceso de mejora. Adicionalmente, el modelo nace con la premisa esencial que los modelos utilizados sean ligeros y basados en estándares internacionales acordes a las características, idiosincrasia y circunstancias de la realidad socio-económica de la industria de software en Colombia [56].

Agile SPI es un Framework que se caracteriza por:

- Guiar la mejora de los procesos de desarrollo de software, manteniendo el nivel de agilidad que la empresa lo desee.
- Ser un Framework basado en modelos livianos, que soporte un programa de mejoramiento continuo, a través de un proceso de mejora ágil.
- Estar acorde con una industria dinámica, creativa, innovadora e incierta como lo es la industria del software. Una industria donde el conocimiento y el talento humano son elementos fundamentales para garantizar su éxito.

Y plantea los siguientes componentes en su arquitectura:

- **Light SPI Evaluation Model:** Un modelo ligero de evaluación del proceso productivo.
- **Light SPI Metrics Quality Model:** Un modelo ligero de métricas del proceso productivo.
- **Light SPI Quality Model:** Un modelo de calidad ligero.
- **Framework PDS:** Un marco conceptual y tecnológico para soportar procesos.
- **Agile SPI Process:** Un proceso ágil que guía a una mejora de procesos.

Es importante mencionar que Agile SPI – Process adopta los principios del manifiesto ágil y las características para un SPI liviano, lo cual hace que los

programas de mejora se enfoquen en arrojar resultados ágiles y de acuerdo a las necesidades de las empresas. [37], [38]

Adicionalmente, el modelo Agile SPI-Process es un proceso iterativo e incremental basado en casos de mejoramiento, los cuales tienen la característica de entregar resultados rápidos de mejora porque permiten crear mini programas de mejoramiento que incluyan casos de mejora dentro del programa de mejora global [39].

El modelo Agile SPI se puede dividir en cinco fases, las cuales son: [37]

- **Instalación del Programa:** En esta fase se crea una propuesta de mejora basada en las necesidades del negocio, la cual ayudará a guiar a la organización a través de cada una de las fases siguientes, esta propuesta debe ser aprobada por la gerencia para garantizar así la asignación de los recursos necesarios para el proyecto de mejora. Durante esta fase también se definen unos objetivos, los cuales son establecidos desde las necesidades de la empresa.
- **Diagnóstico:** En esta fase se realizan actividades de valoración para saber cuál es el estado general de los procesos de la empresa, además de un análisis de los resultados que permitan establecer la prioridad de los casos de mejora,
- **Formulación:** En esta fase son clasificados y priorizados los casos de mejora identificados y se realiza la planificación de una primera iteración de mejora, esto con el fin de realizar una medida del esfuerzo que sirva de base para la estimación del esfuerzo que tomará llevar a cabo el resto del proyecto de mejora.
- **Mejora:** En esta fase, todo el esfuerzo de los casos de mejora es administrado basado en la estimación y planificación hecha en el plan de ejecución de mejora creado en la fase previa. En un documento se registra la ejecución de los pilotos de prueba y se indican los resultados.
- **Revisión del Programa:** En esta fase todas las lecciones aprendidas y las métricas desarrolladas para medir el logro de los objetivos sirven como conocimiento base de información para las personas envueltas en los siguientes ciclos de mejora.

2.4.3. PmCOMPETISOFT

Es un proceso de mejora de procesos de software que guía la ejecución de un ciclo de mejora de procesos de software en pequeñas y medianas empresas (PyMEs). Se caracteriza por ser liviano para su aplicabilidad en las PyMEs de

software y está basado en Agile SPI, el cual es un framework creado para la industria del software de Colombia y enfocado a las micro, pequeñas y medianas empresas – PyMEs [40].

PmCompetisoft integra el modelo de mejora y es un proceso explícito que proporciona una guía, paso a paso, para llevar a cabo esfuerzos de mejora de procesos [41]. Siendo un proceso, iterativo e incremental organizado a través de iteraciones de mejora que abarcan casos de mejora dentro de un ciclo de mejora global. El objetivo de esta estructuración es obtener resultados rápidos de mejora.

Adicionalmente, PmCompetisoft es un proceso que sigue un enfoque iterativo e incremental, y planea satisfacer los siguientes principios [42]:

- Entrega temprana y continua de mejoras.
- Diagnóstico continuo y rápido de procesos.
- Medición básica de procesos.
- Colaboración efectiva entre grupos.
- Aprendizaje continuo.

PmCompetisoft se compone de uno o más ciclos de mejora. Cada ciclo de mejora consta de 5 fases: instalación del programa, diagnóstico de procesos, formulación de mejoras, ejecución de mejoras y revisión del programa. A continuación se describirán estas fases [40]:

- **Instalación del ciclo:** En esta fase se construye una propuesta de Mejora alineada con la planeación estratégica de la organización plasmada en el Plan Estratégico. Esta propuesta guía a la organización a través de cada una de las fases siguientes de la iniciativa de mejora. Además se establece o se actualiza la propuesta, la cual debe contener: el proceso de mejora, los objetivos de mejora generales y los recursos necesarios para llevar a cabo la iniciativa de mejora al interior de la organización.
- **Diagnóstico de los procesos:** En esta fase se realiza la actividad de valoración (evaluación interna) de procesos para conocer el estado general de los procesos de la empresa y analizar los resultados con el objetivo de establecer los casos de mejora y sus prioridades.
- **Formulación del ciclo de mejora:** En esta fase se realiza la primera iteración del ciclo de mejora con el fin de obtener una medida del esfuerzo de conducir esta iteración. Esta información se utiliza como base para la estimación del esfuerzo, costo y tiempo que demandarán las demás iteraciones del ciclo de mejora.

- **Mejora de procesos:** En esta fase se gestiona y ejecuta las mejoras correspondientes a la iteración piloto de acuerdo con los planes establecidos. Si la planificación de la iteración piloto se ha desarrollado satisfactoriamente se acepta e institucionaliza los nuevos procesos en la organización. Esta actividad puede realizarse una o varias veces en un ciclo de mejora
- **Revisión del ciclo de mejora:** En esta fase se corrige o se ajustan todos los elementos relacionados con la ejecución de cada una de las iteraciones piloto de mejora y también se hace un análisis post-mortem del trabajo realizado en todo el ciclo de mejora.

Es importante resaltar que el modelo IDEAL se ha convertido en un referente en la definición de otros modelos de mejora, tales como: Ágil SPI [64], IAPM [65], SPIMS [66] e IMPACT [67], debido a su capacidad de adoptar nuevos conceptos, por tanto, este modelo de mejora se convirtió en la guía base para el mejoramiento del proceso de requisitos, el cual fue complementado en su ciclo de mejora, con prácticas de casos de mejora e Investigación – Acción (Capítulo 4). Con el propósito de analizar e intervenir diferentes problemáticas del proceso, como problemas independientes y de esta manera obtener planes de mejora más específicos e integrales según las particularidades de cada problemática y así lograr que el modelo de mejora de procesos esté más acorde a las necesidades de la Organización.

Por tanto, el modelo IDEAL fue seleccionado como principal instrumento para apoyar esta iniciativa, puesto que su objetivo principal es proporcionar una guía a las organizaciones para la mejora de sus procesos, basándose en experiencias adquiridas con clientes de la Industria y el Gobierno [16].

Además, los siguientes aspectos lo respaldan positivamente:

- Proporciona un conjunto de actividades coherentes para sustentar la adopción de las prácticas recomendadas por el CMMI, el cual es el modelo de procesos seleccionado en este programa de mejora.
- Posee una adecuada documentación, donde se especifica detalladamente lo que se debe tener en cuenta para emprender un programa de mejora.
- El modelo ha sido utilizado en varios programas de mejoramiento con resultados positivos.
- El modelo es flexible en cuanto a ejecución, puesto que las prácticas sugeridas pueden variar dependiendo del contexto de la Organización.

2.5. Factores de Éxito

Como indican en [43], el éxito de las iniciativas de mejora de procesos software se ve influenciado por un conjunto de factores críticos de éxito, así como también debe enfrentar una serie de factores de resistencia, barreras y obstáculos, tanto a nivel organizacional como de las personas involucradas y afectadas por el cambio.

El trabajo mencionado [43], realizó un estudio secundario con el objetivo de identificar factores que afectaron positiva o negativamente las iniciativas de mejora de procesos de software realizadas entre 1995 y 2011. Esta investigación consolidó resultados de 42 estudios, obteniendo como resultado los factores que se muestran en la tabla 2:

Factor	Tipo Factor	Cantidad de Estudios	Total
Positivo	Éxito	26	32
	Motivadores	6	
Negativo	Barreras	12	22
	Fracaso	2	
	Resistencia	2	
	Obstáculos	3	
	Desmotivadores	3	

Tabla 2. Tipos de factores (Adaptación de [43]).

Por último se realizó el proceso de categorización de los factores identificados, el cual dio como resultado 23 categorías que agrupan factores de éxito, motivadores, barreras, fracaso, resistencia, obstáculos y desmotivadores. En la tabla 3, se describen cada categoría con su respectivo significado y la cantidad de veces que los factores fueron citados en la literatura analizada, donde el orden indica la importancia relativa de la categoría.

Categoría	Factores Positivos	Factores Negativos	Significado
Compromiso de la alta gerencia	24	22	Refiere al compromiso y apoyo de los ejecutivos y de la alta gerencia a la iniciativa de mejora de procesos software.
Recursos	23	15	Refiere a la provisión de los recursos humanos, materiales y financieros necesarios para el desarrollo de las actividades de mejora de procesos.
Estrategia de implementación	22	14	Refiere al establecimiento de una metodología formal y de un plan para la implementación de la iniciativa de mejora de procesos software.
Objetivos de MPS claros	23	13	Refiere al establecimiento de objetivos precisos, relevantes y alcanzables para la

Categoría	Factores Positivos	Factores Negativos	Significado
			iniciativa de mejora de procesos software.
Participación del personal	25	9	Refiere al grado en que el personal dedicado a la mejora de procesos participa y se involucra en la realización de las actividades de mejora y en la toma de decisiones.
Conciencia y percepción de beneficios	16	18	Refiere a que el personal involucrado en la mejora de procesos perciba los beneficios y esté informado de los avances actuales obtenidos con el esfuerzo que se está realizando para mejorar los procesos software.
Comunicación	24	10	Refiere a establecer mecanismos para el intercambio de información entre los involucrados en la mejora de procesos software, y a la disseminación de esa información a nivel organizacional.
Estándares y procedimientos	24	9	Refiere a generar procedimientos de trabajo formales y procesos documentados.
Personal con habilidades, conocimientos	15	17	Refiere a que las personas dedicadas a las actividades de mejora posean conocimientos y experiencia en mejora de procesos software, así como habilidades adecuadas para gestionar la iniciativa en su conjunto.
Gestión del cambio cultural y organizac.	12	16	Refiere a gestionar adecuadamente los cambios a niveles cultural y organizacional derivados de la implementación de mejoras de procesos software.
Capacitación y Mentoring	18	5	Refiere, por un lado, a la provisión de entrenamiento relativo a mejora de procesos software al personal dedicado a las actividades de mejora y, por el otro, al establecimiento de relaciones de tipo mentor-tutelado para la transferencia de conocimientos y experiencia en mejora de procesos software.
Tiempo	8	14	Refiere a la dedicación de suficiente tiempo para las actividades de mejora de procesos software.
Autonomía	18	0	Refiere el grado en el que se habilita al personal dedicado a la mejora de procesos a llevar a cabo sus actividades sin prescripción de roles o funciones específicos, así como el nivel de responsabilidad y autoridad que se les otorgue para diseñar y modificar los procesos.
Grupo de trabajo	11	4	Refiere a establecer un grupo de trabajo

Categoría	Factores Positivos	Factores Negativos	Significado
dedicado a la MPS			permanente cuyos integrantes estén dedicados específicamente a las actividades de mejora de procesos software.
Carrera profesional	15	0	Refiere a las expectativas de que la mejora de procesos software permita a los involucrados progresar laboral y profesionalmente, dentro y fuera de la organización.
Políticas organizacionales	10	4	Refiere a las decisiones y acciones que, a nivel organizacional, pueden influir en la iniciativa de mejora de procesos software.
Gestión del proyecto de mejora	5	7	Refiere a considerar la iniciativa de mejora de procesos software como un proyecto con todas sus características y gestionarlo consecuentemente.
Liderazgo	7	3	Refiere a la actitud de impulsar de manera constante y sistemática las actividades de mejora de procesos.
Compensaciones	10	0	Refiere a recompensar en forma específica a las personas involucradas en las actividades de mejora de procesos.
Mediciones	7	2	Refiere a contar con métricas que permitan realizar el seguimiento de las actividades de mejora de procesos y detectar desvíos en los avances previstos.
Personal de MPS respetado	7	0	Refiere al prestigio y consideración con que las personas dedicadas a la mejora de procesos software son percibidas en la organización.
Rotación de personal	0	6	Refiere a la ocurrencia de cambios en el conjunto de las personas dedicadas a las actividades de mejora de procesos software.
OTROS	25	2	Incluye aquellos factores que, o bien se encontraron reportados una sola vez en los estudios analizados, o bien no tienen cabida en las categorías precedentes.

Tabla 3. Factores identificados en programas de mejora según [43].

2.6. Casos de Mejora

Los casos de mejora son unidades atómicas de mejora en las áreas de procesos que se han sido seleccionadas de acuerdo al nivel de prioridades establecido por el negocio y por medio de ellos se puede lograr resultados de mejora a corto plazo, puesto que permiten establecer mini-programas de mejora dentro de un programa de mejoramiento global [37], y de esta manera afrontar el mejoramiento

de forma incremental e iterativa hasta ejecutar completamente el programa de mejoramiento del proceso software.

En modelos de mejoramiento de procesos como Competisoft y Agile SPI, han hecho uso del concepto de casos de mejora, debido a que les permite planificar iterativamente entregas de valor a la organización y enfocar el esfuerzo inicialmente en las áreas más estratégicas para la organización [3], [38].

Además, los casos de mejora también fueron utilizados con éxito en un programa masivo de mejora realizado en Colombia, donde se identificó que la principal ventaja o factor de éxito fue precisamente que los casos de mejora se constituyeron como la estrategia natural de como el equipo de mejora (conformada en buena parte por desarrolladores) segmentaba las mejoras en unidades significativas, haciendo una analogía con los casos de uso [3].

2.7. Experimentación en Ingeniería de Software

Actualmente, las teorías y modelos que se emplean en el desarrollo de software carecen de evidencias sobre su exactitud, adecuación, límites, cualidades, costes y riesgos [45]. Por tanto la experimentación en Ingeniería de Software es apropiada para proporcionar evidencia empírica que permita la evaluación y validación de estos atributos, convirtiéndose en una de las estrategias claves, para buscar acortar la brecha entre la teoría y el estado de la práctica de la ingeniería de software [53].

La experimentación le permite a la Ingeniería de Software enriquecer su cuerpo de conocimiento con mayor precisión, como lo ha realizado exitosamente en otras disciplinas de la ciencia como Física, Química, Medicina y más recientemente en disciplinas tradicionalmente no experimentales, tales como la Economía Experimental o la Psicología Experimental, puesto que el propósito de la experimentación es identificar las causas por las que se producen determinados resultados [46].

Este tipo de estudios ha cobrado gran importancia en la región, particularmente en la rama de la Ingeniería de Software, por ejemplo, Brasil ha sido uno de los países latinoamericanos que ha promovido el diseño de estudios experimentales en este ámbito de la ciencia. Vale la pena destacar el papel que tiene el grupo de Ingeniería de Software Experimental de la Universidad Federal de Rio de Janeiro (<http://lens-ese.cos.ufrj.br/ese/index.php?lang=en>), que es el grupo que da el soporte académico al programa de mejora que Brasil ha venido impulsando a través de MPS.Br.; algunos de los estudios que ha generado este grupo en los últimos años son los siguientes:

- Knowledge-Sharing Issues in Experimental Software Engineering [69].

- Estudos Primários e Secundários apoiando a busca por Evidência em Engenharia de Software [70].
- Aplicando uma Metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software [71].

Además, en Latinoamérica, por medio de “Latin American and Caribbean Collaborative ICT Research Federation - LACCIR” se está llevando a cabo el siguiente proyecto de investigación:

- LACXSER (Latin American Collaboratory of eXperimental Software Engineering Research - LACXSER), el cual promueve el uso de técnicas de CSCL (Computer Supported Collaborative Learning) para enseñar temas propios de Ingeniería del Software [72]. En la página oficial del proyecto (<http://www.lacxser.org/>), se pueden encontrar las publicaciones realizadas en el 2009 y 2010 por los integrantes del proyecto.

Y para finalizar, en Colombia también se ha trabajado sobre el tema, como lo evidencia estas iniciativas:

- Aplicación de Investigación – Acción y Casos de estudio utilizando Competisoft [58].
- Estudio empírico de aplicación de la metodología PSP en estudiantes de un programa de Tecnología en Sistemas con diferentes niveles de formación [59].
- Estado actual de la estimación de software en compañías colombianas que han adoptado CMMI [60].

Es importante resaltar que en la Ingeniería de Software Experimental se hace uso de métodos y técnicas experimentales como instrumentos para la investigación, entre los que se encuentra la Observación y la Investigación – Acción.

2.8. Investigación – Acción

El término "investigación acción" proviene del autor *Kurt Lewis* y fue utilizado por primera vez en 1944 para describir una forma de investigación que podía ligar el enfoque experimental de la ciencia social con programas de acción social que respondiera a los problemas sociales principales de entonces. Mediante la investigación – acción, Lewis argumentaba que se podía lograr en forma simultánea avances teóricos y cambios sociales [47], pero Lomax en 1990, define la investigación-acción como “*Una intervención en la práctica profesional con la*

intención de ocasionar una mejora” [47], el cual es el objetivo de los programas de mejoramiento de procesos (SPI).

A modo de síntesis, la investigación-acción es una espiral de ciclos de investigación y acción constituidos por las siguientes fases: planificar, actuar, observar y reflexionar, que tiene como fin mejorar la práctica actual. Ver figura 7.

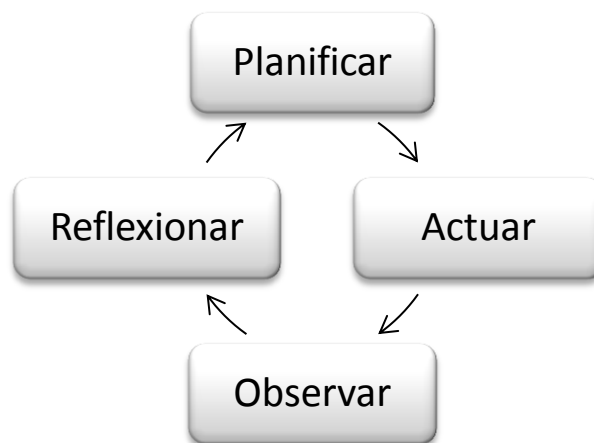


Figura 7. Investigación – Acción (Adaptación de [47]).

- **Desarrolla un plan de acción** para mejorar la práctica actual. El plan debe ser flexible, de modo que permita la adaptación a efectos imprevistos.
- **Actúa** para implementar el plan, que debe ser deliberado y controlado.
- **Observa** la acción para recoger evidencias que permitan evaluarla. La observación debe planificarse, y llevar un diario para registrar los propósitos. El proceso de la acción y sus efectos deben observarse y controlarse individual o colectivamente.
- **Reflexiona** sobre la acción registrada durante la observación, ayudada por la discusión entre los miembros del grupo. La reflexión del grupo puede conducir a la reconstrucción del significado de la situación y proveer la base para una nueva planificación y continuar otro ciclo.

Se podría decir que gran parte de la investigación en ingeniería de software aplica conceptos de la investigación-acción, puesto que muchas de las ideas claves en esta área de la ciencia fueron desarrolladas originalmente para ponerlas en práctica en proyectos reales de desarrollo, e informar sobre las experiencias obtenidas. En este orden de ideas, Dittrich [48] describe el desarrollo de sistemas de cooperación como una forma de investigación-acción ideal para la ingeniería de software empírica; si se adopta el marco de la investigación-acción de manera

más explícita, es probable generar conocimiento sobre fenómenos complejos de la Ingeniería del Software [49].

La investigación-acción es también un marco atractivo para la investigación mezclado con actividades profesionales, especialmente para los profesionales interesados en reflexionar sobre sus experiencias y la transmisión de los resultados de aprendizaje para el beneficio de la Industria, Academia y la Ciencia.

2.9. Observación

La observación desempeña un papel importante en la investigación, no sólo es un elemento fundamental de la ciencia que hace parte del método científico y le permite al investigador acumular hechos que le ayudan a identificar un problema y descubrir pautas para elaborar una solución teórica de su problema.

El método de observación normalmente es un mecanismo que permite detectar aspectos interesantes al interior de los proyectos de software y sirven de base para nuevos proyectos experimentales como lo describen en el libro [50].

Hay dos tipos de observación: la pasiva y la participativa; las cuales son utilizadas frecuentemente en programas de mejoramiento de proceso de Software. En la pasiva, el investigador adquiere el conocimiento, principalmente, a partir de la percepción de los sentidos acerca de una realidad que aparentemente le es ajena, conscientemente trata de evitar introducir su presencia en el desarrollo de las circunstancias, es capaz de recibir información sin casi exigirla explícitamente de los sujetos que integran su objeto de estudio. En la participativa, el investigador se involucra directamente con su objeto de estudio, participando conscientemente en el desarrollo de los hechos [51].

Es importante mencionar que la Investigación – Acción y la Observación son algunos de los métodos más relevantes en Software Experimental, puesto que permite recopilar experiencias y evidencias de las teorías planteadas por los Investigadores, por tanto, este método permitirá abordar la mejora del proceso cualitativamente, a través de la observación y experiencia obtenida en el transcurso del programa, dejando evidencia de ellos como en un estudio formal.

De otro lado, a través de la investigación – acción y la Observación, los modelos de mejoramiento puede lograr un enfoque ágil mediante los siguientes aspectos:

- Establecer comunicación cara a cara entre los integrantes del programa de mejoramiento.
- Obtener aprendizaje continuo por medio de la observación y reflexión.
- Establecer un esquema de trabajo colaborativo y participativo.
- Realizar retrospectivas del avance del programa.

- Mejorar los planes de mejora contruidos por medio de la participación y colaboración de los integrantes del equipo.
- Minimizar la cantidad de documentación a realizar mientras se refina la solución.

3. Contextualización

En este capítulo se describe la Organización y el departamento de informática en estudio, además, se describen las fases del proceso software utilizado actualmente por el área de desarrollo con sus debilidades, asimismo, se especifican las restricciones establecidas por la Organización para la ejecución del proyecto y las limitaciones que enfrentará el programa de mejora.

3.1. Contexto de la Organización

La Organización en estudio, en adelante la Organización, fue fundada hace más de 35 años con el propósito de garantizar a los Asociados y Ahorradores la seguridad en sus ahorros y aportes sociales, y convertirse en una alternativa social y económica para la comunidad Antioqueña, canalizando los recursos del Estado Colombiano para financiar actividades productivas.

Con base a lo anterior, la Organización definió su objetivo estratégico teniendo en cuenta su Misión y Visión, manifestando que desean cumplir el objetivo de permanecer, impactando positivamente la calidad de vida de los Asociados, Empleados y la Comunidad en general, mediante una eficiente gestión administrativa, financiera y comercial; a partir de esto, la Organización ha construido su plan estratégico donde concentrarán sus esfuerzos en los siguientes objetivos específicos:

- Lograr la excelencia operacional.
- Entregar productos y servicios basándose en las buenas prácticas de Tecnología.
- Crear cultura cooperativa fundamentada en la participación y el compromiso de los asociados.
- Ser la Entidad Cooperativa con mayor reconocimiento por la excelencia de su servicio.
- Fomentar las microfinanzas como un modelo de desarrollo.

En la definición de sus objetivos específicos se puede evidenciar que el Departamento de Informática se encuentra directamente involucrado por medio del segundo objetivo estratégico: “Entregar productos y servicios basándose en las buenas prácticas de tecnología.”

Debido a esto, el Departamento de Informática definió como principal objetivo, proveer a la Organización, la tecnología necesaria para la alcanzar los objetivos estratégicos, facilitando así su desarrollo integral, en correspondencia con la visión estratégica; brindando el soporte requerido en la búsqueda de satisfacción de necesidades y expectativas de clientes internos y externos.

Este departamento está conformado por 17 personas que dan soporte a la infraestructura tecnológica de la Organización; a partir de la segmentación de actividades según diferentes necesidades, se observa la distribución de personal en las siguientes áreas, lo cual indica la relevancia que tiene el área de desarrollo teniendo en cuenta que buena parte de las aplicaciones de software han sido construidas internamente:

- Mesa de Ayuda (2).
- Área de Operaciones (2).
- Área de Infraestructura (2).
- Área de Desarrollo (9).
- Área de Producción (2).

Las diferentes secciones del departamento están en contacto continuo con los usuarios finales de las diferentes soluciones tecnológicas, generando frecuentemente nuevas versiones de estas soluciones que permiten mejorar el servicio prestado por el departamento y la organización, y cumplir con los requisitos de ley a los que está sujeta; además se pone a disposición nuevas soluciones que apoyan y mejoran la operatividad y desempeño de la organización.

Con el propósito de dar una orientación a los integrantes del Departamento de informática, este último estableció los siguientes objetivos específicos:

- Mantener una plataforma tecnológica vigente de acuerdo con las condiciones del entorno y las necesidades de la organización en estudio.
- Mejorar la capacidad de respuesta del Departamento de Informática para atender necesidades y solicitudes relacionadas con aspectos de Desarrollo de Software e Infraestructura en términos cuantitativos y cualitativos.
- Desarrollar, implantar, mantener y soportar los sistemas de información necesarios para la organización.
- Promover y proporcionar capacitación sobre aspectos tecnológicos que permitan una mayor eficiencia del talento humano en la ejecución de sus tareas.
- Investigar, proponer, adaptar y aplicar buenas prácticas que permitan a la organización en estudio mantener un marco competitivo que contribuya a su supervivencia y crecimiento.
- Asegurar la disponibilidad y calidad de la información con los niveles apropiados de seguridad.
- Lograr un adecuado nivel de desempeño del personal vinculado al Departamento de Informática, mediante la distribución de funciones y responsabilidades, acorde con el perfil de cada cargo.

3.2. Proceso Software actual

Dado que el proyecto de mejora será ejecutado dentro del Proceso Software, es importante establecer el contexto de este, con el fin de que el lector identifique el estado actual del mismo.

El Proceso Software definido por el Departamento de Informática, establece las fases de Requisitos, Diseño, Construcción, Pruebas y Liberación, las cuales se ejecutan secuencialmente y adoptan conceptos de las metodologías tradicionales. La principal característica de este Proceso Software, es que permite ejecutar nuevamente las fases previas, es decir, si los Analistas se encuentran en la fase de Diseño y requieren clarificar o analizar nuevamente ciertos requisitos, pueden devolverse a la fase de Requisitos para resolver los inconvenientes.

El disparador del proceso son la necesidades de los usuarios, donde ellos deben especificar los requisitos funcionales y no funcionales claramente en una plantilla definida por el Departamento Informática; después, los usuarios deben crear una solicitud en el sistema de gestión de requerimientos adjuntando la plantilla, para estas sean evaluadas por el Subgerente de Desarrollo o por el Comité de Informática, dependiendo del alcance de la solicitud.

Posteriormente, si la solicitud es aceptada, de inmediato se crea una instancia del Proceso Software para que sea atendida por el Departamento de Informática, el cual se muestra gráficamente en la figura 8. Al lado izquierdo de cada fase se encuentran los insumos que se requieren para su ejecución y en el lado derecho los productos generados por cada una de ellas.

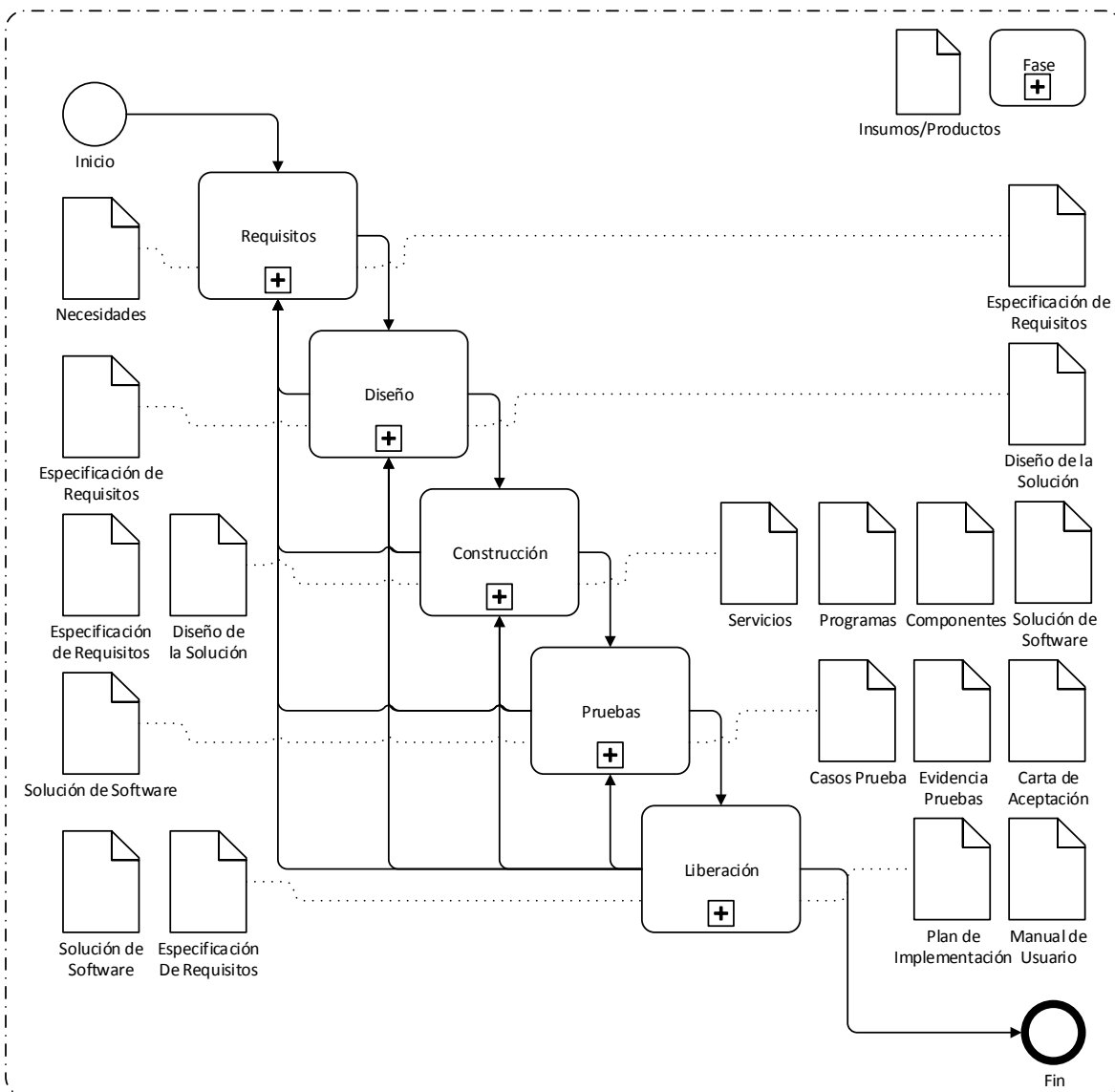


Figura 8. Modelo del Proceso Software (Construcción propia).

A continuación, en la tabla 4, se describe a gran escala las actividades de cada una de las fases que componen el proceso software:

Fase	Actividad	Descripción
Requisitos	Especificar Requisitos	El Dueño de Proceso y/o Interesados son los responsables de detallar sus necesidades en una plantilla definida por el Departamento de Informática, el cual se establece como el documento de especificación de requisitos.
	Analizar Requisitos	El documento de especificación de requisitos es leído por el Ingeniero de Requisitos, con el propósito de encontrar ambigüedades e

Fase	Actividad	Descripción
		identificar inconsistencias, las cuales son resueltas en compañía del Dueño de Proceso y/o Interesados. Este último es el responsable de realizar las modificaciones en el documento de especificación de requisitos, aclarando las ambigüedades y resolviendo las inconsistencias
Diseño	Diseñar Solución	El diseñador debe realizar el diseño de la solución, tomando como apoyo el documento de especificación de requisitos a través de la herramienta establecida por la Organización, lo cual permitirá visualizar con claridad la estructura de la solución.
	Verificar modelos	El diseñador par verifica que el diseño este cumpliendo los lineamientos establecidos en el Equipo de Desarrollo.
Construcción	Codificar solución	El Desarrollador debe escribir el código con los estándares de programación establecidos para el lenguaje en el cual desarrolla, según las consideraciones del diseño de la solución.
	Pruebas Unitarias	El Desarrollador debe realizar pruebas unitarias de lo construido para disminuir la cantidad de errores en la fase de pruebas.
	Refactorizar código	El Desarrollador debe realizar una revisión del código y analizar si éste se puede construir de una forma más óptima, pero debe garantizar que el funcionamiento no cambie.
Pruebas	Realizar pruebas de humo	El Desarrollador debe realizar las primeras pruebas sobre la solución de software a modo de exploración.
	Elaborar set de pruebas	El Tester debe construir el documento de set de pruebas, tomando como referencia la plantilla definida por el Departamento de Informática, el cual contiene los aspectos que el usuario va a probar.
	Ejecutar set de pruebas	El Tester debe ejecutar cada aspecto definido en el documento de set de pruebas y marcar si fue exitoso o no. Si el aspecto probado no fue exitoso, debe agregar una descripción del por qué no lo fue.
	Reportar resultado de las pruebas	El Tester debe reportar los resultados obtenidos después de ejecutar el set de pruebas; además el Tester debe adjuntar el documento de set de pruebas en el sistema de gestión de requerimientos.
Liberación	Construir plan de implantación	El Desarrollador debe construir el Plan de implantación, el cual contiene las actividades

Fase	Actividad	Descripción
		que se deben de llevar a cabo para que la solución de software sea implementada en el ambiente de producción.
	Verificar plan de implantación	El Liberador debe ejecutar una lista de chequeo sobre el Plan de implantación, con el propósito de verificar su completitud.
	Actualizar manual de usuario	El Ingeniero de Procesos debe actualizar los manuales de usuario que fueron modificados por la solución de software construida.
	Actualizar mapa de procesos	El Ingeniero de Procesos debe actualizar el mapa de procesos, siempre y cuando se modifique o se cree un proceso o procedimiento.

Tabla 4. Actividades del Proceso Software (Construcción Propia).

Como se describió anteriormente, el Departamento de Informática cuenta con un proceso definido, pero este carece de actividades transversales que lo soporten para que sea más eficiente, proactivo y predecible.

Después de analizar cada una de las fases que lo conforman, el Equipo de trabajo identificó las siguientes debilidades:

- Inmadurez del Proceso de Ingeniería de Requisitos.
- No se tiene una adecuada planeación de los Proyectos de Software.
- Inmadurez en la Gestión de Proyectos de Software.
- Inmadurez en la estimación de los Proyectos de Software.
- No se tiene establecido actividades de Gestión de la Configuración.
- No se tiene una adecuada estrategia para la Gestión del Conocimiento.
- No se tiene ambientes adecuados para la ejecución de las Pruebas.
- No se tiene definido actividades de aseguramiento de calidad.

3.3. Problemáticas

A partir de las debilidades identificadas, el Equipo de Desarrollo toma la iniciativa de solucionar los inconvenientes relacionados con la Ingeniería de Requisitos, dado que los requisitos son la materia prima o la base para el desarrollo de software, permitiendo determinar el alcance del problema y las funcionales a construir, lo cual es el insumo principal para la Planeación y Ejecución del proyecto [4].

Se realizó un análisis interno por medio de la observación, con el propósito de conocer cómo los integrantes del Equipo de Desarrollo obtienen las necesidades de los usuarios, especifican, validan, verifican y gestionan los requisitos, permitiendo identificar dificultades que afectan negativamente la productividad del

Equipo de trabajo y la satisfacción de los usuarios. Los problemas identificados fueron los siguientes:

- **Problemática 1:** Los usuarios son los responsables de especificar los requisitos funcionales y no funcionales, el Analista solo se limita a construir lo especificado en la plantilla de solicitud.
- **Problemática 2:** Los analistas no realizan actividades relacionadas con elicitación, verificación y validación de requisitos.
- **Problemática 3:** Alcanzar un acuerdo entre los analistas y usuarios, cuando este último solicita cambios en los requisitos es complicado, debido a que demanda mucho tiempo y esfuerzo por parte de los analistas, puesto que se desconoce el impacto que este cambio genera dentro del proyecto.
- **Problemática 4:** Los usuarios no hacen uso de funcionalidades que ofrecen los sistemas de información actuales, las cuales fueron desarrolladas por el Área de Desarrollo según las necesidades de los mismos.
- **Problemática 5:** Los usuarios no están satisfechos con las soluciones entregadas por el Área de Desarrollo, dado que no satisfacen totalmente sus necesidades.
- **Problemática 6:** Los usuarios no se encuentran satisfechos, debido a que las soluciones no se entregan en los tiempos pactados ni con el alcance establecido.
- **Problemática 7:** Después de realizar la entrega de la solución a los usuarios, inmediatamente éstas reciben 4 solicitudes de cambios en promedio, cada solicitud de cambio requiere entre de ocho (8) y veinticuatro (24) horas adicionales para su análisis e implementación.
- **Problemática 8:** El índice de satisfacción de los usuarios es tan solo del 45%, el cual es calculado a partir de la oportunidad y calidad de la soluciones entregadas por el área de desarrollo.

Este sondeo inicial fue el punto de partida para concientizar al equipo de trabajo sobre la situación actual y para conseguir el apoyo de los directivos, y de este modo iniciar un programa de mejora con urgencia que permitiera obtener resultados tangibles, visibles y a corto plazo, dado que si esta situación no mejoraba pronto, era probable que los procesos de negocio de la Organización se vieran afectados por no encontrar en el Departamento de Informática un aliado estratégico que ayudará a aumentar la competitividad e innovación del negocio.

3.4. Restricciones Organizacionales

La Organización estableció para este proyecto las siguientes restricciones:

3.4.1. Restricciones de Tiempo

- Dentro de la jornada laboral solo se destinó 24 horas semanales para la ejecución de las actividades del proyecto, así:
- 2 horas de reunión con el comité directivo.
- 2 horas de seguimiento a las actividades y avance del proyecto.
- 16 horas para la ejecución de las actividades planeadas.
- 4 horas de gestión y generación de informes de avance.
- El líder del proyecto podía hacer uso de hasta 16 horas semanales, fuera de la jornada laboral para hacer actividades de investigación de buenas prácticas, procesos, metodologías de análisis, diagnóstico y evaluación, y planes de acción, ya sea individualmente o con ayuda del consultor.
- La ejecución del proyecto no debía tomar más de 12 meses.

3.4.2. Restricciones de Horario

- Todas las actividades del proyecto se realizaron dentro de la jornada laboral de los participantes, es decir, de Lunes a Viernes entre las 8:00 am y las 12:30 pm, y las 2:00 pm y las 6:00 pm; y los días sábados entre las 8:00 am y las 1:00 pm, en los casos en que se considere necesario.
- Las reuniones de seguimiento o grupos primarios se debían realizar los días lunes de cada semana a las 9:00 am, en caso de ser un día festivo se trasladaba para el día martes de la semana en curso y a la misma hora.

3.4.3. Restricciones de Personal

Para la ejecución del proyecto se disponía del siguiente personal de la Organización:

- 2 Analistas de desarrollo.
- 1 Analista de infraestructura.
- 1 Analista líder.
- 1 Analista de procesos.
- 1 Auditor de sistemas.
- 1 Líder de proyecto.

3.4.4. Restricciones de Seguridad Local

- Solo se podía hacer uso de la información de la organización dentro de las instalaciones de la misma.
- Se debió garantizar la confidencialidad de la información generada y obtenida.
- Los entregables generados se debían mantener en formato digital con carácter confidencial.
- Los entregables oficiales del proyecto se debían entregar en formato PDF, los archivos originales en formato .doc debían ser entregados al área de procesos para su archivamiento.

3.4.5. Restricciones de Recursos

Los implementos o materiales para la ejecución del proyecto, fueron aquellos asignados a cada participante para la ejecución de sus actividades diarias dentro de la organización.

3.5. Limitaciones

En este punto se describen las limitaciones a las que se enfrentó el programa de mejoramiento dentro de la Organización:

- **Conocimiento:** Las personas que conforman el equipo de Desarrollo poseían conocimientos heterogéneos, esto no era del todo malo, pero si este grupo de personas realizaban las mismas actividades, esto ya no era tan positivo, dado que existían personas que tienen un nivel superior de conocimiento en comparación con el grupo, pero también existían personas con un nivel muy bajo, por tal motivo, el conocimiento del Equipo se consideró un limitante del Proceso Software, dado que era complicado nivelar estos conocimientos dentro del grupo y no era posible alcanzar el propósito de desarrollar productos con calidad, independiente de la persona que lo realice.
- **Cultura:** Las personas de la Organización tenían como concepción la siguiente frase: “Déjelo como está... Toda la vida ha funcionado así y si así lo definieron fue por algo”. Dado esto, las personas no se atrevían a investigar o intentar hacer cosas nuevas.
- **Resistencia al cambio:** Al inicio del proyecto existían personas que se encontraban en una zona de confort, por tal motivo eran reacios al cambio, dado que era muy probable que modificaran sus conocimientos, habilidades o comportamientos.

- **Autonomía:** El Equipo de Desarrollo no era autónomo en sus decisiones que toma con respecto al Proceso Software, dado que estas debían ser aceptadas por el Área de producción y el Subgerente de Desarrollo.

4. Definición del Marco de Trabajo para la Mejora

En este capítulo se detallan los modelos utilizados para llevar a cabo el programa de mejora; se seleccionó un modelo de referencia y un modelo de evaluación, debido a que ellos hacen parte de los componentes primarios de un programa de mejora y son los responsables de establecer un punto de referencia y definir un método de evaluación entre la práctica actual y marco de referencia respectivamente.

Además, se adoptó un modelo de mejora que sirviera de guía para realizar la mejora al proceso de requisitos. En la figura 9, se muestra los principales componentes del programa de mejora y como estos interactúan entre sí.

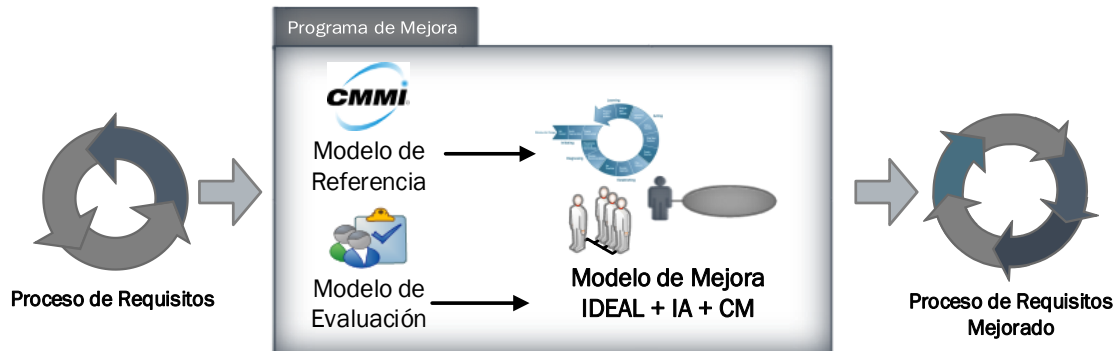


Figura 9. Componentes del programa de mejora (Construcción propia).

A continuación se presenta cada uno de los componentes que se integraron en este marco de trabajo.

4.1. Modelo de Referencia

Como ya fue presentado, CMMI-DEV (Capability Maturity Model Integration – Development) es un modelo de referencia internacional que busca evaluar las áreas de proceso involucradas en el desarrollo de software, a partir de buenas prácticas identificadas por expertos en cada área del proceso. CMMI – DEV presenta un esquema definido y estructurado con el cual se define el nivel de capacidad de la organización en cada área del proceso y el nivel de madurez de la misma en cuanto al desarrollo de software general.

Se ha seleccionado CMMI – DEV 1.3 como el modelo de referencia para el diagnóstico de la capacidad, dado que la estructura de áreas de proceso, prácticas y subprácticas definidas por CMMI – DEV, representa una excelente alternativa

para evaluar el estado del proceso software en la Organización, pues es posible individualizar cada una de las áreas de proceso para hacer un diagnóstico objetivo del proceso definido en la organización, dado que el propósito de CMMI-DEV es ayudar a las organizaciones a mejorar sus procesos de desarrollo y de mantenimiento, tanto para los productos como para los servicios [7].

Dentro de las áreas definidas por CMMI, se encuentran “Desarrollo de Requisitos” y “Gestión de Requisitos”, las cuales fueron el punto de referencia para el desarrollo del programa de mejora.

- *“El propósito de la Gestión de Requisitos (**REQM**) es gestionar los requisitos de los productos y de los componentes del producto del proyecto, e identificar inconsistencias entre esos requisitos y los planes y productos de trabajo del proyecto” [7].*
- *“El propósito del Desarrollo de requisitos (**RD**) es producir y analizar los requisitos de cliente, de producto y de componente del producto” [7].*

Además, la propuesta de CMMI presenta para cada de área de proceso un conjunto de metas específicas y prácticas específicas que permiten identificar el nivel de ejecución o el nivel de capacidad asociado a las actividades propias de cada área de proceso, donde las metas específicas son requeridas para el área de proceso, mientras que las prácticas específicas son opcionales.

Para gestión de requisitos, se presenta una meta específica que contiene 5 prácticas específicas, las cuales recomiendan líneas de acción de la organización respecto a lo definido por CMMI.

SG 1 Gestionar los requisitos.

SP 1.1 Obtener una comprensión de los requisitos

SP 1.2 Obtener el compromiso sobre los requisitos.

SP 1.3 Gestionar los cambios de los requisitos.

SP 1.4 Mantener la trazabilidad bidireccional de los requisitos.

SP 1.5 Identificar las inconsistencias entre el trabajo del proyecto y los requisitos.

Mientras, para Desarrollo de Requisitos, se establecen 3 metas específicas que agrupan 10 prácticas específicas, y permiten estipular el nivel de madurez de la Organización en esta área de proceso.

SG 1 Desarrollar los requisitos del cliente.

SP 1.1 Obtener las necesidades.

SP 1.2 Desarrollar los requisitos del cliente.

SG 2 Desarrollar los requisitos del producto.

SP 2.1 Establecer requisitos del producto y de componentes del producto.

SP 2.2 Asignar los requisitos de componentes del producto.

SP 2.3 Identificar los requisitos de la interfaz.

SG 3 Analizar y validar los requisitos.

SP 3.1 Establecer los conceptos operativos y los escenarios.

SP 3.2 Establecer una definición de la funcionalidad requerida.

SP 3.3 Analizar los requisitos.

SP 3.4 Analizar los requisitos para alcanzar el equilibrio.

SP 3.5 Validar los requisitos.

Cada una de las prácticas presentadas, están asociadas a un conjunto de subprácticas, que corresponden a actividades más detalladas para alcanzar el objetivo de cada práctica y aportar a la consecución de cada una de las metas específicas.

4.2. Modelo de Evaluación

Los modelos de evaluación son vitales en los programas de mejora, puesto que permiten establecer el grado de madurez y capacidad del proceso actual con respecto a un marco de referencia, igualmente por medio de un análisis identificar las fortalezas y debilidades del proceso; y con base a estas debilidades se puede dar inicio a un programa de mejora.

Como ya fue mencionado, existen diversos modelos de evaluación que sirven de apoyo para determinar las fortalezas y oportunidades de mejora, entre los que se encuentran SCAMPI e ISO/IEC 15504.

Dado que el objetivo de la Organización no era obtener una valoración o una certificación con respecto a un marco de referencia, y además, se quería realizar un proceso guiado por los miembros de la organización, sin incurrir en costos de consultoría y focalizado en el proceso de requisitos, fueron descartados los modelos de evaluación formal y se decidió construir un instrumento propio de evaluación.

Se estableció como instrumento una encuesta cerrada, la cual es considerada un método de primer grado, debido a que permite establecer una relación directa entre el investigador, los encuestados y la colección de datos. La encuesta está conformada de preguntas cerradas; donde las preguntas cerradas limitan a los encuestados a seleccionar una única respuesta en un conjunto limitado de opciones, esto con el propósito de obtener datos cualitativos que faciliten el entendimiento de las fortalezas y debilidades respecto al área de proceso evaluada.

Para la evaluación del Proceso de Requisitos se decidió involucrar los usuarios, con el propósito de conocer su percepción sobre el proceso actual, debido a que ellos juegan un papel muy importante en este proceso, además, el Equipo de Desarrollo se autoevaluó con respecto al marco de referencia para establecer la brecha entre la practica actual y las practicas recomendadas por CMMI, esto con el fin de tener diferentes puntos de vista del problema y de este modo establecer oportunidades de mejora más articuladas a las necesidades de la Organización.

En primer lugar, para conocer la perspectiva de los usuarios se estableció una encuesta cerrada (Anexo 1), la cual fue dirigida a cada área del cuerpo administrativo de la Organización y tuvo como objetivo responder los siguientes interrogantes:

- **Interrogante 1:** ¿Cuál es el nivel de conocimiento del usuario en el proceso de requisitos?
- **Interrogante 2:** ¿Cuál es el grado de articulación al contexto de la organización con que los usuarios definen los requisitos?
- **Interrogante 3:** ¿Cuál es el grado de importancia que el usuario reconoce al proceso de requisitos?
- **Interrogante 4:** ¿Cuál es el grado de responsabilidad que el usuario tiene con respecto al proceso de requisitos?
- **Interrogante 5:** ¿Reconocen los usuarios oportunidades de mejora al proceso de requisitos?

Asimismo, para la autoevaluación del Equipo de Desarrollo se definió otra encuesta cerrada (Anexo 2), con el objetivo determinar la brecha entre las prácticas implementadas por el Departamento de Informática sobre la Gestión y Desarrollo de Requisitos en el Proceso Software y las prácticas sugeridas por el Modelo CMMI-DEV 1.3.

Este instrumento es totalmente estructurado, dado que las preguntas fueron planeadas, analizadas y evaluadas antes de la ejecución de las encuestas por expertos de la academia, igualmente, el instrumento adoptó conceptos de la encuesta “Current Requirements Practice Self-Assessment” formulada por Karl E. Wiegers [68], la cual está enfocada en el cómo se realizan las actividades de ingeniería de requisitos en la organización, donde cada respuesta permite clasificar las prácticas en insuficiente, débil, bueno o excelente.

La encuesta está conformada por 20 preguntas y con ellas se desea responder los siguientes interrogantes:

- **Interrogante 6:** ¿Cuál es el Nivel de adopción de la practicas sugeridas por CMMI-DEV 1.3 para el Desarrollo de Requisitos dentro de las actividades del proceso software definido por la Organización?
- **Interrogante 7:** ¿Cuál es el Nivel de adopción de la practicas sugeridas por CMMI-DEV 1.3 para la Gestión de Requisitos dentro de las actividades del proceso software definido por la Organización?
- **Interrogante 8:** ¿Cuáles son las prácticas menos utilizadas por la Organización sugeridas por CMMI-DEV 1.3 para la Ingeniería de Requisitos?

Para evaluar la adopción de las prácticas sugeridas por el modelo CMMI-DEV con respecto a las áreas de proceso “Desarrollo de Requisitos” y “Gestión de Requisitos”, se construyeron preguntas para cada práctica específica del Área de Proceso, donde las respuestas fueron definidas tomando como base las subpracticadas recomendadas en las prácticas específicas.

Por ejemplo, para evaluar la práctica específica “*SP 3.3 Analizar los Requisitos*”, CMMI-DEV recomienda las siguientes subpracticadas:

1. Analizar las necesidades, las expectativas, las restricciones y las interfaces externas de las partes interesadas para eliminar conflictos y para organizarlos en temas relacionados.
2. Analizar los requisitos para determinar si satisfacen los objetivos de los requisitos de nivel más alto.
3. Analizar los requisitos para asegurarse de que son completos, factibles, realizables y verificables.
4. Identificar los requisitos claves que tienen una fuerte influencia en el coste, calendario, funcionalidad, riesgos o rendimiento.
5. Identificar las medidas de rendimiento técnico que serán seguidas durante el esfuerzo de desarrollo.
6. Analizar los conceptos operativos y los escenarios para refinar las necesidades, las restricciones y las interfaces del cliente, y para descubrir nuevos requisitos.

En función de estas subpracticadas, se construyeron las siguientes preguntas en el Instrumento:

- Pregunta 17: ¿Cómo realizan el análisis de los requisitos para determinar si es viable su implementación?
- Pregunta 18: Si han decidido implementar los requisitos del producto, ¿Cómo analizan el impacto, el rendimiento técnico y las restricciones de los requisitos?

Donde la pregunta 17 abarca las subpracticadas 1, 2 y 3, mientras que la pregunta 18 agrupa las subpracticadas 4, 5 y 6; y las respuestas para cada pregunta se encuentran relacionadas directamente con las subpracticadas de la siguiente forma:

- a. No adopta ninguna subpracticada.
- b. Adopta la primera subpracticada.
- c. Adopta la primera y segunda subpracticada.
- d. Adopta la primera, segunda y tercera subpracticada.

Como se puede apreciar, las respuestas de las preguntas son incrementales, es decir, la respuesta "d" indica que se ha adoptado más subpracticadas con respecto a la respuesta "b" o "c", por tanto contribuyen en mayor medida en el nivel de adopción de la práctica específica (SP) evaluada y en consecuencia la Meta Específica (SG) tiene un mayor nivel.

En la tabla 5, se muestra la relación entre las preguntas del instrumento y las prácticas recomendadas por el CMMI-DEV.

Área de Proceso	Meta Específica	Práctica Específica	# Pregunta
Gestión de Requisitos (REQM).	SG 1 Gestionar los requisitos.	SP 1.1 Obtener una comprensión de los requisitos	1, 2, 3
		SP 1.2 Obtener el compromiso sobre los requisitos.	3
		SP 1.3 Gestionar los cambios de los requisitos.	4, 5
		SP 1.4 Mantener la trazabilidad bidireccional de los requisitos.	6, 7
		SP 1.5 Identificar las inconsistencias entre el trabajo del proyecto y los requisitos.	8
Desarrollo de Requisitos (RD).	SG 1 Desarrollar los requisitos del cliente.	SP 1.1 Obtener las necesidades.	9
		SP 1.2 Desarrollar los requisitos del cliente.	10
	SG 2 Desarrollar los requisitos del producto.	SP 2.1 Establecer requisitos del producto y de componentes del producto.	11
		SP 2.2 Asignar los requisitos de componentes del producto.	12
		SP 2.3 Identificar los requisitos de la interfaz.	13
	SG 3 Analizar y validar los requisitos.	SP 3.1 Establecer los conceptos operativos y los escenarios.	14
		SP 3.2 Establecer una definición de la	15, 16

Área de Proceso	Meta Específica	Práctica Específica	# Pregunta
		funcionalidad requerida.	
		SP 3.3 Analizar los requisitos.	17, 18
		SP 3.4 Analizar los requisitos para alcanzar el equilibrio.	19
		SP 3.5 Validar los requisitos.	20

Tabla 5. Áreas de proceso e instrumento de evaluación (Construcción propia).

4.3. Modelo de Mejora

Para lograr los objetivos del programa, fue necesario adaptar el modelo de mejora a las necesidades de la Organización, considerando las condiciones de contexto del caso de estudio; con el propósito de generar resultados tangibles y a corto plazo y de este modo sustentar la continuación del programa de mejora y mantener la motivación del equipo de trabajo.

Partiendo de las bondades que ofrece el modelo IDEAL, se complementó con casos de mejora e investigación – acción, con el fin de obtener resultados positivos a corto plazo; siempre partiendo de los principios de simplicidad y practicidad, esto con el propósito de analizar e intervenir diferentes problemáticas del proceso como problemas independientes, y de este modo obtener planes de acción más específicos e integrales según las particularidades de cada problemática y lograr que el modelo de mejora de procesos esté más acorde a las necesidades de la Organización, en el cual los protagonistas principales son los mismos actores de proceso.

En la figura 10 se sintetiza la manera como se hizo la integración de estos modelos, lo cual se puede resumir en las siguientes acciones:

- Se preservaron las fases generales del modelo IDEAL.
- Se definió un conjunto de metas medibles y alcanzables.
- Se obtuvo el respaldo de la dirección a través de un análisis financiero.
- Se estableció un equipo de trabajo colaborativo y participativo.
- Se realizó un diagnóstico desde la perspectiva del usuario y del Equipo de Desarrollo.
- Se estableció un conjunto de recomendaciones que abordaban las debilidades identificadas en la evaluación de la práctica actual.
- Se complementó la fase de Establecimiento con casos de mejora.
- Se complementó la fase de Acción con Investigación – Acción
- Se establecieron espacios de reflexión para el análisis de los resultados obtenidos a medida que se implementaban los casos de mejora.
- Se determinó un conjunto de actividades transversales que ayudaran a tener un mayor control sobre la ejecución del programa de mejora.

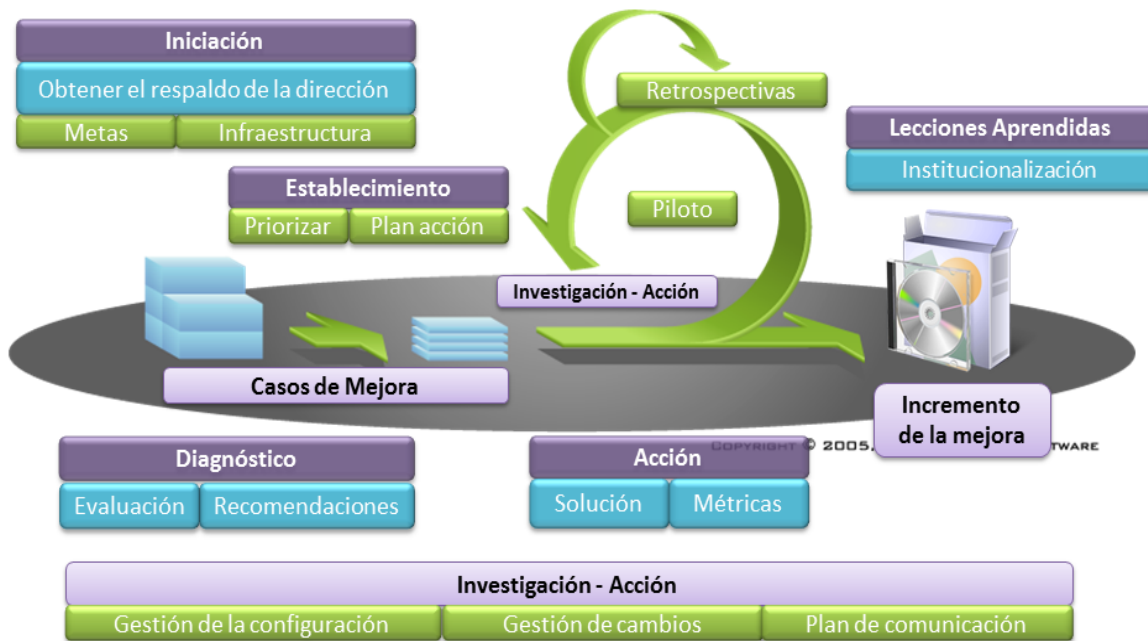


Figura 10. Ciclo de vida de mejora (Construcción propia).

Además, los casos de mejora fueron definidos y estructurados basándose en las fases que conforman el método de Investigación – Acción, dado que éstos permiten mejorar iterativamente los planes de acción, cuando estos no cumplen con los objetivos trazados. En la tabla 6, se muestra la estructura de un caso de mejora tomando como referente las fases de I-A, se indican las actividades que se deben desarrollar en cada de ellas con sus respectivos entregables y se establece en qué fase del modelo IDEAL se deben realizar dichas actividades.

Fase del Modelo IDEAL	Fase	Casos de mejora	
		Actividades	Entregable
Establecimiento	Planear	- Definir los objetivos del caso mejora.	- Objetivos.
		- Establecer un plan de acción que permite alcanzar dichos objetivos.	- Plan de acción.
Acción	Actuar	- Implementar el plan de acción definido en un proyecto piloto.	- Evidencia de los seguimientos.
	Observar	- Realizar seguimientos constantes a las actividades que se desarrollan.	
		- Planificar la observación.	- Metas de la observación.
		- Recoger evidencias que permitan evaluar el éxito de	- Evidencias de la

		las actividades o del plan de acción. - Registrar los eventos que suceden a medida que ejecuten el plan de acción.	observación.
	Reflexionar	- Analizar y discutir las evidencias registradas durante la observación entre los miembros del equipo. - Plantear alternativas para el plan de acción, si este no se llevó a cabo exitosamente. - Identificar lecciones aprendidas que permitan mejorar las definiciones de los siguientes casos de mejora. - Redefinir el plan de acción y continuar con otra ejecución del caso de mejora, si es necesario.	- Lecciones aprendidas. - Nuevo plan de acción (si se requiere otra iteración del caso de mejora).

Tabla 6. Estructura de un caso de mejora (Construcción propia).

Como se indicó anteriormente, las fases de Establecimiento y Acción fueron las fases que se complementaron con el concepto de casos de mejora y el método de Investigación – Acción respectivamente, puesto que son las responsables de definir los planes de mejora, priorizar las actividades, ejecutar la solución, establecer pilotos de prueba y desplegar la solución.

Asimismo, las fases de Iniciación, Diagnóstico y Aprendizaje también fueron consideradas dentro de la iniciativa; si bien no fueron impactadas con nuevas actividades, debido a que estas fases no están directamente relacionadas con la definición y ejecución de los planes de mejora, se hizo un esfuerzo importante para adecuar estas fases a las necesidades particulares de la organización, buscando construir solo los documentos estrictamente necesarios en cada fase y de este modo realizar un menor esfuerzo, evitando actividades que no agregan valor al programa de mejoramiento.

De otra parte, el método de Investigación – Acción estuvo presente en todas las fases del modelo IDEAL, esto es desde la fase de iniciación hasta la fase de aprendizaje, por medio de la observación y reflexión, con el propósito de identificar y corregir oportunamente anomalías dentro del programa, y así evitar desvíos que impidieran alcanzar las metas establecidas.

En síntesis, los cambios realizados sobre el modelo IDEAL para este caso de estudio, fueron los siguientes: Desde la perspectiva de los casos de mejora, se agregaron las siguientes actividades, **establecer casos de mejora** y **priorizar**

casos de mejora; desde la perspectiva de la Investigación-Acción, las actividades de **establecer posibles soluciones piloto** y **refinar solución** fueron enriquecidas, con el propósito de facilitar la comunicación entre los integrantes y brindar espacios de observación y reflexión sobre los planes de acción planteados, permitiendo enriquecerlos progresivamente a medida que se identifiquen situaciones que no se tienen contempladas inicialmente.

A partir del complemento del modelo IDEAL con Investigación - Acción, se estableció el siguiente esquema de trabajo para el programa de mejora (ver figura 11), en el cual se pueden observar los diferentes grupos que participaron en este proyecto: El grupo de investigación estuvo conformado por el autor de este trabajo, como estudiante de maestría y como asesora participó la tutora del presente trabajo; el grupo interdisciplinario estuvo compuesto por el Subgerente de Desarrollo, Jefe de Procesos, Analista de Auditoria y Líder de Proyecto; y el equipo de trabajo de la organización estuvo conformado por dos Analistas de Desarrollo, un Analista de Procesos y un Analista Líder.



Figura 11. Esquema de trabajo (Adaptación de [10]).

Estos grupos de trabajo fueron propuestos de este modo, con el objetivo de definir una estructura liviana de trabajo que facilite la comunicación y gestión entre los integrantes del proyecto.

En la tabla 7, se indican las principales actividades y se presenta la estrategia de intervención utilizada en cada una de las fases del Modelo:

Fase	Actividades	Estrategia de Intervención
Inicial	<ul style="list-style-type: none"> - Establecer Metas del Programa. - Obtener respaldo de la Dirección de la Organización. - Establecer los recursos necesarios para llevar a cabo el proyecto 	<ul style="list-style-type: none"> - Obtener el apoyo de la dirección por medio de presentaciones, por medio de los beneficios técnicos, operacionales y económicos. - Definir conjuntamente con la dirección las metas del programa de mejora haciendo uso Goal Question Metric (GQM) [63]. - Definir los objetivos del programa, la infraestructura requerida, los roles y las responsabilidades de los integrantes del Equipo de trabajo, contando solo con el personal interno.

<p>Diagnostico</p>	<ul style="list-style-type: none"> - Caracterizar la práctica actual. - Establecer la brecha entre lo actual y lo deseado. - Desarrollar recomendaciones de mejora. 	<ul style="list-style-type: none"> - Caracterizar el proceso utilizado por los empleados para la ingeniería de requisitos por medio de un modelo de procesos. - Establecer las prácticas recomendadas por el marco de referencia para el proceso de requisitos. - Realizar un estudio cuidadoso sobre las áreas de proceso REQM y RD. - Diseñar los instrumentos de evaluación (encuestas) para establecer la brecha entre la práctica actual y la recomendada por CMMI. Un instrumento para los usuarios y el otro para los desarrolladores. - Aplicar los instrumentos en cada grupo de trabajo. - Verificar, validar y consolidar los datos consignados en los instrumentos, esto con el fin de evitar desviaciones en los datos. - Identificar fortalezas y oportunidades de mejora del proceso de requisitos mediante un análisis exhaustivo de los datos. - Construir un conjunto de recomendaciones, las cuales deberán ser expuestas al Grupo Interdisciplinario, al Departamento de Informática y a la Dirección de la Organización.
<p>Establecimiento</p>	<ul style="list-style-type: none"> - Establecer prioridades de mejora. - Establecer propuesta de mejora (Casos de mejora). - Establecer casos de mejora (Casos de mejora). - Priorizar casos de mejora. - Establecer plan de acción. 	<ul style="list-style-type: none"> - Priorizar las recomendaciones realizadas en la fase anterior y como estas serán abordadas en el programa de mejora. - Definir una estrategia de mejora y un plan general del programa de mejora. - Definir los casos de mejora a partir de las recomendaciones y su prioridad. - Priorizar los casos de mejora definidos utilizando el método pair-wise [73]. - Definir los planes de acción que se deben de ejecutar para cada caso de mejora, donde se indica claramente el objetivo de esa mejora y cómo ésta contribuirá al mejoramiento del proceso de requisitos.
<p>Acción</p>	<ul style="list-style-type: none"> - Establecer solución. - Establecer posibles soluciones pilotos (I-A). - Refinar la solución (I-A). - Implementar solución. 	<ul style="list-style-type: none"> - Analizar las condiciones que se requieren en los proyectos para llevar a cabo el plan de acción de los casos de mejora. - Determinar los proyectos que servirán de pilotos para los casos de mejora prioritarios, donde éstos implementaron las acciones establecidas en el plan de acción.

		<ul style="list-style-type: none"> - Depurar los planes de acción cuando estos no satisfagan las recomendaciones establecidas, o no se ajuste a las necesidades del proceso, o no este alineada con las condiciones de la Organización, mediante el método de investigación – acción. - Realizar acompañamiento a la mejora por medio del método investigación – acción. - Institucionalizar las prácticas establecidas en el caso de mejora en compañía del área de procesos, cuando la ejecución del proyecto piloto finalice satisfactoriamente.
Aprendizaje	<ul style="list-style-type: none"> - Analizar y validar la mejora. - Establecer acciones futuras. 	<ul style="list-style-type: none"> - Realizar una reunión entre los integrantes del Equipo de trabajo y Grupo Interdisciplinario para identificar lecciones aprendidas, después de finalizar la ejecución de un caso de mejora. - Retroalimentar los planes de acción que no han sido ejecutados.

Tabla 7. Estrategia de intervención del caso de estudio (Construcción propia).

Luego de finalizar la ejecución de los planes de acción de cada caso de mejora, se procedió a realizar nuevamente las encuestas, con el propósito de determinar el impacto de la mejora en el proceso de requisitos y de este modo cuantificar el nuevo nivel de adopción, en el capítulo 6 se presenta el resultado de este análisis.

Además, debido a que el programa de mejoramiento del proceso software fue considerado como uno de los proyectos más importantes en la Organización, fue necesario establecer un proceso de gestión de la configuración, un proceso de gestión de cambios, un plan de comunicación y un plan de calidad, esto con el fin de tener un mayor control del proyecto y facilitar que el programa de mejora pudiera ser ejecutado sin mayores contratiempos.

A continuación se detallan cada uno de los procesos transversales o de soporte, los cuales fueron definidos tomando como referencia las prácticas recomendadas por el PMBOK [19].

4.3.1. Gestión de la configuración

En el plan de gestión de la configuración se definieron los elementos configurables, ¿dónde serán almacenados?, ¿cuáles requieren de un control de cambios? y ¿cómo se controlarán los cambios a estos elementos?

En la tabla 8 se describen los artefactos que conforman el conjunto de ítems de configuración, estos artefactos fueron seleccionados, debido a su relevancia para

la ejecución de las actividades del Equipo y de esta manera no afectar el desarrollo de las tareas planificadas.

Documento	Ubicación
Carta de Proyecto	/Info/MPS/Proyecto/Proyecto.doc
Solicitudes de Cambio	/Info /MPS/SolicitudesCambio/
Cronograma & Presupuesto	/Info /MPS/Proyecto/Cronograma.mpp
Bitácora de Proyecto	/Info /MPS/Proyecto/Bitacora.doc
Riesgos	/Info /MPS/Proyecto/Riesgos.xls
Gestión del Cambio	/Info /MPS/SolicitudesCambio/HistorialCambio.doc
Informes	/Info /MPS/Proyecto/Informes/
Entregables	/Info /MPS/Entregables/Inicial/ /Info /MPS/Entregables/Diagnostico/ /Info /MPS/Entregables/Establecimiento/ /Info /MPS/Entregables/Acción/ /Info /MPS/Entregables/Aprendizaje/
Actas	/Info /MPS/Proyecto/Actas/
Documentos	/Info /MPS/Proyecto/Referencia/
Calidad	/Info /MPS/Proyecto/Calidad/
Otros	/Info /MPS/Proyecto/Otros/

Tabla 8. Ítems de configuración (Construcción propia).

Los documentos fueron clasificados por la Organización basándose en el estándar internacional ISO 27001, el cual tiene por objeto garantizar que los controles implementados son adecuados en relación con la confidencialidad, integridad y disponibilidad de la información, además, serán gestionados por medio de la herramienta Microsoft Visual SourceSafe.

4.3.1.1. Gestión de cambios

Tomando como referencia las buenas prácticas de Gestión de Proyectos del PMI se estableció un sistema de gestión de la configuración acompañado de un control integrado de cambios, puesto que éste proporciona una manera normalizada, efectiva y eficiente de gestionar de forma centralizada los cambios y las líneas base aprobados dentro de un proyecto; por tanto se definió los siguientes lineamientos:

Las solicitudes de cambio se deben realizar por medio del sistema de gestión de cambios o reporte de incidencias definido en la organización, describiendo claramente el cambio requerido, el motivo del cambio, y la información de contacto del solicitante.

Estos cambios deben ser gestionados directamente por el líder del proyecto quien tras hacer un análisis de las características específicas de la solicitud de cambio y realizar el análisis de impacto correspondiente decidirá si puede aprobar o rechazar el cambio, o si lo debe redirigir al comité directivo.

Según las características de cada una de las solicitudes, el líder del proyecto las clasificará en una de las siguientes categorías:

- Planeación: Solicitudes relacionadas con la distribución de las actividades del Proceso, la generación de entregables, modificación de formatos, etc.
- Proyecto: Solicitudes relacionadas con el alcance del proyecto, su presupuesto, cronograma, la inclusión o separación de áreas de la organización interesadas en el proyecto, etc.
- Equipo: Conformación de equipos, distribución de actividades, distribución de materiales y/o implementos, etc.

Las solicitudes clasificadas en las categorías “Planeación” y/o “Equipo” deben ser resueltas por el líder del proyecto, quien notificará al solicitante la respuesta a su solicitud y gestionará en el equipo de trabajo la implementación del cambio solicitado. Si la solicitud es clasificadas como “Proyecto” deben ser dirigidas al comité directivo y/o al patrocinador para su aceptación o rechazo, anexando el análisis de impacto correspondiente; en este caso, el comité directivo y/o el patrocinador dará solución a la solicitud de cambio y notificará al líder del proyecto, y será este último el encargado de notificar al solicitante y gestionar en el equipo de trabajo su implementación, en caso de ser aprobada.

Las solicitudes de cambio pueden ser aprobadas o rechazadas en dos instancias:

1. Líder del proyecto: Cambios que tengan que ver con las actividades internas del proyecto y la distribución de actividades y recursos.
2. Comité directivo o patrocinador: Cambios concernientes al alcance del proyecto, presupuesto, cronograma general y demás cambios que no pueda responder directamente el líder del proyecto.

Todas las solicitudes deben ser atendidas en un plazo máximo de 8 días, tiempo en el cual el líder del proyecto debe analizar y clasificar la solicitud, y dar respuesta aquellas que le correspondan; si una solicitud debe ser atendida por el comité directivo y no es posible que este se reúna en el plazo de los 8 días, se debe consultar con patrocinador la respuesta más adecuada para la solicitud, ya sea aprobar o rechazar, o informar al solicitante la fecha en la que se dará respuesta a su solicitud (próxima reunión del comité directivo). Tras la aprobación o rechazo de una solicitud se debe registrar en el sistema de reporte de incidencias la respuesta a la solicitud y el plan a seguir; si la solicitud es clasificada como “Proyecto”, además del registro en el sistema de incidencias se debe notificar por medio escrito al solicitante incluyendo la siguiente información:

- Estado de la solicitud (“Aprobada”, “Rechazada”).
- Fecha de respuesta

- Si fue rechazada, el motivo del rechazo.
- Si fue aprobada, los pasos a seguir y la fecha en la que se realizará su implementación.
- Quien dio resolución a la solicitud (Comité directivo o líder de proyecto).

Cuando una solicitud es aprobada, se debe divulgar a los equipos de trabajo, las acciones a seguir para implementar el cambio, o como se va a reflejar este cambio en sus actividades diarias, y los responsables de su implementación en caso de ser necesario. Tras la implementación del cambio, se debe notificar al solicitante y al grupo interdisciplinario que el cambio ha sido implementado, además de la implementación del cambio, también se debe notificar las acciones que se llevaron a cabo y las consecuencias que se han identificado tras su implementación.

4.3.2. Plan de comunicación

Un plan de comunicación permitirá establecer un enfoque más eficaz y eficiente para comunicarse entre los interesados del proyecto. Una comunicación eficaz significa que la información se suministra en el formato adecuado, en el momento justo y con el impacto apropiado, mientras que una comunicación eficiente significa proporcionar únicamente la información necesaria [19].

Debido a esto, se estableció un plan de comunicaciones para determinar la información que necesitan los interesados del proyecto, ¿el cómo la van a obtener?, ¿quién será el responsable de suministrarla?, ¿cuándo la van a obtener? y ¿por cuál medio se proporcionará?, esto con el propósito de evitar algunos inconvenientes, como:

- Demoras en la entrega de información.
- Información confidencial dirigida a la audiencia equivocada.
- Falta de comunicación con algunos de los interesados involucrados.

De acuerdo a lo anterior, en la tabla 9 se muestra el plan de comunicación definido para el proyecto:

Origen	Destino	Plantilla / Formato	Periodicidad
Comité Directivo	Patrocinador	AVAN-001.rtf	Quincenal
Patrocinador	Comité Directivo	AVAN-002.rtf	Quincenal
Patrocinador	Líder de Proyecto	AVAN-003.rtf	Quincenal
Líder de Proyecto	Patrocinador	AVAN-004.rtf	Quincenal
Auditoría	Comité Directivo	AUD-001.rtf	Quincenal
Auditoría	Líder de Proyecto	AUD-002.rtf	Quincenal
Líder de Proyecto	Comité Directivo	AVAN-005.rtf	Quincenal
Comité Directivo	Líder de Proyecto	AVAN-006.rtf	Quincenal
Líder de Proyecto	Consultor	CO-INF001.rtf	Semanal

Origen	Destino	Plantilla / Formato	Periodicidad
Consultor	Líder de Proyecto	CO-INF002.rtf	Semanal
Líder de Proyecto	Equipo Desarrollo	GA-GC001.rtf	Semanal
Líder de Proyecto	Equipo Infraestructura	GA-GC002.rtf	Semanal
Equipo Desarrollo	Equipo Infraestructura	Formato Libre	Diario
Equipo Infraestructura	Equipo Desarrollo	Formato Libre	Diario

Tabla 9. Plan de comunicación (Construcción propia).

4.3.3. Plan de calidad

El plan de calidad tiene como objetivo garantizar la calidad de los artefactos construidos a lo largo del proyecto, mediante la identificación temprana de defectos. El plan de calidad contiene los artefactos que serán inspeccionados en el transcurso del proyecto, el tipo de artefacto, responsable de construirlo, responsable de inspeccionarlo, fecha de inspección y observaciones, para este se recomendó hacer uso del documento PC-PC001.rtf.

Todas las actividades del plan de calidad deben ser planificadas y organizadas al inicio del proyecto, puesto que se deben establecer cuando y quienes serán los responsables de hacer las inspecciones de calidad. Estas inspecciones serán ejecutadas por un integrante del área de calidad, con el objetivo de verificar la adherencia a los estándares y lineamientos establecidos por la organización, verificar completitud de los artefactos y sugerir oportunidades de mejora.

Este proceso de inspección está basado en reglas y se utilizarán las listas de chequeo para cada una de las fases del Modelo IDEAL, estas listas de chequeo están disponibles en la intranet de la organización en el repositorio de calidad.

Una vez realizadas las inspecciones, el analista de calidad deberá dejar registro de las no conformidades en el documento PC-NC001.rtf, las cuales serán clasificadas de la siguiente manera:

- Acción correctiva: Error actual.
- Acción preventiva: Actualmente no es un error pero a futuro tiene una probabilidad alta de serlo.
- Acción de Mejora: Consideración o sugerencia.

Es importante aclarar que antes de avanzar a la siguiente fase del ciclo de mejora, el Equipo de trabajo debió corregir las no conformidades reportadas, puesto que se realizaba seguimiento continuo a dichas no conformidades, es decir, si una inspección de una fase tiene asociada por lo menos una no conformidad correctiva, el líder del proyecto no pudo continuar con la siguiente fase hasta que esta no sea corregida, esto con el fin de garantizar un avance del proyecto sin traumatismo y complejidades.

5. Experiencia de mejora del proceso de requisitos

En este capítulo se comparte la experiencia obtenida siguiendo las fases sugeridas por el modelo IDEAL y aplicando propuesta del marco de trabajo presentada en el capítulo anterior. En primer lugar, se muestran las metas del programa de mejora y como se establecieron los roles y responsabilidades. En segundo lugar, se describe como se llevó a cabo la evaluación del proceso a través de los interrogantes planteados, los cuales ayudarán a determinar el estado actual del proceso. En tercer lugar, se comparte la experiencia de trabajo haciendo uso de los casos de mejora e Investigación – Acción en las fases de Establecimiento y Acción. Y por último, se comparte las lecciones aprendidas obtenidas después de mejorar el proceso de Requisitos.

5.1. Fase de Inicio

Un aspecto importante de esta fase es lograr el respaldo de la dirección de la Organización, y para lograrlo, el líder de desarrollo en compañía con el subgerente de desarrollo realizaron una presentación donde dejaban en evidencia los beneficios que se pueden lograr después de mejorar el proceso de requisitos, además, presentaron un análisis financiero, indicando el ROI (Retorno/Inversión) que se puede obtener 6 meses después de haber finalizado la ejecución del programa de mejora.

Después de discutir dichos argumentos entre los asistentes a la reunión, la dirección de la Organización decidió apoyar el proyecto de mejora, indicando que su ejecución se debe convertir en un modelo a seguir para futuros proyectos.

Una vez obtenido el respaldo de la dirección, se procedió a construir las metas del programa, basándose en la metodología GQM, la cual establece que una meta debe estar compuesta por:

- Objeto: Es el objeto de estudio, puede ser Productos, Procesos o Recursos.
- Propósito: Indica cual es el propósito de la meta, por ejemplo: Comprender, predecir, planificar, controlar, comparar o evaluar algún aspecto de la calidad o productividad de un objeto.
- Parámetro: Indica el parámetro de medición de la meta.
- Punto de Vista: Determina quién o quiénes son los interesados de los resultados, debido a que la perspectiva puede variar dependiendo del interesado.

Las metas establecidas para el programa de mejora, fueron:

- Aumentar en un 30% la efectividad y eficacia del Equipo de trabajo en la ejecución del Proceso de Requisitos en un periodo de tiempo de 6 meses después de la ejecución del proyecto, desde el punto de vista del subgerente de desarrollo.
- Formular un Proceso de Requisitos que permita aumentar en un 30% la satisfacción, aceptación y confianza de los usuarios involucrados, a partir de la definición de requisitos del sistema con mayor calidad, después del punto de vista de los interesados.
- Reducir en un 30% los reprocesos generados por una mala aplicación de los conceptos de la Ingeniería de Requisitos, dentro de los 6 meses próximos a la ejecución del proyecto, por medio de la capacitación y concienciación a los miembros del Equipo de Desarrollo en los aspectos de la Ingeniería de Requisitos, desde el punto de vista del subgerente de desarrollo.

Posteriormente se continuó con la definición de los grupos de trabajo con sus respectivas responsabilidades, en la tabla 10, se indican las responsabilidades de los roles involucrados en el esquema de trabajo. En esta definición se decidió incluir los roles de auditor y patrocinador, dado que estos son transversales al programa de mejora y son mencionados durante la experiencia de mejora.

Rol	Responsabilidad
Grupo interdisciplinario	Planear y controlar la ejecución de las actividades.
	Gestionar la ejecución de las actividades que incluyen la participación de otras áreas de la organización o clientes.
	Hacer seguimiento al cumplimiento del cronograma del proyecto y la generación de los entregables definidos.
	Presentar al patrocinador del proyecto los informes de avance y los entregables generados en cada fase.
	Controlar la distribución adecuada del presupuesto del proyecto.
	Ejecutar los planes de comunicación con los diferentes usuarios e interesados en el proyecto.
	Planear, gestionar y controlar los riesgos asociados al proyecto y vigilar la ejecución oportuna de los planes de contingencia asociados a cada riesgo identificado.
	Autorizar y gestionar la ejecución de la propuesta establecida por los investigadores.
	Autorizar, gestionar y divulgar los planes de institucionalización de las mejoras definidas en las propuestas.
Equipo de trabajo	Ejecutar las actividades de la fase de iniciación del proyecto en el que se encuentre asignado.
	Ejecutar las actividades de la fase de diagnóstico del proyecto en el

Rol	Responsabilidad
	que se encuentre asignado.
	Ejecutar las actividades de la fase de establecimiento del proyecto en el que se encuentre asignado.
	Ejecutar las actividades de la fase de acción del proyecto en el que se encuentre asignado.
	Ejecutar las actividades de la fase de pruebas del proyecto en el que se encuentre asignado.
	Ejecutar las actividades de la fase de despliegue del proyecto en el que se encuentre asignado.
	Generar los entregables asignados al proyecto.
	Generar los artefactos necesarios para construir los entregables definidos para cada fase del proyecto.
	Ejecutar los planes de acción definidos para cada caso de mejora.
	Ejecutar las actividades que le sean accionadas en cada uno de los proyectos piloto.
	Responder las encuestas generadas para obtener el diagnóstico del proceso de requisitos.
	Participar de la implementación de los casos de mejora donde se encuentren directa e indirectamente involucrados.
	Investigador + Asesor
Discutir sobre la mejora de proceso para definir una propuesta que permita mejorar las debilidades del proceso actual.	
Modificar métodos de mejora existente para obtener resultados a corto plazo de manera ágil.	
Realizar análisis, observación y reflexión de la problemática, con el propósito de identificar fortalezas y debilidades del Equipo de trabajo.	
El asesor realizará varianzas empíricas a las propuestas planteadas por los investigadores cuando lo considere pertinente para el éxito del proyecto.	
Patrocinador	Gestionar la asignación del presupuesto necesario para el proyecto.
	Hacer seguimiento y vigilancia a la ejecución y avance del proyecto.
	Promover el proyecto en las diferentes áreas de la organización.
Auditor	Garantizar que las actividades y entregables generados por el proyecto estén acordes a las políticas de calidad definidas en la Organización.
	Garantizar que se generen todos los entregables definidos para el proyecto.
	Verificar el avance del proyecto y generar los informes respectivos.

Tabla 10. Asignación de responsabilidades (Construcción propia).

5.2. Fase de Diagnóstico

Para la ejecución de las encuestas, el grupo interdisciplinario realizó varias actividades preliminares, las cuales permitieron evaluar la efectividad de los

instrumentos y garantizar que toda la población encuestada tuviera un conocimiento homogéneo respecto a la práctica evaluada, así:

1. Se contextualizó a la población encuestada, sobre las prácticas de la Ingeniería de Requisitos.
2. Se realizó un piloto para cada instrumento con un integrante de la muestra para mejorarlo.
3. Se resolvieron inquietudes respecto a las preguntas de los instrumentos.
4. Se aclaró a los participantes, que las encuestas debían ser resuelta con base a las prácticas actuales de la Organización, y no a las que consideraran se deberían implementar.

5.2.1. Evaluación desde la perspectiva del usuario

La encuesta fue diligenciada por 17 personas que hacen parte del cuerpo administrativo de la organización, donde se encuentran Directores, Jefes, Coordinadores y Analistas.

Para dar respuesta al **interrogante 1**, se analizaron las respuestas de las preguntas 1, 2 y 3 del instrumento 1, permitiendo establecer que el 94.12% de los encuestados conocen el significado de un requisito y cuál es su principal objetivo en el proceso de requisitos, como se muestra en la figura 12.

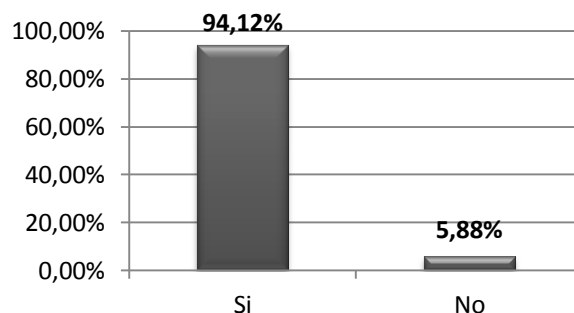


Figura 12. Significado de un requisito y su objetivo (Construcción propia).

Por otro lado, mientras que el 58.82% de los encuestados siempre tienen claro el resultado que esperan al finalizar la implementación de requisito y el 41.18% lo saben casi siempre (figura 14), sólo el 29.41% siempre conocen la información que se necesita para la definición de un buen requisito (figura 13), lo que conllevaría a un riesgo de alto impacto en el proceso de requisitos, debido a que se pueden generar ambigüedades o requisitos incompletos, afectando considerablemente el nivel de reprocesos.

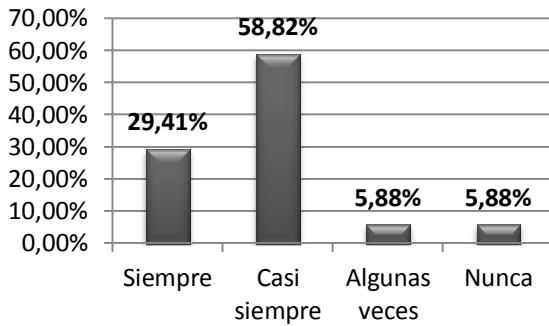


Figura 13. Conocimiento de información (Construcción propia).

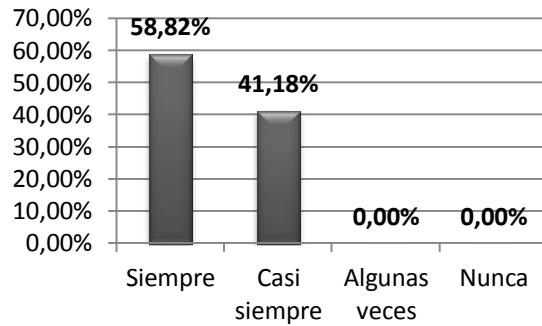


Figura 14. Conocimiento de los resultados (Construcción propia).

En conclusión, generalmente los usuarios saben con certeza el resultado final que desea, sin embargo, no siempre conocen de la información que se requiere para lograr dicho resultado.

Con respecto al **interrogante 2**, se evaluó el grado de articulación de los requisitos planteados por los usuarios con relación al contexto de la Organización, es decir, si los requisitos definidos tienen en cuenta los lineamientos establecidos por los entes de control y por la Organización.

Por medio de la información suministrada, se identificó que sólo el 29,41% de los encuestados no tiene en cuenta las restricciones de los procesos de negocio y los lineamientos legales para definir los requisitos, igualmente, tampoco involucran a otras áreas de negocio que pueden verse afectadas por estos requisitos. Véase Figura 15.

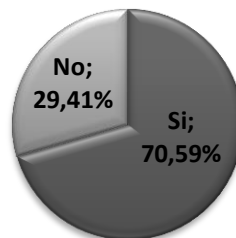


Figura 15. Involucran a otras áreas de negocio (Construcción propia).

Igualmente, por medio de este análisis, se pudo establecer que los encuestados casi siempre conocen los procesos de negocio y políticas definidas por la Organización al momento de definir nuevos requisitos, no obstante una pequeña porción de la población considera que no tienen un buen conocimiento sobre estos aspectos, como se puede ver en las figuras 16 y 17.

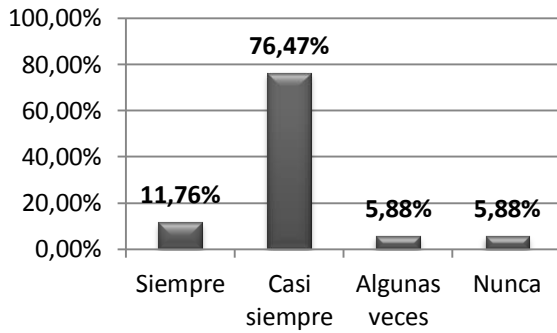


Figura 16. Conocimiento de Procesos (Construcción propia).

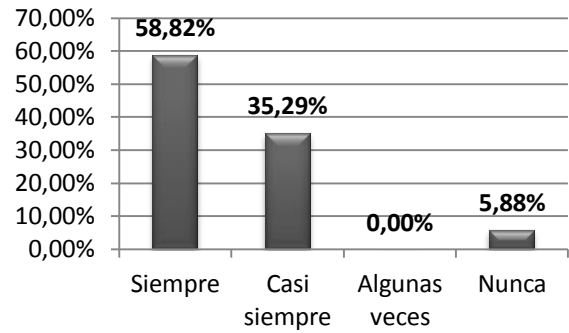


Figura 17. Conocimiento de Políticas (Construcción propia).

Por tanto, se puede afirmar que el grado de articulación de los requisitos con el contexto de la organización es aceptable, debido a que más del 70% de los encuestados tienen en cuenta los procesos y políticas establecidos por la Organización.

De otro lado, se analizaron las preguntas que tienen relación con actividades que involucran a los usuarios en el proceso de requisitos, con el fin de determinar el grado de importancia que tiene el proceso de requisitos para ellos y de esto modo responder el **interrogante 3**.

La mayoría de los usuarios piensan que un mal requisito puede afectar considerablemente el desarrollo del proyecto (figura 18), debido a esto, el 70,59% de los encuestados son conscientes de los recursos que se requieren y saben el impacto que éste puede generar al interior del proyecto (figura 19), además, el 64,71% especifican claramente la reglas de validación que se deben realizar para el desarrollo de los requisitos (figura 20), sin embargo, es de anotar que el 70,59% no prioriza los requisitos (figura 21).

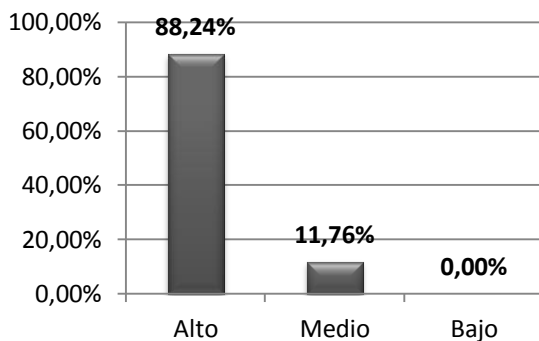


Figura 18. Impacto de un mal requisito (Construcción propia).

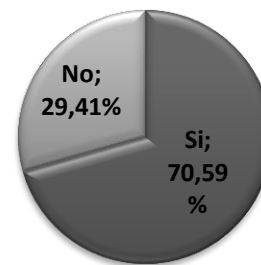


Figura 19. Conocimiento del impacto y recursos (Construcción propia).

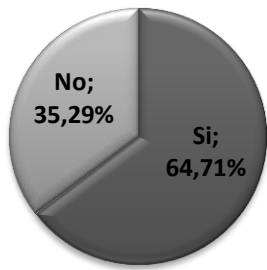


Figura 20. Establecen validaciones (Construcción propia).

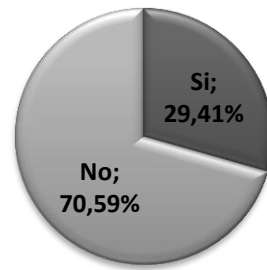


Figura 21. Priorizan los requisitos (Construcción propia).

Gracias a este análisis, se pudo establecer que más de las dos terceras partes de los usuarios consideran que el proceso de requisitos es importante para el éxito de un proyecto, aunque la mayoría no prioricen los requisitos.

Las preguntas asociadas a la responsabilidad que se debe definir dentro del proceso requisitos, permitieron responder el **interrogante 4**, donde el 64.71% consideran que los responsables de definir los requisitos deben ser los dueños de procesos (figura 22), pero la gran mayoría manifiesta que lo deben realizar en compañía de los analistas de desarrollo y procesos, puesto que más del 90% los usuarios consideran que la intervención del analista de desarrollo facilitaría la toma de requisitos (figura 23), mientras que más del 80% les parece pertinente la cooperación de los analista de procesos en la definición de requisitos (figura 24).

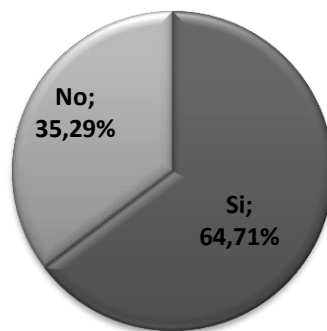


Figura 22. Responsable de los de requisitos (Construcción propia).

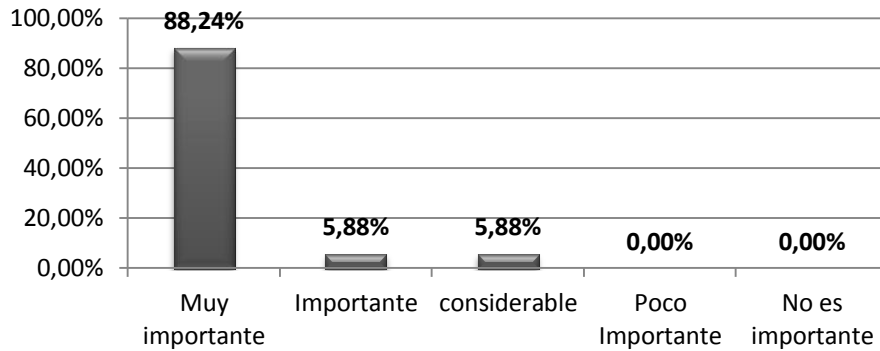


Figura 23. Importancia del Analista de Desarrollo (Construcción propia).

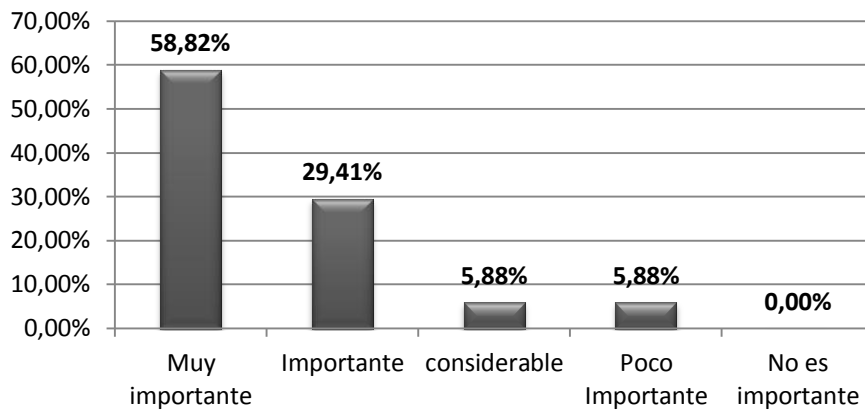


Figura 24. Importancia del Analista de Procesos (Construcción propia).

Para finalizar la evaluación desde la perspectiva de los usuarios, las preguntas asociadas a la percepción que tienen los usuarios respecto al Proceso de Requisitos, permitieron dar respuesta al **interrogante 5**, donde se logró establecer que en promedio, el 80% de los encuestados consideran que se deben hacer mejoras a las practicas actuales que se ejecutan dentro de la Organización para poder lograr un enfoque simple. Figura 25.

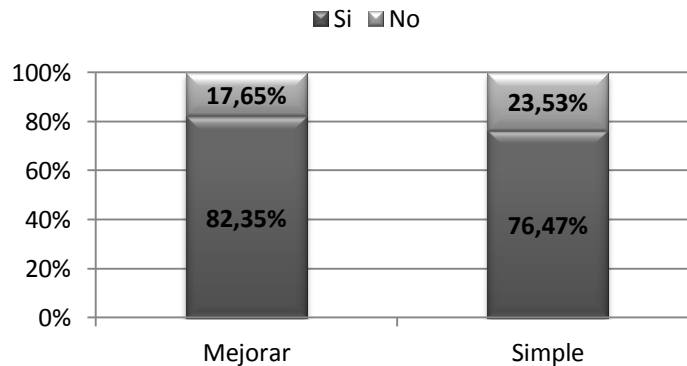


Figura 25. Mejorar el proceso de requisitos (Construcción propia).

5.2.2. Evaluación desde la perspectiva de los Analistas de Desarrollo

La encuesta fue realizada a siete (7) integrantes del Área de Desarrollo y de Producción, con el fin de cubrir las áreas que están involucradas directa e indirectamente en el proceso de requisitos. En la tabla 11 se describen los cargos, nivel jerárquico y la cantidad de años trabajados en la Organización de cada uno de los encuestados, por acuerdos de confidencialidad no se muestra información personal de los encuestados, lo cual no afecta el diagnóstico.

	Cargo	Nivel Jerárquico	Antigüedad
Encuestado 1	Analista de Producción	Analista	2 años
Encuestado 2	Analista de Desarrollo	Analista	4 años
Encuestado 3	Analista de Desarrollo	Analista	4 meses
Encuestado 4	Analista de Desarrollo	Analista	15 años
Encuestado 5	Analista de Desarrollo	Analista	4 años
Encuestado 6	Analista de Desarrollo	Analista	6 años
Encuestado 7	Líder de Desarrollo	Coordinador	5 años

Tabla 11. Información de los encuestados (Construcción propia).

Una vez finalizadas las actividades preliminares, se continuó con la ejecución de la encuesta. A continuación se presentan los resultados obtenidos tras ejecutar, tabular y analizar la encuesta aplicada a los participantes, y a través de estos resultados se logró resolver los interrogantes planteados como base para la generación del instrumento.

Así, de acuerdo al **interrogante 6** y a la información suministrada por los encuestados, se pudo identificar el nivel de ejecución de las prácticas, el nivel de cumplimiento de las metas específicas, y finalmente concluir con el nivel de capacidad respecto al desarrollo de requisitos.

En la figura 26, se puede observar las prácticas que tienen una mejor o menor porcentaje de ejecución, según la opinión de los analistas encuestados, entre las que se pueden destacar *SP1.1 Obtener las necesidades* y *SP 3.1 Establecer los conceptos operativos y los escenarios*, por tener un porcentaje mayor de ejecución sobre las demás.

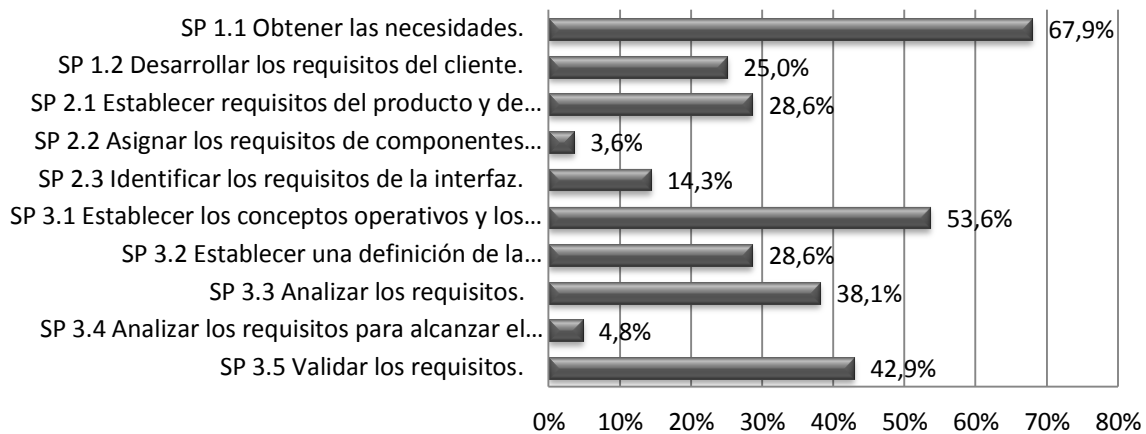


Figura 26. Cumplimientos de las prácticas de RD (Construcción propia).

Respecto a las metas específicas sugeridas para el área de Desarrollo de Requisitos, se presentan porcentajes bajos de cumplimiento respecto a estas metas, estando todos por debajo del 50% de cumplimiento, como se aprecia en la figura 27.; donde la meta *SG 1. Desarrollar los requisitos del cliente*, se destaca sobre las demás metas, por tener un porcentaje de cumplimiento superior.

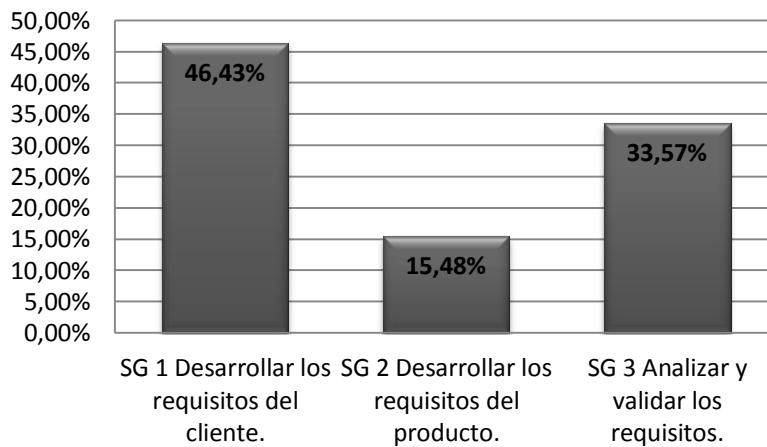


Figura 27. Cumplimiento de metas específicas de RD (Construcción propia).

Finalmente, se puede concluir a partir de porcentaje de cumplimiento de las metas específicas, que la organización solo ejecuta en un 31,83% de sus actividades y procesos, prácticas apropiadas para el desarrollo de los requisitos, lo cual es coherente con el estado actual y los problemáticas identificadas. Ver figura 28.

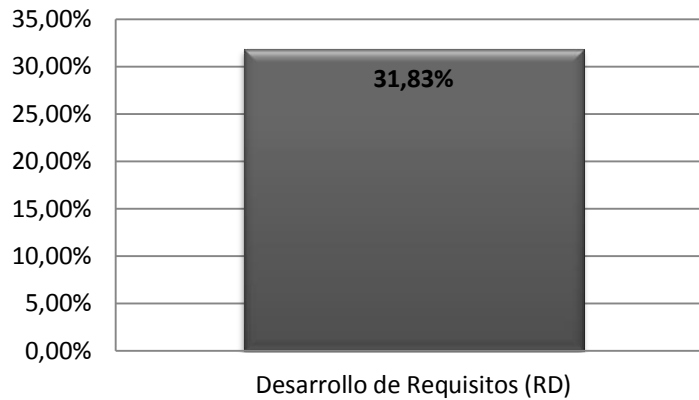


Figura 28. Nivel de adopción de prácticas de RD (Construcción propia).

Para dar respuesta al **Interrogante 7** fue necesario evaluar las respuestas de varias preguntas, para establecer el nivel de ejecución prácticas y cumplimiento de las metas específicas. En la figura 29, se muestra el porcentaje de ejecución de cada una de las prácticas específicas recomendadas por el área de proceso Gestión de Requisitos.

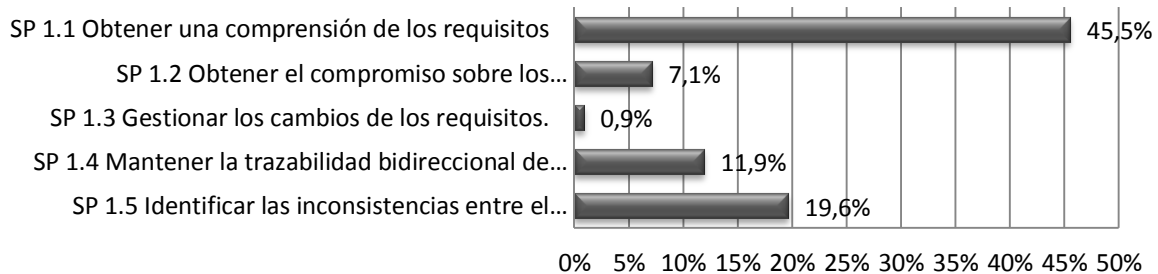


Figura 29. Cumplimientos de las prácticas de REQM (Construcción propia).

Desde el punto de vista de los analistas, se puede determinar que la práctica específica más adoptada es: *SP1.1 Obtener una comprensión de los requisitos*, pero uno de los aspectos más alarmante de este análisis, es que la práctica específica menos utilizada es *SP1.3 Gestionar los cambios a los requisitos*, lo cual es concordante con la problemática de reprocesos expuesta anteriormente.

Con relación a las metas específicas, se identificó que el nivel de adopción de *SG1 Gestionar los requisitos* es muy bajo, puesto que el porcentaje de ejecución de las prácticas sugeridas por esta meta específica, es solo del 17%. Ver figura 30.

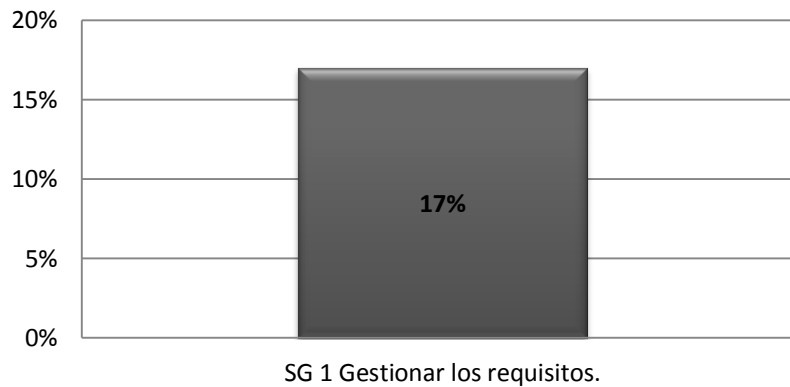


Figura 30. Cumplimiento de metas específicas de REQM (Construcción propia).

Por tanto, el porcentaje de ejecución de las prácticas establecidas por el CMMI en el área de proceso “Gestión de Requisitos (REQM)”, es tan sólo del 17%, justificando de este modo la situación que padece actualmente el Departamento de Informática.

De otro lado, se identificaron las seis (6) prácticas menos utilizadas por la Organización en su proceso de requisitos, esto con el fin de dar respuesta al **interrogante 8**. En la tabla 12, se muestran las prácticas identificadas:

Area de Proceso	Práctica Específica	Porcentaje Adopción
REQM	SP 1.3 Gestionar los cambios de los requisitos.	0,9%
RD	SP 2.2 Asignar los requisitos de componentes del producto.	3,6%
RD	SP 3.4 Analizar los requisitos para alcanzar el equilibrio.	4,8%
REQM	SP 1.2 Obtener el compromiso sobre los requisitos	7,1%
REQM	SP 1.4 Mantener la trazabilidad bidireccional de los requisitos.	11,9%
RD	SP 2.3 Identificar los requisitos de la interfaz.	14,3%

Tabla 12. Prácticas específicas menos adoptadas (Construcción propia).

Después de analizar el alcance de dichas prácticas, se puede concluir que éstas tienen relación directa con las problemáticas identificadas en el capítulo 3, por ejemplo:

- Las prácticas “*SP 1.3 Gestionar los cambios de los requisitos*” y “*SP1.4 Mantener la trazabilidad bidireccional de los requisitos*” tiene relación con la problemática “*Alcanzar un acuerdo entre los analistas y usuarios, cuando este último solicita cambios en los requisitos es complicado, debido a que demanda mucho tiempo y esfuerzo por parte de los analistas, puesto que se desconoce el impacto que este cambio genera dentro del proyecto*”, además,

- Las prácticas “SP 3.4 Analizar los requisitos para alcanzar el equilibrio” y “SP 2.3 Identificar los requisitos de la interfaz” se encuentran relacionadas con la problemática “Los usuarios no están satisfechos con las soluciones entregadas por el Área de Desarrollo, dado que no satisfacen totalmente sus necesidades”.

Lo que indica que los planes de mejora deben estar direccionados en aumentar el nivel de adopción estas prácticas en el proceso de requisitos. Para contrarrestar los efectos negativos de dichas problemáticas.

5.2.3. Recomendaciones

Después de analizar la información suministrada por los usuarios y los analistas, los investigadores y el asesor procedieron a construir un conjunto de recomendaciones que permitieran lograr las metas establecidas para el programa de mejora. Estas recomendaciones fueron enfocadas a los aspectos del saber y del hacer, debido a que estos son fundamentales en los programa de mejora para lograr un equilibrio entre el conocimiento y los quehaceres, como se muestra en la figura 31.

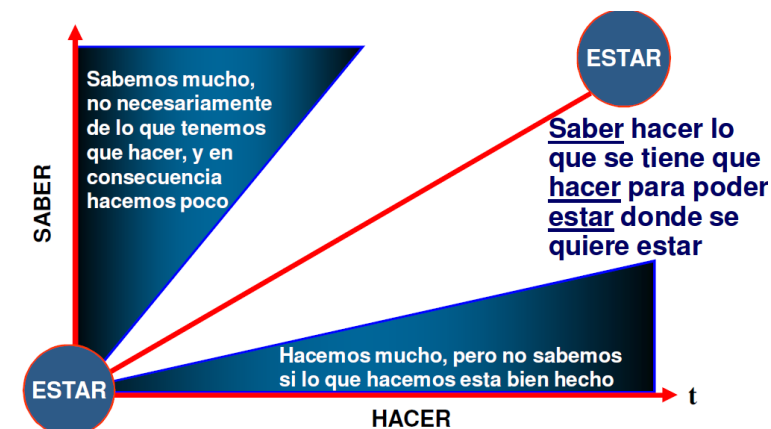


Figura 31. Saber vs Hacer [36].

Una vez construidas las recomendaciones, estas fueron compartidas al Grupo Interdisciplinario y al Equipo de trabajo para su análisis y evaluación. A continuación se listan las recomendaciones sugeridas por los investigadores y el asesor:

5.2.3.1. Recomendaciones de conocimiento

1. Los analistas de desarrollo deben desarrollar habilidades para la Elicitación de Requisitos.
2. Los analistas de desarrollo deben adquirir conocimiento para el análisis de negocio.

3. Los analistas de desarrollo deben aprender a especificar los requisitos a través de los casos de uso.
4. Los analistas de desarrollo deben saber cómo establecer el impacto que puede generar un cambio en los requisitos.
5. Los analistas de desarrollo deben ilustrarse en como validar y verificar los requisitos.

5.2.3.2. Recomendaciones de prácticas

1. Los analista de desarrollo deben ser los responsables de elicitar lo requisitos en compañía de los dueños de procesos o interesados.
2. Los analistas de desarrollo debe hacer uso de técnicas para la elicitación de requisitos, por ejemplo: Lluvia de ideas, entrevistas, encuestas, análisis grupales, entre otras.
3. Los analistas de desarrollo deben construir un documento donde se especifiquen los requisitos funcionales y no funcionales por medio del modelo de casos de mejora.
4. El Equipo de trabajo debe establecer un método de priorización de los requisitos funcionales.
5. Los analistas de desarrollo deben utilizar una herramienta CASE que les permita construir los diagramas UML que requieren para su análisis.
6. El líder del equipo debe definir un conjunto de atributos de calidad para cada uno de los entregables del proceso de requisitos, esto con el fin de establecer un método objetivo de verificación.
7. Los usuarios deben ser los responsables de validar los requisitos funcionales y no funcionales establecidos en la especificación de requisitos.
8. El líder del equipo debe ser el responsable de gestionar cualquier cambio sobre los requisitos definidos, esto con el propósito de tener un adecuado control sobre los requisitos.
9. El líder del equipo debe hacer uso de la matriz de trazabilidad, con el fin de tener una herramienta que le facilite realizar el análisis de impacto de un cambio sobre los requisitos.
10. El Equipo de trabajo debe implementar métricas que les permita medir el estado del proceso de requisitos.
11. El Equipo de trabajo debe realizar modificaciones al sistema de gestión de requerimientos – SGR para que este quede articulado con el proceso de requisitos.
12. El Equipo de trabajo debe emplear de un método de estimación para establecer el esfuerzo que se requiere para desarrollar los requisitos, entre los que se recomiendan se encuentran Juicio de Experto y PERT.
13. El Grupo interdisciplinario debe procurar construir un Proceso colaborativo, ágil y adaptable.
14. El Equipo de trabajo debe establecer una estrategia de Gestión de Configuración para los artefactos construidos en el Proceso de Requisitos.

Las anteriores recomendaciones fueron discutidas con el asesor y el grupo interdisciplinario y después de su aprobación, se procedió a dar inicio a la fase de establecimiento.

5.3. Fase de Establecimiento

Siguiendo las recomendaciones de conocimiento, el Grupo Interdisciplinario en compañía del Asesor, decidieron capacitar al Equipo de Desarrollo en los aspectos de la Ingeniería de Requisitos, esto con el fin de nivelar el conocimiento de todos los integrantes.

Este curso fue dictado por una reconocida academia de formación de la ciudad. El curso tenía establecido una duración de 40 horas teórico/práctico y abordaba los siguientes temas:

1. Descripción general del Análisis de Negocios.
2. Análisis corporativo.
3. Planeación y gestión de requisitos.
4. Obtención de requisitos.
5. Análisis y documentación de requisitos.
6. Comunicación de los requisitos.
7. Evaluación y Validación de la solución.
8. Fundamentos del Analista de Negocios (BA).
9. Casos de Estudio y Ejercicios.

Posteriormente, debido al alto costo de implementar todas las recomendaciones del hacer al tiempo, el asesor en compañía del grupo de trabajo definió los casos de mejora que se requieren para llevar a cabo las recomendaciones sugeridas por los investigadores. La agrupación de las recomendaciones se realizó teniendo en cuenta el dominio a la que pertenece en la Ingeniería de Requisitos, es decir, a Elicitación, Análisis, Especificación, Validación, Verificación y Gestión de Requisitos.

Cada caso de mejora agrupa un conjunto de recomendaciones, y es considerado una unidad de trabajo que no puede superar los 20 días hábiles de esfuerzo, esto con el fin de establecer metas a corto plazo y cuantificables.

Se hace uso de los conceptos de SCRUM [44] para facilitar el entendimiento de cómo se trabajó el concepto de los casos de mejora en esta experiencia: El *product backlog* está compuesto por casos de mejora, los cuales son establecidos a partir de las dificultades, debilidades, necesidades y problemáticas identificadas en la evaluación del proceso y son abordadas en las recomendaciones, mientras que el *Sprint Backlog* solo contiene los casos de mejora que serán trabajados en una iteración en particular con sus respectivas actividades, como se muestra en la figura 32.

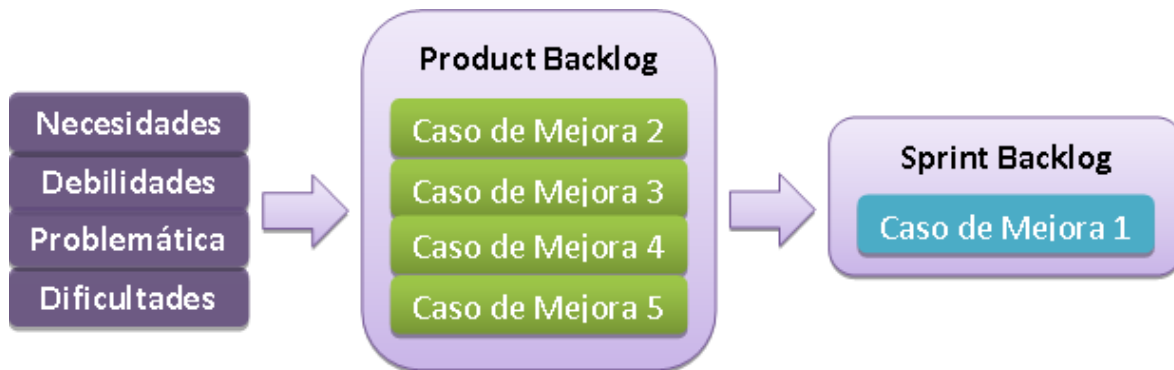


Figura 32. Analogía Casos de mejora y SCRUM (Construcción propia).

Surgieron 7 casos de mejora, los cuales se muestran en la tabla 13; cada caso de mejora tenía el propósito de articular dentro del proceso, las recomendaciones que surgieron de la fase de diagnóstico.

Casos de Mejora	Recomendaciones
Caso de Mejora 1	<ul style="list-style-type: none"> - Los analista de desarrollo deben ser los responsables de elicitar lo requisitos en compañía de los dueños de procesos o interesados. - Los analistas de desarrollo debe hacer uso de técnicas para la elicitation de requisitos, por ejemplo: Lluvia de ideas, entrevistas, encuestas, análisis grupales, entre otras.
Caso de Mejora 2	<ul style="list-style-type: none"> - Los analistas de desarrollo deben construir un documento donde se especifiquen los requisitos funcionales y no funcionales por medio del modelo de casos de uso. - Los analistas de desarrollo deben utilizar una herramienta CASE que les permita construir los diagramas UML que requieren para su análisis.
Caso de Mejora 3	<ul style="list-style-type: none"> - Los usuarios deben ser los responsables de validar los requisitos funcionales y no funcionales identificados. - El líder del equipo debe definir un conjunto de atributos de calidad de los entregables del proceso de requisitos, esto con el fin de establecer un método objetivo de verificación.
Caso de Mejora 4	El Equipo de trabajo debe establecer un método de priorización de los requisitos funcionales.
Caso de Mejora 5	<ul style="list-style-type: none"> - El líder del equipo debe ser el responsable de gestionar cualquier cambio sobre los requisitos definidos, esto con el propósito de tener un adecuado control sobre los requisitos. - El líder del equipo debe hacer uso de la matriz de trazabilidad, con el fin de tener una herramienta que le facilite realizar el análisis de impacto de un cambio sobre los requisitos.
Caso de Mejora 6	- El Equipo de trabajo debe emplear de un método de estimación para establecer el esfuerzo que se requiere para desarrollar los requisitos, entre los que se recomiendan se encuentran Juicio de Experto y PERT.

Casos de Mejora	Recomendaciones
	<ul style="list-style-type: none"> - El Grupo interdisciplinario debe procurar construir un Proceso colaborativo, ágil y adaptable. - El Equipo de trabajo debe establecer una estrategia de Gestión de Configuración para los artefactos construidos en el Proceso de Requisitos.
Caso de Mejora 7	<ul style="list-style-type: none"> - El Equipo de trabajo debe implementar métricas que les permita medir el estado del proceso de requisitos. - El Equipo de trabajo debe realizar modificaciones al sistema de gestión de requerimientos – SGR para que este quede articulado con el proceso de requisitos.

Tabla 13. Relación casos de mejora y recomendaciones (Construcción propia).

Como se identificaron siete casos de mejora, fue necesario hacer uso de la técnica **pair-wise** [73] (basado en el proceso de jerarquía analítico), para establecer la prioridad relativa de cada uno con respecto a las necesidades de la Organización.

Para establecer la prioridad, el grupo interdisciplinario definió como criterios de priorización, el riesgo de fracaso en su implementación, el esfuerzo que se requiere hacer para llevar a cabo el caso de mejora y la importancia de éste. El riesgo de fracaso es la probabilidad de que en la implementación del caso de mejora suceda algo inesperado que afecte negativamente el plan de acción. En la tabla 14, se muestra el peso asignado a cada criterio.

Criterios	Comparación			Normalización			Prioridad
	Riesgo	Esfuerzo	Importancia	Riesgo	Esfuerzo	Importancia	
Riesgo	1,00	2,00	0,33	0,22	0,33	0,20	25,19%
Esfuerzo	0,50	1,00	0,33	0,11	0,17	0,20	15,93%
Importancia	3,00	3,00	1,00	0,67	0,50	0,60	58,89%
Total	4,50	6,00	1,67	1,00	1,00	1,00	100%

Tabla 14. Criterios de priorización (Construcción propia).

A continuación se indican los pasos que se realizaron para determinar la prioridad relativa de cada caso de mejora:

1. Se realizó una comparación de cada caso de mejora contra el resto de los casos de mejora identificados con el criterio: ¿Cuál es el más importante entre ellos? Ver tabla 15.

	CM1	CM2	CM3	CM4	CM5	CM6	CM7
CM1	1,00	4,00	2,00	4,00	1,00	2,00	2,00
CM2	0,25	1,00	0,50	2,00	0,50	1,00	2,00
CM3	0,50	2,00	1,00	2,00	0,33	1,00	2,00

	CM1	CM2	CM3	CM4	CM5	CM6	CM7
CM4	0,25	0,50	0,50	1,00	0,33	2,00	2,00
CM5	1,00	2,00	3,00	3,00	1,00	3,00	3,00
CM6	0,50	1,00	1,00	0,50	0,33	1,00	2,00
CM7	0,50	0,50	0,50	0,50	0,33	0,50	1,00
TOTAL	4,00	11,00	8,50	13,00	3,83	10,50	14,00

Tabla 15. Comparación casos de mejora por Importancia (Construcción propia).

2. Se normalizó las comparaciones, con el propósito de determinar la prioridad de acuerdo a la importancia (Columna PRD). Ver tabla 16.

	CM1	CM2	CM3	CM4	CM5	CM6	CM7	PRD
CM1	0,25	0,36	0,24	0,31	0,26	0,19	0,14	25,02%
CM2	0,06	0,09	0,06	0,15	0,13	0,10	0,14	10,50%
CM3	0,13	0,18	0,12	0,15	0,09	0,10	0,14	12,91%
CM4	0,06	0,05	0,06	0,08	0,09	0,19	0,14	9,49%
CM5	0,25	0,18	0,35	0,23	0,26	0,29	0,21	25,38%
CM6	0,13	0,09	0,12	0,04	0,09	0,10	0,14	9,95%
CM7	0,13	0,05	0,06	0,04	0,09	0,05	0,07	6,76%
TOTAL	1,00	1,00	1,00	1,00	1,00	1,00	1,00	100%

Tabla 16. Prioridad de casos de mejora por Importancia (Construcción propia).

3. Se comparó cada caso de mejora entre sí, con el criterio: ¿Cuál caso de mejora requiere menor esfuerzo para su implementación? Ver tabla 17.

	CM1	CM2	CM3	CM4	CM5	CM6	CM7
CM1	1,00	0,50	0,50	0,50	3,00	4,00	5,00
CM2	2,00	1,00	0,50	0,50	3,00	3,00	4,00
CM3	2,00	2,00	1,00	0,50	3,00	3,00	3,00
CM4	2,00	2,00	2,00	1,00	3,00	4,00	4,00
CM5	0,33	0,33	0,33	0,33	1,00	2,00	3,00
CM6	0,25	0,33	0,33	0,25	0,25	1,00	2,00
CM7	0,20	0,25	0,33	0,25	0,33	0,50	1,00
TOTAL	7,78	6,42	5,00	3,33	13,58	17,50	22,00

Tabla 17. Comparación de casos de mejora por esfuerzo (Construcción propia).

4. Se hizo la normalización de cada comparación, para determinar la prioridad de acuerdo al esfuerzo (Columna PRD). Ver tabla 18.

	CM1	CM2	CM3	CM4	CM5	CM6	CM7	PRD
CM1	0,13	0,08	0,10	0,15	0,22	0,23	0,23	16%

	CM1	CM2	CM3	CM4	CM5	CM6	CM7	PRD
CM2	0,26	0,16	0,10	0,15	0,22	0,17	0,18	17,68%
CM3	0,26	0,31	0,20	0,15	0,22	0,17	0,14	20,68%
CM4	0,26	0,31	0,40	0,30	0,22	0,23	0,18	27,15%
CM5	0,04	0,05	0,07	0,10	0,07	0,11	0,14	8,35%
CM6	0,03	0,05	0,07	0,08	0,02	0,06	0,09	5,60%
CM7	0,03	0,04	0,07	0,08	0,02	0,03	0,05	4,34%
TOTAL	1,00	1,00	1,00	1,00	1,00	1,00	1,00	100%

Tabla 18. Prioridad de casos de mejora por esfuerzo (Construcción propia).

5. Los casos de mejora fueron comparados entre ellos, con el criterio: ¿cuál tiene menos riesgo en su implementación? Ver tabla 19.

	CM1	CM2	CM3	CM4	CM5	CM6	CM7
CM1	1,00	0,33	0,50	0,33	1,00	2,00	2,00
CM2	3,00	1,00	2,00	0,50	3,00	3,00	3,00
CM3	2,00	0,50	1,00	0,50	3,00	3,00	3,00
CM4	3,00	2,00	2,00	1,00	3,00	3,00	3,00
CM5	1,00	0,33	0,33	0,33	1,00	2,00	2,00
CM6	0,50	0,33	0,33	0,33	0,50	1,00	1,00
CM7	0,50	0,33	0,33	0,33	0,50	1,00	1,00
TOTAL	11,00	4,82	6,49	3,32	12,00	15,00	15,00

Tabla 19. Comparación de casos de mejora por riesgo (Construcción propia).

6. Las comparaciones fueron normalizadas, con el propósito de establecer la prioridad con respecto al riesgo (Columna PRD). Ver tabla 20.

	CM1	CM2	CM3	CM4	CM5	CM6	CM7	PRD
CM1	0,09	0,07	0,08	0,10	0,08	0,13	0,13	9,80%
CM2	0,27	0,21	0,31	0,15	0,25	0,20	0,20	22,70%
CM3	0,18	0,10	0,15	0,15	0,25	0,20	0,20	17,72%
CM4	0,27	0,41	0,31	0,30	0,25	0,20	0,20	27,81%
CM5	0,09	0,07	0,05	0,10	0,08	0,13	0,13	9,42%
CM6	0,05	0,07	0,05	0,10	0,04	0,07	0,07	6,27%
CM7	0,05	0,07	0,05	0,10	0,04	0,07	0,07	6,27%
TOTAL	1,00	1,00	1,00	1,00	1,00	1,00	1,00	100%

Tabla 20. Prioridad de casos de mejora por riesgo (Construcción propia).

Por último, se realizó una ponderación de cada prioridad dada para cada uno de los criterios definidos (Tabla 21), el cual es distribuido según el peso de cada criterio, con el fin de determinar la prioridad relativa de los casos de mejora:

	Comparación			Normalización			Prioridad
	Importancia	Esfuerzo	Riesgo	Importancia	Esfuerzo	Riesgo	
CM1	25,02%	16,19%	9,80%	14,73%	2,58%	2,47%	19,78%
CM5	25,38%	8,35%	9,42%	14,95%	1,33%	2,37%	18,65%
CM4	9,49%	27,15%	27,81%	5,59%	4,32%	7,01%	16,92%
CM3	12,91%	20,68%	17,72%	7,60%	3,29%	4,46%	15,36%
CM2	10,50%	17,68%	22,70%	6,18%	2,82%	5,72%	14,71%
CM6	9,95%	5,60%	6,27%	5,86%	0,89%	1,58%	8,33%
CM7	6,76%	4,34%	6,27%	3,98%	0,69%	1,58%	6,25%
Total	100%	100%	100%	59%	16%	25%	100%

Tabla 21. Prioridad relativa por caso de mejora (Construcción propia).

Después de establecer la prioridad de cada caso de mejora, el asesor procedió a definir el plan de acción, el cual consistía en definir para cada caso de mejora, los recursos necesarios para su ejecución, es decir, establecer las personas que harán parte del equipo de trabajo del caso de mejora y la infraestructura requerida para el éxito del mismo. Además, en el plan de acción se estableció el cronograma de trabajo, los hitos de control, responsabilidades, mecanismos de seguimiento y el análisis de riesgo de acuerdo a las prioridades establecidas.

Una vez definido el plan de acción del programa de mejoramiento, éste fue presentado al Grupo Interdisciplinario para su evaluación, el cual, después de una semana de deliberaciones aprobó dicho plan. En este punto no se brinda más detalle sobre el plan de acción, dado que se requiere dar cumplimiento al acuerdo de confidencialidad establecido con la Organización.

5.4. Fase de Acción

Luego de obtener la aprobación del plan de mejora, el Asesor y el Grupo Interdisciplinario evaluaron los proyectos de software en curso y los consignados en el Plan Estratégico del Departamento de Informática, esto con el fin de establecer los proyectos que serán los pilotos para cada caso de mejora. Es importante anotar que en esa evaluación, se determinó que cada caso de mejora será probado en un proyecto piloto, es decir, se seleccionó un proyecto para cada caso de mejora, y de este modo disminuir el riesgo de deserción o resistencia al cambio por parte del Equipo de trabajo o usuarios.

Posteriormente a la definición de los proyectos pilotos para cada caso de mejora, el Asesor procedió a definir el plan de trabajo de cada caso de mejora según el orden de prioridad definido en la fase de establecimiento, donde se establecen los talleres, seguimientos, herramientas, conocimientos y responsabilidades que se requieren para el éxito de éste.

Antes de que los participantes iniciaran oficialmente el proyecto, se realizaba una reunión de aproximadamente de 5 horas, donde el Asesor explicaba el alcance del caso de mejora, indicando claramente las prácticas que se van a adoptar y como se deben llevar a cabo, apoyándose en talleres que le permitiera simular las prácticas que iban a emplear en el transcurso de proyecto.

Después de dar inicio al proyecto piloto, el Asesor observaba constantemente como el Equipo de trabajo del proyecto llevaba a cabo la solución de mejora desde una perspectiva externa, es decir, sin interactuar con ellos y de este modo recopilar información que le permitiera realizar mejoras al plan de trabajo del caso de mejora.

Además, el Asesor se reunía con el Equipo de trabajo del proyecto los martes de cada semana por un espacio de 2 horas, para aclarar dudas conceptuales o prácticas, y también para conocer los inconvenientes que se les presentaron en el transcurso de la semana. En esta reunión, el Asesor en compañía del Equipo de trabajo reflexionaban sobre el plan de trabajo propuesto para el caso de mejora y evaluaban alternativas para ajustar la solución inicialmente planteada a los conocimientos, herramientas y habilidades que disponía la Organización.

En todos los casos de mejora se replanteó su plan de acción propuesto, dado que la solución inicial no satisfacía completamente todas las recomendaciones que agrupaba, de esta manera fue necesario realizar como mínimo otra iteración del caso de mejora en otro proyecto piloto o en el mismo para mejorar el resultado obtenido después de llevar a cabo el plan inicial.

En la tabla 22, se muestra el número de redefiniciones que se realizaron en cada caso de mejora hasta lograr una solución satisfactoria, sin llegar a la perfección.

Caso de Mejora	Redefiniciones
Caso de Mejora 1	1
Caso de Mejora 2	2
Caso de Mejora 3	1
Caso de Mejora 4	1
Caso de Mejora 5	2
Caso de Mejora 6	2
Caso de Mejora 7	1

Tabla 22. Redefiniciones por caso de mejora (Construcción propia).

Los casos de mejora que tuvieron más de dos redefiniciones, se debió a la complejidad que estos tenían, con respecto al conocimiento y habilidades que se debían adquirir, por tanto, fue necesario realizar otra iteración para refinar las soluciones planteadas. Es importante resaltar que en esta fase del ciclo de mejora, los planes de acción para cada caso de mejora fueron refinados a través del ciclo

de vida propuesto por el método de Investigación – Acción, expuesto anteriormente. Ver figura 33.

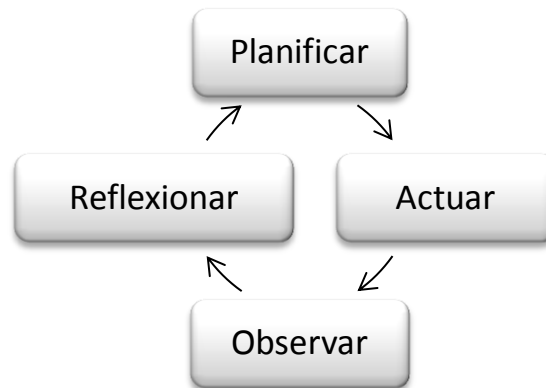


Figura 33. Ciclo de vida de Investigación – Acción (Adaptación de [47]).

Una vez finalizada la implantación de cada caso de mejora, se realizaba un reunión donde los involucrados en los proyectos pilotos contaba las experiencias obtenidas, los beneficios identificados, los inconvenientes que tuvieron y como los solucionaron a los otros integrante del equipo de trabajo.

Después de ver el éxito en proyecto piloto y de exponer los beneficios que se obtuvieron, se procedió en compañía del Área de Procesos a documentar el nuevo proceso y posteriormente a divulgarlo e Institucionalizarlo en la Organización. La herramienta utilizada para especificar y publicar el nuevo proceso de requisitos fue EFPC, dado permite crear un sitio web interactivo con el proceso diseñado, el cual era publicado en la Intranet de la Organización cuando este era actualizado.

Por último, en la figura 34, se puede visualizar como se llevó a cabo la ejecución del plan de mejora en la Organización, indicando las redefiniciones de los planes de acción, los proyectos que sirvieron de pilotos para la implementación de la solución y el tiempo de implementación de cada uno de los casos de mejora.

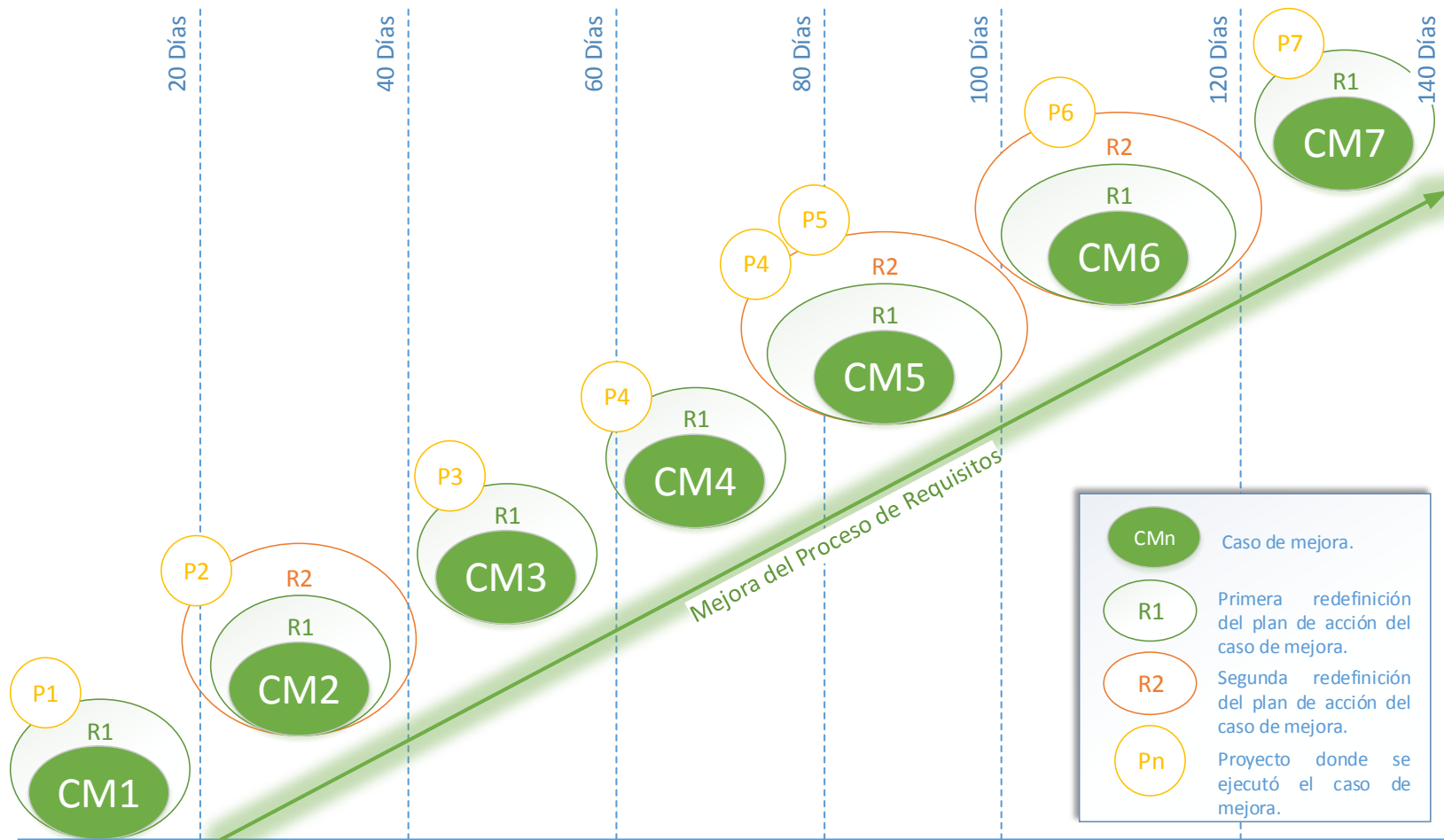


Figura 34. Ejecución del plan de mejora (Construcción propia).

5.5. Fase de Aprendizaje

Asimismo, después de finalizar la implantación de cada caso de mejora, en la reunión de cierre, donde participaban el Asesor, Grupo Interdisciplinario y el Equipo de trabajo, se establecieron los logros alcanzados, el esfuerzo invertido y las lecciones aprendidas, esto con el fin de aumentar el conocimiento de la Organización sobre programas de mejora, las cuales fueron consignadas en documento oficial y almacenadas en el repositorio establecido por la Gestión de la Configuración, y de esto modo retroalimentar y enriquecer los siguientes proyectos pilotos antes de que iniciaran.

Entre las lecciones aprendidas más destacadas se encuentran:

- Realizar reuniones de retrospectivas durante la ejecución del proyecto piloto, permitió mejorar los resultados obtenidos.
- Realizar talleres al inicio de cada proyecto piloto, le permitió a los integrantes del Equipo de trabajo del proyecto interiorizar el objetivo del caso de mejora y mejorar el conocimiento sobre la Ingeniería de Requisitos.
- Establecer una estrategia que agilice la aprobación del plan de trabajo para cada caso de mejora, puesto que en varias ocasiones los integrantes del Grupo Interdisciplinario no se podían reunir por motivos Organizacionales, lo cual retrasaba el programa de mejora.

6. Análisis de los resultados del Programa de Mejora

En este capítulo se presentan los niveles de adopción de CMMI y el índice de satisfacción del usuario logrados después de ejecutar completamente el programa de mejora, además, se comparte el análisis financiero donde se establece el ROI alcanzado después de finalizar el proyecto.

Igualmente, se recopilan los factores de éxito y problemas identificados en esta experiencia de mejora y por último, se da respuesta a cada una de las preguntas de investigación planteadas para este proyecto.

6.1. Análisis de los resultados de mejora

Al mes de haber finalizado el programa de mejora, se evaluó nuevamente el proceso de requisitos desde la perspectiva de los analistas, con el propósito de cuantificar el nivel de avance de las áreas de proceso establecidas por el CMMI para la Ingeniería de Requisitos.

Para el área de procesos Desarrollo de Requisitos – RD se logró un gran avance, donde la meta específica “SG3 Analizar y validar los requisitos” supero el nivel de adopción del 80%, además la meta específica “SG2 Desarrollar los requisitos del producto” aumento en un 50% su nivel de adopción, mientras que la meta específica “SG1 Desarrollar los requisitos del cliente” logró un nivel mayor a 70%, lo que indica que las recomendaciones establecidas para el proceso de requisitos estaban articuladas a las practicas recomendadas por CMMI. Ver figura 34.

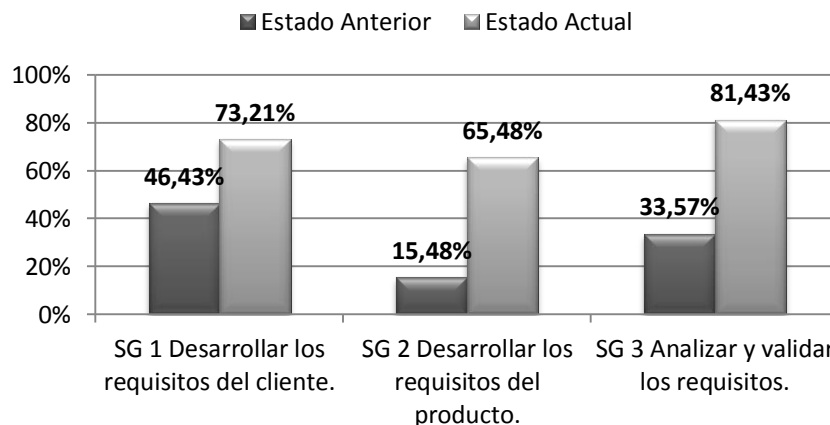


Figura 35. Nivel de adopción de metas específicas RD (Construcción propia).

Igualmente, se realizó la comparación entre el estado actual y anterior para el área de proceso *Gestión de Requisitos – REM*, donde se alcanzó un destacado avance,

puesto que se pasó de un 17,02% a un 70,18% de nivel de adopción de las practicas recomendadas por el CMMI para la gestión de Requisitos, como se muestra en la figura 35.

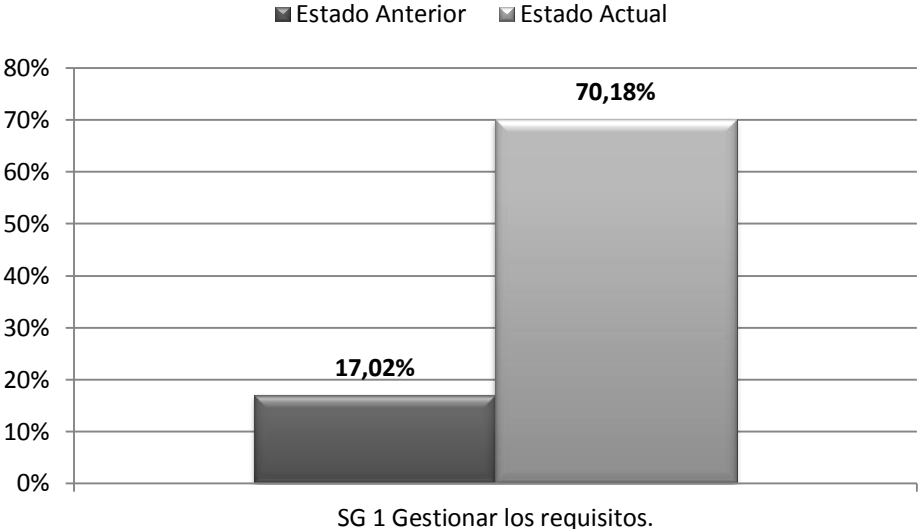


Figura 36. Nivel de adopción de metas específicas REQM (Construcción propia)

Además, las áreas de proceso establecidas por CMMI para la ingeniería de requisitos lograron alcanzar un nivel de adopción superior al 70%, lo cual es un excelente resultado para el programa de mejora. Ver figura 36.

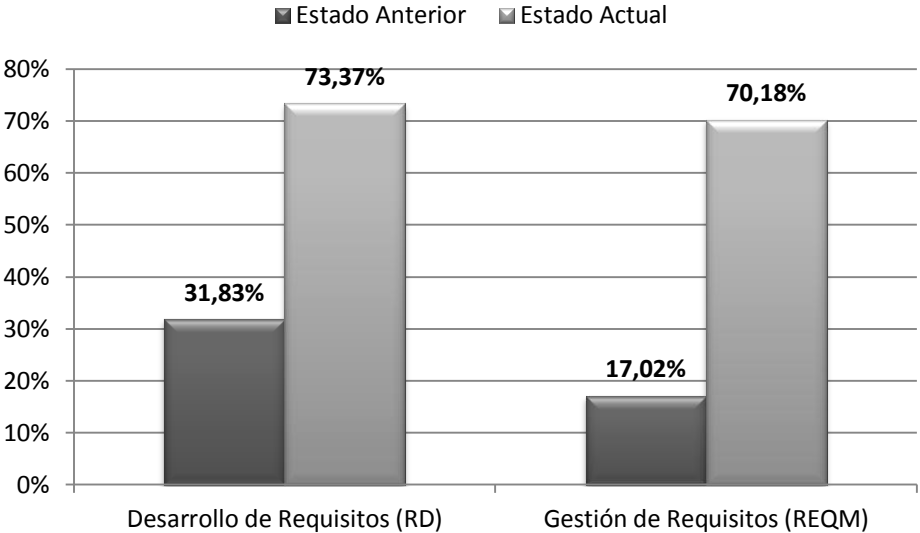


Figura 37. Estado actual de RD y REQM (Construcción propia)

6.2. Análisis de las metas del programa de mejora

Antes de iniciar el programa de mejora, un analista de desarrollo empleaba mensualmente el 60% del tiempo laboral (144 horas) en actividades efectivas sobre los proyectos de software, mientras que las 96 horas restantes eran empleadas de la siguiente manera:

- 24 horas en promedio se destinaban en actividades de Soporte de segundo nivel sobre las aplicaciones de negocio.
- 65 horas en promedio estaban siendo destinadas en atender solicitudes de cambio y reprocesos en las soluciones construidas.
- 7 horas en promedio se invertían en actividades Organizacionales.

Nota: La información expuesta anteriormente fue obtenida del sistema de gestión de requerimientos - SGR, donde los Analistas diligencian diariamente las actividades que realizan en el transcurso del día, indicando el tipo de actividad realizada.

Por medio de un seguimiento continuo a los proyectos ejecutados durante los próximos 6 meses después de finalizar el programa de mejora, se obtuvo la siguiente información. Ver tabla 23.

Concepto	Evaluación Inicial	Mes					
		1	2	3	4	5	6
Tiempo invertido en proyecto (Hora)	144	145	159	167	177	184	190
Reprocesos y Solicitudes de cambio (Hora)	65	62	58	53	49	45	45

Tabla 23. Distribución de horas por actividades (Construcción propia).

Donde se evidencia que la **primera y tercera meta** del programa de mejora, fueron alcanzadas, puesto que se logró que los analistas invirtieran un 32% más de su tiempo en actividades del proyecto y se disminuyó en un 31% el tiempo destinado para solucionar inconsistencias.

Por tanto, la adopción de las nuevas prácticas al proceso de requisitos establecido por la Organización, permitió aumentar la productividad del Equipo de Desarrollo en los proyectos en lo que participan y disminuir los reprocesos relacionados con la Ingeniería de Requisitos con respecto al Proceso Software desde el punto de vista del Subgerente de Desarrollo, el cual acepto satisfactoriamente el éxito del proyecto.

Igualmente, se recopiló información del sistema de gestión de requerimientos mes a mes durante 6 meses, para determinar el índice de satisfacción del usuario, el cual es calculado automáticamente por sistema al finalizar la solicitud, donde el usuario califica la oportunidad y calidad del producto entregado por el área de desarrollo entre 0 y 100% donde 0 es totalmente insatisfecho y 100% es totalmente satisfecho.

Como se indicó al inicio del documento, el índice de satisfacción se encontraba en un 45% y fue ascendiendo poco a poco hasta lograr un máximo de 81%, lo que indica que las prácticas adoptadas y los conocimientos adquiridos fueron madurando en el transcurso del tiempo. Ver figura 37.

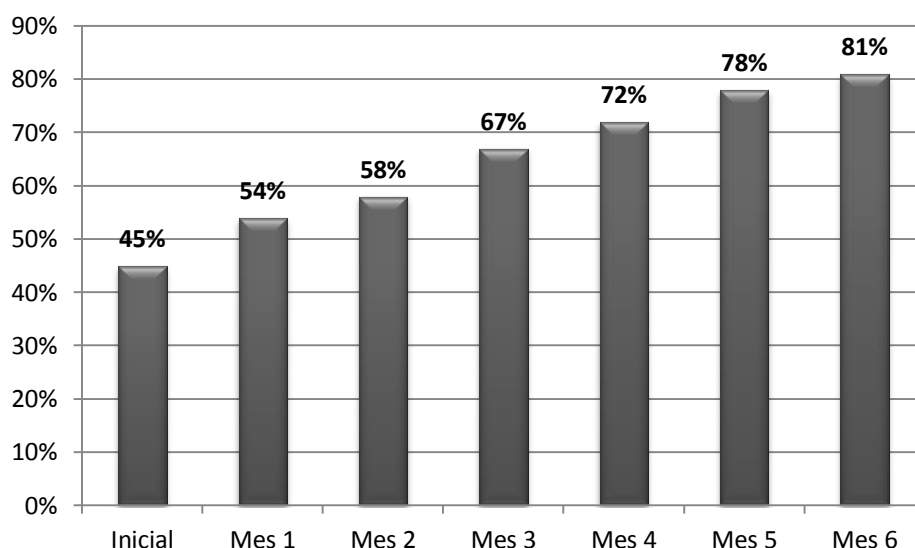


Figura 38. Índice de satisfacción del usuario (Construcción propia).

Además es importante resaltar que este nuevo índice de satisfacción permitió alcanzar la **segunda meta** del programa de mejora de aumentar en un 30% de satisfacción del usuario, puesto que este índice aumento un 36%.

6.3. Análisis Financiero

Para realizar el análisis financiero del programa de mejora, el Asesor cuantificó la inversión realizada por la Organización en aspectos económicos, humanos y de tiempo, asimismo, los beneficios obtenidos después de la ejecución del programa de mejora, y de este modo establecer el ROI logrado.

Los costos asociados mes a mes durante la implementación del programa de mejora se presentan en la tabla 24, donde se indica la inversión realizada en capacitación, adopción de prácticas de CMMI, adecuación del sistema de

información que apoya el proceso software y las licencias de nuevas herramientas de apoyo para el proceso de Requisitos.

Concepto de Inversión	Personas /Cantidad	Hora x Mes	Valor Unidad	Mes						Total	
				1	2	3	4	5	6		
Capacitación											
Curso de Ingeniería de Requisitos	8		600.000	4.800.000							4.800.000
Tiempo interrupción de labores	8	40	18.000	5.760.000							5.760.000
Mejorar Proceso de Requisitos											
Investigar	1	30	18.000	270.000	540.000	540.000					1.350.000
Establecer propuesta	3	30	18.000		810.000	1.620.000	1.620.000				4.050.000
Ejecutar propuesta	3	30	18.000			1.620.000	1.620.000	810.000			4.050.000
Evaluar resultados	1	20	18.000			180.000	360.000	360.000			900.000
Institucionalizar cambios	3	30	18.000				1.620.000	1.620.000			3.240.000
Adecuación Sistema de Gestión de Requisitos	2	40	18.000					720.000	1.440.000		2.160.000
Licencia EA	8		250.000				2.000.000				2.000.000
Total				10.830.000	1.350.000	3.960.000	7.220.000	3.510.000	1.440.000		28.310.000

Tabla 24. Costos económicos por mes (Construcción propia).

Además, en la tabla 25, se muestran los valores que corresponden al beneficio que se obtuvo después de la implementación del programa de mejora, con base en el mejoramiento de la Productividad, la disminución de Reprocesos y la disminución de las solicitudes de Cambio a atender.

Concepto	Mes					
	1	2	3	4	5	6
Beneficio por productividad	160.000	2.400.000	3.680.000	5.280.000	6.400.000	7.360.000
Beneficio por disminución de reprocesos	480.000	1.120.000	1.920.000	2.560.000	3.200.000	3.200.000
Beneficio Por Mes	640.000	3.520.000	5.600.000	7.840.000	9.600.000	10.560.000
Beneficio Acumulado	640.000	4.160.000	9.760.000	17.600.000	27.200.000	37.760.000

Tabla 25. Beneficios económicos por mes (Construcción propia).

De esta manera se establece que entre los meses 5 y 6 después de la implementación del programa de mejora, se recupera la inversión efectuada. Igualmente los beneficios se seguirán obteniendo de manera permanente, teniendo como base el escenario del sexto mes, es decir al menos mensualmente se tendrá un beneficio de **\$10.560.000**, valor que sería actualizado de acuerdo con los ajustes salariales que la Organización determine anualmente.

Por último se estableció la razón beneficio/costo, en nuestro caso, dado que los beneficios perdurarán en el tiempo, es importante tomar un corte para comparar estos valores, por lo cual tomaremos el corte del sexto mes después de la implementación, puesto que seis (6) meses fue el periodo establecido como horizonte para la evaluación de las metas. Siendo así, la razón económica Beneficio/Costo fue la siguiente:

$$ROI = \frac{\text{Beneficios}}{\text{Costos}} = \frac{\$37'760.000}{\$28'310.000}$$

$$ROI = 1,33$$

Con este resultado se determina que los beneficios obtenidos por la implementación del proyecto cubren **1.33** veces los costos asociados a la implementación del mismo, además, es importante resaltar que los beneficios económicos perduran en el tiempo.

6.4. Inconvenientes presentados durante el programa de mejora

En el conjunto de actividades que se realizaron para implementar el programa de mejora, se presentaron varios problemas. A continuación se describirán los problemas de mayor relevancia con su respectiva estrategia para mitigar su impacto.

Problema 1: Los encuestados tenían inconvenientes en la interpretación de conceptos técnicos que se mencionaban en las preguntas y respuestas de la encuesta.

Para solucionar este inconveniente, el grupo de encuestadores implemento la estrategia de compartir las dificultades, es decir, si un encuestado tenía inconvenientes para interpretar un concepto, inmediatamente los encuestadores explicaban su definición y se contextualizaba con eventos reales de la Organización, además la explicación de este concepto se replicaba sobre los demás encuestados, con el propósito de mantener la homogeneidad en el conocimiento para solucionar la encuesta.

Problema 2: Algunos encuestados empezaron a solucionar el instrumento tomando como referencia el mundo ideal y no las actividades de la Ingeniería de Requisitos que ejecutaban en el Proceso Software de la Organización, lo cual afectaría considerablemente el resultado del diagnóstico, puesto que el objetivo es determinar la brecha entre las practicas actuales y las sugeridas por el modelo CMMI-DEV.

Inmediatamente que se identificó este problema, los encuestadores realizaron un explicación a cada uno de los encuestados, indicándoles que deben solucionar el instrumento tomando como referencia las actividades que realizan actualmente en el proceso software que tengan relación con la Ingeniería de Requisitos y así determinar la brecha entre las practicas actuales y recomendada.

Esta situación se consideró anormal, puesto que al momento de la entrega de la encuesta, se le indico a cada encuestado que debían tomar como referencia el proceso software utilizado por la Organización para desarrollar o mantener software.

Problema 3: Los encuestados manifestaban la poca disponibilidad de tiempo que tenían para resolver un instrumento, lo cual afectaría el resultado del diagnóstico, dado que el instrumento podía ser solucionado sin realizar un análisis cuidadoso y a consciencia por algún integrante.

Para mitigar el inconveniente de que la encuesta fuera resuelta a la ligera, se le explicó la importancia del proceso de la Ingeniería de Requisitos y del instrumento; y que beneficios traería si este último fuese solucionado con un análisis detallado de cada pregunta y sus respectivas respuestas.

Problema 4: El Equipo de desarrollo no llegaba a un acuerdo sobre la mejor solución que se debía emplear para cada caso de mejora, lo que impedía que el programa de mejora avanzara a lo planificado.

El Asesor indicio que la solución no debe ser perfecta desde el inicio, puesto que a medida que se ejecute la solución en el proyecto piloto, estará será mejorada a través del ciclo de vida del método Investigación – Acción.

Problema 5: Los integrantes de los proyectos pilotos no seguían el plan de trabajo establecido para el proyecto.

Para solucionar este inconveniente, fue necesario realizar una reunión con la Dirección de la Organización, para que concienciara a los integrantes de los Equipos de trabajo de la importancia del programa de mejora para el negocio y el Departamento de Informática. Después de esta reunión el Equipo de trabajo tomó con mayor seriedad el programa de mejora.

Problema 6: Los analistas de proceso no disponían de una herramienta para diseñar el nuevo proceso de requisitos de forma que éste fuera interactivo para el Departamento de Informática y usuarios.

Después de analizar varias alternativas para diagramar el proceso, entre las que se encontraba BPA Suite, Aris y Microsoft Visio, se optó por diseñarlo en EPFC, además de que era gratuito, permitía publicar un sitio web interactivo con el proceso diseñado.

7. Conclusiones, Lecciones Aprendidas y trabajos futuros

Se ha mostrado la experiencia de un programa de mejora aplicada en el área de informática dedicada a ofrecer servicios financieros. Este caso de estudio fue realizado siguiendo un marco de trabajo que fue definido teniendo en cuenta las condiciones particulares de la Organización. En este último capítulo se comparten las conclusiones obtenidas al finalizar la experiencia de mejora del proceso de requisitos. En primer lugar se retoman las preguntas de investigación y se da respuesta a la manera como éstas fueron abordadas en el trabajo, en segundo lugar se sintetizan las lecciones aprendidas de este programa de mejora que se realizó de manera que sirvan como referente a organizaciones que emprendan programas de mejora y especialmente en los casos en que se trata de organizaciones internas de una empresa de servicio. Finalmente se recomiendan trabajos futuros para la Organización en estudio.

7.1. Conclusiones

En esta sección del documento se dará respuesta a las preguntas de investigación planteadas para el proyecto de Maestría a partir de la experiencia obtenida en el programa de mejora del proceso de requisitos.

1. ¿De qué manera integrar el Modelo IDEAL con Casos de Mejora y la Investigación – Acción para mejorar el proceso de requisitos?

El conjunto de actividades del modelo IDEAL enriquecidas con el concepto de casos de mejora y el método de Investigación – Acción, contribuyeron de manera positiva a lograr los resultados esperados del proyecto. La manera como fue integrado en el modelo IDEAL los casos de mejora y la Investigación acción, fue explicada de manera detallada en el capítulo 4 y 5, lo cual puede resumirse en los siguientes aspectos:

- Se agilizaron las fases de Inicio, Diagnóstico y Aprendizaje.
- Se definieron nuevas actividades en las fases de Establecimiento y Acción para trabajar con casos de mejora.
- Se enriquecieron las actividades de la Fase de Acción con el método de Investigación – Acción.
- Se establecieron proyectos pilotos acorde a las recomendaciones sugeridas en la Fase de Establecimiento.
- Se priorizaron los casos de mejora de acuerdo a las necesidades de la Organización.

Además de la integración del modelo IDEAL, esta experiencia integró a lo largo de las fases un conjunto de prácticas que resultaron beneficiosas; entre las cuales se puede destacar:

- Definir responsabilidades tomando como referente la matriz RACI.
- Definir metas a través de la técnica GQM.
- Definir criterios para priorizar los casos de mejora por medio de Pair-Wise.
- Establecer una estrategia de institucionalización iterativa y evolutiva.
- Realizar retrospectivas a medida que se implementaban los casos de mejora.

2. ¿De qué manera evaluar logros tempranos en el proceso de mejora de requisitos?

En esta experiencia de mejora se evidencia que afrontar un programa de mejoramiento a través de casos de mejora, permite obtener resultados a corto plazo que motivan al Equipo de trabajo en continuar trabajando en pro del proyecto.

Igualmente, facilita establecer un esquema de institucionalización progresiva, mitigando los riesgos de resistencia al cambio y desmotivación de los integrantes del programa, dado que a medida que se evidenciaba el éxito de la solución en proyectos pilotos, se iniciaba la institucionalización de las prácticas sugeridas en el caso de mejora. Y por medio de la institucionalización la Dirección de la Organización y los usuarios percibían que el proyecto de mejora avanzaba y brindaba resultados positivos al Departamento de Informática, los cuales se propagaban al interior de la Organización.

3. ¿Cuáles son las herramientas y prácticas de apoyo que facilitan un proceso de adopción liviana por parte del equipo de desarrollo?

Las prácticas que permitieron establecer un proceso de mejora liviano en este proyecto, fueron:

- Dividir la complejidad del problema, para simplificar los desafíos a los que se enfrentaba el programa de mejora y de este modo construir planes de trabajo más acoplados a las circunstancias.
- Realizar entregas frecuentes de mejora al proceso de requisitos, y de este modo mostrar tangiblemente el avance del programa de mejoramiento.
- Establecer comunicación cara a cara entre los integrantes del programa de mejoramiento.

- Obtener aprendizaje continuo por medio de la observación y reflexión, con el objetivo de mejorar poco a poco la solución planteada hasta llegar a su punto de aceptación.
- Establecer una infraestructura liviana sustentada en un esquema colaborativo y participativo, siempre partiendo de los principios de simplicidad y practicidad.
- Realizar retrospectivas del avance del programa semanalmente, esto con el fin de orientar a los integrantes del proyecto hacia las metas del plan de mejora.
- Minimizar la cantidad de documentación a realizar mientras se refina la solución, partiendo del principio de sólo construir los documentos estrictamente necesarios.
- Formalizar el proceso, luego de aplicar los conceptos de mejora en los proyectos pilotos.

Además, entre las herramientas de apoyo que se pueden resaltar se encuentra:

- Tablero de trabajo, donde el Asesor informaba diariamente sobre las actividades pendientes, en proceso y finalizadas con su respectivo responsable, muy similar al taskboard de Scrum. Esta herramienta fue de vital apoyo para el programa, debido a que el Equipo de trabajo se mantenía actualizado del estado y del avance de éste, sin necesidad de establecer otros espacios de discusión.

Al principio el tablero de trabajo tuvo resistencia por parte de algunos integrantes del proyecto, puesto que los dejaba en evidencia ante los demás, por no cumplir con las actividades asignadas, pero al final, se convirtió en una herramienta de motivación, debido a que podían visualizar que el proyecto avanzaba y las metas eran alcanzables.

- La Aplicación EPFC (Eclipse Process Framework Composer) permitió modelar el proceso de requisitos de una forma ágil e interactiva, además, facilita la creación de un sitio web que contiene la definición del proceso.
- Además, de la agilidad del programa de programa de mejora, también se requieren de herramienta que permitan dar orden, establecer controles y brindar formalidad al programa en los aspectos Organizacionales, como la gestión de la configuración, gestión del cambio y gestión del costo, para ello, se hizo de uso de herramientas como Visual Source Safe y Excel.

4. ¿Cuál es el papel que pueden jugar los clientes del producto en un proyecto de mejora de procesos?

Dado que son los clientes del producto los que conocen con mayor certeza la calidad que requieren del mismo y perciben directamente los beneficios que éste ofrece, éstos pueden brindar información vital para los programas de mejora; conllevando a que la calidad exigida del producto genere mayor calidad del proceso como tal.

Así mismo son los clientes del producto los que determinan en algunos casos la necesidad de mejorar los procesos de software, debido a que pueden señalar las necesidades que no están siendo satisfechas, convirtiéndose así en evaluadores del proceso software.

Particularmente en el programa de mejora de proceso de requisitos, se torna muy importante la vinculación activa de los clientes de las soluciones, pues ellos juegan un papel relevante dentro del proceso

5. ¿Cuáles fueron los factores de éxito del proyecto de mejora?

El Grupo Interdisciplinario, después de analizar las lecciones aprendidas y las experiencias obtenidas en el transcurso del programa de mejora destacaron los siguientes factores de éxito:

- Un factor de éxito importante en esta experiencia, fue el compromiso que brindó la Dirección de la Organización, puesto que fue el organismo que respaldaba todas las decisiones tomadas, igualmente, le otorgaba credibilidad e importancia al proyecto dentro de la Organización.
- Otro factor de éxito trascendental, fue la generación de resultados a corto plazo, debido a que la Organización y los Equipo de trabajo podían percibir los beneficios que se lograban a medida que el programa de mejora avanzaba.
- Otro factor de éxito significativo, fue haber establecido desde el inicio las metas del programa de mejora, dado que los integrantes de los Equipos de trabajo sabían hacia donde iban, por tanto, concentraban todos sus esfuerzos en lograr meta trazada.
- Además, uno de los principales factores de éxito de este programa de mejora, fue la capacitación sobre Ingeniería de Requisitos, puesto que le permitió al Equipo de trabajo obtener nuevos conocimientos, los cuales eran necesarios en el transcurso del proyecto y permitió enriquecer las soluciones planteadas a través de la observación, reflexión y discusión.
- Asimismo, haber establecido una estrategia basada en la colaboración y participación, permitió que los integrantes del proyecto se apersonaran del

mismo, puesto que sintieron que sus aportes, conocimientos e ideas eran importantes para el éxito del proyecto.

Por último, en la tabla 26, se indica el nivel y el tipo de incidencia de los factores de éxito identificados en la literatura que se presentan en el Capítulo 2, con el propósito de evaluar la impacto de estos sobre este programa de mejora.

Factor de Éxito	Tipo de Incidencia	Nivel de Incidencia
Compromiso de la alta gerencia	Positivo	Alto
Recursos	Positivo	Alto
Estrategia de implementación	Positivo	Alto
Objetivos de MPS claros	Positivo	Alto
Participación del personal	Positivo	Alto
Conciencia y percepción de beneficios	Positivo	Alto
Comunicación	Positivo	Medio
Estándares y procedimientos	Positivo	Bajo
Personal con habilidades, conocimientos	Negativo	Medio
Gestión del cambio cultural y organizacional.	Negativo	Medio
Capacitación y Mentoring	Positivo	Alto
Tiempo	Negativo	Medio
Autonomía	Positivo	Medio
Grupo de trabajo dedicado a la MPS	Negativo	Medio
Carrera profesional	Positivo	Bajo
Políticas organizacionales	Negativo	Bajo
Gestión del proyecto de mejora	Positivo	Medio
Liderazgo	Positivo	Medio
Compensaciones	Negativo	Bajo
Mediciones	Positivo	Bajo
Personal de MPS respetado	Positivo	Bajo
Rotación de personal	Negativo	Bajo

Tabla 26. Nivel de incidencia de los factores de éxito (Construcción propia)

7.2. Lecciones Aprendidas

A partir de los resultados obtenidos y de la experiencia vivida en el programa de mejora se puede concluir lo siguiente:

- El complemento entre el modelo IDEAL, casos de mejora e investigación acción, permiten establecer un enfoque ágil de mejora, puesto que proveen un proceso de mejora iterativo e incremental que facilita la obtención de resultados a corto plazo.
- El respaldo de la alta dirección de la organización es vital para el éxito del proyecto, puesto que inyecta credibilidad de éste al interior de la Organización.

- Por medio de la observación y reflexión, se logra establecer un esquema de trabajo colaborativo y participativo, facilitando la cooperación entre todos los integrantes del proyecto de mejora.
- El nivel de adopción de las prácticas de requisitos sugeridas por CMMI-DEV, permite establecer el grado de madurez que la Organización tiene con respecto a las áreas de proceso.
- Realizar una reunión donde explica detalladamente el instrumento antes de iniciar la evaluación, permite unificar la interpretación del instrumento, lo cual aumentó la confiabilidad de los datos.
- En términos financieros, es mayor el beneficio obtenido al desarrollar el programa de mejora propuesto, que la inversión que éste requirió para su implementación.
- El cliente jugó un papel vital dentro del programa de mejora, debido a la información que brindó, por consiguiente es de gran importancia la participación activa de él.
- Es importante medir el retorno de la inversión, aunque se requiere alto esfuerzo en mediciones, al final permite tener evidencias cuantitativas acerca del impacto económico que generó la mejora del proceso.

7.3. Trabajo futuro

Para trabajos futuros dentro de la misma organización, se recomienda continuar haciendo esfuerzos en adoptar un mayor nivel de las prácticas específicas mostradas en la tabla 27, debido a que estas se encuentran en un nivel de adopción menor a un 65% y fueron las que obtuvieron un menor porcentaje de adopción.

Área de Proceso	Práctica Especifica	Porcentaje Adopción
REQM	SP 1.3 Gestionar los cambios de los requisitos.	49,11%
RD	SP 2.3 Identificar los requisitos de la interfaz.	50,0%

Tabla 27. Prácticas específicas menos adaptadas (Construcción propia).

Además, se recomienda dar inicio a otros programas de mejora del proceso software que estén alienados con las problemáticas identificadas en la sección 3, puesto que allí se indicó las problemáticas que afectan al proceso software definido por la Organización. Por ejemplo, se recomienda iniciativas en las áreas de la Gestión de Proyectos, Gestión del Conocimiento, Gestión de la

Configuración o en la fase de análisis y diseño, y de modo alcanzar progresivamente un grado de madurez acorde a las necesidades de la Organización.

Para finalizar, se sugiere hacer uso del complemento entre el modelo IDEAL, casos de mejora, investigación – acción en otro programa de mejoramiento, con el propósito de recopilar evidencia de que éste complemento facilita definir un enfoque ágil basándose en la simplicidad y practicidad.

8. Bibliografía

[1] Abran, Alain; Moore, Jame W. "Guide to the Software Engineering Body of Knowledge - SWEBOK", IEEE Computer Society. Version 2004.

[2] Standish Group Chaos, Standish Research Paper. Href <http://www.standishgroup.com>.

[3] Anaya, Raquel; Gómez, Liliana. "Lecciones Aprendidas en el Acompañamiento Masivo para Mejora de Procesos en Empresas de Software: Un Caso Colombiano". Memorias CIBSE 2011. Springer-Verlag Berlin Heidelberg. 2011.

[4] Arias Chaves, Michael. "La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software". Julio 2006.

[5] Hickey, A., and A. Davis, "The Role of Requirements Elicitation Techniques in Achieving Software Quality", International Workshop on Requirements Engineering: Foundations for Software Quality (REFSQ). September 2002.

[6] Paulk, M., "The Capability Maturity Model for Software: Guidelines for Improving the Software Process". SEI Series in Software Eng. 1995.

[7] Herbsleb, J. D., & Goldenson, D. R., "A Systematic Survey of CMM Experience and Results", Proc. 18th Int'l Conf. Software Eng., 323–330. 1996.

[8] Damian, D., Zowghi, D., Vaidyanathasamy, L., & Pal, Y. "An industrial case study of immediate benefits of requirements engineering process improvement at the Australian Center for Unisys Software". Empirical Software Engineering, 45-75. 2004.

[9] Diaz, M., & Sligo, J., "How software process improvement helped Motorola". IEEE Software, 75-81. 1997,

[10] Pino, F. J., García, F., & Piattini, M., "Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas". REICS Revista española de innovación, calidad e ingeniería de software, 6-23. 2006.

[11] Sawyer, P., Sommerville, I., and Viller, S., "Capturing the Benefits of Requirements Engineering", IEEE Software, 16(2), pp. 78-85. 1999.

[12] Sawyer, P., Sommerville, I., and Viller, S., "Improving the Requirements Process", Cooperative System Engineering Group. 1998.

[13] ZIEHE, Ted. "Software Process Improvement (SPI) Guidelines for Improving Software: Release 5.0", SEMATECH, October 1996.

[14] VAN SOLINGE, Rini. "Measuring the ROI of Software Process Improvement", IEE Software, 2004.

[15] Humphrey, W. S., Snyder, T. S., & Willis, R. R., "Software process improvement at Hughes Aircraft". IEEE Software, 11-23. 1991.

[16] McFeeley, Bob. "IDEALSM: A User's Guide for Software Process Improvement", Software Engineering Institute, Febrero 1996.

[17] Cabrera, Armando; Solano, Raquel; Montalván, Mayra. "Procesos de ingeniería del software". Universidad Técnica Particular de Loja. 2009.

[18] Chrissis, Mary Beth; Konrad, Mire; Shrum, Sandy. "CMMI Guía para la integración de procesos y la mejora de productos". Traducción: Cátedra de Mejora de Procesos de Software en el Espacio Iberoamericano de la Universidad Politécnica de Madrid. 2009.

[19] "Guía de los fundamentos para la dirección de proyectos - PMBOK", Project Management Institute, Inc. Cuarta Edición. 2008.

[20] Informe Inversión en el sector Software y Servicios de TI en Colombia. Href <http://www.inviertaencolombia.com.co/sectores/servicios/software-y-servicios-de-ti.html>

[21] "Proexport – Oportunidades de la Industria de Software - TI Colombia" Href <http://www.inviertaencolombia.com.co/sectores/servicios/software-y-servicios-de-ti.html>

[22] Revista IT Manager. "¿Hacia dónde va la industria del Software?". Href <http://fedesoft.org/novedades/hacia-donde-va-la-industria-del-software>

[23] VonSperling, Daniel L; Prado, Darci. "MATURITY AND SUCCESS IN I.T. Summary Report." Slide 12. April 30th 2010. Href www.maturityresearch.com.

[24] Richardson, Ita; Ryan, Kevin. "Software Process Improvements in a Very Small Company". University of Limerick, Ireland. 2001.

[25] Piattini, M; García, F. "Calidad en el desarrollo y mantenimiento del software". Madrid: RA-MA Editorial. 2003.

[26] Alarcón, Andrea; Gonzáles, Juan; Rodríguez, Sandra. "Guía para pymes desarrolladoras de software, basada en la norma ISO/IEC 15504". Revista – Universidad Católica del Norte. 2011.

[27] Beecham, Sarah; Hall, Tracy; Rainer, Austen. "Defining a Requirements Process Improvement Model". Department of Computer Science, University of Hertfordshire. Software Quality Journal - SQJ , vol. 13, no. 3, pp. 247-279, 2005.

[28] SEI Administrative Agent. "CMMI para desarrollo, Versión 1.3 – Mejora de los procesos para el desarrollo de mejores productos y servicios". Editorial Universitaria Ramón Areces. Noviembre 2010.

[29] Oktaba, Hanna; Piattini, Mario; Pino, Franciso J; Orozco, Maria Julia; Alquicira; Claudia. "Mejora de Procesos Software para pequeñas y medianas empresas y proyectos. Madrid: RA-MA Editorial. 2008.

[30] NMX-I-059/02-NYCE-2011 - MoProSoft. Href <http://www.moprosoft.com.mx/>

[31] Métrica 3 - Inicio/Documentación/Metodologías y Guías/Métrica v.3. Href <http://administracionelectronica.gob.es>.

[32] MPS.BR – Melhoria de Procesos do Software Brasileiro. Href <http://www.softex.br/mpsbr/ES/guias/default.asp>

[33] Members of the Assessment Method Integrated team. "Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition", SEI. Diciembre 2001.

[34] Sánchez Lorenzo, Gonzalo Alonso. “Mejora del proceso software de una pequeña Empresa desarrolladora de software: Caso Competisoft-perú tau”. Lima, Perú. Pontificia Universidad Católica del Perú. 2008.

[35] Software Engineering Institute – SEI. “The IDEAL Model”. Carnegie Mellon University. 2009. Href <http://www.sei.cmu.edu/library/assets/idealmodel.pdf>

[36] Piattini, Mario; “Calidad de Sistemas de Información”. Sección: Material - Evaluación y mejora de procesos. Universidad de Castilla La Mancha. 2009. Href <http://alarcos.inf-cr.uclm.es/doc/calidad/>

[37] Vidal, Juan; Hurtado, Julio; Pino, Francisco; Oktaba, Hanna; Piattini, Mario. “Mejora de Procesos – IT. 13”. Competisoft. Agosto 2006. Href http://alarcos.inf-cr.uclm.es/competisoft/publico/downloads/Inf_T%C3%A9cnicos/COMPETISOFT_IT%2013_Mejora%20de%20Procesos.pdf

[38] Pardo, César; Hurtado, Julio; Collazos, César. “Mejora de procesos de software ágil con Agile - SPI Process”. Revista Dyna. Julio 2009. Href <http://www.scielo.org.co/pdf/dyna/v77n164/a25v77n164.pdf>

[39] Oktaba, Hanna; Piattini, Mario. “Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies”. IGI Global, USA 2008.

[40] Pino, Francisco; Vidal, Juan; Hurtado, Julio. “Guía del consultor para la mejora de procesos software – IT. 18”. Competisoft. Marzo 2007. Href http://alarcos.inf-cr.uclm.es/competisoft/publico/downloads/Inf_T%C3%A9cnicos/COMPETISOFT_IT_18%20Marzo%2015%202007%20Guia%20del%20consultor.pdf

[41] Palomino Vásquez, Marco Antonio Ibsen. “Mejora del proceso de una pequeña empresa desarrolladora de software: caso Competisoft - Peru LIM.LAMBDA, segundo ciclo”. Lima, Perú. Pontificia Universidad Católica del Perú. Noviembre 2011.

[42] Pino, Francisco; Vidal, Juan; García, Félix; Piattini, Mario. “Modelo para la implementación de mejora de procesos en pequeñas Organizaciones Software”. MBI do Brasil. 2007

[43] Maturro, Gerardo; Saavedra; Jimena. “Factores que Inciden en la Mejora de Procesos Software. Un mapeo sistemático de la literatura”. ClbSE 2012: pp 84-97.

[44] Palacio, Juan. "Flexibilidad con Scrum". SafeCreative. Noviembre 2007. Href http://www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf

[45] Jedlitschka, Andreas; Ciolkowski, Marcus. "Towards Evidence in Software Engineering". International Symposium on Empirical Software Engineering. IEEE Computer Society, 2004.

[46] Apa, Cecilia; Robaina, Rosana; De León, Stephanie; Vallespir, Diego. "Conceptos de Ingeniería de Software Empírica". Instituto de Computación – Facultad de Ingeniería, Universidad de la Republica, Montevideo, Uruguay. 2010.

[47] Murillo, Francisco Javier. "Métodos de investigación en Educación Especial". Universidad Autónoma de Madrid. Curso Investigación - Acción 2010-2011. Href http://www.uam.es/personal_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Curso_10/Inv_accion_trabajo.pdf

[48] Dittrich, Yvonne. "Doing empirical research on software development: Finding a path between understanding, intervention, and method development". Social thinking: Software practice, MIT Press Cambridge. 2002.

[49] Easterbrook, Steve; Singer, Janice; Storey; Margaret-Anne, Damian, Daniela. "Selecting Empirical Methods for Software Engineering Research". University of Toronto. 2007. Href <http://www.cs.toronto.edu/~sme/papers/2007/SelectingEmpiricalMethods.pdf>

[50] Cavero, José; Vela, Belén; Martínez, Esperanza. "Aspectos filosóficos, psicológicos y metodológicos de la informática". Ciencia Experimental y la Tecnología. Universidad del Rey Juan Carlos. Madrid, 2005.

[51] Arrieta, Guillermo; Araque, César. "La observación: base metodológica de la investigación". Revista INIA, Venezuela. Diciembre 2006.

[52] Páez Anaya, Ivan Dario. "Estudio empírico del estado actual de la estimación de software en Pymes de Colombia". Universidad Eafit. Mayo 2012.

[53] Anaya, Raquel. "¿Cómo acortar la brecha entre la teoría y la práctica de la Ingeniería de Software?" Universidad EAFIT. LACREST Medellín 2012.

[54] Wang, Yingxu. "Software Engineering Standards: Review and Perspectives". World Scientific Publishing. Ed. De la Villa M., Ruiz M., Ramos I., Modelos de Evaluación y Mejora de Procesos: Análisis Comparativo. 2002.

[55] Romero, Felipe; Blanco, Mónica. "Mejoramiento de Procesos de Software en pequeñas empresas". Departamento de Ingeniería de Sistemas. Universidad de los Andes Revista electrónica Paradigma en construcción de Software. Enero 2008. Href

http://paradigma.uniandes.edu.co/index.php?option=com_content&task=view&id=33&Itemid=33&lang=es&showall=1

[56] Pino, Francisco; García, Félix; Piattini, Mario. "Adaptación de las normas ISO/IEC 12207:2002 y ISO/IEC 15504:2003 para la evaluación de la madurez de procesos software en países en desarrollo". IEEE América Latina, Vol. 4, N° 2, Abril 2006, pp. 17-24

[57] ISO – International Organization for Standardization. "ISO/IEC 15504". 2004. Href <http://www.iso.org>.

[58] Pino, Francisco J; García, Félix; Piattini, Mario. "Aplicación de Investigación – Acción y Casos de estudio utilizando Competisof". ClbSE. 2009. páginas.167-180.

[59] Niño, Jhon; "Estudio empírico de aplicación de la metodología PSP en estudiantes de un programa de Tecnología en Sistemas con diferentes niveles de formación". Universidad Eafit. 2012.

[60] Páez, Ivan; Anaya, Raquel; Travasso, Guillermo. "Estado actual de la estimación de software en compañías colombianas que han adoptado CMMI". Ponencia Workshop latinoamericano de software Experimental. 2012.

[61] Humphrey, W. S., Snyder, T. S., & Willis, R. R., "Software process improvement at Hughes Aircraft". IEEE Software, pp 11-23. 1991.

[62] Diaz, M., & Sligo, J., "How software process improvement helped Motorola". IEEE Software, pp 75-81. 1997.

[63] R. van Solingen, E. Berghout, The Goal/Question/Metric Method: a Practical Guide for Quality Improvement of Software Development. McGraw Hill, London, 1999.

[64] Pardo, César; Hurtado, Julio; Collazos, César. “Mejora de procesos de software ágil con Agile - SPI Process”. Revista Dyna. Julio 2009.

[65] Visconti, Marcello; Cook, Curtis R. “An Ideal Process Model for Agile Methods”. 5th International Conference, PROFES 2004, Kansai Science City, Japan. 2004. pp 431 – 441.

[66] Tagra, Dinesh. “SPIMS: Software Process Improvement Model for Small Scale Industries”. Computer Science and Engineering Department. Thapar University, Patiala, India. 2009.

[67] Sulayman, Muhammad. “A systematic literature review of software process improvement for small and medium web companies”, Department of Computer Science, University of Auckland, 2009.

[68] WIEGERS, Karl E. “Process Goodies / Process Improvement” Href: <http://www.processimpact.com>

[69] Shull, Forrest; Mendonça, Manoel G.; Basili, Victor; Carver, Jeffrey; Maldonado, José C.; Fabbri, Sandra; Travassos, Guilherme Horta; Ferreira, Maria Cristina. “Knowledge-Sharing Issues in Experimental Software Engineering”. Journal, Empirical Software Engineering. Volume 9 Issue 1-2, Marzo 2004, pp 111 - 137.

[70] Mafra, Sômulo Nogueira; Travassos, Guilherme Horta. “Estudos Primários e Secundários apoiando a busca por Evidência em Engenharia de Software”. Programa de Engenharia de Sistemas e Computação. Rio de Janeiro. Marzo 2006.

[71] Mafra, Sômulo Nogueira; Barcelos, Rafael Ferreira; Travassos, Guilherme Horta.” Aplicando uma Metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software”. XX Simpósio Brasileiro de Engenharia de Software. Florianópolis, SC, Brasil. 2006.

[72] Proyecto LACXSER (Latin American Collaboratory of eXperimental Software Engineering Research). Href: <http://www.lacxser.org/>

[73] Karlsson, Joachim. “Software Requirements Prioritizing”. Requirements Engineering, 1996., Proceedings of the Second International Conference on. Abril 1996.

9. Anexos

9.1. Anexo 1

Encuesta de diagnóstico del Proceso de Requisitos. Perspectiva de los Usuarios.

1. ¿Sabe usted que es un requisito y cuál es su principal objetivo? Si la respuesta es "Sí" explique brevemente que significa.

Sí No

2. En la definición de los requisitos, ¿conoce claramente la información que se requiere para obtener el resultado esperado del requisito?

Siempre Casi Siempre Algunas veces Nunca

3. En la definición de los requisitos, ¿conoce claramente el resultado que se debe obtener al finalizar el requisito?

Siempre Casi Siempre Algunas veces Nunca

4. ¿Tiene usted en cuenta las restricciones del sistema, de los procesos y aspectos legales para la definición de los requisitos?

Sí No

5. En la definición de los requisitos, ¿usted conoce los procesos de la Organización que serán afectados?

Siempre Casi Siempre Algunas veces Nunca

6. En la definición de los requisitos, ¿usted tiene en cuenta las políticas definidas por la Organización para esta actividad?

Siempre Casi Siempre Algunas veces Nunca

7. ¿Usted tiene en cuenta las demás áreas, la manera como intervienen y se ven afectadas con los requisitos en curso? Si la respuesta es "Sí" explique brevemente como hace partícipe a las áreas afectadas.

Sí No

8. ¿Conoce usted qué impacto tiene un requisito mal definido en el desarrollo de un proyecto?
- Bajo Medio Alto
9. Al momento de definir los requisitos, ¿usted tiene en cuenta las estimaciones de tiempo, recursos e impactos que estos conllevan para la Organización?
- Sí No
10. En la definición de los requisitos, ¿usted especifica claramente las validaciones que se deben de realizar para garantizar su objetivo?
- Sí No
11. ¿Usted como usuario prioriza los requisitos solicitados al Departamento de informática? **Si su respuesta es afirmativa, describir cómo.**
- Sí No
- ¿Cómo?: _____
-
12. ¿Considera usted que los dueños de procesos deberían ser los encargadas de definir los requisitos y hacer la solicitud al Departamento de informática?
- Sí No
13. ¿Usted conoce el proceso definido por el Departamento de informática para la toma de requisitos?
- Sí No
14. En caso de conocer el proceso de requisitos, clasifique la importancia de este proceso en el desarrollo de nuevas funcionalidades y/o servicios para la Organización. **En una escala del 1 al 5, dónde 5 es “Muy importante” y 1 es “No es importante”**
- 1 2 3 4 5
15. ¿Considera usted que la compañía del analista de desarrollo facilita y mejora la toma de requisitos? **En una escala del 1 al 5, dónde 5 es “Muy importante” y 1 es “No es importante”**
- 1 2 3 4 5
16. ¿Considera usted que es importante la participación de un analista de procesos como apoyo en la definición y revisión de los requisitos? **En una escala del 1 al 5, dónde 5 es “Muy importante” y 1 es “No es importante”**
- 1 2 3 4 5

17. ¿Cómo considera el nivel de equipamiento de la Organización para la toma de requisitos? **En una escala del 1 al 5, dónde 5 es “Excelente” y 1 es “Insuficiente”**

- Herramientas para crear solicitudes: 1 2 3 4 5
Oportunidad de informática: 1 2 3 4 5
Disponibilidad de informática: 1 2 3 4 5
Manuales para definir requisitos: 1 2 3 4 5
Capacitación para definir requisitos: 1 2 3 4 5

18. ¿Considera que el análisis y las especificaciones de los requisitos pueden hacerse de manera más simple? Si la respuesta es “Sí” indique como podría facilitarse esta actividad.

Sí No

¿Cómo?: _____

19. ¿Cree usted que la manera como se vienen realizando la toma de requisitos en la Organización debe mejorar? Si la respuesta es “Sí” indique como podría facilitarse esta actividad.

Sí No

¿Cómo?: _____

9.2. Anexo 2

Encuesta de diagnóstico del Proceso de Requisitos. Perspectiva del Área de Desarrollo.

1. ¿Cómo son identificados y caracterizados los grupos de usuarios que se encuentran involucrados en el proceso de requisitos?
 - a. No se sabe con exactitud quiénes serán los usuarios.
 - b. Los representantes de los usuarios creen que ellos saben quiénes son los usuarios.
 - c. Los usuarios que realizaron la solicitud, son agrupados de acuerdo a su importancia en el proceso de requisitos.
 - d. Los interesados directos e indirectos son identificados por grupos de usuario, cuyas características se resumen en la especificación de requisitos de software.
2. ¿Qué criterios utilizan para la evaluación y aceptación de los requisitos?
 - a. Se utiliza la intuición del analista.
 - b. Si los requisitos están en un documento de especificación de requisitos, son aceptados.
 - c. Se verifica que los requisitos sean claros y no generen ambigüedades.
 - d. Se verifica que los requisitos descritos en la especificación de requisitos se encuentren claramente establecidos, completos, sin ambigüedades, apropiados para implementar, rastreables y verificables.
3. Después de tener un total entendimiento sobre los requisitos ¿Cómo se vinculan a los usuarios al proceso de requisitos?
 - a. Los usuarios no son vinculados, porque los desarrolladores ya saben qué deben construir.
 - b. Los representantes de los usuarios creen que pueden proporcionar la perspectiva del usuario.
 - c. Los grupos de usuarios identificados son encuestados o entrevistados constantemente, con el propósito de tener una comunicación continua con el usuario, pero los usuarios no se encuentran comprometidos totalmente.
 - d. Se acuerdan responsabilidades, niveles de autoridad y compromisos con las personas que representan los diferentes grupos de usuarios que participan en el proyecto, estos acuerdos son registrados en la especificación de requisitos y son evaluados nuevamente si se presentan cambios en los requisitos.
4. ¿Cómo son gestionados los cambios en los requisitos?
 - a. Cambios no controlados, se ingresan nuevos cambios en el proyecto cada vez que alguien tiene una idea nueva o se da cuenta de que se les olvidó algo.

- b. No se realizan cambios a los requisitos después de que la fase de requisitos está completa, pero todavía se hacen acuerdos informales de cambio.
 - c. Utilizamos un formato definido para la presentación de las solicitudes de cambio. El líder del proyecto decide qué cambios se van a incorporar.
 - d. Los cambios se realizan de acuerdo a nuestro documentado de control de cambios. Utilizamos una herramienta para recoger, almacenar y comunicar las solicitudes de cambio. El impacto de cada cambio se evalúa antes de que el grupo de control de cambios decide si se aprueba.
5. ¿Cómo es definida la línea base para los requisitos y como es gestionada en cada proyecto?
- a. No se define una línea base.
 - b. Los usuarios y gestores establecen totalmente los requisitos, pero el desarrollo sigue recibiendo muchos cambios y reclamaciones.
 - c. Se define una línea de base en los requisitos iniciales en la especificación de requisitos, pero no siempre se actualiza cuando se realizan cambios a los requisitos.
 - d. Los requisitos son almacenados en una base de datos y se define una línea de base inicial. La especificación de requisitos es actualizada cuando los cambios son aprobados. Mantenemos un historial de cambios sobre la línea base.
6. ¿Cómo son clasificadas las diferentes versiones de los documentos de requisitos?
- a. El documento es impreso y la fecha se genera automáticamente.
 - b. Usamos una secuencia de números como 1.0, 1.1, y así sucesivamente para cada versión del documento.
 - c. Tenemos un esquema de identificación manual, que permite distinguir entre las versiones preliminares, de las versiones de la línea base, considerando el nivel de importancia.
 - d. Los documentos de requisitos utilizan el control de versiones y son almacenados en un sistema de gestión de la configuración, o en una herramienta de gestión de requisitos, que permite mantener un historial de los documentos, incluyendo los cambios.
7. ¿Cómo se rastrea el origen de los requisitos de software?
- a. Los requisitos no son rastreables.
 - b. Sabemos la procedencia de muchos requisitos.
 - c. Todos los requisitos tienen un origen identificado.
 - d. Tenemos rastreabilidad en dos vías. El origen de todos los requisitos de software, y los artefactos que fueron generados a partir de ellos (Ejemplo: caso de uso, reglas de negocio, arquitectura o componentes).
8. ¿Cómo se identifican las inconsistencias entre los requisitos y el producto final?

- a. Después de finalizar la solución, esta es entregada al usuario.
 - b. La solución es evaluada por el usuario utilizando como instrumento la especificación de requisitos, identificando las inconsistencias y reportándolas al equipo del proyecto.
 - c. El producto final es evaluado contra la especificación de requisitos, con el propósito de identificar las inconsistencias y los motivos de porque se presentaron. Estas inconsistencias son documentadas y se identifican los cambios que se deben realizar.
 - d. El producto final es confrontado contra los requisitos incluyendo los cambios realizados en el transcurso del proyecto, lo cual permite identificar las inconsistencias y sus razones. Inmediatamente se modifica el plan de proyecto para realizar los cambios que deben ser ejecutados para corregir dichas inconsistencias.
9. ¿Qué métodos se utilizan para identificar todos los requisitos de software?
- a. Comenzamos con una idea general, escribir código, y modificar el código hasta que haya terminado.
 - b. Los representantes de los usuarios proporcionan un concepto del producto, y los desarrolladores empiezan a escribir los requisitos.
 - c. Los representantes de los usuarios manifiestan las características y funciones que contener el producto en un documento no estandarizado.
 - d. Llevamos a cabo las entrevistas estructuradas de obtención de requisitos con representantes de los diferentes grupos de usuarios. Contamos con casos de uso para entender las tareas del usuario, y obtener de allí los requisitos funcionales.
10. ¿Cómo son documentados los requisitos de software?
- a. Se tiene una historia oral de las reuniones, mensajes de correo, notas de entrevistas y notas de la reunión.
 - b. Escribimos documentos no estructurados en texto narrativo, o creamos diagramas de casos de uso para el análisis.
 - c. Escribimos los requisitos en un lenguaje natural estructurado, utilizando plantillas estandarizadas para la especificación de requisitos. En ocasiones nos apoyamos con algunos modelos de análisis usando notaciones estándar.
 - d. Son almacenadas las necesidades del usuario en una base de datos o una herramienta de gestión de requisitos, incluyendo modelos de análisis en una herramienta CASE y son identificadas las restricciones.
11. ¿Los requisitos del usuario son analizados técnicamente para determinar los requisitos del producto y las componentes del producto?
- a. Lo requisitos del usuario no son analizados técnicamente.
 - b. Los requisitos de los usuarios son analizados técnicamente por los analistas, con el objetivo de brindar insumo al diseño del producto y así establecer las componentes del producto.

- c. Después de realizar un análisis de los requisitos en lenguaje técnico (UML) por parte de los analistas, se puede llegar a identificar nuevos requisitos derivados de las decisiones del diseño (Ej: Selección de una tecnología).
 - d. Se realiza el análisis de los requisitos por medio de diagramación (UML) y son utilizados para las decisiones de diseño, además estos son relacionados con el propósito de ayudar a la evaluación del impacto de los cambios.
12. ¿Cómo son asignados los requisitos del sistema a las componentes del producto?
- a. Los analistas después de definir los requisitos del producto, proceden inmediatamente a desarrollar la funcionalidad.
 - b. Después de realizar un análisis de los requisitos, estos son asignados por funciones.
 - c. Un analista o un arquitecto del sistema analiza los requisitos y decide cuáles van a ser asignados en cada componente del producto, teniendo en cuenta las restricciones.
 - d. Los requisitos del sistema son agrupados por funciones y son asignados a las componentes del producto, incluyendo las limitaciones del diseño. Esta información se encuentran explícitamente definida y documentada.
13. ¿Cómo son integradas las componentes del producto entre sí, o como permiten la integración del producto con otros productos externos?
- a. No se realiza análisis de este tipo.
 - b. Se analizan las componentes del producto para determinar las interfaces internas que requiere para su buen funcionamiento.
 - c. Se identifican las interfaces que permitan la integración del producto internamente y externamente.
 - d. Se identifican las interfaces externas e internas del producto y se adicionan los requisitos de esta interfaces en la especificación de requisitos.
14. ¿Se establecen los conceptos operativos y los escenarios donde el producto interactúa con el entorno, con los usuarios y/o otras componentes?
- a. Se identifican los conceptos operativos y los escenarios que incluyen información sobre la instalación, operación, mantenimiento y conceptos de soporte del producto o componentes del producto, o si es necesario como debe ser retirado o desinstalado.
 - b. Además de que se identifican los conceptos operativos y los escenarios del literal a. También se define el entorno donde funcionara el producto, teniendo en cuenta las restricciones.
 - c. Se identifican los conceptos operativos, los escenarios y el entorno del producto. Además estos son revisados, con el propósito de verificar que estén acordes a los requisitos y si es posible identificar nuevos requisitos.

- d. Se identifican y se revisan los conceptos operativos, los escenarios, el entorno del producto y como este producto interactúa con el usuario final, apoyándose en UML para su análisis.
15. ¿Cómo realizan el análisis funcional o definición de la funcionalidad, con el propósito de tener agrupaciones lógicas y su asociación con los requisitos?
- a. No se realiza un análisis funcional.
 - b. Se analizan y se cuantifican las funcionalidades que requiere el usuario.
 - c. Se realiza la práctica mencionada en el literal b. Además se analizan los requisitos del producto con el propósito de identificar particiones lógicas o funcionalidades (Ej: subfunciones).
 - d. Después de identificar las funcionalidades del usuario y de los requisitos, estos últimos son agrupados, basándose en criterios como funcionalidades similares o por acoplamiento, para facilitar el análisis de requisitos, apoyándose en los diagramas de casos de uso.
16. ¿Cómo realizan el análisis funcional o definición de la funcionalidad, con el propósito de determinar lo que debe hacer el producto?
- a. No se realiza un análisis funcional.
 - b. Se identifica la secuencia de las funciones críticas del producto apoyándose en el diagrama de actividades.
 - c. Después de identificar la secuencia de las funciones críticas, los requisitos del usuario son asociados a las particiones funcionales o elementos de soporte.
 - d. Se realizan las actividades del literal c. y se asignan los requisitos funcionales a las funciones y subfunciones previamente identificadas.
17. ¿Cómo realizan el análisis de los requisitos para determinar si es viable su implementación?
- a. No realizamos análisis, empezamos inmediatamente a desarrollar.
 - b. Los requisitos son analizados con el fin de detectar y eliminar las ambigüedades que se pueden representar y son organizados por funcionalidades.
 - c. Se eliminan las ambigüedades entre los requisitos y son analizados para determinar si están acorde a las necesidades de los usuarios.
 - d. Los requisitos son analizados con el propósito de garantizar que se encuentren completos, sean factibles, realizables, verificables, que no presenten ambigüedades y que estén alineados con las necesidades de los usuarios.
18. Si han decidido implementar los requisitos del producto, ¿Cómo analizan el impacto, el rendimiento técnico y las restricciones de los requisitos?
- a. No se realiza ningún tipo de análisis.
 - b. Se identifican los requisitos claves, teniendo en cuenta criterios como: Costo, tiempo, complejidad de implementación, riesgo o rendimiento.

- c. Se identifican los requisitos claves y las medidas de rendimiento técnico que serán seguidas durante el esfuerzo del desarrollo.
 - d. Se identifican los requisitos claves y las medidas de rendimiento técnico, y se analizan los conceptos operativos y los escenarios, con el propósito de refinar las necesidades y restricciones
19. ¿Qué métodos son utilizados para analizar el equilibrio entre las necesidades y las restricciones?
- a. No se utiliza ningún tipo de método de análisis.
 - b. El análisis se apoya en algunos modelos, simulaciones y/o prototipos usando notación estándar del grupo.
 - c. Se realiza un análisis sobre algunos modelos y prototipos; posteriormente se realizan estudios de impacto sobre los requisitos con relación a la arquitectura funcional que se tiene.
 - d. Se realiza la actividad descrita en el literal b. Además se identifican los riesgos que pueden surgir si la arquitectura funcional no está acorde a los requisitos teniendo en cuenta los ciclos del producto.
20. ¿Cómo son validados los requisitos?
- a. Los requisitos no son validados por los interesados.
 - b. Se tienen listas definidas sobre las validaciones básicas para evaluar los diseños contra el entorno del producto, con el fin de mitigar el riesgo y el impacto.
 - c. Después de determinar el riesgo del producto, los analistas y los interesados evalúan los requisitos contra la especificación de requisitos apoyándose en modelos (Diagramas Casos de Uso, Diagrama de actividades).
 - d. Inspeccionamos los documentos de requisitos y modelos de diseño, con los participantes involucrados, tales como: Usuarios, analistas, desarrolladores y tester. Evaluando la exactitud de los requisitos existentes contra las necesidades de los interesados, con el propósito de obtener una retroalimentación de los usuarios y posiblemente definir nuevos requisitos que no han sido especificados.