

UNIVERSIDAD CATÓLICA DE SANTA MARÍA

FACULTAD DE CIENCIAS E INGENIERÍAS FÍSICAS Y FORMALES

PROGRAMA PROFESIONAL DE INGENIERÍA DE SISTEMAS



TEMA:

“PROPUESTA DE UN MODELO DE INTEGRACIÓN PARA DESARROLLAR Y GESTIONAR REQUISITOS EN EL DESARROLLO DE PROYECTOS DE SOFTWARE BASADO EN EL ESTÁNDAR CMMI-DEV Y SWEBOK”

Presentado por:

JOSÉ LUIS GUTIÉRREZ VILCA

Para optar el Título profesional de:

Ingeniero de Sistemas

**AREQUIPA – PERÚ
2014**

DEDICATORIA

A mis padres por su apoyo, comprensión e inmenso amor el cual me brindan día a día.

A mis hermanos por estar siempre conmigo.

A mis abuelos Martin y Susana que me cuidan y guían desde el cielo.

Y a toda mi familia por su apoyo incondicional.

AGRADECIMIENTOS

A mis padres por su apoyo e infinita paciencia.

Al Dr. Cesar Baluarte Araya por su asesoría y apoyo incondicional durante el desarrollo del presente trabajo.

A la Mgter. Karim Guevara Puentes de la Vega por el apoyo en la verificación del presente trabajo.

A Boris Mutafelija, Alfraino de Souza, Rosana Vaccare y Estela Morales por compartir sus consejos e ideas las cuales fueron un valioso aporte en la realización de este trabajo.

ÍNDICE

LISTA DE FIGURAS	iv
LISTA DE TABLAS	v
LISTA DE ABREVIATURAS.....	vi
RESUMEN	vii
ABSTRACT	viii
INTRODUCCIÓN.....	ix
CAPITULO I: PLANTEAMIENTO TEÓRICO	1
1.1. TÍTULO DEL PROYECTO	1
1.2. DESCRIPCIÓN DEL PROBLEMA.....	1
1.3. FORMULACIÓN INTERROGATIVA DEL PROBLEMA.....	2
1.4. JUSTIFICACIÓN DE LA INVESTIGACIÓN	3
1.5. OBJETIVOS DE LA INVESTIGACIÓN	3
1.5.1. Objetivo general	3
1.5.2. Objetivos específicos.....	3
1.6. HIPÓTESIS	3
1.7. VARIABLES E INDICADORES	4
1.7.1. Variable independiente.....	4
A. Indicadores	4
1.7.2. Variable dependiente.....	4
A. Indicadores	4
1.8. ÁREA Y LÍNEA DE INVESTIGACIÓN	4
1.8.1. Área de investigación	4
1.8.2. Línea de investigación.....	4
1.9. TIPO Y NIVEL DE INVESTIGACIÓN	5
1.9.1. Tipo de investigación	5
1.9.2. Nivel de investigación	5
1.10. ÁMBITO	5
CAPITULO II: MARCO TEÓRICO	6
2.1. REVISIÓN HISTÓRICA.....	6
2.1.1. Historia y evolución del CMMI	6
2.1.2. Historia y evolución del SWEBOK.....	7
2.1.3. Historia y evolución de IDEAL.....	7
2.2. CONCEPTOS	8

2.2.1.	Modelo.....	8
2.2.2.	Requisito.....	8
2.2.2.1.	Requisito de Software	8
2.2.2.2.	Desarrollo de requisitos.....	9
2.2.2.3.	Gestión de requisitos	9
2.2.3.	Estándar	9
2.2.4.	Proyecto	10
2.2.5.	Mejora de Procesos de Software	10
2.3.	ANTECEDENTES INVESTIGATIVOS	10
2.4.	ESTADO DEL ARTE.....	13
2.4.1.	Requisitos de Software según SWEBOK 3.0.....	13
2.4.1.1.	Fundamentos de los Requisitos de Software.....	15
2.4.1.2.	Proceso de los Requisitos	17
2.4.1.3.	Educción de Requisitos	18
2.4.1.4.	Análisis de Requisitos	20
2.4.1.5.	Especificación de Requisitos.....	22
2.4.1.6.	Validación de Requisitos.....	23
2.4.1.7.	Consideraciones Prácticas	25
2.4.1.8.	Herramientas de Requisitos de Software	27
2.4.2.	Desarrollo de Requisitos según CMMI-DEV 1.3.....	28
2.4.2.1.	Desarrollar los Requisitos del Cliente (SG 1).....	32
2.4.2.2.	Desarrollar los Requisitos de Producto (SG 2)	35
2.4.2.3.	Analizar y Validar Requisitos (SG 3)	38
2.4.3.	Gestión de Requisitos según CMMI-DEV 1.3	44
2.4.3.1.	Gestionar los Requisitos (SG 1).....	45
2.4.4.	Modelo IDEAL.....	51
2.4.4.1.	Inicio	52
2.4.4.2.	Diagnóstico	57
2.4.4.3.	Establecimiento	58
2.4.4.4.	Acción	60
2.4.4.5.	Aprendizaje	62
CAPITULO III: PROPUESTA DEL MODELO		64
3.1.	PROPUESTA DEL MODELO DE INTEGRACIÓN.....	64
3.1.1.	Introducción.....	64

3.1.2.	Objetivo	64
3.1.3.	Estructura del modelo de integración propuesto	65
3.1.4.	Practicass Fundamentales	66
3.1.4.1.	Estructura y fuentes	67
3.1.4.2.	Comparación entre el modelo CMMI-DEV y la Guía SWEBOK	69
3.1.4.3.	Desarrollo de las Prácticas Fundamentales	73
3.1.5.	Estrategia de implementación.....	93
3.1.5.1.	Desarrollo de la estrategia de implementación	94
CAPITULO IV: REALIDAD DE LA EMPRESA	101
4.1.	DESCRIPCIÓN DE LA EMPRESA	101
4.2.	EVALUACIÓN DE LA SITUACIÓN ACTUAL.....	102
4.2.1.	Presentación de resultados.....	104
CAPITULO V: VALIDACIÓN	115
5.1.	DESARROLLO DE LA VALIDACIÓN DEL MODELO	115
5.1.1.	Fase Inicio	115
5.1.2.	Fase Diagnostico	117
5.1.3.	Fase Establecimiento	127
5.1.4.	Fase Acción	130
5.1.5.	Fase de Aprendizaje.....	133
5.2.	EVALUACIÓN DEL MODELO	134
CONCLUSIONES	146
SUGERENCIAS	147
BIBLIOGRAFÍA	148
ANEXO 1	150
ANEXO 2	151
ANEXO 3	154

LISTA DE FIGURAS

Figura 1. Resultados de resolución de proyectos de la investigación CHAOS.....	1
Figura 2. La historia de los CMMs.....	7
Figura 3. Distribución de temas para los Requisitos de Software.....	15
Figura 4. Estructura de las representaciones continua y por etapas.....	29
Figura 5. Componentes del modelo CMMI.....	31
Figura 6. Resumen de Metas y Prácticas Específicas.....	32
Figura 7. Resumen de Metas y Prácticas Específicas.....	45
Figura 8. Fases y tareas por fase del modelo IDEAL.....	51
Figura 9. Componentes y fuentes del modelo propuesto.....	65
Figura 10. Estructura área de proceso.....	66
Figura 11. Estructura de prácticas fundamentales.....	67
Figura 12. Estructura y fuentes de las prácticas fundamentales.....	68
Figura 13. Significado de requisito y su objetivo.....	104
Figura 14. Conocimiento de los resultados.....	105
Figura 15. Conocimiento de la información.....	105
Figura 16. Consideración de restricciones del sistema, procesos y aspectos legales.....	106
Figura 17. Conocimiento de procesos afectados.....	107
Figura 18. Conocimiento de políticas.....	107
Figura 19. Impacto de un requisito mal definido.....	108
Figura 20. Conocimiento de tiempo, recursos, impacto.....	109
Figura 21. Establecer validaciones.....	109
Figura 22. Priorización de requisitos.....	110
Figura 23. Experto responsable de los requisitos.....	111
Figura 24. Equipamiento para toma requisitos.....	111
Figura 25. Mejorar el proceso de requisitos.....	112
Figura 26. Cumplimiento de Prácticas específicas – Desarrollo de Requisitos.....	120
Figura 27. Cumplimiento de Metas específicas – Desarrollo de Requisitos.....	121
Figura 28. Nivel de Adopción – Desarrollo de Requisitos.....	121
Figura 29. Cumplimiento de Prácticas específicas – Gestión de Requisitos.....	122
Figura 30. Cumplimiento de Meta específica – Gestión de Requisitos.....	123
Figura 31. Interpretación de resultados – Pregunta 1.....	136
Figura 32. Interpretación de resultados – Pregunta 2.....	137
Figura 33. Interpretación de resultados – Pregunta 3.....	138
Figura 34. Interpretación de resultados – Pregunta 4.....	139
Figura 35. Interpretación de resultados – Pregunta 5.....	140
Figura 36. Interpretación de resultados – Pregunta 6.....	141
Figura 37. Interpretación de resultados – Pregunta 7.....	142
Figura 38. Interpretación de resultados – Pregunta 8.....	143
Figura 39. Interpretación de resultados – Pregunta 9.....	144
Figura 40. Interpretación de resultados – Pregunta 10.....	145

LISTA DE TABLAS

Tabla 1. Principales factores de fracaso en los proyectos	2
Tabla 2. Áreas de conocimiento del SWEBOK	13
Tabla 3. Áreas de proceso, categorías y niveles de madurez	30
Tabla 4. Fases del modelo IDEAL	52
Tabla 5. Roles, responsabilidades y habilidades de cada grupo del proyecto	55
Tabla 6. Comparación SWEBOK y CMMI-DEV	70
Tabla 7. Desarrollo de Requisitos - Meta específica 1 – Práctica específica 1.1	73
Tabla 8. Desarrollo de Requisitos - Meta específica 1 – Práctica específica 1.2	75
Tabla 9. Desarrollo de Requisitos - Meta específica 2 – Práctica específica 2.1	76
Tabla 10. Desarrollo de Requisitos - Meta específica 2 – Práctica específica 2.2	78
Tabla 11. Desarrollo de Requisitos - Meta específica 2 – Práctica específica 2.3	78
Tabla 12. Desarrollo de Requisitos - Meta específica 3 – Práctica específica 3.1	79
Tabla 13. Desarrollo de Requisitos - Meta específica 3 – Práctica específica 3.2	80
Tabla 14. Desarrollo de Requisitos - Meta específica 3 – Práctica específica 3.3	82
Tabla 15. Desarrollo de Requisitos - Meta específica 3 – Práctica específica 3.4	84
Tabla 16. Desarrollo de Requisitos - Meta específica 3 – Práctica específica 3.5	86
Tabla 17. Gestión de Requisitos - Meta específica 1 – Práctica específica 1.1.....	87
Tabla 18. Gestión de Requisitos - Meta específica 1 – Práctica específica 1.2.....	88
Tabla 19. Gestión de Requisitos - Meta específica 1 – Práctica específica 1.3.....	89
Tabla 20. Gestión de Requisitos - Meta específica 1 – Práctica específica 1.4.....	90
Tabla 21. Gestión de Requisitos - Meta específica 1 – Práctica específica 1.5.....	91
Tabla 22. Desarrollo y Gestión de Requisitos - Meta genérica 1 – Práctica genérica 1.1....	92
Tabla 23. Actividades Principales de la fase Inicio.....	94
Tabla 24. Actividades Principales de la fase Diagnóstico.....	95
Tabla 25. Ejecución del Diagnóstico.....	96
Tabla 26. Actividades Principales de la fase Establecimiento	97
Tabla 27. Actividades Principales de la fase Acción.....	99
Tabla 28. Actividades Principales de la fase Aprendizaje.....	100
Tabla 29. Roles y responsabilidades y habilidades de cada grupo del proyecto	116
Tabla 30. Relación entre prácticas recomendadas y preguntas de la encuesta.....	119
Tabla 31. Prácticas menos utilizadas en Proceso de Requisitos.....	124
Tabla 32. Priorización de las recomendaciones.....	128
Tabla 33. Relación entre Variables, Indicadores e Ítems del cuestionario.....	134
Tabla 34. Tabulación de respuestas.....	135

LISTA DE ABREVIATURAS

CMM	Capability Maturity Model (Modelo de madurez y de capacidad)
CMMI	Capability Maturity Model Integration (Modelo de madurez y de capacidad Integrado)
CMMI-DEV	CMMI for Development (CMMI para Desarrollo)
IDEAL	Iniciación, Diagnostico, Establecimiento, Accion, Aprendizaje
IEEE	Institute of Electrical and Electronics Engineers (Instituto de Ingeniería Eléctrica y Electrónica)
ISO	International Organization for Standardization (Organización Internacional de Normalización)
SEI	Software Engineering Institute (Instituto de Ingeniería de Software)
SW-CMM	Capability Maturity Model for Software (Modelo de madurez y de capacidad para Software)
SWECC	Software Engineering Coordinating Committee (Comité para la coordinación de la ingeniería de software)
SWEBOK	Software Engineering Body of Knowledge (Guía al Cuerpo de Conocimiento de la Ingeniería de Software)

RESUMEN

Los procesos de desarrollo y gestión de requisitos se encuentran entre las principales fuentes de los problemas encontrados durante el desarrollo de proyectos de software. Uno de los aspectos críticos para el éxito en el desarrollo de estos procesos es el uso de técnicas, metodologías y normas para la implementación de estas actividades.

En este trabajo se propone un modelo de integración, cuya base son las buenas prácticas del modelo CMMI-DEV y la guía SWEBOK. Integrandos ambos conjuntos de conocimientos, este modelo se enfocará específicamente en los procesos de requisitos de software con el objetivo de facilitar y mejorar la aplicación en los procesos de desarrollo y gestión de requisitos.

Por último, para la implementación de estos procesos, se define una estrategia basada en el modelo IDEAL, el cual proporciona un conjunto de actividades coherentes para sustentar la adopción de las buenas prácticas propuestas.

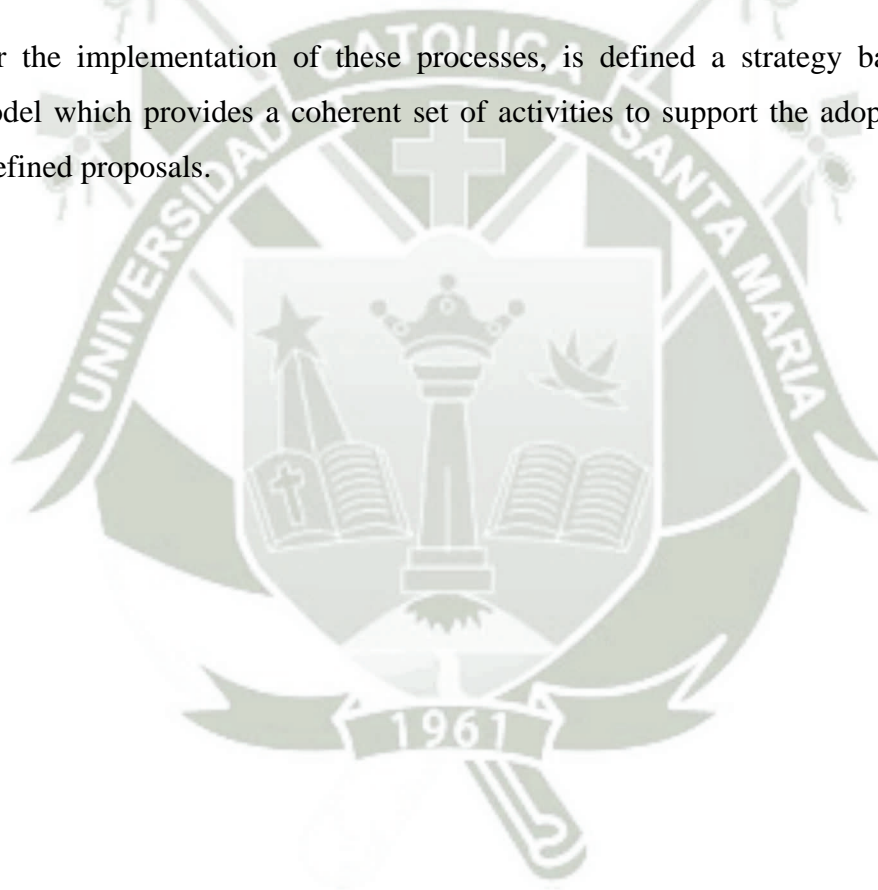


ABSTRACT

Development processes and requirements management are among the main sources of the problems encountered during the development of software projects. One of the critical aspects for the successful in the development of these processes is the use of techniques, methodologies and standards for the implementation of these activities.

In this paper is proposed an integration model whose base are the good practices of CMMI-DEV model and SWEBOK guide. Integrating both sets of knowledge, this model will specifically focus on the processes of software requirements in order to facilitate and improve the application development processes and requirements management.

Finally, for the implementation of these processes, is defined a strategy based on the IDEAL model which provides a coherent set of activities to support the adoption of best practices defined proposals.



INTRODUCCIÓN

El objetivo principal del presente trabajo es proponer un modelo de integración para desarrollar y gestionar requisitos teniendo como referencia las buenas prácticas recomendadas por el modelo CMMI-DEV “Mejora de los procesos para el desarrollo de mejores productos y servicios” y la guía SWEBOK “Guía al Cuerpo de Conocimiento de la Ingeniería del Software”.

Como parte de este trabajo se define una estrategia basada en el modelo IDEAL para implementar las buenas prácticas en la organización.

Con este trabajo se espera que las organizaciones dedicadas al desarrollo de software faciliten y mejoren la aplicación de los procesos de requisitos.

El resto de este trabajo de investigación se organiza de la siguiente manera:

El primer capítulo, Planteamiento teórico, trata sobre los aspectos generales de la investigación.

El segundo capítulo, Marco teórico, se abordan los aspectos teóricos y conceptos básicos relacionados a la investigación.

El tercer capítulo, Propuesta del modelo, se desarrolla la propuesta del modelo de integración.

El cuarto capítulo, Realidad de la empresa, se hace una breve descripción de la empresa, luego se realiza una evaluación de los procesos de requisitos actuales de la organización.

El quinto capítulo, Validación, se valida el modelo propuesto en una realidad.

Finalmente, se expone las conclusiones, sugerencias y hacia donde nos enfocamos en un futuro.

CAPITULO I: PLANTEAMIENTO TEÓRICO

1.1.TÍTULO DEL PROYECTO

Propuesta de un modelo de integración para desarrollar y gestionar requisitos en el desarrollo de proyectos de software basado en el estándar CMMI-DEV y SWEBOK.

1.2.DESCRIPCIÓN DEL PROBLEMA

El Standish Group ha logrado recopilar información sobre casos de proyectos en entornos de TI de la vida real y proyectos de desarrollo de software desde 1985. El reporte CHAOS es el informe de investigación más famoso sobre por qué los proyectos tienen éxito o fracasan. Esta investigación se basa en datos de aproximadamente 50000 proyectos. El reporte CHAOS clasifica los proyectos en éxito, fracaso o deficiente.

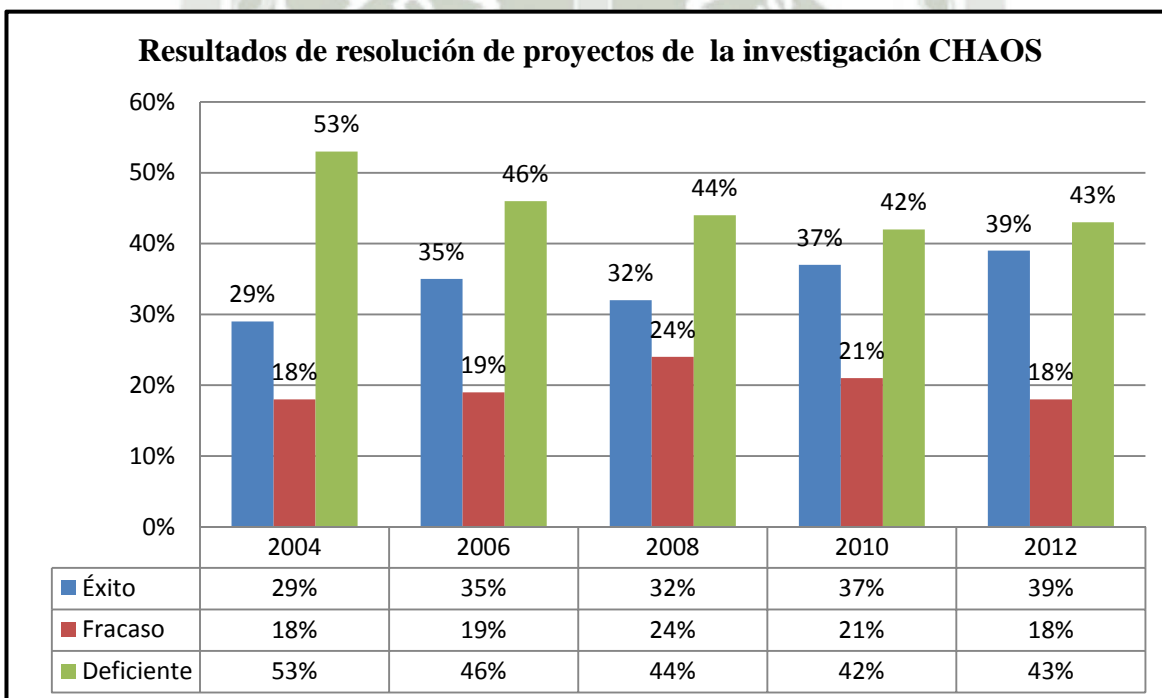


Figura 1. Resultados de resolución de proyectos de la investigación CHAOS
Fuente: [StandishGroup, 2013]

Tal como se observa en la figura 1, la edición 2013 del informe CHAOS muestra que en el año 2012 los proyectos de software tuvieron una tasa de éxito del 39% (fue entregado a tiempo, en el presupuesto y con todas sus funciones). Por otra parte, el 18% de los proyectos fracasaron (se cancelaron o se finalizaron pero el producto nunca se usó)

mientras que el 43% fueron deficientes (con retraso, por encima del presupuesto y/o con menos de los requisitos esperados).

En uno de estos estudios se encuestó a los jefes de TI para obtener sus opiniones acerca de las principales razones del fracaso de un proyecto. Estos fueron los resultados:

Tabla 1. Principales factores de fracaso en los proyectos

Principales factores de fracaso	% de Respuestas
1. Requisitos incompletos.	13.1 %
2. Deficiencia en el involucramiento del usuario.	12.4 %
3. Deficiencia de recursos.	10.6 %
4. Expectativas no realistas.	9.9%
5. Deficiencia en soporte ejecutivo.	9.3%
6. Cambios en los requisitos y especificaciones.	8.7%
7. Deficiencia en la planeación.	8.1%
8. Ya no se necesita más.	7.5%
9. Deficiencia en administración de TI.	6.2%
10. Desconocimiento en tecnología.	4.3%
11. Otros.	9.9%

Fuente: [StandishGroup, 1995]

Como se puede ver en la tabla 1, entre los factores principales que conducen al fracaso en los proyectos de software se encuentran los requisitos incompletos, la falta de participación de los usuarios, expectativas poco realistas y los cambios en los requisitos y especificaciones.

Estos factores en conjunto hacen un total del 44.1 % de las causas del fracaso de los proyectos y están relacionadas con deficiencias en las actividades que se desarrollan durante el desarrollo y gestión de requisitos.

1.3.FORMULACIÓN INTERROGATIVA DEL PROBLEMA

¿De qué manera la propuesta de un modelo de integración basado en el estándar CMMI-DEV y SWEBOK apoyaría a desarrollar y gestionar requisitos en el desarrollo de proyectos de software?

1.4.JUSTIFICACIÓN DE LA INVESTIGACIÓN

En la actualidad muchas empresas se esfuerzan por comprender, documentar y gestionar los requisitos de software. Requisitos incompletos, requisitos cambiantes, desconocimiento de los objetivos del negocio son las razones principales por las cuales muchos proyectos de desarrollo de software no terminan siendo exitosos. Para mejorar estos procesos se debe seguir modelos que proporcionen una orientación para aplicar las buenas prácticas en el desarrollo de productos y servicios de calidad.

1.5.OBJETIVOS DE LA INVESTIGACIÓN

1.5.1. Objetivo general

Proponer un modelo de integración para desarrollar y gestionar requisitos en el desarrollo de proyectos de software basado en el estándar CMMI-DEV y SWEBOK.

1.5.2. Objetivos específicos

- Analizar las buenas prácticas de requisitos de software propuestas por el modelo CMMI-DEV y la Guía del Cuerpo de Conocimientos de la Ingeniería de Software SWEBOK.
- Validar el modelo propuesto en una realidad.

1.6.HIPÓTESIS

Es probable que la propuesta de un modelo de integración para desarrollar y gestionar requisitos en el desarrollo de proyectos de software basado en el estándar CMMI-DEV y SWEBOK logre mejorar el proceso de desarrollo y gestión de requisitos en el desarrollo de proyectos de software.

1.7.VARIABLES E INDICADORES

1.7.1. Variable independiente

Modelo de integración para desarrollar y gestionar requisitos de software.

A. Indicadores

- Claridad.
- Mejora del desarrollo de requisitos.
- Mejora de la gestión de requisitos.
- Comprensión.
- Satisfacción de expectativas.

1.7.2. Variable dependiente

Desarrollo de proyectos de software.

A. Indicadores

- Desarrollo y gestión de requisitos en el desarrollo de proyectos de software.

1.8.ÁREA Y LÍNEA DE INVESTIGACIÓN

1.8.1. Área de investigación

Ingeniería de Software.

1.8.2. Línea de investigación

Procesos de desarrollo de software.

1.9. TIPO Y NIVEL DE INVESTIGACIÓN

1.9.1. Tipo de investigación

Aplicada, porque se busca la aplicación de los conocimientos a la solución de un problema práctico inmediato.

1.9.2. Nivel de investigación

Descriptiva, porque se tiene como objetivo determinar las características de un fenómeno, así como establecer relaciones entre algunas variables permitiendo tener un conocimiento actualizado del fenómeno tal como se presenta.

1.10. ÁMBITO

El ámbito de aplicación son las empresas de desarrollo de software, tomándose el muestreo no probabilístico y como caso de estudio a BanTotal-Centro de Desarrollo de software.



CAPITULO II: MARCO TEÓRICO

2.1.REVISIÓN HISTÓRICA

2.1.1. Historia y evolución del CMMI

- **1985.** El SEI (Instituto de Ingeniería del Software) empieza a trabajar en un marco de madurez de procesos que permita evaluar a las empresas productoras de software. La investigación evoluciona hacia el “Modelo de Madurez de las Capacidades (CMM)”.
- **1991.** El SEI publica la versión 1.0 del Modelo de Madurez de las Capacidades para el Software (SW-CMM, Capability Maturity Model for Software).
- **1993.** El SEI publica la versión 1.1 de SW-CMM.
- **1997.** El SEI publica la versión 2 de SW-CMM.
- **2000.** El modelo SW-CMM fue integrado y relevado por el nuevo modelo “Integración de Modelos de madurez de las Capacidades (CMMI)”. El SEI publica la versión 1.02 del CMMI.
- **2002.** El SEI publica la versión 1.1 del CMMI.
- **2006.** El SEI publica la versión 1.2 denominándola CMMI-DEV “CMMI para Desarrollo”.
- **2010.** El SEI publica la versión 1.3 del CMMI-DEV.

La figura 2 muestra la historia y evolución del CMMI.

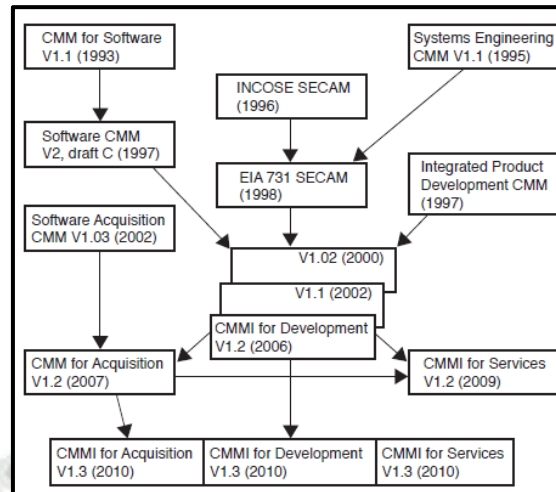


Figura 2. La historia de los CMMs
Fuente: [SEI, 2010]

2.1.2. Historia y evolución del SWEBOK

- **1993.** La IEEE Computer Society creó el SWECC (Comité para la coordinación de la ingeniería de software) dedicado a evaluar, planear y coordinar acciones relacionadas a establecer la Ingeniería de Software como una profesión.
- **1998.** El SWECC empieza a trabajar en el proyecto SWEBOK.
- **2004.** El SWECC promovido por la IEEE publica la versión 2004 de la Guía del Cuerpo de Conocimientos de la Ingeniería del Software (SWEBOK, Software Engineering Body of Knowledge).
- **2013.** El SWECC publica la versión 3 del SWEBOK.

2.1.3. Historia y evolución de IDEAL

- **1996.** El SEI publica la versión 1.0 del modelo IDEAL: A User's Guide for Software Process Improvement.
- **2001.** El SEI publica la versión 1.1 del modelo IDEAL: A Practical Guide for Improvement.

2.2. CONCEPTOS

2.2.1. Modelo

Según [García, 2013], cuando un objeto se sustituye o se representa por otro para estudiarlo en una situación diferente, al segundo objeto lo llamamos “modelo” del primero. Entendemos entonces que la acción básica de un modelo es representar o sustituir a un objeto dado.

Entonces, “el modelo es algo que sustituye a lo modelado”, para manipularlo y entender al mundo como “es”; es decir, el modelo de un fenómeno es una herramienta que se usa para describirlo, interpretarlo, predecir comportamiento en diferentes situaciones específicas, validar hipótesis y elaborar estrategias para la intervención”.

2.2.2. Requisito

Según [RAE, 2001], un requisito es una “circunstancia o condición necesaria para algo”.

2.2.2.1. Requisito de Software

Según la definición existente en el glosario de la [IEEE, 1990], un requisito de software es:

- “Una condición o capacidad necesitada por un usuario para solucionar un problema o lograr un objetivo”.
- “Una condición o capacidad que debe cumplir o poseer un producto, servicio, componente de producto o componente de servicio para satisfacer un acuerdo de proveedor, un estándar, una especificación u otros documentos impuestos formalmente”.

Según el SWEBOK [IEEE, 2013], un requisito de software es:

- “Una característica que debe ser expuesta con el fin de resolver algún problema en el mundo real”.

2.2.2.2.Desarrollo de requisitos

Según el CMMI-DEV [SEI, 2010], el propósito del Desarrollo de Requisitos es educir, analizar y establecer los requisitos de cliente, de producto y de componente de producto.

2.2.2.3.Gestión de requisitos

Según el CMMI-DEV [SEI, 2010], el propósito de la Gestión de Requisitos es gestionar los requisitos de los productos y los componentes de producto del proyecto, y asegurar la alineación entre esos requisitos, los planes y los productos de trabajo del proyecto.

2.2.3. Estándar

Según [PMI 2013], por estándar se entiende:

- “Documento formal que describe normas, métodos, procesos y prácticas establecidos”.
- “Documento que provee, para uso común y repetitivo, las reglas, pautas o características que deberían cumplir las actividades (o sus resultados), a fin de obtener un óptimo grado de orden en un contexto dado”.

Según [Sommerville, 2002], los estándares en el software son útiles porque:

- Agrupan lo mejor y más apropiado de las buenas prácticas y usos del desarrollo de software.
- Engloban los “conocimientos” que son patrimonio de una organización.
- Proporcionan un marco para implementar procedimientos de aseguramiento de la calidad.
- Proporcionan continuidad entre el trabajo de distintas personas.

2.2.4. Proyecto

Según [PMI, 2013], un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. La naturaleza temporal de los proyectos indica un principio y un final definidos. El final se alcanza cuando se logran los objetivos del proyecto o cuando se termina el proyecto porque sus objetivos no se cumplirán o no pueden ser cumplidos, o cuando ya no existe la necesidad que dio origen al proyecto.

2.2.5. Mejora de Procesos de Software

Según [Raninen, 2014], la mejora de procesos de software es un conjunto de actividades que conducen a un mejor proceso de software y por consiguiente a una entrega de software de mayor calidad de una manera más oportuna.

2.3. ANTECEDENTES INVESTIGATIVOS

- **Título:** “Propuesta de un modelo de asociación en el gerenciamiento de riesgos de proyectos informáticos basado en el estándar PMBOK Y CMMI-SW”. [Guillen, 2011]

Autor: Jorge Guillen Amado.

Institución: Universidad Católica de Santa María, Arequipa - Perú.

Año: 2011

Objetivo: Proponer un modelo de asociación en el gerenciamiento de riesgos de proyectos informáticos basados en el estándar PMBOK y CMMI-SW.

Resultado: La propuesta del modelo logro impactar a los encuestados logrando opiniones favorables con respecto a la planificación y gerenciamiento de los riesgos en los proyectos informáticos, ya que existen, en el estado del arte, distintos enfoques en la gestión de riesgos, sin embargo pocas organizaciones lo utilizan dada su complejidad y la demanda de recursos que ello implica.

Conclusiones: Mediante el modelo de asociación en el gerenciamiento de riesgos de proyectos informáticos basados en los estándares validados a nivel mundial, podemos realizar el seguimiento de los riesgos desde las fases iniciales del proyecto ahorrándonos inconvenientes y disminuyendo la probabilidad de que los proyectos informáticos fracasen.

- **Título:** “PRO-REQ: a facilitator guide to implement CMMI-Dev requirements engineering and management areas”. [Souza, Sanches y Vaccare, 2009]

Autores: Alfraino Souza, Rosely Sanches y Rosana Vaccare.

Institución: Universidad de São Paulo, São Paulo - Brasil.

Año: 2009

Objetivo: En este trabajo se presenta una guía llamada PRO-REQ, cuyo objetivo es facilitar la aplicación de los procesos de ingeniería de requisitos en las organizaciones que no pueden pagar sus altos costos. La Guía se basa en las prácticas de CMMI, pero también considera otras fuentes de buenas prácticas en la gestión e ingeniería de requisitos.

Resultado: Se extrajeron las buenas prácticas PRO-REQ de varias fuentes, pero su estructura se basa en las áreas de proceso de ingeniería y gestión de requisitos de CMMI –DEV. Por lo tanto, contiene las mismas 2 metas específicas, 10 prácticas y 51 sub- prácticas de CMMI -Dev, pero proporciona aproximadamente 200 buenas prácticas, junto con los productos de trabajo de entrada y salida previstos

Conclusiones: Utilizando partes de la guía PRO-REQ fue posible definir procesos de requisitos en una pequeña organización brasileña de software. Algunos de los proyectos llevados a cabo en esa organización después de la definición de estos procesos presentan buenos resultados, con menos re- trabajo hecho con respecto a los problemas de los requisitos, mientras que otros resultados de los proyectos no se puede comparar aún, pero ya se presentó los productos de trabajo de proceso y sus principales características, lo que caracteriza la evidencia de la ejecución del proceso.

- **Título:** “Architecting Standard Processes with SWEBOK and CMMI”. [Mutafelija y Stromberg, 2005]

Autores: Boris Mutafelija y Harvey Stromberg.

Institución: Systems and Software Consortium, Columbia - Estados Unidos.

Año: 2005

Resumen: Este artículo resume las similitudes y diferencias entre SWEBOK y CMMI y proporciona pautas para usar SWEBOK a la hora de desarrollar una arquitectura de procesos. También se abordan brevemente otros cuerpos de conocimientos existentes y emergentes, tales como PMBOK, Ingeniería de Sistemas BOK y Aseguramiento de la Calidad BOK, y su uso en el desarrollo de arquitecturas de proceso.

A partir de las investigaciones y experiencias expuestas queda en evidencia la necesidad de ahondar en el tema, incluso con todos estos conocimientos surge la necesidad de mejorar y actualizar estos conocimientos con el objetivo de minimizar los daños causados por una pobre gestión de los requisitos.

2.4. ESTADO DEL ARTE

2.4.1. Requisitos de Software según SWEBOK 3.0

El SWEBOK es una guía creada por el SWECC (Comité para la coordinación de la ingeniería del software), promovido por la IEEE Computer Society, que describe el conocimiento que existe de la disciplina de la ingeniería del software. Supone un paso esencial hacia el desarrollo de la profesión porque representa un amplio consenso respecto a los contenidos de la disciplina. Divide a la ingeniería del software en 15 áreas de conocimiento (Knowledge area o KA) tal como se observa en la tabla 2. [IEEE, 2013]

Tabla 2. Áreas de conocimiento del SWEBOK

Áreas de Conocimiento del SWEBOK
Requisitos de Software
Diseño de Software
Construcción de Software
Pruebas de Software
Mantenimiento de Software
Gestión de la Configuración del Software
Gestión de Ingeniería de Software
Ingeniería de Procesos de Software
Modelos y Métodos de Ingeniería del Software
Calidad del Software
La Práctica Profesional de la Ingeniería del Software
La Economía de la Ingeniería del Software
Fundamentos de la Informática
Fundamentos Matemáticos
Fundamentos de la Ingeniería

Fuente: [IEEE, 2013]

La guía al cuerpo de conocimiento de la ingeniería del software se estableció con los siguientes 5 objetivos: [IEEE, 2013]

- Promover una visión consistente de la Ingeniería del Software en todo el mundo.
- Para especificar el alcance de, y clarificar la posición de la Ingeniería del Software respecto a otras disciplinas, como las Ciencias de la Computación, la Gestión de proyectos, la Ingeniería informática y las Matemáticas.
- Caracterizar los contenidos de la Ingeniería del Software.
- Proveer acceso a través de las temáticas al cuerpo de conocimientos de la Ingeniería del Software.
- Proveer una base para su desarrollo curricular para la certificación individual y material autorizado.

El área del conocimiento de los requisitos del software se refiere a la educación, análisis, especificación y validación de los requisitos del software así como la gestión de los requisitos durante todo el ciclo de vida del producto de software.

Los requisitos de software expresan las necesidades y las limitaciones impuestas en un producto de software que contribuyen a la solución de un cierto problema del mundo real. [IEEE, 2013]

La figura 3 muestra la estructura del área de conocimiento de requisitos de software.

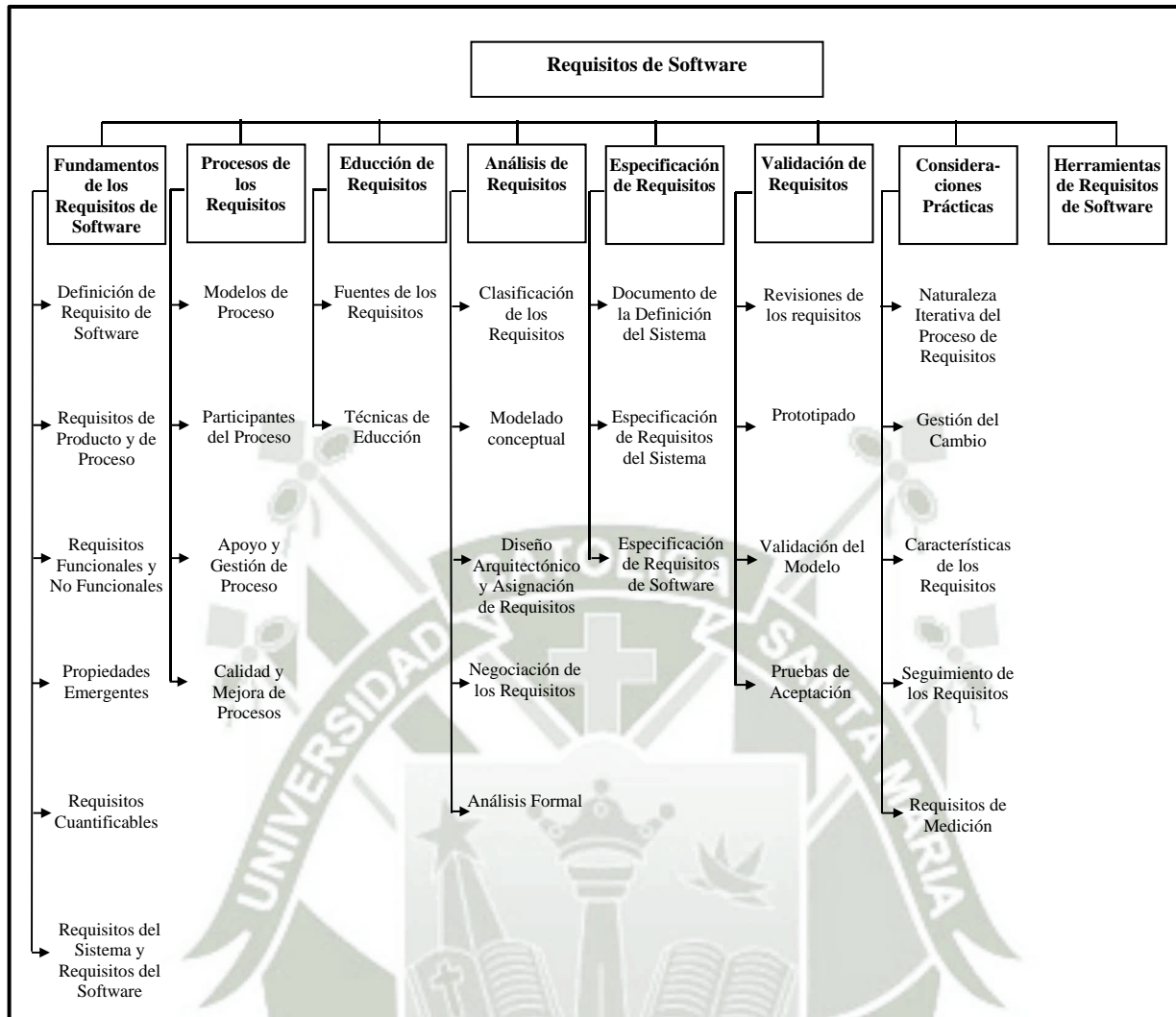


Figura 3. Distribución de temas para los Requisitos de Software
Fuente: [IEEE, 2013]

2.4.1.1. Fundamentos de los Requisitos de Software

A. Definición de Requisito de Software

Básicamente, un requisito de software es una característica que debe ser expuesta con el fin de resolver algún problema en el mundo real.

Una característica esencial de todos los requisitos de software es que sean comprobables como una característica individual como requisito funcional o a nivel del sistema como un requisito no funcional. [IEEE, 2013]

B. Requisitos de Producto y de Proceso

Un requisito de producto es una necesidad o restricción en el software a desarrollar (por ejemplo, "El software debe verificar que un estudiante cumple con todos los requisitos previos antes de que se registre en un curso").

Un requisito de proceso es esencialmente una restricción en el desarrollo del software (por ejemplo, "el software será desarrollado usando el proceso RUP"). [IEEE, 2013]

C. Requisitos Funcionales y No Funcionales

Los requisitos funcionales describen las funciones que el software va a ejecutar; por ejemplo, formatear un texto o modular una señal. Se conocen también como capacidades o características.

Los requisitos no funcionales son los que actúan para limitar la solución. Los requisitos no funcionales son a veces conocidos como restricciones o requisitos de calidad. [IEEE, 2013]

D. Propiedades Emergentes

Algunos requisitos representan propiedades emergentes de software - es decir, los requisitos que no pueden ser abordadas por un solo componente, sino que dependen de cómo todos los componentes de software interoperan. El requisito de rendimiento para un centro de llamadas, por ejemplo, dependería de cómo el sistema telefónico, el sistema de información y los operadores interactúan en condiciones reales de funcionamiento. [IEEE, 2013]

E. Requisitos Cuantificables

Los requisitos de software deben expresarse claramente e inequívocamente como sea posible, y, en su caso, en términos cuantitativos. Es importante evitar los requisitos vagos e inverificables

que dependen para su interpretación de un juicio subjetivo ("el software debe ser confiable", "el software debe ser fácil de usar"). Un ejemplo de los requisitos cuantificables es el siguiente: El software de un centro de llamadas debe aumentar el rendimiento del centro en un 20%. [IEEE, 2013]

F. Requisitos del Sistema y Requisitos del Software

Los requisitos del sistema son los requisitos para el sistema en su totalidad. En un sistema que contiene los componentes de software, los requisitos de software derivan de los requisitos del sistema.

Los requisitos del sistema abarcan los requisitos de usuario, los requisitos de otros stakeholders (como las autoridades reguladoras), y requisitos sin fuente humana identificable. [IEEE, 2013]

2.4.1.2. Proceso de los Requisitos

A. Modelos de Proceso

El objetivo de este tema es proporcionar una comprensión del proceso de los requisitos.

- Es un proceso iniciado al comienzo de un proyecto que continúa perfeccionándose durante todo el ciclo de vida.
- Identifica los requisitos de software como elementos de configuración, y los maneja usando las mismas prácticas de gestión de configuración del software como de otros productos de los procesos del ciclo de vida del software.
- Necesita ser adaptado a la organización y al contexto del proyecto.

En particular, el tema se refiere a cómo las actividades de educación, análisis, especificación y validación se configuran para los diferentes tipos de proyectos y restricciones. [IEEE, 2013]

B. Participantes del Proceso

En este tema se presentan las funciones de las personas que participan en el proceso de requisitos. Este proceso es fundamentalmente interdisciplinario, y el especialista de requisitos tiene que mediar entre el dominio del stakeholder y de la ingeniería de software.

Los ejemplos típicos de los stakeholders del software incluyen (pero no se limitan a) los siguientes: usuarios, clientes, analistas de mercado e Ingenieros de software. [IEEE, 2013]

C. Apoyo y Gestión de Proceso

Esta parte presenta los recursos de la gestión de proyectos requeridos y consumidos por el proceso de requisitos. Su propósito principal es establecer una relación entre las actividades de proceso identificadas en los Modelos de proceso y los temas de coste, recursos humanos, capacitación, y herramientas. [IEEE, 2013]

D. Calidad y Mejora de Procesos

Este tema se refiere a la evaluación de la calidad y de la mejora del proceso de los requisitos. Su propósito es enfatizar el papel principal que el proceso de requisitos desempeña en términos de coste y puntualidad de un producto de software, y de la satisfacción del cliente. Ayudará a orientar el proceso de los requisitos con estándares de calidad y modelos de mejora de procesos para el software y sistemas. [IEEE, 2013]

2.4.1.3. Educción de Requisitos

La educación de requisitos se refiere al origen de los requisitos de software y cómo el ingeniero de software puede recogerlos. Es la primera etapa en la construcción de una comprensión del problema del software que se requiere solucionar. Es fundamentalmente una actividad humana y es donde se identifican los stakeholders y donde se establecen las relaciones entre el equipo de desarrollo y el cliente. También se conoce como “captura de

requisitos,” “descubrimiento de requisitos,” y “adquisición de requisitos.” [IEEE, 2013]

A. Fuentes de los Requisitos

Los requisitos tienen muchas fuentes en el software normal, y es esencial que todas las fuentes posibles estén identificadas y evaluadas. Este tema está destinado a promover el conocimiento de las diversas fuentes de requisitos de software y de los frameworks para la gestión de los mismos. Los principales puntos tratados son los siguientes: [IEEE, 2013]

- Metas.
- Conocimiento del dominio.
- Stakeholders.
- Reglas de negocio.
- El entorno operativo.
- El entorno de la organización.

B. Técnicas de Educación

Una vez que se hayan identificado las fuentes de los requisitos, el ingeniero de software puede comenzar la obtención de información de requisitos de estas fuentes. Este tema se centra en las técnicas para obtener stakeholders humanos para expresar la información relevante de los requisitos. Una serie de técnicas existen para la educación de requisitos, los principales son los siguientes: [IEEE, 2013]

- Entrevistas.
- Escenarios.
- Prototipos.

- Reuniones facilitadas.
- Observación.
- Historias de usuario.
- Otras técnicas.

2.4.1.4. Análisis de Requisitos

Este tema se refiere al proceso de analizar requisitos para:

- Detectar y resolver conflictos entre los requisitos.
- Descubrir los límites del software y cómo debe interactuar con su entorno organizacional y operacional.
- Elaborar los requisitos del sistema para derivar los requisitos de software.

Se debe tener cuidado para describir los requisitos con exactitud suficiente para permitir que los requisitos sean validados, su implementación sea verificada, y sus costos estimados. [IEEE, 2013]

A. Clasificación de los Requisitos

Los requisitos se pueden clasificar en varios aspectos. Algunos ejemplos son los siguientes: [IEEE, 2013]

- Si el requisito es funcional o no funcional.
- Si el requisito se deriva de uno o más requisitos de alto nivel o una propiedad emergente.
- Si el requisito está en el producto o el proceso.
- La prioridad del requisito: Cuanto mayor sea la prioridad, más esencial es el requisito para el cumplimiento de los objetivos generales del software.

- El alcance del requisito: El alcance se refiere al grado en que un requisito afecta al software y a los componentes de software.
- Volatilidad / estabilidad: Algunos requisitos cambiarán durante el ciclo de vida del software, e incluso durante el propio proceso de desarrollo.

B. Modelado Conceptual

Los modelos conceptuales comprenden modelos de entidades del dominio del problema, configurado para reflejar sus relaciones y dependencias del mundo real.

Varios tipos de modelos se pueden desarrollar. Estos incluyen los diagramas de casos de uso, los modelos de flujo de datos y muchos otros. Muchas de estas notaciones de modelado son parte del Lenguaje Unificado de Modelado (UML).

Los factores que influyen en la elección de la notación de modelado incluyen lo siguiente: [IEEE, 2013]

- La naturaleza del problema: Algunos tipos de software exigen que ciertos aspectos sean analizados rigurosamente.
- La experiencia del ingeniero de software.
- Los requisitos de proceso del cliente: Los clientes pueden imponer su notación o método favorito o prohibir cualquiera que le sea desconocido.

C. Diseño Arquitectónico y Asignación de Requisitos

El diseño arquitectónico es el punto en el que el proceso de requisitos se mezcla con el diseño de software o de sistema.

La asignación es importante para permitir un análisis detallado de los requisitos. El diseño arquitectónico se identifica estrechamente con el modelado conceptual. [IEEE, 2013]

D. Negociación de los Requisitos

Esto se refiere a la resolución de problemas con requisitos donde los conflictos se producen entre dos stakeholders que requieren funciones incompatibles entre sí, entre los requisitos y los recursos, o entre los requisitos funcionales y no funcionales, por ejemplo. La priorización de requisitos es necesaria. [IEEE, 2013]

E. Análisis Formal

El Análisis formal ha hecho un impacto en algunos dominios de aplicación, en particular de los sistemas de alta integridad. La expresión formal de los requisitos exige un lenguaje con semántica definida formalmente. El uso de un análisis formal para los requisitos de expresión tiene dos beneficios. En primer lugar, permite requisitos expresados en el idioma a ser especificado de forma precisa y sin ambigüedades, por lo que (en principio) evitar la posibilidad de una mala interpretación. En segundo lugar, los requisitos se pueden razonar sobre, permitir propiedades deseadas del software especificado a ser probado. [IEEE, 2013]

2.4.1.5. Especificación de Requisitos

En la ingeniería de software, "la especificación de requisitos de software" se refiere normalmente a la producción de un documento que pueda ser revisado de forma sistemática, evaluada y aprobada. Para los sistemas complejos, particularmente aquellos que involucran componentes sustanciales no-software, se producen hasta tres diferentes tipos de documentos: definición del sistema, requisitos del sistema y requisitos de software. [IEEE, 2013]

A. Documento de la Definición del Sistema

Este documento (a veces conocido como documento de requisitos de usuario o documento de concepto de operaciones) registra los requisitos del sistema. En él se definen los requisitos del sistema de alto nivel desde la perspectiva de dominio. El documento enumera los requisitos del sistema, junto con la información de fondo acerca de los objetivos generales del sistema, su entorno de destino, y una declaración de las restricciones, hipótesis y requisitos no funcionales. [IEEE, 2013]

B. Especificación de Requisitos del Sistema

En este punto, se especifican los requisitos del sistema, los requisitos de software se derivan de los requisitos del sistema, y luego se especifican los requisitos para los componentes de software. [IEEE, 2013]

C. Especificación de Requisitos de Software

La especificación de requisitos de software establece la base para un acuerdo entre los clientes y los contratistas o proveedores en lo que el producto de software debe hacer, así como lo que no se espera que haga.

La especificación de requisitos de software permite una evaluación rigurosa de los requisitos antes de que pueda comenzar el diseño y reduce el rediseño más tarde. También proporciona una base informada para la transferencia de un producto de software para nuevos usuarios o plataformas de software. Por último, puede proporcionar una base para la mejora de software. [IEEE, 2013]

2.4.1.6. Validación de Requisitos

Los documentos de requisitos pueden estar sujetos a los procedimientos de validación y verificación. Los requisitos pueden ser validados para asegurar que el ingeniero de software ha entendido los requisitos, es también importante verificar que un documento de requisitos este conforme a los estándares de la compañía y que sea comprensible, coherente y completo.

Los diferentes stakeholders, incluidos los representantes del cliente y el desarrollador, deben revisar el documento(s).

La validación de requisitos tiene que ver con el proceso de examinar el documento de requisitos para garantizar que se defina el software adecuado (es decir, el software que los usuarios esperan). [IEEE, 2013]

A. Revisiones de los Requisitos

Un grupo de examinadores es asignado en breve para buscar errores, suposiciones erróneas, falta de claridad y desviación de la práctica estándar. La composición del grupo que lleva a cabo la revisión es importante (al menos un representante del cliente debe ser incluido para el proyecto), y esto puede ayudar a proporcionar orientación sobre lo que debe buscar en forma de listas de verificación. [IEEE, 2013]

B. Prototipado

El Prototipo es comúnmente un medio para validar la interpretación del ingeniero de software de los requisitos de software, así como para la obtención de nuevos requisitos. La ventaja de los prototipos es que pueden hacer más fácil la interpretación de las hipótesis de los ingenieros de software y, en caso necesario, dar información útil sobre por qué están equivocados. [IEEE, 2013]

C. Validación del Modelo

Por lo general es necesario para validar la calidad de los modelos desarrollados durante el análisis. Por ejemplo, en modelos de objetos, es útil llevar a cabo un análisis estable para verificar la existencia de rutas de comunicación entre los objetos, dominio de los stakeholders, intercambio de datos. [IEEE, 2013]

D. Pruebas de Aceptación

Una propiedad esencial de los requisitos de software es que debe ser posible validar que el producto final satisfaga esto. Los requisitos que no se pueden validar en realidad sólo son "deseos." Por tanto, una tarea importante es la planificación de cómo verificar cada requisito. En la mayoría de los casos, el diseño de las pruebas de aceptación se hace por cómo los usuarios finales suelen realizar negocios utilizando el sistema. [IEEE, 2013]

2.4.1.7. Consideraciones Prácticas

El proceso de requisitos abarca todo el ciclo de vida del software. Cambia la gestión y el mantenimiento de los requisitos en un estado que refleja con precisión el software a construir, o que se ha construido, son la clave para el éxito del proceso de ingeniería de software.

La documentación de los requisitos y la gestión del cambio son la clave para el éxito de cualquier proceso de requisitos. [IEEE, 2013]

A. Naturaleza Iterativa del Proceso de Requisitos

Hay una presión general en la industria de software para que los ciclos de desarrollo cada vez sean más cortos.

Los requisitos normalmente iteran hacia un nivel de calidad y detalle que es suficiente para permitir las decisiones, el diseño y adquisición a tomar. En algunos proyectos, esto puede dar lugar a que los requisitos se establezcan antes de que se hayan comprendido todas sus propiedades. Se corre el riesgo de un reproceso costoso si surgen problemas al final del proceso de ingeniería de software.

En casi todos los casos, la comprensión de los requisitos sigue evolucionando como diseño y continuo desarrollo. Esto a menudo conduce a la revisión de los requisitos al final del ciclo de vida. Tal vez

el punto más importante en la comprensión de los requisitos de software es que una proporción significativa de los requisitos cambiará.

El cambio tiene que ser manejado para asegurar que los cambios propuestos pasan por una revisión definida, por el proceso de aprobación y por la aplicación cuidadosa de seguimiento de requisitos, análisis de impacto, y la gestión de configuración de software. Por lo tanto, el proceso de requisitos no es sólo una tarea para el usuario en el desarrollo de software, se extiende por todo el ciclo de vida del software. En un proyecto típico, las actividades de requisitos de software evolucionan con el tiempo de la educación a la gestión del cambio. [IEEE, 2013]

B. Gestión del Cambio

La gestión del cambio es fundamental para la gestión de requisitos. En este tema se describe el papel de la gestión del cambio, los procedimientos que deben estar en su lugar, y el análisis que se debe aplicar a los cambios propuestos. [IEEE, 2013]

C. Características de los Requisitos

Los requisitos deben consistir no sólo de una especificación de lo que se requiere, sino también de información auxiliar, que ayude a manejar e interpretar los requisitos. Las características de los requisitos deben ser definidas, registradas y actualizadas como el software bajo desarrollo o evolución de mantenimiento. La característica más importante de los requisitos, sin embargo, es un identificador que permite a los requisitos ser identificados de forma única e inequívoca. [IEEE, 2013]

D. Seguimiento de los Requisitos

El seguimiento de requisitos tiene que ver con la recuperación de la fuente de los requisitos y la predicción de los efectos de los requisitos. El seguimiento es fundamental para la realización de análisis de impacto cuando los requisitos cambian.

El mantenimiento de una matriz gráfica o de trazabilidad- es una actividad que debe ser considerada durante todo el ciclo de vida de un producto. Si la información de trazabilidad no se actualiza y los cambios en los requisitos siguen ocurriendo, la información de trazabilidad se convierte en poco confiable para el análisis de impacto. [IEEE, 2013]

E. Requisitos de Medición

En la práctica, suele ser útil tener algún concepto de "cantidad" de los requisitos para un producto de software en particular. Este número es útil para evaluar el "tamaño" de un cambio en los requisitos, en la estimación del coste de desarrollo o en la tarea de mantenimiento, o simplemente para su uso común en otras mediciones. [IEEE, 2013]

2.4.1.8. Herramientas de Requisitos de Software

Las herramientas para hacer frente a los requisitos de software se dividen en dos categorías: herramientas para el modelado y las herramientas para la gestión de requisitos.

Las herramientas de gestión de requisitos normalmente soportan una gama de actividades, incluyendo la documentación, seguimiento, y la gestión del cambio. De hecho, el seguimiento y la gestión del cambio son en realidad posibles si son apoyados de una herramienta. La gestión de requisitos es fundamental para la buena práctica de los requisitos. [IEEE, 2013]

2.4.2. Desarrollo de Requisitos según CMMI-DEV 1.3

El modelo CMMI-DEV es un modelo desarrollado por el Software Engineering Institute que proporciona 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. [SEI, 2010]

CMMI da soporte a dos caminos de mejora denominadas “representaciones” usando niveles: [SEI, 2010]

- Representación continua: Permite a las organizaciones mejorar de forma incremental los procesos que corresponden a un área de proceso individual (o grupo de áreas de proceso) seleccionada por la organización.

El uso de la representación continua permite alcanzar “niveles de capacidad”.

- Representación por etapas: Permite a las organizaciones mejorar un conjunto de procesos relacionados tratando, de forma incremental, conjuntos sucesivos de áreas de proceso.

El uso de la representación por etapas permite alcanzar “niveles de madurez”.

La figura 4 muestra la estructura de las representaciones continua y por etapas.

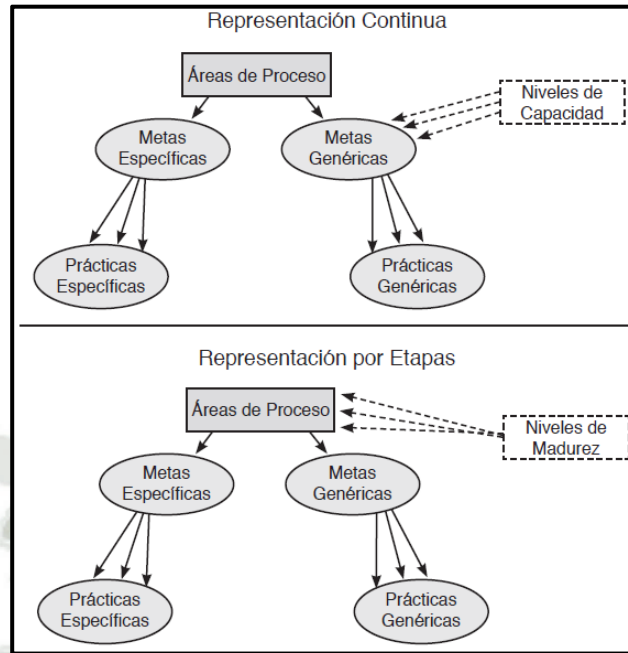


Figura 4. Estructura de las representaciones continua y por etapas
Fuente: [SEI, 2010]

CMMI para Desarrollo aborda las prácticas que cubren el ciclo de vida del producto desde la concepción hasta la entrega y el mantenimiento.

CMMI-DEV contiene 22 áreas de proceso tal como se muestra en la tabla 3.

Tabla 3. Áreas de proceso, categorías y niveles de madurez

Área de Proceso	Categoría	Niveles de madurez
Análisis Causal y Resolución (CAR)	Soporte	5
Gestión de Configuración (CM)	Soporte	2
Análisis de Decisiones y Resolución (DAR)	Soporte	3
Gestión Integrada del Proyecto (IPM)	Gestión de proyectos	3
Medición y Análisis (MA)	Soporte	2
Definición de Procesos de la Organización (OPD)	Gestión de procesos	3
Enfoque en Procesos de la Organización (OPF)	Gestión de procesos	3
Gestión del Rendimiento de la Organización (OPM)	Gestión de procesos	5
Rendimiento de Procesos de la Organización (OPP)	Gestión de procesos	4
Formación en la Organización (OT)	Gestión de procesos	3
Integración del Producto (PI)	Ingeniería	3
Monitorización y Control del Proyecto (PMC)	Gestión de proyectos	2
Planificación del Proyecto (PP)	Gestión de proyectos	2
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)	Soporte	2
Gestión Cuantitativa del Proyecto (QPM)	Gestión de proyectos	4
Desarrollo de Requisitos (RD)	Ingeniería	3
Gestión de Requisitos (REQM)	Gestión de proyectos	2
Gestión de Riesgos (RSKM)	Gestión de proyectos	3
Gestión de Acuerdos con Proveedores (SAM)	Gestión de proyectos	2
Solución Técnica (TS)	Ingeniería	3
Validación (VAL)	Ingeniería	3
Verificación (VER)	Ingeniería	3

Fuente: [SEI, 2010]

Un área de proceso (Ver figura 5) es un grupo de prácticas relacionadas dentro de un área que, cuando se implementan conjuntamente, satisface un conjunto de metas consideradas importantes para mejorar esa área. Está clasificada en 4 categorías: Ingeniería, Gestión de proyectos, Soporte y Gestión de procesos. [SEI, 2010]

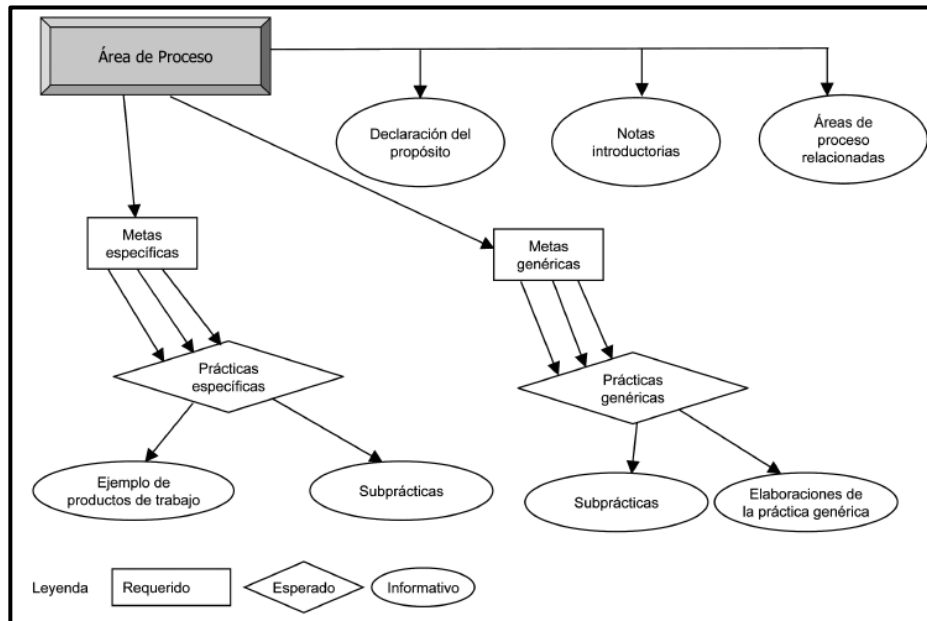


Figura 5. Componentes del modelo CMMI
Fuente: [SEI, 2010]

El área de proceso de Desarrollo de Requisitos describe tres tipos de requisitos: de cliente, de producto y de componente de producto.

Todos los proyectos de desarrollo tienen requisitos. Los requisitos son la base para el diseño. El desarrollo de los requisitos incluye las siguientes actividades: [SEI, 2010]

- Educción, análisis, validación y comunicación de las necesidades, las expectativas y las restricciones del cliente para obtener los requisitos priorizados de cliente que constituyen una comprensión de lo que satisfará a las partes interesadas.
- Recopilación y coordinación de las necesidades de las partes interesadas.
- Desarrollo de los requisitos del ciclo de vida del producto.
- Establecimiento de los requisitos funcionales de cliente y de los requisitos de los atributos de calidad.

- Establecimiento de los requisitos iniciales de producto y de componente de producto consistentes con los requisitos de cliente.

La figura 6 muestra la estructura del área de proceso de Desarrollo de Requisitos.

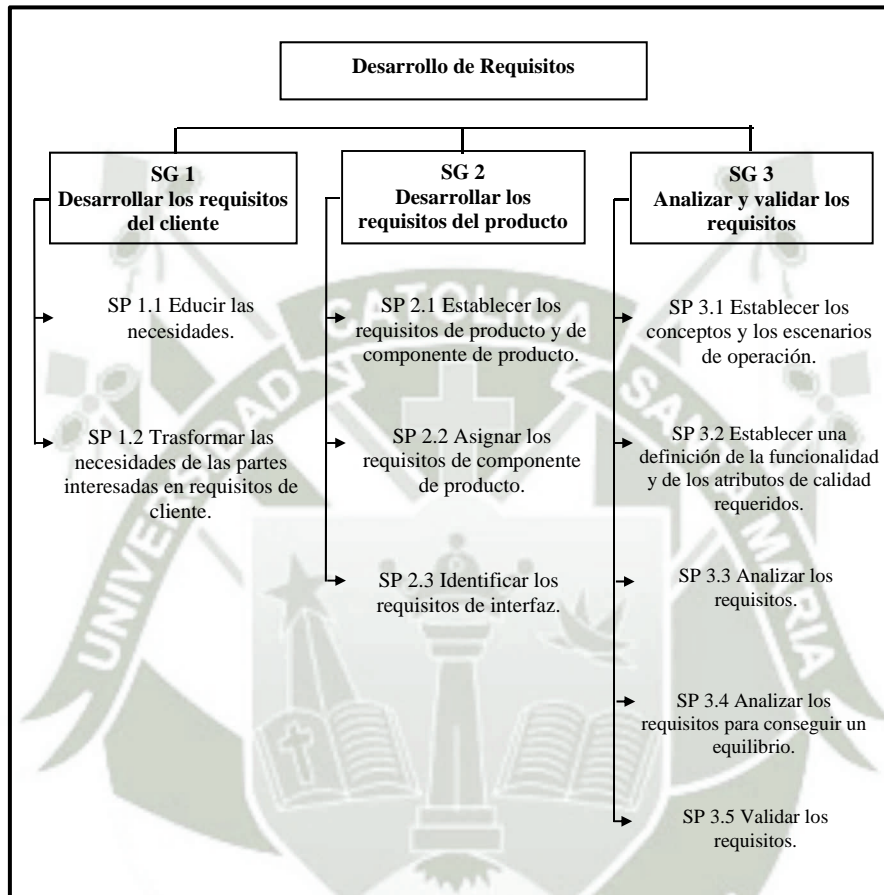


Figura 6. Resumen de Metas y Prácticas Específicas

Fuente: [SEI, 2010]

Adaptado por: [Elaboración propia]

Prácticas Específicas por Meta

2.4.2.1. Desarrollar los Requisitos del Cliente (SG 1)

Las necesidades de las partes interesadas (p. ej., clientes, usuarios finales, proveedores, desarrolladores, personal de pruebas, fabricantes, personal de soporte logístico) son la base para determinar los requisitos de cliente. Las

necesidades, las expectativas, las restricciones, las interfaces, los conceptos operacionales y los conceptos de producto de las partes interesadas se analizan, ajustan, refinan y elaboran para traducirlos en un conjunto de requisitos de cliente. [SEI, 2010]

A. Educir las Necesidades (SP 1.1)

Educir va más allá de la recopilación de requisitos mediante la identificación proactiva de requisitos adicionales no explícitamente proporcionados por los clientes. Los requisitos adicionales deberían tratar las distintas actividades del ciclo de vida del producto y sus impactos en el producto. [SEI, 2010]

Algunos ejemplos de técnicas para educir las necesidades son:

- Cuestionarios, entrevistas y escenarios (operacional, soporte y desarrollo) obtenidos de los usuarios finales.
- Prototipos y modelos.
- Tormenta de ideas.
- Pruebas beta.
- Casos de uso.
- Historias de usuario.

Algunos ejemplos de fuentes de requisitos que pueden no ser identificadas por el cliente son:

- Políticas de negocio.
- Estándares.
- Tecnología.

Ejemplo de productos de trabajo

1. Resultados de las actividades de educación de requisitos

Subprácticas

1. Comprometer a las partes interesadas relevantes usando métodos para educir las necesidades, las expectativas, las restricciones y las interfaces externas.

B. Transformar las Necesidades de las Partes Interesadas en Requisitos de Cliente (SP 1.2)

Se deberían consolidar las distintas entradas provenientes de las partes interesadas relevantes, se debería recuperar la información que falta y se deberían resolver los conflictos a medida que se desarrollan y priorizan los requisitos de cliente. Los requisitos de cliente pueden incluir las necesidades, las expectativas y las restricciones con respecto a la verificación y a la validación. [SEI, 2010]

Ejemplos de productos de trabajo

1. Requisitos de cliente priorizados.
2. Restricciones de cliente para llevar a cabo la verificación.
3. Restricciones de cliente para llevar a cabo la validación.

Subprácticas

1. Traducir las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas en requisitos documentados del cliente.
2. Establecer y mantener una priorización de los requisitos funcionales de cliente y de los atributos de calidad.
3. Definir las restricciones para la verificación y la validación.

2.4.2.2. Desarrollar los Requisitos de Producto (SG 2)

Los requisitos de cliente se analizan conjuntamente con el desarrollo del concepto de operación para inferir conjuntos de requisitos más detallados y precisos llamados “requisitos de producto y de componente de producto”. Los requisitos de producto y de componente de producto tratan las necesidades asociadas con cada fase del ciclo de vida del producto. [SEI, 2010]

A. Establecer los Requisitos de Producto y de Componente de Producto (SP 2.1)

Los requisitos funcionales y los atributos de calidad del cliente pueden expresarse en términos del cliente y pueden ser descripciones no técnicas. Los requisitos de producto son la expresión de estos requisitos en términos técnicos que pueden utilizarse para las decisiones de diseño.

Los requisitos de producto y de componente de producto tratan la satisfacción del cliente, el negocio, y los objetivos del proyecto y sus atributos asociados, tales como eficacia y asequibilidad. [SEI, 2010]

Ejemplos de productos de trabajo

1. Requisitos derivados.
2. Requisitos de producto.
3. Requisitos de componente de producto.
4. Requisitos de arquitectura, que especifican o restringen las relaciones entre componentes de producto.

Subprácticas

1. Desarrollar los requisitos en los términos técnicos necesarios para el diseño del producto y de los componentes de producto.
2. Inferir los requisitos resultantes de las decisiones de diseño.

3. Desarrollar los requisitos de arquitectura capturando los atributos críticos de calidad y las medidas de atributos de calidad necesarios para establecer la arquitectura y el diseño del producto.

Algunos ejemplos de medidas de los atributos de calidad son:

- Respuesta dentro de 1 segundo.
 - Disponibilidad del sistema en un 99% del tiempo.
4. Establecer y mantener las relaciones entre los requisitos para su consideración durante la gestión del cambio y la asignación de los requisitos.

B. Asignar los Requisitos de Componente de Producto (SP 2.2)

La arquitectura del producto proporciona la base para asignar los requisitos de producto a los componentes de producto. Los requisitos para los componentes de producto de la solución definida incluyen la asignación de la prestación del producto; las restricciones de diseño; y el ajuste, la forma y la función para cumplir los requisitos y facilitar la producción. [SEI, 2010]

Ejemplos de productos de trabajo

1. Hojas de asignación de requisitos.
2. Asignaciones provisionales de requisitos.
3. Restricciones de diseño.
4. Requisitos inferidos.
5. Relaciones entre requisitos inferidos.

Subprácticas

1. Asignar los requisitos a las funciones.
2. Asignar los requisitos a los componentes de producto y a la arquitectura.
3. Asignar las restricciones de diseño a componentes de producto y a la arquitectura.
4. Asignar requisitos a las entregas incrementales.
5. Documentar las relaciones entre requisitos asignados.

C. Identificar los Requisitos de Interfaz (SP 2.3)

Se identifican las interfaces entre las funciones (o entre objetos u otras entidades lógicas).

Se definen los requisitos de interfaz entre los productos o componentes de producto identificados en la arquitectura del producto. Estos requisitos se controlan como parte de la integración del producto y del componente de producto y son una parte integral de la definición de la arquitectura. [SEI, 2010]

Ejemplo de productos de trabajo

1. Requisitos de interfaz.

Subprácticas

1. Identificar las interfaces tanto externas como internas al producto (p. ej., entre las particiones funcionales u objetos).
2. Desarrollar los requisitos para las interfaces identificadas.

2.4.2.3. Analizar y Validar Requisitos (SG 3)

Las prácticas específicas asociadas a esta meta específica cubren el análisis y la validación de los requisitos con respecto al entorno previsto del usuario final. Los análisis se realizan para determinar qué impacto tendrá el entorno de operación previsto sobre la capacidad para satisfacer las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas.

Los requisitos se validan para incrementar la probabilidad de que el producto resultante funcione según lo previsto en el entorno de uso. [SEI, 2010]

A. Establecer los Conceptos y los Escenarios de Operación (SP 3.1)

Un escenario es normalmente una secuencia de eventos que podrían ocurrir en el desarrollo, uso o soporte del producto, el cual se usa para hacer explícitas algunas de las necesidades funcionales o de los atributos de calidad de las partes interesadas. Por el contrario, un concepto operacional para un producto depende generalmente tanto de la solución de diseño como del escenario.

Los conceptos y los escenarios de operación documentan la interacción de los componentes de producto con el entorno, con los usuarios finales y con otros componentes de producto, independientemente de la disciplina de ingeniería. Estos conceptos se deberían documentar para todos los modos y los estados dentro de las operaciones, del desarrollo del producto, del despliegue, de la entrega, del soporte (incluyendo el mantenimiento y soporte), de la formación y de la retirada. [SEI, 2010]

Ejemplos de productos de trabajo

1. Concepto operacional.
2. Desarrollo, instalación, operación, mantenimiento y conceptos de soporte del producto o componente de producto.

3. Conceptos de retirada.
4. Casos de uso.
5. Escenarios de cronología.
6. Nuevos requisitos.

Subprácticas

1. Desarrollar los conceptos y los escenarios de operación que incluyan operaciones, instalación, desarrollo, mantenimiento, soporte y retirada según sea apropiado.
2. Definir el entorno en el que funcionará el producto o componente de producto, incluyendo los límites y las restricciones.
3. Revisar los conceptos y los escenarios de operación para refinar y descubrir requisitos.
4. Desarrollar un concepto de operación detallado, a medida que se seleccionan los productos y los componentes de producto, que define la interacción del producto, del usuario final y del entorno, y que satisfaga las necesidades operativas, de mantenimiento, de soporte y de retirada.

B. Establecer una Definición de la Funcionalidad y de los Atributos de Calidad Requeridos (SP 3.2)

Una aproximación para definir la funcionalidad y los atributos de calidad requeridos es analizar escenarios utilizando lo que llamamos “análisis funcional” para describir lo que se pretende que haga el producto.

Esta descripción funcional puede incluir acciones, secuencia, entradas, salidas u otra información que comunique la forma en la que se utilizará el producto.

Algunos atributos de calidad que surgirán serán significativos para la arquitectura y así conducirán el desarrollo de la arquitectura del producto. [SEI, 2010]

Ejemplos de productos de trabajo

1. Definición de funcionalidad y de atributos de calidad requeridos.
2. Arquitectura funcional.
3. Diagramas de actividad y casos de uso.
4. Análisis orientado a objetos con servicios o métodos identificados.
5. Requisitos de los atributos de calidad significativos para la arquitectura.

Subprácticas

1. Determinar la misión clave y los factores de negocio.
2. Identificar la funcionalidad y los atributos de calidad deseados.
3. Determinar los atributos de calidad significativos para la arquitectura en base a la misión clave y los factores del negocio.
4. Analizar y cuantificar la funcionalidad requerida por los usuarios finales.
5. Analizar los requisitos para identificar las particiones lógicas o funcionales (p. ej., subfunciones).
6. Dividir los requisitos en grupos, en base a los criterios establecidos (p. ej., funcionalidad similar, requisitos similares de atributos de calidad, acoplamiento) para facilitar y enfocar el análisis de requisitos.

7. Asignar los requisitos de cliente a las particiones funcionales, objetos, personal o elementos de soporte para dar apoyo a la síntesis de las soluciones.
8. Asignar los requisitos a las funciones y a las subfunciones (u otras entidades lógicas).

C. Analizar los Requisitos (SP 3.3)

Teniendo en cuenta el concepto y los escenarios de operación, los requisitos para el nivel uno de la jerarquía del producto se analiza para determinar si son necesarios y suficientes para cumplir con los objetivos de niveles más altos de la jerarquía del producto. [SEI, 2010]

Ejemplos de productos de trabajo

1. Informes de defectos de los requisitos.
2. Cambios propuestos a los requisitos para resolver defectos.
3. Requisitos clave.
4. Medidas de rendimiento técnico.

Subprácticas

1. Analizar las necesidades, las expectativas, las restricciones y las interfaces externas de las partes interesadas para organizarlas en temas relacionados y eliminar conflictos.
2. Analizar los requisitos para determinar si satisfacen los objetivos de los requisitos de nivel más alto.
3. Analizar los requisitos para asegurarse que son completos, factibles, realizables y verificables.
4. Identificar los requisitos clave que tienen una fuerte influencia en el coste, el calendario, el rendimiento o el riesgo.

5. Identificar las medidas técnicas de rendimiento que serán seguidas durante el esfuerzo de desarrollo.
6. Analizar los conceptos y los escenarios de operación para refinar las necesidades, las restricciones y las interfaces del cliente, y para Inferir nuevos requisitos.

D. Analizar los Requisitos para Conseguir un Equilibrio (SP 3.4)

Las necesidades y las restricciones de las partes interesadas pueden tratar temas como coste, calendario, rendimiento del proyecto o producto, funcionalidad, prioridades, componentes reutilizables, facilidad de mantenimiento o riesgo. [SEI, 2010]

Ejemplo de productos de trabajo típicos

1. Evaluación de los riesgos relativos a los requisitos.

Subprácticas

1. Usar modelos, simulaciones y prototipos probados para analizar el equilibrio entre las necesidades y las restricciones de las partes interesadas.
2. Realizar una evaluación de riesgos sobre los requisitos y la definición de funcionalidad y atributos de calidad requeridos.
3. Examinar los conceptos del ciclo de vida del producto en cuanto a los impactos de los requisitos en los riesgos.
4. Evaluar el impacto de los requisitos de los atributos de calidad significativos para la arquitectura en el producto y en los costes y riesgos del desarrollo del producto.

E. Validar los Requisitos (SP 3.5)

La validación de los requisitos se ejecuta en etapas tempranas del desarrollo con los usuarios finales para ganar confianza en que los requisitos sean capaces de guiar un desarrollo que dé como resultado una validación final de éxito. [SEI, 2010]

Algunos ejemplos de técnicas usadas para la validación de los requisitos son:

- Análisis.
- Simulaciones.
- Prototipos.
- Demostraciones.

Ejemplo de productos de trabajo

1. Registro de los métodos y resultados del análisis.

Subprácticas

1. Analizar los requisitos para determinar el riesgo de que el producto resultante no funcione apropiadamente en el entorno de uso previsto.
2. Explorar la adecuación y la completitud de los requisitos desarrollando representaciones del producto (p. ej., prototipos, simulaciones, modelos, escenarios, guiones gráficos) y obteniendo realimentación sobre ellos de las partes interesadas relevantes.
3. Evaluar el diseño a medida que madure en el contexto del entorno de validación de los requisitos para identificar las cuestiones de validación, y para exponer necesidades y requisitos de cliente sin especificar.

2.4.3. Gestión de Requisitos según CMMI-DEV 1.3

Los procesos de gestión de requisitos gestionan todos los requisitos recibidos o generados por el proyecto, incluyendo tanto los requisitos técnicos como los no técnicos, así como los requisitos impuestos al proyecto por la organización. [SEI, 2010]

Cuando un proyecto recibe requisitos de un proveedor de requisitos aprobado, éstos se revisan con dicho proveedor para resolver las cuestiones y para prevenir malentendidos antes de que los requisitos se incorporen en los planes del proyecto. Una vez que el proveedor y el receptor de los requisitos alcanzan un acuerdo, se obtiene un compromiso sobre los requisitos por parte de los participantes en el proyecto. El proyecto gestiona los cambios a los requisitos a medida que evolucionan e identifica inconsistencias que ocurren entre los planes, los productos de trabajo y los requisitos. [SEI, 2010]

Una parte de la gestión de requisitos es documentar los cambios a los requisitos y su análisis razonado, y mantener la trazabilidad bidireccional entre los requisitos fuente, todos los requisitos de producto y de componente de producto, y otros productos de trabajo especificados. [SEI, 2010]

La figura 7 muestra la estructura del área de proceso de Gestión de Requisitos.

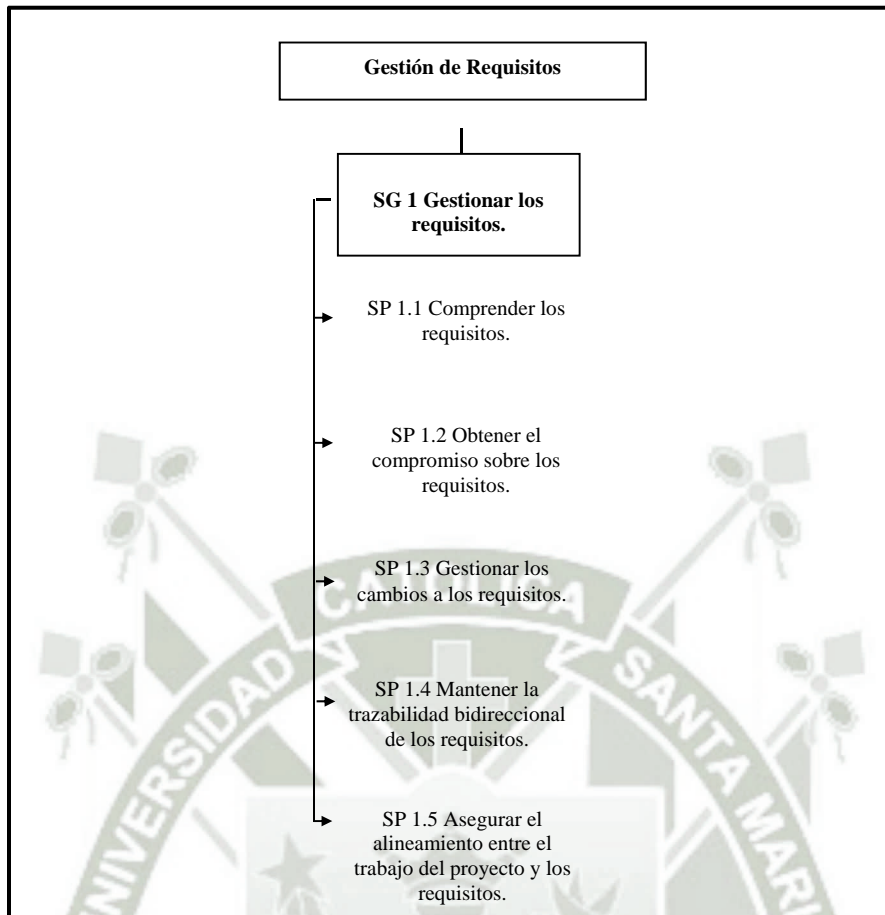


Figura 7. Resumen de Metas y Prácticas Específicas

Fuente: [SEI, 2010]

Adaptado por: [Elaboración Propia]

Prácticas Específicas por Meta

2.4.3.1. Gestionar los Requisitos (SG 1)

El proyecto mantiene un conjunto actual y aprobado de requisitos durante la vida del proyecto haciendo lo siguiente: [SEI, 2010]

- Gestionando todos los cambios a los requisitos.
- Manteniendo las relaciones entre los requisitos, los planes del proyecto y los productos de trabajo.

- Asegurando la alineación entre los requisitos, los planes del proyecto y los productos de trabajo.
- Tomando acciones correctivas.

A. Comprender los Requisitos (SP 1.1)

A medida que madura el proyecto y se derivan los requisitos, todas las actividades o disciplinas recibirán requisitos. Para evitar el flujo continuo de requisitos, se establecen criterios para designar los canales apropiados o las fuentes oficiales desde las que se reciben los requisitos. Aquellos que reciben los requisitos, los analizan con el proveedor para asegurar que se alcanza una comprensión compatible y compartida del significado de los requisitos. El resultado de estos análisis y diálogos es un conjunto de requisitos aprobados. [SEI, 2010]

Ejemplos de productos de trabajo

1. Listas de criterios para distinguir a los proveedores apropiados de requisitos.
2. Criterios para la evaluación y la aceptación de los requisitos.
3. Resultados del análisis frente a los criterios.
4. Un conjunto de requisitos aprobados.

Subprácticas

1. Establecer criterios para distinguir a los proveedores apropiados de requisitos.
2. Establecer criterios objetivos para la evaluación y aceptación de los requisitos.
3. Analizar los requisitos para asegurar que se cumplen los criterios establecidos.

4. Alcanzar una comprensión de los requisitos con los proveedores de requisitos para que los participantes del proyecto puedan comprometerse con ellos.

B. Obtener el Compromiso Sobre los Requisitos (SP 1.2)

La práctica específica anterior se ocupó de alcanzar una comprensión con los proveedores de los requisitos. Esta práctica específica se ocupa de los acuerdos y compromisos entre aquellos que llevan a cabo las actividades necesarias para implementar los requisitos. Los requisitos evolucionan a lo largo del proyecto. A medida que los requisitos evolucionan, esta práctica específica asegura que los participantes del proyecto se comprometen con los requisitos actuales y aprobados, y con los cambios resultantes en los planes, actividades y productos de trabajo del proyecto. [SEI, 2010]

Ejemplos de productos de trabajo

1. Evaluaciones del impacto de los requisitos.
2. Compromisos documentados de los requisitos y de sus cambios.

Subprácticas

1. Evaluar el impacto de los requisitos sobre los compromisos existentes.
2. Negociar y registrar los compromisos.

C. Gestionar los Cambios a los Requisitos (SP 1.3)

Los requisitos cambian por diversas razones. A medida que cambian las necesidades y avanza el trabajo, es posible que se tengan que hacer cambios a los requisitos existentes. Es esencial gestionar estas adiciones y cambios, eficiente y eficazmente. Para analizar con eficacia el impacto de los cambios, es necesario que se conozca la fuente de cada requisito y

que esté documentado el análisis razonado de cualquier cambio. El proyecto puede querer seguir medidas apropiadas de volatilidad de los requisitos para juzgar si es necesario un enfoque nuevo o modificado para el control de cambios. [SEI, 2010]

Ejemplos de productos de trabajo

1. Petición de cambio de requisitos.
2. Informes de impacto del cambio de requisitos.
3. Estado de los requisitos.
4. Base de datos de requisitos.

Subprácticas

1. Documentar todos los requisitos y los cambios a los requisitos que se reciben o generan por el proyecto.
2. Mantener una historia de cambios de los requisitos, incluyendo el análisis razonado de los cambios.
3. Evaluar el impacto de los cambios de requisitos desde el punto de vista de las partes interesadas relevantes.
4. Poner a disposición del proyecto los requisitos y los datos de los cambios.

D. Mantener la Trazabilidad Bidireccional de los Requisitos (SP 1.4)

La intención de esta práctica específica es mantener la trazabilidad bidireccional de los requisitos. Cuando se gestionan bien los requisitos, se puede establecer la trazabilidad desde un requisito fuente hasta sus requisitos de más bajo nivel y desde estos requisitos de más bajo nivel de vuelta hasta sus requisitos fuente. Esta trazabilidad bidireccional ayuda a determinar si todos los requisitos fuente se han tratado totalmente y si

todos los requisitos de nivel más bajo pueden trazarse hacia una fuente válida.

La trazabilidad de los requisitos también cubre las relaciones a otras entidades, tales como productos de trabajo intermedios y finales, cambios en la documentación de diseño y planes de pruebas. La trazabilidad puede cubrir relaciones horizontales, tales como relaciones entre interfaces, así como relaciones verticales. La trazabilidad es particularmente necesaria al evaluar el impacto de los cambios de los requisitos sobre las actividades del proyecto y los productos de trabajo. [SEI, 2010]

Algunos ejemplos de aspectos de trazabilidad a considerar son:

- Alcance de la trazabilidad: los límites dentro de los cuales es necesaria la trazabilidad.
- Definición de trazabilidad: los elementos que necesitan relaciones lógicas.
- Tipos de trazabilidad: cuándo es necesaria la trazabilidad horizontal y la vertical.

Ejemplos de productos de trabajo

1. Matriz de trazabilidad de los requisitos.
2. Sistema de seguimiento de los requisitos.

Subprácticas

1. Mantener la trazabilidad de los requisitos para asegurar que la fuente de los requisitos de nivel más bajo (es decir, inferidos) está documentada.
2. Mantener la trazabilidad de los requisitos desde un requisito a sus requisitos inferidos y a la asignación a los productos de trabajo.

3. Generar una matriz de trazabilidad de requisitos.

E. Asegurar el Alineamiento Entre el Trabajo del Proyecto y los Requisitos (SP 1.5)

Esta práctica específica encuentra las inconsistencias entre los requisitos, los planes del proyecto y los productos de trabajo, e inicia acciones correctivas para resolverlas. [SEI, 2010]

Ejemplos de productos de trabajo

1. Documentación de inconsistencias entre los requisitos y los planes del proyecto y los productos de trabajo, incluyendo fuentes y condiciones.
2. Acciones correctivas.

Subprácticas

1. Revisar los planes del proyecto, las actividades y los productos de trabajo en cuanto a la consistencia con los requisitos y los cambios realizados sobre ellos.
2. Identificar la fuente de la inconsistencia (si existe).
3. Identificar cualquier cambio que se debería realizar a los planes y a los productos de trabajo resultantes de los cambios a la línea base de requisitos.
4. Iniciar cualquier acción correctiva necesaria.

2.4.4. 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 de un método integral para la gestión y mejora de la capacidad. [SEI, 1996]

Adicionalmente, este modelo de mejora ofrece un conjunto de actividades que permite mejorar o adoptar las prácticas recomendadas por CMMI-DEV, 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. [SEI, 1996]

El modelo IDEAL está dividido conforme a la figura 8, en la cual se muestran todas sus fases y orden de ejecución.

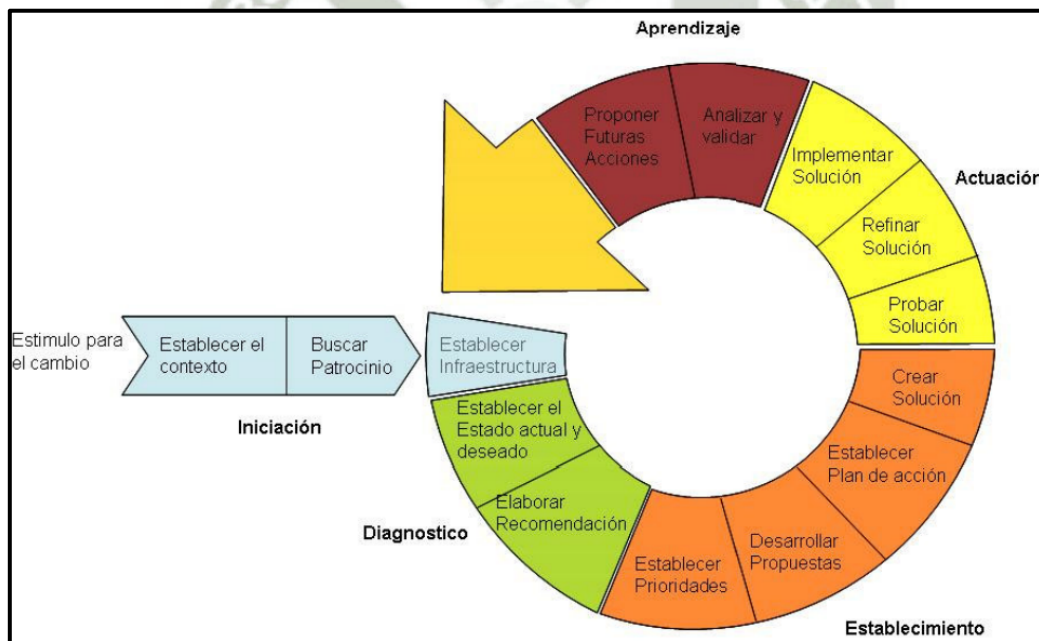


Figura 8. Fases y tareas por fase del modelo IDEAL
Fuente: [SEI, 1996]

En la figura anterior se puede apreciar que el modo de ejecución es cíclico, mostrando que los procesos nunca terminan de mejorar.

El modelo IDEAL tiene su nombre basado en sus cinco fases, tal y como se muestra en la tabla 4 a continuación:

Tabla 4. Fases del modelo IDEAL

Letra inicial	Fase	Descripción
I	Iniciación (Initiating)	Definir la base para un proceso exitoso de mejora.
D	Diagnóstico (Diagnosing)	Identificar dónde está posicionada la Organización y a dónde quiere llegar.
E	Establecimiento (Establishing)	Planificar las acciones a ejecutar para alcanzar el estado deseado.
A	Acción (Acting)	Ejecutar el Plan.
L	Aprendizaje (Learning)	Aprender de la experiencia realizada y visualizar oportunidades de mejora.

Fuente: [SEI, 1996]

Adaptado por: [Elaboración Propia]

A continuación se detalla cada una de las fases:

2.4.4.1. Inicio

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. [Torres y Arbeláez, 2008]

Estímulo para el cambio

El estímulo para el cambio puede estar dado por: [Torres y Arbeláez, 2008]

- Acontecimientos o circunstancias inesperadas.
- Orden de alta jerarquía de la organización.
- Información generada de las actividades de Benchmarking.

Establecer el contexto

Una vez que el estímulo ha sido identificado, se debe establecer el contexto donde se ejecutará el esfuerzo dentro de la organización.

Para esto se deben formular las siguientes preguntas: [Torres y Arbeláez, 2008]

- ¿Qué metas y objetivos específicos del negocio serán observados o apoyados por este cambio?

Los objetivos serán los argumentos de presentación o aprobación del proyecto, el propósito es proveer una lista de características que deben cumplir dichos objetivos, con el fin de asegurar su factibilidad y su alineación con la estrategia de negocio.

Según [Doran, 1981], los objetivos SMART deben ser:

- ✓ Específicos: Claros sobre qué, dónde, cuándo y cómo va a cambiar la situación.
 - ✓ Medibles: Debe ser posible cuantificar los fines y beneficios.
 - ✓ Realizables–Factibles: Que sea posible lograr los objetivos (conociendo los recursos y las capacidades a disposición del equipo de trabajo).
 - ✓ Realistas: Que sea posible obtener el nivel de cambio reflejado en el objetivo.
 - ✓ Limitados en tiempo: Estableciendo el periodo de tiempo en el que se debe completar cada uno de ellos.
- ¿Qué beneficio se obtendrá?

Estos beneficios pueden ser:

- ✓ Incremento de la Calidad.

- ✓ Gestión del conocimiento.
- ✓ Estandarización de procesos.
- ✓ Mejorar el nivel de conocimiento de la compañía.
- ✓ Reducción de costos de producción.
- ✓ Satisfacción de Clientes.

Buscar patrocinio

El patrocinador es bajo quien está la decisión de desembolsar los recursos para el proyecto. Este puede ser:

- Gerente del área o proceso al cual pertenece la organización.
- Gerente de la organización.
- Una persona designada.

Es importante mantener buenos niveles de patrocinio a través de todo el proceso, así como el otorgamiento de recursos por parte del patrocinador. [Torres y Arbeláez, 2008]

Establecer Infraestructura

Una vez que la razón del cambio y el contexto se han establecido y los patrocinadores están confiados en el proyecto, la organización debe instalar un mecanismo para manejar los detalles de la puesta en práctica del proceso de mejoramiento.

También es fundamental definir los recursos materiales y el recurso humano requerido para el proyecto, para esto se recomienda definir roles y perfiles. [Torres y Arbeláez, 2008]

La tabla 5 muestra un ejemplo de los roles, responsabilidades y habilidades de cada grupo del proyecto.

Tabla 5. Roles, responsabilidades y habilidades de cada grupo del proyecto

Grupo:	Comité Directivo	
Rol	Responsabilidades	Habilidades Deseadas
<p>Instancia necesaria de coordinación capaz de tomar acciones correctivas apropiadas respecto al cumplimiento de los objetivos del programa de mejora.</p>	<ul style="list-style-type: none"> ✓ Alinear el proyecto de mejoramiento con la visión, misión y objetivos de la compañía. ✓ Definir las prioridades con respecto a las prácticas a ser adoptadas. ✓ Dar soporte a la implementación de las recomendaciones aprobadas, apoyando el uso permanente de dichas recomendaciones. ✓ Dar soluciones a las dificultades identificadas. ✓ Monitorear periódicamente el progreso del programa de mejoramiento. 	<ul style="list-style-type: none"> ✓ Capacidades administrativas. ✓ Sensatez y coherencia en los actos y las sugerencias para con los demás. ✓ Gran capacidad de orientación al logro de objetivos. ✓ Buena capacidad para de persuasión (siempre bajo parámetros estratégicos de la organización).
Grupo:	Grupo de Ingeniería de Procesos	
Rol	Responsabilidades	Habilidades Deseadas
<p>Grupo responsable de liderar el mejoramiento de los procesos y facilitar la definición e implementación de las prácticas definidas en los procesos.</p>	<ul style="list-style-type: none"> ✓ Proveer la capacitación necesaria en los procesos organizacionales así como métodos y tecnologías relacionadas. ✓ Liderar los ciclos de mejoramiento de procesos incluyendo las actividades de diagnóstico, evaluaciones y certificaciones de los procesos de la organización. ✓ Elaborar planes de acción de mejoramiento de procesos. ✓ Monitorear, controlar y comunicar periódicamente el estado de avance del proyecto de mejoramiento. ✓ Establecer y mantener una librería de activos de procesos. ✓ Apoyar a los equipos de proyectos en el uso de los procesos estándar de la compañía. 	<ul style="list-style-type: none"> ✓ Creatividad ✓ Actitud crítica y abierta hacia el cambio. ✓ Planeación. ✓ Conocer bien los procesos actuales y hacia donde se deben dirigirse. ✓ Comunicación. ✓ Escucha.

Grupo:		Equipo de Definición de Procesos	
Rol	Responsabilidades	Habilidades Deseadas	
Grupo ejecutor de la revisión, mejoramiento y/o definición de las actividades asociadas con los proceso de desarrollo.	<ul style="list-style-type: none"> ✓ Desarrollar un plan de implementación por cada área de proceso. ✓ Aportar ideas de mejoras a los procesos organizacionales actuales. ✓ Definir los procesos, contribuyendo cada miembro del equipo con su punto de vista de acuerdo con la labor que desempeña en la organización. ✓ Revisar e integrar los procesos que se definan con los otros procesos ya definidos de la organización. ✓ Preparar y realizar la capacitación necesaria al resto de la organización. ✓ Realizar tareas de institucionalización de las políticas y procedimientos definidos en coordinación con los Directores de Proyecto. ✓ Reportar el estado de sus actividades al Comité Directivo y al Grupo de ingeniería de procesos. 	<ul style="list-style-type: none"> ✓ Consistencia ✓ Atención al Detalle ✓ Experiencia en Organización estructurada y diseño. ✓ Habilidades de organización. ✓ Capacidad de rastreabilidad. ✓ Habilidades de Investigación. ✓ Habilidad de crear y seguir estándares de documentación. ✓ Experiencia y habilidades para indagar y extraer información de expertos. ✓ Comunicación ✓ Trabajo en equipo 	
Grupo:		Gerentes/Jefes de Proyectos	
Rol	Responsabilidades	Habilidades Deseadas	
Dar continuidad a los cambios en los proyectos.	<ul style="list-style-type: none"> ✓ Responsables de institucionalizar las prácticas definidas en sus respectivos proyectos o líneas de negocios dentro de la organización. ✓ Asegurar el debido involucramiento de cada profesional en el logro de los objetivos de la empresa. ✓ Ayudar al correcto entendimiento y uso del proceso definido. ✓ Coordinar eventuales ajustes o mejoras con el Grupo de ingeniería de procesos. 	<ul style="list-style-type: none"> ✓ Correcta administración de recursos. ✓ Coherencia. ✓ Habilidad para Motivar a los demás. ✓ Capacidad para transmitir el conocimiento. 	

Grupo:	Equipo Evaluador	
Rol	Responsabilidades	Habilidades Deseadas
El equipo que tiene la responsabilidad del diagnóstico.	<ul style="list-style-type: none"> ✓ Documentar y mantener los insumos del diagnóstico. ✓ Comprometerse con el Acuerdo de Confidencialidad y de no Atribución. ✓ Estar familiarizados con la estructura de la organización. ✓ Asegurarse que el diagnóstico sea conducido de acuerdo con lo que se formalizó en el documento de planeación. ✓ Difundir la Metodología del proceso de diagnóstico a los Miembros del Equipo Evaluador. ✓ Preparar a las personas a entrevistar o las personas encargadas de suministrar información. 	<ul style="list-style-type: none"> ✓ Imparcialidad. ✓ Atención al detalle. ✓ Buena capacidad para transmitir el conocimiento. ✓ Organización. ✓ Capacidad para trabajo en equipo.

Fuente: [Torres y Arbeláez, 2008]
Adaptado por: [Elaboración Propia]

2.4.4.2. 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. [Torres y Arbeláez, 2008]

Establecer el estado actual y el deseado

La evaluación debe concentrarse en los elementos críticos de los cambios que son introducidos y no en cada aspecto del trabajo de la organización.

Como referencia se tienen las fases de ejecución del diagnóstico propuesto por [Arias y Mejía, 2006]:

- Preparación.
- Recolección de información.
- Análisis de información.
- Presentación de resultados.
- Cierre de evaluación.

Elaborar recomendaciones

Elaborar recomendaciones y tomarlas en cuenta para proceder en las actividades subsecuentes. [Torres y Arbeláez, 2008]

2.4.4.3. Establecimiento

En la fase de establecimiento, los problemas que la organización identificó 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. [Torres y Arbeláez, 2008]

Establecer prioridades

Al establecer prioridades se debe tener en cuenta los siguientes factores: [Torres y Arbeláez, 2008]

- Recursos limitados.
- Dependencias entre actividades recomendadas.
- Factores externos.
- Prioridades globales de la organización.

Se debe evaluar a conciencia las necesidades que se hicieron evidentes en la fase de Diagnóstico. Reorganizar las prioridades acorde con las nuevas necesidades.

Para establecer adecuadamente las prioridades, se deben tener en cuenta las necesidades con las que se justificaron los objetivos del proyecto y las recomendaciones que entregó el diagnóstico, para luego priorizar las acciones a realizar. Para esta priorización, es útil considerar los siguientes aspectos:

- Restricciones de tiempo.
- Restricciones de recursos.
- Restricciones de personal.
- Consecuencias de las acciones en los procesos.
- Tareas o necesidades que dependan de alguna otra.
- Importancia para la estrategia organizacional.

Desarrollar la propuesta

El objetivo de esta tarea es identificar la estrategia para abordar el proyecto de implementación de tal manera que no vayan en contra de las particularidades de la cultura organizacional, procurando no perder las buenas prácticas que actualmente se tienen.

Para entender el significado de Estrategia, Peter Drucker define a la estrategia como una respuesta a dos preguntas:

- ¿Qué es nuestro negocio?, cómo se encuentra la organización hoy (Diagnóstico).
- ¿Qué debería ser?, dónde la quiere llevar (Acción) y CÓMO va a llegar allá.

Al definir la estrategia para abordar el proyecto, primero se debe identificar qué es lo que se debe lograr (las prioridades) y luego cómo se logrará. [Torres y Arbeláez, 2008]

Elaborar un plan de acción

En esta parte se debe elaborar el plan detallado de la puesta en práctica del proyecto, teniendo en cuenta la estrategia de la organización mencionada anteriormente.

El plan detallado describe las acciones a realizar sobre cada uno de los procesos y se estructura de la siguiente manera:

- Introducción del documento.
- El objetivo de la implementación.
- Responsables.
- Procesos a abordar.
- Cronograma de implementación para cada área de proceso incluyendo responsables y fechas.
- Plan de Comunicación.
- Control de versiones donde se evidencie: quién elabora, quién aprueba.

Una vez definido el proyecto con su plan detallado, se puede pasar a la acción en la cual se desarrollará y se pondrá a prueba la solución. [Torres y Arbeláez, 2008]

2.4.4.4. Acción

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. [Torres y Arbeláez, 2008]

Crear solución

Crear la mejor solución para resolver las necesidades de la organización. Esto implica la integración de herramientas, conocimiento, información, procesos y habilidades tanto existentes como recién introducidas. [Torres y Arbeláez, 2008]

Probar solución

En esta tarea se procede a realizar una prueba piloto de cada proceso de la organización. Se debe tener especial cuidado en lo que respecta a la capacitación del proceso que se pondrá a prueba, se debe garantizar que el proceso está completamente entendido y que todos los responsables tienen claro su rol. [Torres y Arbeláez, 2008]

Es importante responder a las siguientes preguntas:

- ¿Cómo poner a prueba los nuevos procesos?

Se recomienda ir poniendo a prueba todos los nuevos procesos lo antes posible, sin perder de vista que hay que cumplir con un proyecto que ya estaba pactado previamente, por lo que es muy importante que la prueba de los procesos no impacte negativamente en el proyecto.

Otra camino es esperar un nuevo proyecto y desde su inicio arrancar con la prueba de los nuevos procesos.

- ¿Qué proyectos elegir para dichas pruebas?

Identifique los proyectos candidatos para la prueba analizando los procesos que se podrían probar en dichos proyectos, teniendo en cuenta que el proyecto candidato no sea un proyecto problema.

Tomar en cuenta la retroalimentación, sugerencias y dudas respecto del nuevo proceso. Se debe tener un canal de comunicación para los integrantes de los equipos que estén llevando a cabo el piloto. Las

pruebas piloto arrojarán ciertas eventualidades que deben ser registradas en cada proceso, con el fin de afinarlos. Ej. Herramienta e-Learning.

Refinar solución

Cuando la solución propuesta ha sido aplicada en un proyecto piloto, se puede refinar para reflejar el conocimiento, la experiencia y las lecciones aprendidas en el ensayo. [Torres y Arbeláez, 2008]

Implementar solución

Una vez que la solución se ha refinado y es realizable, se puede poner en práctica en toda la organización.

Al finalizar esta etapa los procesos deben quedar lo suficientemente claros como para ser aplicados a lo largo de la organización. Se debe preparar entonces para una etapa de retroalimentación. [Torres y Arbeláez, 2008]

2.4.4.5. 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. [Torres y Arbeláez, 2008]

Analizar y validar

Es importante plantearse las siguientes preguntas:

- ¿De qué manera el esfuerzo logro su propósito previsto?
- ¿Qué salió bien?
- ¿Qué se podía hacer más con eficacia o eficiencia?

Esta actividad identifica el grado en que el esfuerzo invertido logro los propósitos deseados, las cosas que se trabajaron bien y como se podrían hacer mejor las cosas en el ciclo de mejoramiento siguiente. Las lecciones se analizan, se resumen y se documentan. Se reexaminan las necesidades de la

empresa identificadas en la fase de inicio para ver si fueron satisfechas.
[Torres y Arbeláez, 2008]

Proponer futuras acciones

Una vez realizado el análisis anterior, se deben corregir y tomar acciones sobre cada uno de los procesos para los cuales se haya generado algún tipo de cambio.

Es importante desarrollar y documentar las recomendaciones basadas en el análisis y las validaciones.

Una vez realizados estos ajustes en los procesos o en la manera como se está implantando, es cuestión de seguir el ciclo del modelo IDEAL y volver a la fase de Diagnóstico. [Torres y Arbeláez, 2008]

El modelo IDEAL es un ciclo continuo de aprendizaje que realiza varias iteraciones para lograr la implementación de los procesos.

Esta mejora de procesos debe ser administrada de manera eficiente para asegurar su éxito. Es importante documentar las lecciones aprendidas y utilizar esto como fuente de información para otros proyectos. Se debe lograr con cada ejecución del modelo IDEAL, mejorar la calidad de los procesos.

CAPITULO III: PROPUESTA DEL MODELO

3.1.PROPUESTA DEL MODELO DE INTEGRACIÓN

3.1.1. Introducción

Como se mencionó anteriormente, los procesos de requisitos se encuentran entre las principales fuentes de los problemas encontrados durante el desarrollo de proyectos de software. Muchas organizaciones han mostrado interés en la mejora de este proceso porque creen que este proceso puede ser la clave para el desarrollo de productos exitosos. “Lo más difícil en la construcción de un sistema de software es decidir precisamente qué construir. No existe tarea con mayor capacidad de lesionar al sistema, cuando se hace mal. Ninguna otra tarea es tan difícil de rectificar a posteriori” [Brooks, 1987].

Muchas de las organizaciones a menudo saben "Qué" se debe hacer en este proceso gracias a los modelos existentes, tales como CMMI-DEV, entre otros, pero no saben “Cómo” poner en práctica este tipo de actividad. A partir de lo expuesto, este trabajo tiene por propósito desarrollar un modelo de integración basado en excelentes fuentes de las mejores prácticas que sirvan de apoyo en la implementación y mejora de los procesos de requisitos.

3.1.2. Objetivo

El objetivo del modelo propuesto es de mejorar los procesos de requisitos en los proyectos de desarrollo de software. Para ello, se hizo un estudio del modelo CMMI-DEV, la guía SWEBOK y el modelo IDEAL.

Se utilizará este conocimiento acumulado como referencia para la selección y priorización de las prácticas que una organización en particular necesita. De esta forma facilitar la implementación de procesos de requisitos que logren buenos resultados y por lo tanto mejorar la calidad de este proceso.

3.1.3. Estructura del modelo de integración propuesto

El modelo propuesto está compuesto por dos componentes: un conjunto de prácticas fundamentales y por una estrategia de implementación de mejora de procesos.

El primer componente es un conjunto de prácticas fundamentales relacionadas a los aspectos técnicos de los procesos de requisitos, tales como prácticas de educación, análisis, validación, gestión, etc.

El segundo componente es un ciclo de mejoramiento de procesos conocido como IDEAL [SEI, 1996] propuesto por el SEI, el cual proporciona un conjunto de actividades coherentes para sustentar la adopción de las practicas recomendadas por CMMI-DEV.

La figura 9 representa los componentes del modelo y sus respectivas fuentes de referencias utilizadas como base para la construcción de cada componente.

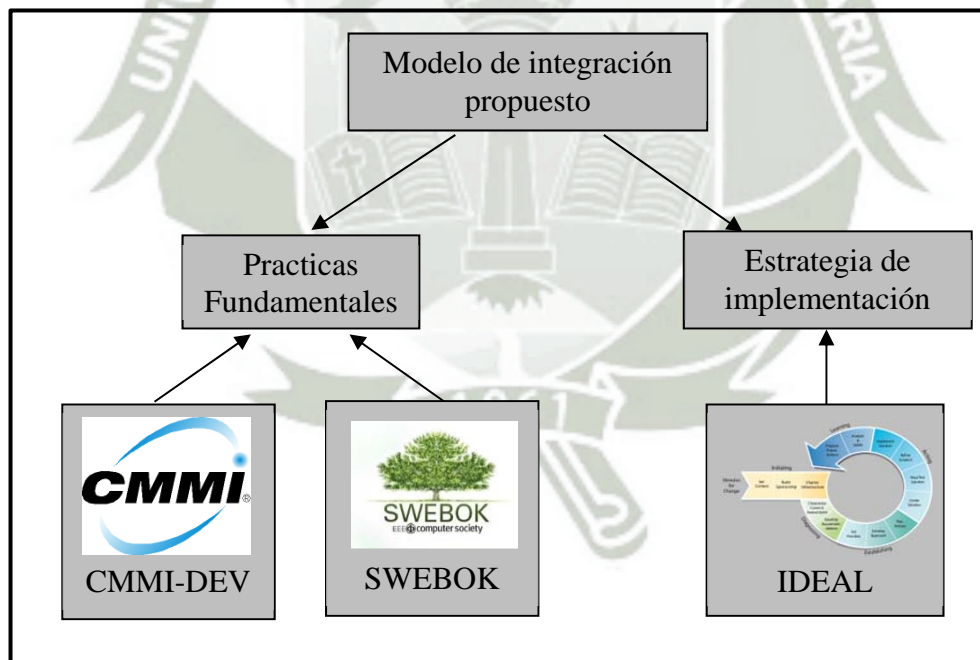


Figura 9. Componentes y fuentes del modelo propuesto
Fuente: [Elaboración Propia]

3.1.4. Prácticas Fundamentales

Las prácticas fundamentales se utilizan para disminuir el nivel de abstracción de las prácticas propuestas por CMMI-DEV, en estas prácticas se dan descripciones más detalladas de como cumplir con las metas de las áreas de proceso de Desarrollo de Requisitos (RD) y Gestión de Requisitos (REQM).

Como CMMI-DEV se basa en las mejores prácticas obtenidas después de muchos años de estudios, se organizaron las prácticas fundamentales de acuerdo a su estructura tal como muestra la figura 10.

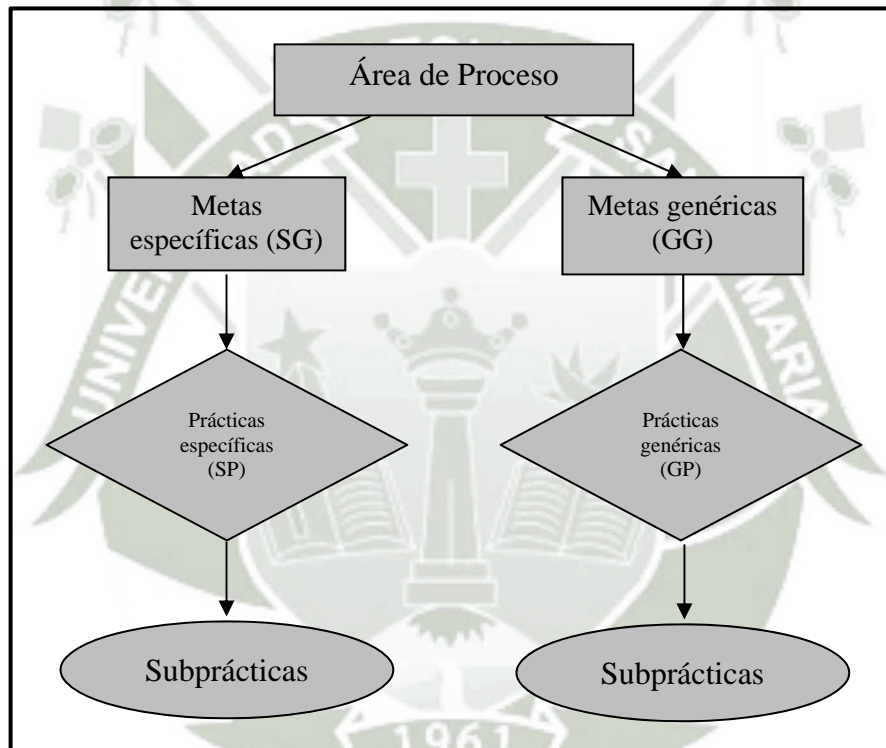


Figura 10. Estructura área de proceso
Fuente: [SEI, 2010]

Además, se agregó un nivel más llamado “prácticas fundamentales” tal como se muestra en la figura 11, permitiendo a las personas responsables saber cómo proceder a ejecutar las Prácticas específicas, Prácticas genéricas, Subprácticas y por lo tanto, cumplir con las metas del área de proceso.

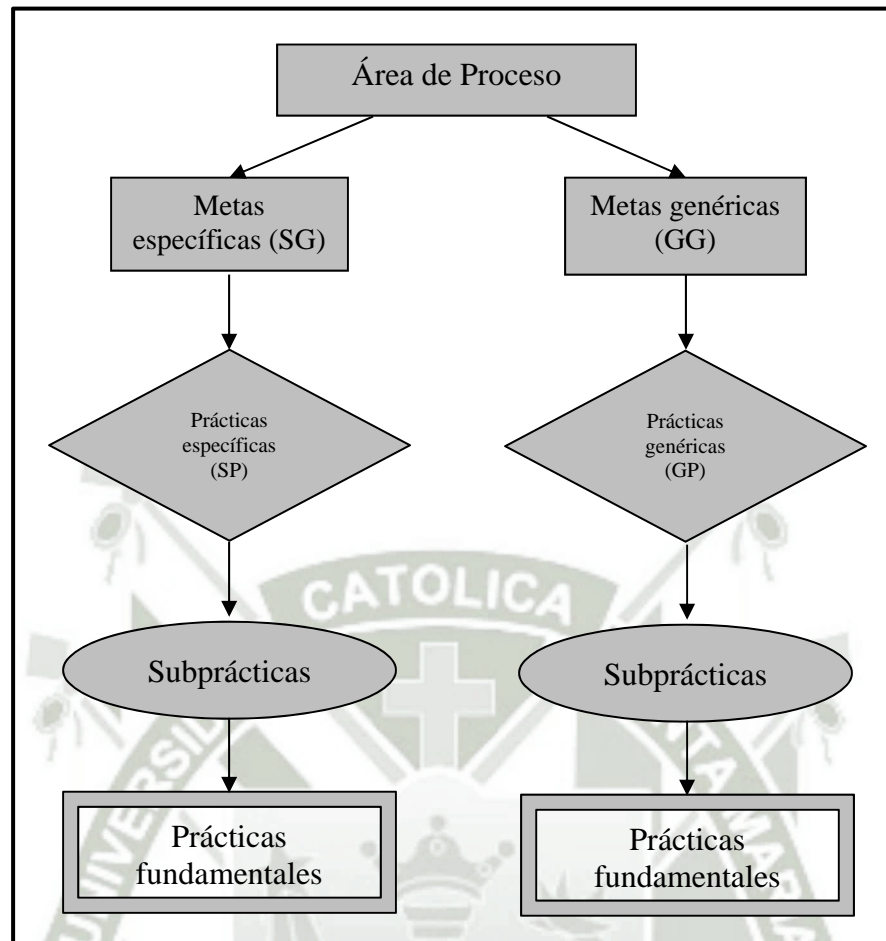


Figura 11. Estructura de prácticas fundamentales

Fuente: [SEI, 2010]

Adaptado por: [Elaboración propia]

Por lo tanto se propone que cada Práctica específica, Práctica genérica y Subprácticas correspondientes deben considerarse al elegir las prácticas fundamentales.

3.1.4.1. Estructura y fuentes

El modelo CMMI-DEV se utilizará para estructurar las áreas de proceso relacionadas con los requisitos - Desarrollo de Requisitos (RD) y la Gestión de Requisitos (REQM). Sobre la base de la estructura de estas áreas de proceso, se utiliza Subprácticas de CMMI-DEV como clasificadores para las buenas prácticas obtenidas de la guía SWEBOK, como se ilustra en la figura 12.

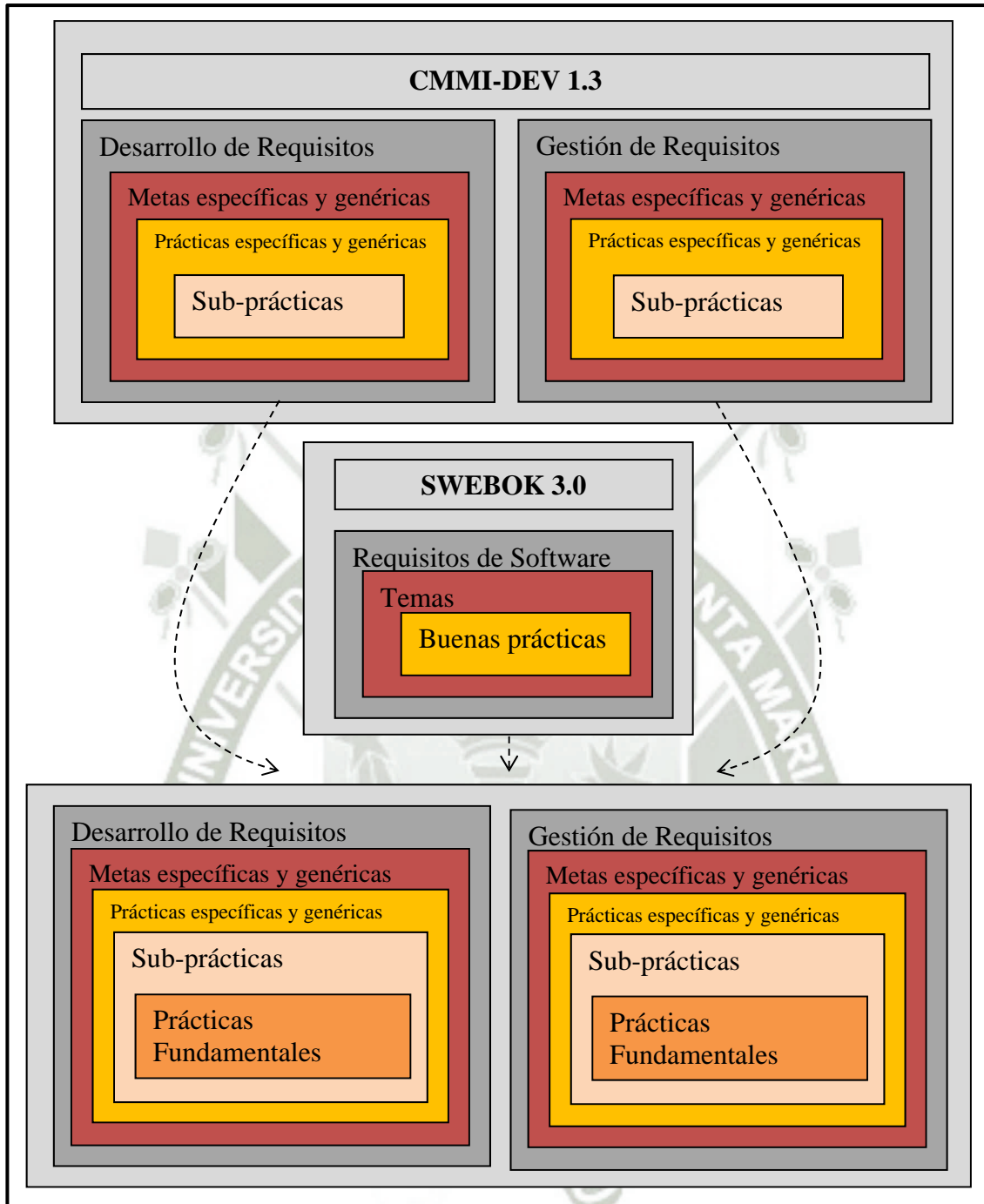


Figura 12. Estructura y fuentes de las prácticas fundamentales
Fuente: [Elaboración propia]

El primer componente del modelo propuesto contiene un conjunto de buenas prácticas que deben seleccionarse y ejecutarse para cumplir con las metas de las áreas de proceso relacionadas con los requisitos. La identificación de estas prácticas se realiza en base al estudio del modelo CMMI-DEV y la guía SWEBOK.

3.1.4.2. Comparación entre el modelo CMMI-DEV y la Guía SWEBOK

La relación de correspondencia entre las prácticas específicas de CMMI-DEV y las buenas prácticas del SWEBOK ha sido definida para las áreas de proceso: Desarrollo de Requisitos (RD) y Gestión de Requisitos (REQM) y para el área de conocimiento de Requisitos de Software.

La tabla 6 muestra la correspondencia entre estas prácticas, es importante tener en cuenta que CMMI-DEV se estructura en metas específicas (SG), cada meta específica puede determinar un conjunto de prácticas específicas (SP), mientras que SWEBOK se estructura en áreas de conocimiento.



Tabla 6. Comparación SWEBOK y CMMI-DEV

	CMMI-DEV																			
	Desarrollo de Requisitos (RD)							Gestión de Requisitos (REQM)												
	SG 1 Desarrollar los requisitos del cliente.	SP 1.1 Educar las necesidades.	SP 1.2 Transformar las necesidades de las partes interesadas en requisitos de cliente	SG 2 Desarrollar los requisitos de producto.	SP 2.1 Establecer los requisitos de producto y de componente de producto	SP 2.2 Asignar los requisitos de componente de producto.	SP 2.3 Identificar los requisitos de interfaz.	SG 3 Analizar y validar los requisitos.	SP 3.1 Establecer los conceptos y los escenarios de operación.	SP 3.2 Establecer una definición de la funcionalidad y de los atributos de calidad requeridos.	SP 3.3 Analizar los requisitos.	SP 3.4 Analizar los requisitos para conseguir un equilibrio.	SP 3.5 Validar los requisitos.	SG 1 Gestionar los requisitos.	SP 1.1 Comprender los requisitos.	SP 1.2 Obtener el compromiso sobre los requisitos.	SP 1.3 Gestionar los cambios a los requisitos.	SP 1.4 Mantener la trazabilidad bidireccional de los requisitos.	SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos	
SWEBOK																				
1. Fundamentos de los Requisitos de Software																				
1.1. Definición de Requisito de Software																				
1.2. Requisitos de Producto y de Proceso					X															
1.3. Requisitos Funcionales y No Funcionales			X																	
1.4. Propiedades Emergentes																				
1.5. Requisitos Cuantificables					X					X										
1.6. Requisitos del Sistema y Requisitos del Software			X																	
2. Proceso de los Requisitos																				
2.1. Modelos de Proceso													X							
2.2. Participantes del Proceso												X				X				
2.3. Apoyo y Gestión de Proceso																				
2.4. Calidad y Mejora de Procesos													X							

CMMLI-DEV

	CMMLI-DEV																		
	Desarrollo de Requisitos (RD)							Gestión de Requisitos (REQM)											
	SG 1 Desarrollar los requisitos del cliente.	SP 1.1 Educar las necesidades.	SP 1.2 Transformar las necesidades de las partes interesadas en requisitos de cliente	SG 2 Desarrollar los requisitos de producto.	SP 2.1 Establecer los requisitos de producto y de componente de producto	SP 2.2 Asignar los requisitos de interfaz.	SG 3 Analizar y validar los requisitos.	SP 3.1 Establecer los conceptos y escenarios de operación.	SP 3.2 Establecer una definición de la funcionalidad y de los atributos de calidad requeridos.	SP 3.3 Analizar los requisitos.	SP 3.4 Analizar los requisitos para conseguir un equilibrio.	SP 3.5 Validar los requisitos.	SG 1 Gestionar los requisitos.	SP 1.1 Comprender los requisitos.	SP 1.2 Obtener el compromiso sobre los requisitos.	SP 1.3 Gestionar los cambios a los requisitos.	SP 1.4 Mantener la trazabilidad bidireccional de los requisitos.	SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos	
SWEBOOK																			
3. Educación de Requisitos		X						X	X										
3.1. Fuentes de los Requisitos		X						X	X										
3.2. Técnicas de Educación		X						X	X										
4. Análisis de Requisitos																			
4.1. Clasificación de los Requisitos			X						X										
4.2. Modelado Conceptual								X											
4.3. Diseño Arquitectónico y Asignación de Requisitos						X													
4.4. Negociación de los Requisitos			X												X				
4.5. Análisis Formal																			
5. Especificación de Requisitos																			
5.1. Documento de la Definición del Sistema			X																
5.2. Especificación de Requisitos del Sistema			X																
5.3. Especificación de Requisitos del Software			X																
6. Validación de Requisitos																			
6.1. Revisiones de los Requisitos																			
6.2. Prototipado																			
6.3. Validación del Modelo																			
6.4. Pruebas de Aceptación																			

	CMMI-DEV																			
	Desarrollo de Requisitos (RD)									Gestión de Requisitos (REQM)										
	SG 1 Desarrollar los requisitos del cliente.	SP 1.1 Educar las necesidades.	SP 1.2 Transformar las necesidades de las partes interesadas en requisitos de cliente	SG 2 Desarrollar los requisitos de producto.	SP 2.1 Establecer los requisitos de producto y de componente de producto	SP 2.2 Asignar los requisitos de componente de producto.	SP 2.3 Identificar los requisitos de interfaz.	SG 3 Analizar y validar los requisitos.	SP 3.1 Establecer los conceptos y los escenarios de operación.	SP 3.2 Establecer una definición de la funcionalidad y de los atributos de calidad requeridos.	SP 3.3 Analizar los requisitos.	SP 3.4 Analizar los requisitos para conseguir un equilibrio.	SP 3.5 Validar los requisitos.	SG 1 Gestionar los requisitos.	SP 1.1 Comprender los requisitos.	SP 1.2 Obtener el compromiso sobre los requisitos.	SP 1.3 Gestionar los cambios a los requisitos.	SP 1.4 Mantener la trazabilidad bidireccional de los requisitos.	SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos	
7. SWEBOOK																				
7. Consideraciones Prácticas																				
7.1. Naturaleza Iterativa del Proceso de Requisitos																				
7.2. Gestión del Cambio																				
7.3. Características de los Requisitos																				
7.4. Seguimiento de los Requisitos																				
7.5. Requisitos de Medición																				
8. Herramientas de Requisitos de Software																				

Fuente: [Elaboración Propia]

3.1.4.3. Desarrollo de las Prácticas Fundamentales

A continuación se presenta las prácticas fundamentales, clasificadas de acuerdo a las prácticas específicas y genéricas del modelo CMMI-DEV.

Cada práctica específica de CMMI-DEV está compuesta de un conjunto de Subprácticas. Para cada una de estas Subprácticas se asigna un conjunto de prácticas fundamentales obtenidas de la guía SWEBOK.

Por tratarse de una propuesta de modelo para la mejora únicamente de los procesos de requisitos, se tiene por base la representación continua [SEI, 2010].

Tabla 7. Desarrollo de Requisitos - Meta específica 1 – Práctica específica 1.1

Área de Proceso	Desarrollo de Requisitos	Acrónimo	RD
Meta Específica	1. Desarrollar los requisitos del cliente.		
Práctica Específica	1.1. Educir las necesidades.		
Propósito			
Educir las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas para todas las fases del ciclo de vida del producto.			
Subprácticas		Prácticas Fundamentales	
1. Comprometer a las partes interesadas relevantes usando métodos para educir las necesidades, las expectativas, las restricciones y las interfaces externas.		Definir el alcance del proyecto, proporcionando una descripción del software que se especifica y su propósito, priorizando las necesidades de negocios más importantes.	
		Identificar y evaluar todas las posibles fuentes de requisitos. Ej. Clientes, usuarios finales, proveedores, desarrolladores, personal de pruebas, fabricantes, personal de soporte logístico.	
		Evaluar el entorno de la organización, la estructura, la cultura y la política interna de la organización.	
		Ejemplos de fuentes de requisitos que pueden no ser identificadas por el cliente son: <ul style="list-style-type: none"> ✓ Políticas de negocio. ✓ Estándares. ✓ Requisitos de entorno de negocio (p. ej., laboratorios, pruebas y otras instalaciones, infraestructura de tecnología de información). ✓ Tecnología. ✓ Estatutos reguladores. 	

	<p>Establecer una buena comunicación con los diversos stakeholders.</p>
	<p>Utilizar técnicas para educir las necesidades:</p> <ul style="list-style-type: none"> ✓ Entrevistas a los stakeholders. ✓ Escenarios para proporcionar contexto a la obtención de los requisitos del usuario. Ej. Casos de uso. ✓ Prototipos para la especificación de los requisitos ambiguos. Ej. Simulaciones en papel de los diseños de pantalla, versiones de pruebas beta de productos de software. ✓ Reuniones para tratar de lograr un efecto acumulativo del conocimiento de los requisitos de software. ✓ Observación para aprender acerca de las tareas de los usuarios, la interacción de uno con otros, con las herramientas de software y otros recursos. Ej. La etnografía para la edución de requisitos. ✓ Historias de usuario, descripciones breves, en términos del cliente. Ej. "Como <función>, quiero <meta/deseo> de forma que <beneficio>."
	<p>Adquirir o tener apto conocimiento sobre el dominio de aplicación.</p>
	<p>Elaborar lista de Stakeholders.</p>

Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]

Tabla 8. Desarrollo de Requisitos - Meta específica 1 – Práctica específica 1.2

Área de Proceso	Desarrollo de Requisitos	Acrónimo	RD
Meta Específica	1. Desarrollar los requisitos del cliente.		
Práctica Específica	1.2. Trasformar las necesidades de las partes interesadas en requisitos de cliente.		
Propósito			
Transformar las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas en requisitos de cliente priorizados.			
Subprácticas		Prácticas Fundamentales	
1. Traducir las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas en requisitos documentados del cliente.	Elaborar el documento de la definición del sistema (a veces conocido como documento de requisitos de usuario o documento de concepto de operaciones).		
	Definir los requisitos del sistema de alto nivel.		
2. Establecer y mantener una priorización de los requisitos funcionales de cliente y de los atributos de calidad.	El documento puede incluir:		
	<ul style="list-style-type: none"> ✓ Requisitos del sistema. ✓ Información de fondo acerca de los objetivos generales del sistema. ✓ Entorno de destino. ✓ Declaración de las restricciones. ✓ Hipótesis. ✓ Requisitos no funcionales. ✓ Modelos conceptuales. ✓ Flujos de trabajo. 		
	Identificar si el requisito es:		
	<ul style="list-style-type: none"> ✓ Funcional. ✓ No funcional (Atributos de Calidad). Ej. Capacidad de respuesta, mantenimiento, seguridad, fiabilidad, rendimiento, usabilidad, etc. 		
3. Definir las restricciones para la verificación y la validación.	Priorizar los requisitos con el fin de resolver los conflictos y planificar las entregas.		
	Clasificar al requisito por prioridad. Ej. Obligatorio, muy deseable, deseable u opcional.		
	Elaborar lista de requisitos funcionales y no funcionales.		
3. Definir las restricciones para la verificación y la validación.	Especificar lo que el producto de software debe hacer, así como lo que no se espera que haga.		
	Definir restricciones de cliente para llevar a cabo la verificación.		
	Definir restricciones de cliente para llevar a cabo la validación.		

**Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]**

Tabla 9. Desarrollo de Requisitos - Meta específica 2 – Práctica específica 2.1

Área de Proceso	Desarrollo de Requisitos	Acrónimo	RD
Meta Específica	2. Desarrollar los requisitos de producto.		
Práctica Específica	2.1. Establecer los requisitos de producto y de componente de producto.		
Propósito			
Establecer y mantener los requisitos de producto y de componente de producto, basados en los requisitos de cliente.			
Subprácticas		Prácticas Fundamentales	
1. Desarrollar los requisitos en los términos técnicos necesarios para el diseño del producto y de los componentes de producto.		Traducir los deseos del cliente en parámetros técnicos. Ej. House Quality Function Deployment.	
		Expresar los requisitos claramente e inequívocamente como sea posible y en su caso, en términos cuantitativos. Ej. El software de un centro de llamadas debe aumentar el rendimiento del centro en un 20%.	
		Evitar los requisitos vagos e inverificables. Ej. El software debe ser confiable, el software debe ser fácil de usar.	
		Expresar requisitos de forma precisa y sin ambigüedades para evitar la posibilidad de una mala interpretación.	
		Utilizar anotaciones para permitir que los requisitos se describan con la mayor precisión posible.	
		Elaborar el documento de especificación de requisitos de Software.	
		Utilizar el documento de especificación de requisitos para:	
		<ul style="list-style-type: none"> ✓ Proporcionar una base realista para la estimación de los costos del producto, riesgos y horarios. ✓ Para una evaluación rigurosa de los requisitos antes de que pueda comenzar el diseño, de esta manera reducir el rediseño más tarde. ✓ Como base para el desarrollo de planes de verificación y validación. 	
2. Inferir los requisitos resultantes de las decisiones de diseño.		Obtener requisitos derivados a partir de las decisiones de diseño.	

<p>3. Desarrollar los requisitos de arquitectura capturando los atributos críticos de calidad y las medidas de atributos de calidad necesarios para establecer la arquitectura y el diseño del producto.</p>	<p>Algunos ejemplos de atributos de calidad:</p> <ul style="list-style-type: none"> ✓ Disponibilidad del producto. ✓ Capacidad de mantenimiento. ✓ Adaptabilidad. ✓ Oportunidad. ✓ Tasa de transferencia. ✓ Capacidad de respuesta. ✓ Fiabilidad. ✓ Seguridad. ✓ Escalabilidad. <p>Algunos ejemplos de medidas de los atributos de calidad son:</p> <ul style="list-style-type: none"> ✓ Respuesta dentro de 1 segundo. ✓ Disponibilidad del sistema en un 99% del tiempo. ✓ Implementar un cambio con no más de una semana de esfuerzo del personal.
<p>4. Establecer y mantener las relaciones entre los requisitos para su consideración durante la gestión del cambio y la asignación de los requisitos.</p>	<p>Mantener la trazabilidad bidireccional de los requisitos.</p>

Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]

Tabla 10. Desarrollo de Requisitos - Meta específica 2 – Práctica específica 2.2

Área de Proceso	Desarrollo de Requisitos	Acrónimo	RD
Meta Específica	2. Desarrollar los requisitos de producto.		
Práctica Específica	2.2. Asignar los requisitos de componente de producto.		
Propósito			
Asignar los requisitos para cada componente de producto.			
Subprácticas		Prácticas Fundamentales	
1. Asignar los requisitos a las funciones.		Elaborar hojas de asignación de requisitos.	
2. Asignar los requisitos a los componentes de producto y a la arquitectura.		Asignar los requisitos a los componentes de la arquitectura para: <ul style="list-style-type: none"> ✓ Realizar un análisis detallado de los requisitos. ✓ Descubrir nuevos requisitos cuando el componente interactúe con otros componentes. 	
3. Asignar las restricciones de diseño a componentes de producto y a la arquitectura.		Elaborar hojas de asignación de requisitos.	
4. Asignar requisitos a las entregas incrementales.			
5. Documentar las relaciones entre requisitos asignados.		Mantener relaciones entre requisitos inferidos que incluyen las dependencias en que un cambio en un requisito puede afectar a otros requisitos.	

**Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]**

Tabla 11. Desarrollo de Requisitos - Meta específica 2 – Práctica específica 2.3

Área de Proceso	Desarrollo de Requisitos	Acrónimo	RD
Meta Específica	2. Desarrollar los requisitos de producto.		
Práctica Específica	2.3. Identificar los requisitos de interfaz.		
Propósito			
Identificar los requisitos de interfaz.			
Subprácticas		Prácticas Fundamentales	
1. Identificar las interfaces tanto externas como internas al producto (p. ej., entre las particiones funcionales u objetos).		Identificar interfaces internas. Ej. Intercambios de datos entre programas.	
		Identificar interfaces externas. Ej. Intercambios de información entre un sistema de software y otros sistemas de software, bases de datos externas, interface hombre-máquina.	
2. Desarrollar los requisitos para las interfaces identificadas.		Especificar los requisitos de interface.	

**Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]**

Tabla 12. Desarrollo de Requisitos - Meta específica 3 – Práctica específica 3.1

Área de Proceso	Desarrollo de Requisitos	Acrónimo	RD
Meta Específica	3. Analizar y validar los requisitos.		
Práctica Específica	3.1. Establecer los conceptos y los escenarios de operación.		
Propósito			
Establecer y mantener los conceptos y los escenarios de operación asociados.			
Subprácticas		Prácticas Fundamentales	
1. Desarrollar los conceptos y los escenarios de operación que incluyan operaciones, instalación, desarrollo, mantenimiento, soporte y retirada según sea apropiado.	Identificar y desarrollar secuencia de eventos que podrían ocurrir en el desarrollo, uso o soporte del producto, consistentes con las necesidades, las expectativas y las restricciones de las partes interesadas.		
	Formular preguntas sobre las tareas del usuario. Ej. “Qué pasaría sí”, “cómo se hace esto”.		
	Desarrollar escenarios para proporcionar contexto a la obtención de los requisitos del usuario. Ej. Casos de uso.		
	Elaborar modelos conceptuales para entender la situación en la que se produce el problema y representar una solución. Ej. Diagramas de casos de uso, etc.		
		Comenzar por la construcción de un modelo del contexto del software para proporcionar una conexión entre el software y su entorno externo.	
2. Definir el entorno en el que funcionará el producto o componente de producto, incluyendo los límites y las restricciones.	Obtener los requisitos del entorno operativo en el que el software será ejecutado.		
3. Revisar los conceptos y los escenarios de operación para refinar y descubrir requisitos.	Revisar el proceso de concepto y el desarrollo del escenario de operación periódicamente.		
4. Desarrollar un concepto de operación detallado, a medida que se seleccionan los productos y los componentes de producto, que define la interacción del producto, del usuario final y del entorno, y que satisfaga las necesidades operativas, de mantenimiento, de soporte y de retirada.	Elaborar Casos de uso detallado.		

**Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]**

Tabla 13. Desarrollo de Requisitos - Meta específica 3 – Práctica específica 3.2

Área de Proceso	Desarrollo de Requisitos	Acrónimo	RD
Meta Específica	3. Analizar y validar los requisitos.		
Práctica Específica	3.2. Establecer una definición de la funcionalidad y de los atributos de calidad requeridos.		
Propósito			
Establecer y mantener una definición de funcionalidad y de los atributos de calidad requeridos.			
Subprácticas		Prácticas Fundamentales	
1. Determinar la misión clave y los factores de negocio.		Identificar la meta (preocupación de negocio o factor crítico del éxito) en otras palabras, los objetivos generales de alto nivel del software. Ej. Un estudio de factibilidad.	
2. Identificar la funcionalidad y los atributos de calidad deseados.		Identificar a las partes interesadas relevantes. Desarrollar escenarios para proporcionar contexto a la obtención de los requisitos del usuario. Ej. Casos de uso. Elaborar Diagramas de actividad.	
3. Determinar los atributos de calidad significativos para la arquitectura en base a la misión clave y los factores del negocio.		Identificar requisitos de los atributos de calidad significativos para la arquitectura. Algunos ejemplos de atributos de calidad: <ul style="list-style-type: none"> ✓ Disponibilidad del producto. ✓ Capacidad de mantenimiento. ✓ Adaptabilidad. ✓ Oportunidad. ✓ Tasa de transferencia. ✓ Capacidad de respuesta. ✓ Fiabilidad. ✓ Seguridad. ✓ Escalabilidad. 	
4. Analizar y cuantificar la funcionalidad requerida por los usuarios finales.		Expresar los requisitos claramente e inequívocamente como sea posible y en su caso, en términos cuantitativos. Ej. El software de un centro de llamadas debe aumentar el rendimiento del centro en un 20%. Evitar los requisitos vagos e inverificables. Ej. El software debe ser confiable, el software debe ser fácil de usar.	
5. Analizar los requisitos para identificar las particiones lógicas o funcionales (p. ej., subfunciones).			

<p>6. Dividir los requisitos en grupos, en base a los criterios establecidos (p. ej., funcionalidad similar, requisitos similares de atributos de calidad, acoplamiento) para facilitar y enfocar el análisis de requisitos.</p>	<p>Clasificar si el requisito es:</p> <ul style="list-style-type: none"> ✓ Funcional. ✓ No funcional (Atributos de Calidad). Ej. Capacidad de respuesta, mantenimiento, seguridad, fiabilidad, rendimiento, usabilidad, etc. ✓ Deriva de uno o más requisitos de alto nivel o se impone directamente por un stakeholder o alguna otra fuente. ✓ Del producto o del proceso. ✓ Prioridad. Ej. Obligatorio, muy deseable, deseable u opcional. ✓ Alcance o grado en que un requisito afecta al software y a los componentes de software. Ej. Alcance global, alcance limitado. ✓ Volatilidad/estabilidad para establecer un diseño que es más tolerante de cambio.
<p>7. Asignar los requisitos de cliente a las particiones funcionales, objetos, personal o elementos de soporte para dar apoyo a la síntesis de las soluciones.</p>	
<p>8. Asignar los requisitos a las funciones y a las subfunciones (u otras entidades lógicas).</p>	

**Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]**

Tabla 14. Desarrollo de Requisitos - Meta específica 3 – Práctica específica 3.3

Área de Proceso	Desarrollo de Requisitos	Acrónimo	RD
Meta Específica	3. Analizar y validar los requisitos.		
Práctica Específica	3.3. Analizar los requisitos.		
Propósito			
Analizar los requisitos para asegurarse que son necesarios y suficientes.			
Subprácticas		Prácticas Fundamentales	
1. Analizar las necesidades, las expectativas, las restricciones y las interfaces externas de las partes interesadas para organizarlas en temas relacionados y eliminar conflictos.	Asignar un grupo que lleve a cabo la revisión, es importante (al menos un representante del cliente debe ser incluido).		
	Analizar los documentos de requisitos.		
	Elaborar informes de defectos de los requisitos.		
	Desarrollar los cambios propuestos a los requisitos para resolver defectos.		
2. Analizar los requisitos para determinar si satisfacen los objetivos de los requisitos de nivel más alto.	Clasificar si el requisito deriva de uno o más requisitos de alto nivel.		
	Analizar el nivel uno de la jerarquía del producto para determinar si son necesarios y suficientes para cumplir con los objetivos de niveles más altos.		
3. Analizar los requisitos para asegurarse que son completos, factibles, realizables y verificables.	Buscar errores, suposiciones erróneas, falta de claridad y desviación de la práctica estándar.		
	Analizar los requisitos para que: <ul style="list-style-type: none"> ✓ Sean validados. ✓ Su implementación sea verificada. ✓ Sus costos sean estimados. 		
	Asegurar que la calidad de los requisitos sea tan alta como sea posible.		
4. Identificar los requisitos clave que tienen una fuerte influencia en el coste, el calendario, el rendimiento o el riesgo.	Clasificar el requisito por: <ul style="list-style-type: none"> ✓ Prioridad. Ej. Obligatorio, muy deseable, deseable u opcional. ✓ Alcance o grado en que un requisito afecta al software y a los componentes de software. Ej. Alcance global, alcance limitado. ✓ Volatilidad/estabilidad para establecer un diseño que es más tolerante de cambio. 		
	Identificar requisitos clave para realizar el seguimiento del proceso.		

<p>5. Identificar las medidas técnicas de rendimiento que serán seguidas durante el esfuerzo de desarrollo.</p>	<p>Algunos ejemplos de atributos medibles para el rendimiento del proceso son:</p> <ul style="list-style-type: none"> ✓ Tiempo de ciclo. ✓ Porcentaje de tiempo de retrabajo. ✓ Porcentaje de defectos eliminados por las actividades de verificación del producto. ✓ Tasa de defectos no detectados. ✓ Número y gravedad de defectos encontrados (o incidencias comunicadas) en el año siguiente a la entrega del producto (o al inicio del servicio).
<p>6. Analizar los conceptos y los escenarios de operación para refinar las necesidades, las restricciones y las interfaces del cliente, y para Inferir nuevos requisitos.</p>	<p>Analizar los modelos conceptuales. Obtener modelos conceptuales más detallados (nuevos requisitos).</p>

Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]

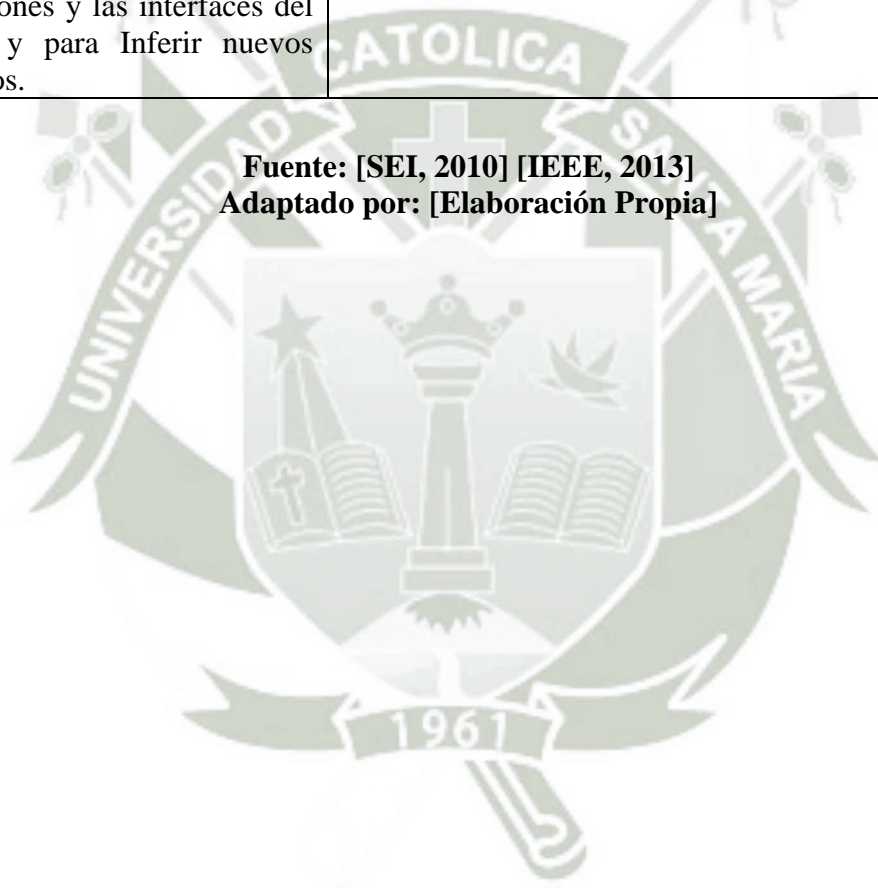


Tabla 15. Desarrollo de Requisitos - Meta específica 3 – Práctica específica 3.4

Área de Proceso	Desarrollo de Requisitos	Acrónimo	RD
Meta Específica	3. Analizar y validar los requisitos.		
Práctica Específica	3.4. Analizar los requisitos para conseguir un equilibrio.		
Propósito			
Analizar los requisitos para equilibrar las necesidades y las restricciones de las partes interesadas.			
Subprácticas		Prácticas Fundamentales	
1. Usar modelos, simulaciones y prototipos probados para analizar el equilibrio entre las necesidades y las restricciones de las partes interesadas.		Reducir el coste del producto y el riesgo del desarrollo del producto.	
2. Realizar una evaluación de riesgos sobre los requisitos y la definición de funcionalidad y atributos de calidad requeridos.		Algunas fuentes típicas de riesgo internas y externas: <ul style="list-style-type: none"> ✓ Requisitos inciertos. ✓ Requisitos de atributos de calidad en competencia que afectan a la selección de la solución y al diseño. ✓ Tecnología no disponible. ✓ Comunicación inadecuada con los clientes actuales o potenciales o con sus representantes. ✓ Restricciones reglamentarias. Ej. Seguridad, protección, entorno. 	
3. Examinar los conceptos del ciclo de vida del producto en cuanto a los impactos de los requisitos en los riesgos.		Consultar a las partes interesadas relevantes si el impacto de los requisitos en costes y riesgos supera el beneficio percibido.	
		Determinar qué cambios pueden ser necesarios.	
4. Evaluar el impacto de los requisitos de los atributos de calidad significativos para la arquitectura en el producto y en los costes y riesgos del desarrollo del producto.		Negociar los cambios que son aceptables dentro del presupuesto, técnico, regulador y otras restricciones.	
		Evaluar tanto la probabilidad como la consecuencia.	
		La probabilidad puede clasificarse como: <ul style="list-style-type: none"> ✓ Remota. ✓ Improbable. ✓ Probable. ✓ Altamente probable. ✓ Casi certeza. 	
		Ejemplos de categorías de consecuencia son: <ul style="list-style-type: none"> ✓ Baja. ✓ Media. ✓ Alta. ✓ Despreciable. ✓ Marginal. ✓ Significativa. ✓ Crítica. 	

	<p>Opciones para tratar los riesgos:</p> <ul style="list-style-type: none">✓ Evitar el riesgo: cambiar o reducir los requisitos mientras se sigan cumpliendo las necesidades del usuario final.✓ Controlar el riesgo: llevar a cabo actividades para minimizar los riesgos.✓ Transferir el riesgo: reasignar requisitos para reducir riesgos.✓ Monitorizar el riesgo: vigilar y reevaluar periódicamente el riesgo en función de los cambios en los parámetros del riesgo asignados.✓ Aceptar el riesgo: reconocer el riesgo pero no tomar ninguna acción.
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]



Tabla 16. Desarrollo de Requisitos - Meta específica 3 – Práctica específica 3.5

Área de Proceso	Desarrollo de Requisitos	Acrónimo	RD
Meta Específica	3. Analizar y validar los requisitos.		
Práctica Específica	3.5. Validar los requisitos.		
Propósito			
Validar los requisitos para asegurar que el producto resultante funcione según lo previsto en el entorno del usuario final.			
Subprácticas	Prácticas Fundamentales		
1. Analizar los requisitos para determinar el riesgo de que el producto resultante no funcione apropiadamente en el entorno de uso previsto.	Examinar el documento de requisitos.		
	Garantizar que se defina el software adecuado (es decir, el software que los usuarios esperan).		
	Verificar que el documento de requisitos este conforme a los estándares de la compañía y que sea: <ul style="list-style-type: none"> ✓ Comprensible. ✓ Coherente. ✓ Completo. 		
	Buscar errores, suposiciones erróneas, falta de claridad y desviación de la práctica estándar.		
	Validar que los requisitos satisfagan las pruebas de aceptación.		
	Planificar la forma de verificar cada requisito.		
	Diseñar las pruebas de aceptación.		
2. Explorar la adecuación y la completitud de los requisitos desarrollando representaciones del producto (p. ej., prototipos, simulaciones, modelos, escenarios, guiones gráficos) y obteniendo realimentación sobre ellos de las partes interesadas relevantes.	Asignar un grupo que lleve a cabo la revisión, es importante (al menos un representante del cliente debe ser incluido).		
	Los diferentes stakeholders, incluidos los representantes del cliente y el desarrollador, deben revisar el documento(s).		
	Identificar los problemas antes que los recursos se han comprometido a hacer frente a los requisitos.		
	Validar la calidad de los modelos conceptuales.		
	Algunos ejemplos de técnicas usadas para la validación de los requisitos son: <ul style="list-style-type: none"> ✓ Análisis. ✓ Simulaciones. ✓ Prototipos. ✓ Demostraciones. 		
3. Evaluar el diseño a medida que madure en el contexto del entorno de validación de los requisitos para identificar las cuestiones de validación, y para exponer necesidades y requisitos de cliente sin especificar.	Perfeccionar el proceso de requisitos durante todo el ciclo de vida del proyecto.		
	Orientar el proceso de los requisitos con estándares de calidad y modelos de mejora de procesos.		
	Planificar e implementar mejoras.		
	Revisar el documento de requisitos de forma sistemática, para su evaluación y aprobación.		

Fuente: [SEI, 2010] [IEEE, 2013]

Adaptado por: [Elaboración Propia]

Tabla 17. Gestión de Requisitos - Meta específica 1 – Práctica específica 1.1

Área de Proceso	Gestión de Requisitos	Acrónimo	REQM
Meta Específica	1. Gestionar los requisitos.		
Práctica Específica	1.1. Comprender los requisitos.		
Propósito			
Desarrollar una comprensión del significado de los requisitos con los proveedores de los requisitos.			
Subprácticas	Prácticas Fundamentales		
1. Establecer criterios para distinguir a los proveedores apropiados de requisitos.	<p>Establecer criterios para designar las fuentes oficiales desde las que se reciben los requisitos.</p> <p>Lista de criterios para distinguir proveedores de requisitos:</p> <ul style="list-style-type: none"> ✓ El cargo que tiene en la empresa. ✓ El conocimiento que tiene sobre la problemática presentada. ✓ El tiempo que lleva en la empresa. ✓ El grado en que empleará el producto a desarrollar. ✓ Su trabajo no se verá perjudicado por el producto. 		
2. Establecer criterios objetivos para la evaluación y aceptación de los requisitos.	<p>Algunos ejemplos de criterios de evaluación y de aceptación son:</p> <ul style="list-style-type: none"> ✓ Claramente y correctamente establecidos. ✓ Completos. ✓ Consistentes unos con otros. ✓ Identificados de forma única. ✓ Consistentes con el enfoque de la arquitectura y con las prioridades de atributos de calidad. ✓ Apropiados para implementar. ✓ Verificables (es decir, que se pueden probar). ✓ Trazables. ✓ Alcanzables. ✓ Vinculados al valor de negocio. ✓ Identificados como una prioridad para el cliente. 		
3. Analizar los requisitos para asegurar que se cumplen los criterios establecidos.	Elaborar documento del análisis de los criterios de aceptación de los requisitos.		
4. Alcanzar una comprensión de los requisitos con los proveedores de requisitos para que los participantes del proyecto puedan comprometerse con ellos.	Asegurar que se alcanza una comprensión compatible y compartida del significado de los requisitos.		
	Elaborar documento de conformidad de requisitos.		

Fuente: [SEI, 2010] [IEEE, 2013]

Adaptado por: [Elaboración Propia]

Tabla 18. Gestión de Requisitos - Meta específica 1 – Práctica específica 1.2

Área de Proceso	Gestión de Requisitos	Acrónimo	REQM
Meta Específica	1. Gestionar los requisitos.		
Práctica Específica	1.2. Obtener el compromiso sobre los requisitos.		
Propósito			
Obtener el compromiso de los participantes del proyecto sobre los requisitos.			
Subprácticas		Prácticas Fundamentales	
1. Evaluar el impacto de los requisitos sobre los compromisos existentes.	Evaluar impacto sobre los participantes del proyecto cuando cambian los requisitos o al principio de un nuevo requisito.		
	Reconocer la inevitabilidad del cambio en los requisitos y tomar medidas para mitigar sus efectos.		
	Estimar el costo de hacer el cambio: <ul style="list-style-type: none"> ✓ En términos de modificaciones en el documento de requisitos. ✓ Diseño e implementación del sistema. 		
	Elaborar documento de evaluación del impacto de los requisitos.		
2. Negociar y registrar los compromisos.	Negociar antes de que los participantes del proyecto se comprometan con un nuevo requisito o un cambio a un requisito.		
	No tomar una decisión unilateral en la resolución de problemas con requisitos.		
	Consultar con los stakeholders para llegar a un consenso sobre un cambio apropiado.		
	Negociar los cambios que son aceptables dentro del presupuesto, técnico, regulador y otras restricciones.		
	Decidir si los beneficios de la implementación de nuevos requisitos se justifican por los costos de implementación.		
	Asegurarse que los cambios pasen por: <ul style="list-style-type: none"> ✓ Una revisión definida. ✓ Un proceso de aprobación. ✓ Por un seguimiento de requisitos. ✓ Análisis de impacto. ✓ Gestión de configuración de software. 		
	Organizar el documento de requisitos para que pueda realizar cambios en él sin mucha reescritura o reorganización.		
	Actualizar documento de conformidad de requisitos con los compromisos documentados de los requisitos y de sus cambios.		

**Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]**

Tabla 19. Gestión de Requisitos - Meta específica 1 – Práctica específica 1.3

Área de Proceso	Gestión de Requisitos	Acrónimo	REQM
Meta Específica	1. Gestionar los requisitos.		
Práctica Específica	1.3. Gestionar los cambios a los requisitos.		
Propósito			
Gestionar los cambios a los requisitos a medida que evolucionan durante el proyecto.			
Subprácticas		Prácticas Fundamentales	
1. Documentar todos los requisitos y los cambios a los requisitos que se reciben o generan por el proyecto.	Utilizar herramientas de gestión de requisitos. Ej. Sistemas de gestión de requisitos, procesadores de texto, hojas de cálculo, bases de datos, etc.		
	Utilizar un identificador que permite a los requisitos ser identificados de forma única e inequívoca.		
	Documentar todos los requisitos existentes.		
2. Mantener una historia de cambios de los requisitos, incluyendo el análisis razonado de los cambios.	Mantener historial de cambios de los requisitos.		
	Seguir la volatilidad de los requisitos.		
3. Evaluar el impacto de los cambios de requisitos desde el punto de vista de las partes interesadas relevantes.	Elaborar informes de impacto del cambio de requisitos.		
4. Poner a disposición del proyecto los requisitos y los datos de los cambios.	Utilizar herramientas de gestión de requisitos. Ej. Sistemas de gestión de requisitos, procesadores de texto, hojas de cálculo, bases de datos, etc.		
	Asegurar que la información sea accesible a todos los involucrados en el proceso.		

**Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]**

Tabla 20. Gestión de Requisitos - Meta específica 1 – Práctica específica 1.4

Área de Proceso	Gestión de Requisitos	Acrónimo	REQM
Meta Específica	1. Gestionar los requisitos.		
Práctica Específica	1.4. Mantener la trazabilidad bidireccional de los requisitos.		
Propósito			
Mantener la trazabilidad bidireccional entre los requisitos y los productos de trabajo.			
Subprácticas		Prácticas Fundamentales	
1. Mantener la trazabilidad de los requisitos para asegurar que la fuente de los requisitos de nivel más bajo (es decir, inferidos) este documentada.		Establecer la trazabilidad desde los requisitos de más bajo nivel hasta sus requisitos fuente. Trazable hacia atrás.	
2. Mantener la trazabilidad de los requisitos desde un requisito a sus requisitos inferidos y a la asignación a los productos de trabajo.		Establecer la trazabilidad desde un requisito fuente hasta sus requisitos de más bajo nivel. Trazable hacia adelante.	
3. Generar una matriz de trazabilidad de requisitos.		Utilizar herramientas de soporte para trazabilidad de requisitos. Ej. Hojas de cálculo, bases de datos, etc.	
		Desarrollar matriz de trazabilidad de requisitos.	
		Actualizar la matriz de trazabilidad durante todo el ciclo de vida de un producto.	

**Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]**

Tabla 21. Gestión de Requisitos - Meta específica 1 – Práctica específica 1.5

Área de Proceso	Gestión de Requisitos	Acrónimo	REQM
Meta Específica	1. Gestionar los requisitos.		
Práctica Específica	1.5. Asegurar el alineamiento entre el trabajo del proyecto y los requisitos.		
Propósito			
Asegurar que los planes del proyecto y los productos de trabajo permanecen alineados con los requisitos.			
Subprácticas		Prácticas Fundamentales	
1. Revisar los planes del proyecto, las actividades y los productos de trabajo en cuanto a la consistencia con los requisitos y los cambios realizados sobre ellos.		Revisar matriz de trazabilidad de requisitos.	
2. Identificar la fuente de la inconsistencia (si existe).		Un procedimiento para la identificación sería: <ul style="list-style-type: none"> ✓ Ocurre un problema o petición de cambio de requisito. ✓ Se analiza y aprueba la petición de cambio. ✓ Se analiza la matriz de trazabilidad, se modifican las entidades relacionadas con dicho requisito y se verifica que los cambios se realicen en todas las entidades relacionadas al requisito. ✓ Si se identifica una inconsistencia documentarla. 	
3. Identificar cualquier cambio que se debería realizar a los planes y a los productos de trabajo resultantes de los cambios a la línea base de requisitos.		Reportar inconsistencias entre los requisitos y los planes del proyecto y los productos de trabajo, incluyendo fuentes y condiciones.	
4. Iniciar cualquier acción correctiva necesaria.		Tomar Acciones correctivas.	

**Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]**

Tabla 22. Desarrollo y Gestión de Requisitos - Meta genérica 1 – Práctica genérica 1.1

Áreas de Procesos	Desarrollo de Requisitos - Gestión de Requisitos	
Meta Genérica	1. Lograr las metas específicas.	
Práctica Genérica	1.1. Realizar las prácticas específicas.	
	Subprácticas	Prácticas Fundamentales
	1. Realizar las prácticas específicas del área de proceso para desarrollar productos de trabajo y proporcionar servicios para lograr las metas específicas del área de proceso.	Producir los productos de trabajo y entregar los servicios que se esperan al realizar (es decir, ejecutar) el proceso. Estas prácticas se pueden hacer de manera informal sin seguir una descripción documentada del proceso o un plan. El rigor con que estas prácticas se realizan depende de las personas que gestionan y realizan el trabajo, y puede variar considerablemente.

**Fuente: [SEI, 2010] [IEEE, 2013]
Adaptado por: [Elaboración Propia]**



3.1.5. Estrategia de implementación

Hasta el momento se ha desarrollado el primer componente, las prácticas fundamentales, las cuales fueron desarrolladas sobre la base de la estructura que plantea CMMI-DEV y las buenas prácticas obtenidas de la guía SWEBOK.

Como siguiente paso es necesario buscar una estrategia para llevar los procesos de la organización a cubrir y aplicar todas las características que propone el primer componente. En la búsqueda de suplir esta necesidad el SEI (Instituto de Ingeniería del Software) ha propuesto un ciclo de mejoramientos de proceso conocido como IDEAL, el cual proporciona un conjunto de actividades coherentes para sustentar la adopción de las prácticas recomendadas por CMMI-DEV. Sencillamente es una respuesta al siguiente caso:

- “Deseo mejorar mis procesos, el modelo que pienso seguir es CMMI-DEV y SWEBOK, ¿Cómo llevo mis procesos a cumplir con las características del modelo?”.

Para la elección de esta estrategia se consideraron los siguientes aspectos:

- IDEAL proporciona un conjunto de actividades coherentes para sustentar la adopción de las prácticas recomendadas por CMMI-DEV.
- Posee una adecuada documentación [SEI, 1996] y material disponible en donde se detalla los pasos a seguir para emprender el programa de mejora.
- El modelo ha sido utilizado en varios programas de mejoramiento con resultados positivos [Morales, 2013].
- El modelo es flexible en cuanto a ejecución, puesto que las prácticas sugeridas pueden variar dependiendo del contexto de la organización.

3.1.5.1. Desarrollo de la estrategia de implementación

A. INICIO

La fase de Inicio es el punto de partida, tiene por propósito identificar las razones de negocio para enfrentar el esfuerzo que requiere el proyecto.

La tabla 23 muestra las principales actividades de la fase Inicio.

Tabla 23. Actividades Principales de la fase Inicio

Fase: Inicio
Reconocer las razones del negocio de cambiar las prácticas actuales del proceso de requisitos.
Identificar las metas y objetivos del negocio que serán observados o apoyados.
Identificar los beneficios que se espera obtener al implementar el proceso de requisitos.
Mantener buenos niveles de patrocinio a través de todo el proceso.
Obtener otorgamiento de recursos por parte del patrocinador.
Establecer un mecanismo capaz de dirigir e implementar el proyecto de mejoramiento.
Definir el recurso humano requerido para el proyecto: <ul style="list-style-type: none"> • Definir los roles que cubrirá. • Definir las características personales (conocimientos, habilidades, destrezas y actitudes). • Agrupar los roles. • Crear perfiles para cada agrupación de roles.
Definir recursos materiales.
Capacitar al personal del proyecto.

Fuente: [SEI, 1996]

Adaptado por: [Elaboración Propia]

Finalizada la fase de Inicio, se procede a dar inicio al proyecto de implementación. Según el modelo IDEAL, en este punto comienzan las etapas iterativas, la primera fase de la iteración es el Diagnóstico.

B. DIAGNOSTICO

En esta fase comienza el proceso continuo de mejora y tiene por propósito establecer una línea base del estado actual de la organización. La tabla 24, muestra las principales actividades de la fase Diagnóstico.

Tabla 24. Actividades Principales de la fase Diagnóstico

Fase: Diagnóstico
Evaluar el estado actual de las prácticas.
Determinar el estado deseado que se pretende alcanzar
Identificar las debilidades y fortalezas de las prácticas actuales.
Elaborar recomendaciones.

Fuente: [SEI, 1996]

Adaptado por: [Elaboración Propia]

Para evaluar el estado actual de las prácticas, la tabla 25 detalla las fases de ejecución del diagnóstico.

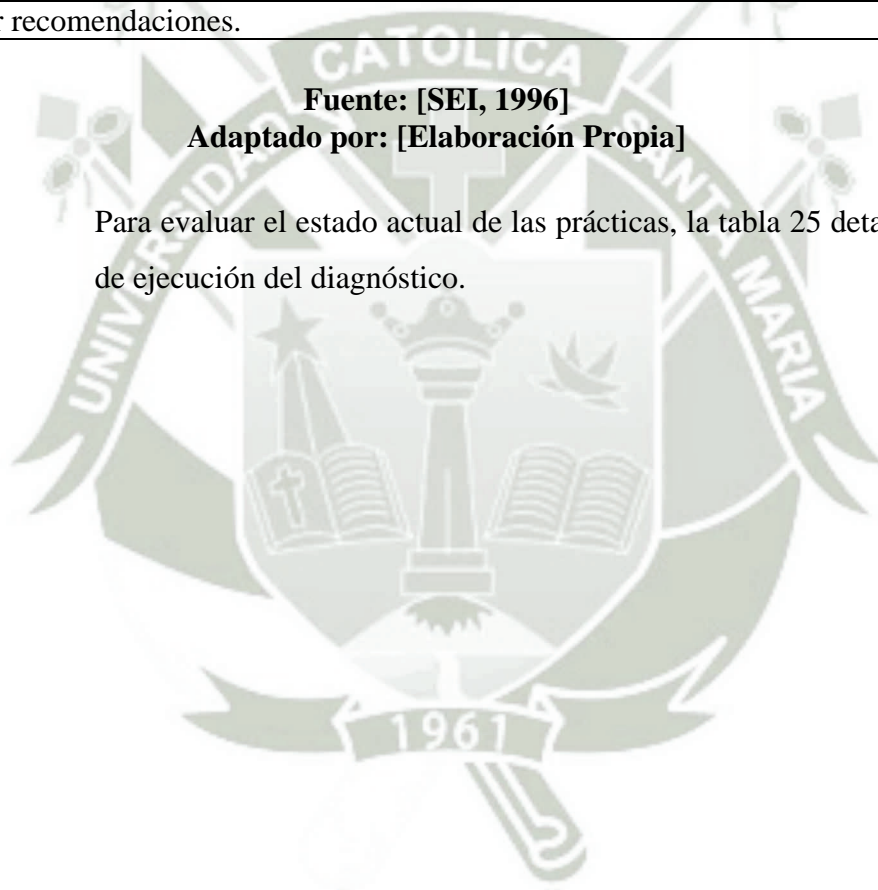


Tabla 25. Ejecución del Diagnóstico

Ejecución del Diagnóstico	
1.	Preparación
	Elaborar material para la recolección de la información.
	Establecer de manera clara cuáles serán las reglas y procedimientos a seguir, para proceder con la evaluación.
	Realizar una introducción inicial a la metodología para llevar a cabo el proceso de diagnóstico.
	Realizar una capacitación en la metodología de diagnóstico a utilizar.
	Verificar que todo lo que se definió en la planeación esté listo
	Verificar que los miembros del equipo tengan todo claro.
2.	Recolección de Información
	Realizar las entrevistas, cuestionarios, etc.
3.	Análisis de Información
	Analiza la información recolectada en la fase anterior, con el fin de poder determinar las debilidades y fortalezas.
	Dar a conocer, la información recolectada, y el análisis efectuado sobre dicha información.
	Iniciar un proceso de análisis de la información integrada.
	Tomar decisiones, para determinar las debilidades y fortalezas.
	Evaluar todas las prácticas y metas que correspondan a cada área de proceso, Desarrollo de requisitos (RD) y Gestión de requisitos (REQM).
	Elaborar informe de resultados del diagnóstico efectuado a la Organización. El informe debe contener: <ul style="list-style-type: none"> ✓ Las debilidades y fortalezas de dicha organización. ✓ Las recomendaciones, sugerencias y pasos a seguir por la misma.
4.	Presentación de resultados
	Presentar a las personas autorizadas, los resultados obtenidos.
	Entregar el informe de los resultados del diagnóstico al Patrocinador.
5.	Cierre de evaluación
	Elaborar documento de lecciones aprendidas.
	El documento debe incluir aspectos negativos como los aspectos positivos que formaron parte del proceso de diagnóstico.
	Entregar el documento al Patrocinador.

Fuente: [Arias y Mejía, 2006]
Adaptado por: [Elaboración Propia]

Una vez finalizada esta fase, la organización debe tener bases firmes para establecer un foco, teniendo clara la capacidad de los procesos a la luz del modelo y qué prácticas de las áreas de proceso Desarrollo de requisitos (RD) y Gestión de requisitos (REQM) están cubiertas por el estado actual; como resultado, se debe saber hacia qué nivel dirigirse y por ende cuáles serán las prácticas en las que se deberán enfocar los esfuerzos de mejora.

C. ESTABLECIMIENTO:

El propósito general de esta fase es establecer un plan de trabajo detallado, con el fin de llevar a la organización hacia el nivel deseado, se fijan las prioridades que reflejen las recomendaciones hechas en la fase Diagnóstico. La tabla 26 muestra las principales actividades de la fase Establecimiento.

Tabla 26. Actividades Principales de la fase Establecimiento

Fase: Establecimiento
Evaluar las necesidades y recomendaciones que se hicieron evidentes en la fase de Diagnóstico.
Establecer las prioridades acorde con las necesidades y recomendaciones, considerando: <ul style="list-style-type: none"> • Restricciones del orden lógico y de implementación de las Áreas de Proceso de requisitos. • Restricciones de tiempo. • Restricciones de recursos. • Restricciones de personal. • Consecuencias de las acciones en los procesos. • Tareas o necesidades que dependan de alguna otra. • Importancia para la estrategia organizacional.
Definir una estrategia de mejora identificando qué es lo que se debe lograr (las prioridades) y luego cómo se logrará.

Fuente: [SEI, 1996]

Adaptado por: [Elaboración Propia]

Una vez definido el proyecto, se puede pasar a la acción en la cual se desarrollará y se pondrá a prueba la solución.

Hasta este momento se cuenta con:

- Recursos definidos y negociados para el proyecto.
- Diagnóstico y recomendaciones.
- Prioridades de cada área de proceso y procesos a ser intervenidos.
- Estrategia de la organización con la cual se abordará el proyecto.

D. ACCIÓN

El propósito de esta fase es implementar la mejora de procesos, se ejecuta todo lo planeado y conceptualizado en las fases anteriores. La tabla 27 muestra las principales actividades de la fase Acción.

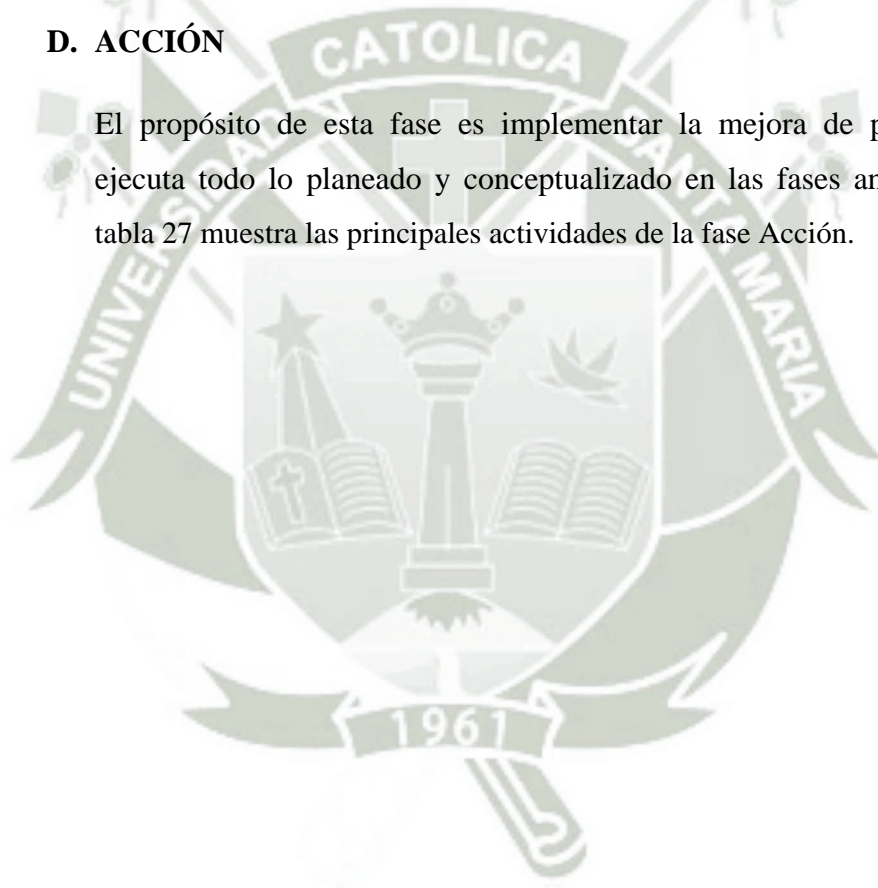


Tabla 27. Actividades Principales de la fase Acción

Fase: Acción
<p>Crear una solución para resolver las necesidades de la organización:</p> <ul style="list-style-type: none"> • Tener presentes las necesidades a cubrir con la modificación del proceso y las áreas de proceso que se van a integrar. • Convocar las reuniones necesarias, asegurándose de contar con los recursos y personal adecuado. • Ejecutar las reuniones informando claramente los objetivos y necesidades a cubrir. • Documentar los procesos modificados. • Aprobar formalmente los cambios a los procesos.
Probar las soluciones en proyectos pilotos
Tomar en cuenta la retroalimentación, sugerencias y dudas respecto del nuevo proceso.
Corregir los problemas que surgieron de la prueba piloto,
Realizar un análisis profundo de todas las sugerencias, adiciones y cambios al proceso.
<p>Generar acciones correctivas en los procesos a partir de las experiencias adquiridas:</p> <ul style="list-style-type: none"> • Generar y ejecutar acciones correctivas que se consideren pertinentes. • Documentar todas las acciones que se pusieron en marcha a partir de las sugerencias hechas por los involucrados. • Comunicar al personal que los cambios y sugerencias valieron la pena e hicieron parte activa del mejoramiento del proceso. • Actualizar la documentación que se tenga de los procesos.
<p>Ejecutar y generar los nuevos procesos refinados y ajustados:</p> <ul style="list-style-type: none"> • Explicar los nuevos procesos ya refinados al resto de la organización. • Mostrar y evaluar el valor agregado que se está generando con el cambio, basado en las pruebas piloto. • Planear e informar la fecha límite para usar cada uno de los nuevos procesos.

Fuente: [SEI, 1996]

Adaptado por: [Elaboración Propia]

Una vez terminada las actividades, se debe tener como objetivos cumplidos, lo siguiente:

- El personal de la organización que ejecutarán cada uno de los procesos entienden el proceso.
- El personal sabe dónde documentarse respecto a cualquier proceso y a quién pedir ayuda, en caso de ser necesario.

- Las personas involucradas se encuentran motivadas para adaptarse al cambio.

Al finalizar esta fase, los procesos deben quedar lo suficientemente claros como para ser aplicados a lo largo de la organización. Se debe preparar entonces para una etapa de retroalimentación.

E. APRENDIZAJE

En esta fase se completa el ciclo de mejoramiento, el propósito de esta fase es aprender de la experiencia del ciclo recién realizado. La tabla 28 muestra las principales actividades de la fase Aprendizaje.

Tabla 28. Actividades Principales de la fase Aprendizaje

Fase: Aprendizaje
Asegurar que las soluciones han sido desarrolladas y aplicadas.
Identificar el grado en que el esfuerzo invertido logro los propósitos deseados.
Examinar los objetivos de la Organización e identificar el nivel de cumplimiento en base a lo construido hasta el momento: <ul style="list-style-type: none"> • Lo que se tiene, se tenía y a lo que se quiere llegar, en términos del estado de los procesos. • El esfuerzo invertido hasta el momento. • Analizar y documentar las lecciones aprendidas.
Difundir el conocimiento adquirido.
Corregir y tomar acciones sobre los procesos para los cuales se haya generado algún tipo de cambio.
Desarrollar y documentar las recomendaciones basadas en el análisis y las validaciones

Fuente: [SEI, 1996]

Adaptado por: [Elaboración Propia]

Esta fase completa el ciclo de mejoramiento, ejecutando esta fase correctamente y ajustando las estrategias, la siguiente ejecución será aún más precisa y efectiva. Esta mejora de procesos debe ser administrada de manera eficiente para asegurar su éxito. Es importante documentar las lecciones aprendidas y utilizar esto como fuente de información para otros proyectos.

CAPITULO IV: REALIDAD DE LA EMPRESA

4.1.DESCRIPCIÓN DE LA EMPRESA

La organización en estudio De Larrobla & Asociados provee desde hace más de 23 años, soluciones tecnológicas para Instituciones Financieras en todo Latinoamérica.

La organización se focaliza en el desarrollo de Sistemas de Misión Crítica para Instituciones Financieras, la Implementación de los mismos en Casa del Cliente, y su mantenimiento.

En la actualidad cuenta con más de 40 instalaciones distribuidas en 12 países de la región, entre las que se encuentran prestigiosos Bancos de renombre internacional.

Su familia de productos Bantotal, comprende soluciones que se caracterizan por el procesamiento integral de todas las operaciones, mediante la integración natural entre sus módulos, con contabilidad on-line y la integridad de la información que administra; explotando las facilidades de las bases de datos relacionales.

Dispone de soluciones “core” para Bancos (tanto mayoristas como minoristas); y soluciones especializadas para las áreas de Microfinanzas, Comercio Exterior, Tesorería (incluyendo operaciones relacionadas con los Mercados de Dinero, Divisas y Valores), Canales de Distribución (Home-Banking), Información Gerencial (basada en tecnología OLAP), e Información a Organismos Reguladores y de Supervisión en cada país donde se instale. Asimismo, cuenta con soluciones para el procesamiento integral de Fondos de Inversión y Empresas Fiduciarias.

El compromiso en la entrega de las soluciones descritas (para el alcance acordado, en plazo y dentro del presupuesto) es el valor principal de la cultura de la compañía.

Los productos de la Organización cuentan con más de 40 instalaciones a nivel de Latinoamérica, siendo los principales países: Argentina, Bolivia, Colombia, Chile, El Salvador, Guatemala, Honduras, México, Panamá, Paraguay, Perú, Uruguay, USA (Miami) y Venezuela.

En la definición del valor principal de la cultura de la compañía se puede evidenciar que el Centro de Desarrollo se encuentra directamente involucrado por medio del compromiso de la organización: “Entregar soluciones para el alcance acordado, en plazo y dentro del presupuesto”.

Debido a esto, el Centro de Desarrollo definió como principal objetivo proveer a la organización las soluciones necesarias para cumplir con estos compromisos.

4.2.EVALUACIÓN DE LA SITUACIÓN ACTUAL

Alcance de la evaluación

La organización cuenta con el Centro de Desarrollo de software llamado “BanTotal Centro de Desarrollo”, área a estudiar en el presente proyecto.

El proceso de desarrollo de software definido por el Centro de Desarrollo, establece las fases de requisitos, diseño, construcción, pruebas y liberación, las cuales se ejecutan secuencialmente y adoptan conceptos de las metodologías tradicionales.

El Alcance de esta evaluación se enfoca en el análisis y propuesta de mejora en el proceso de Requisitos.

Descripción de las fuentes de información utilizadas

Se realizaron entrevistas y encuestas al personal con el objetivo de conocer más a fondo la forma de trabajo, conocer su percepción sobre el proceso actual de requisitos, 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.

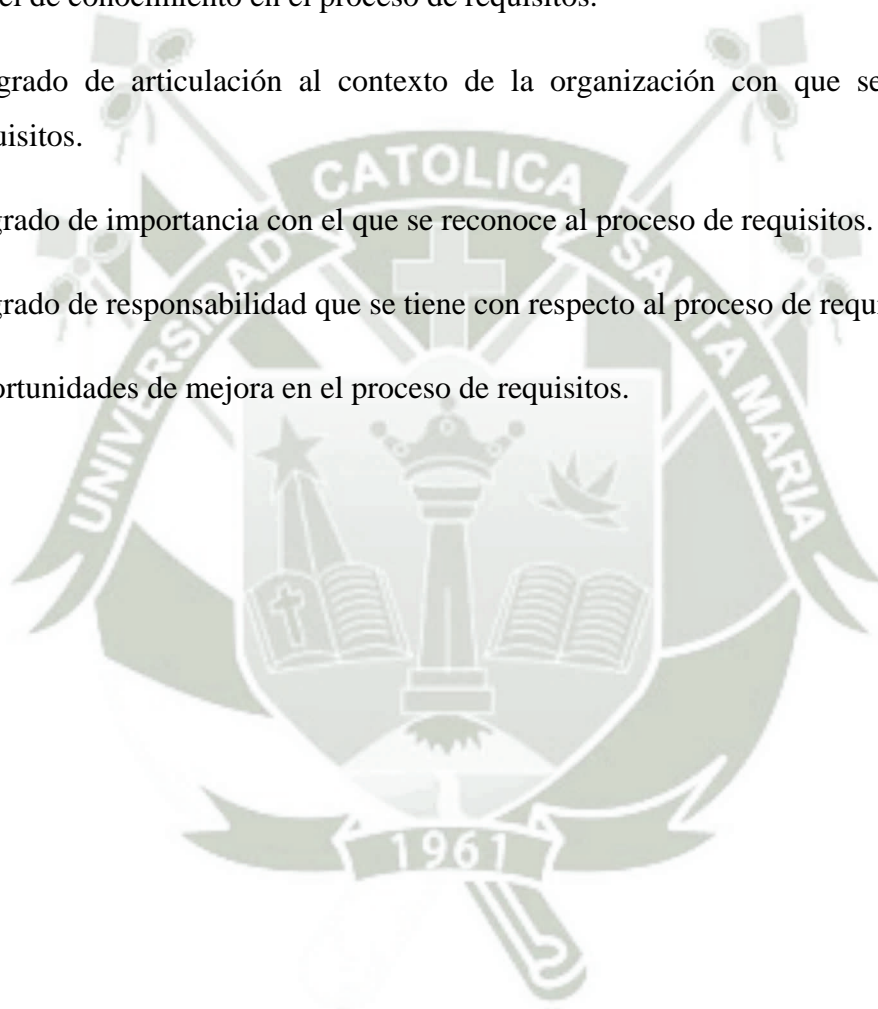
Los roles de las personas entrevistadas fueron:

- Jefes de Desarrollo.
- Analistas funcionales.
- Analistas técnicos.

Se estableció como instrumento una encuesta cerrada con una muestra de 17 participantes (Ver Anexo 1), la cual está conformada de preguntas cerradas, estas preguntas limitan a los encuestados a seleccionar una única respuesta en un conjunto limitado de opciones, esto con el propósito de obtener datos que faciliten el entendimiento de las fortalezas y debilidades respecto al área de proceso evaluada.

Estas respuestas nos permitirán evaluar los siguientes puntos:

- Nivel de conocimiento en el proceso de requisitos.
- El grado de articulación al contexto de la organización con que se define los requisitos.
- El grado de importancia con el que se reconoce al proceso de requisitos.
- El grado de responsabilidad que se tiene con respecto al proceso de requisitos.
- Oportunidades de mejora en el proceso de requisitos.



4.2.1. Presentación de resultados

Se obtuvo los siguientes resultados generales luego de la primera evaluación y entrevistas con los miembros de la organización objeto de estudio.

Nivel de conocimiento en el proceso de requisitos

Para conocer el nivel de conocimiento en el proceso de requisitos, se analizaron las respuestas de las preguntas 1, 2 y 3 del instrumento (Ver Anexo 1), permitiendo establecer que el 100% de los encuestados conocen el significado de un requisito y cuál es su principal objetivo en el proceso de requisitos, tal como se muestra en la figura 13.

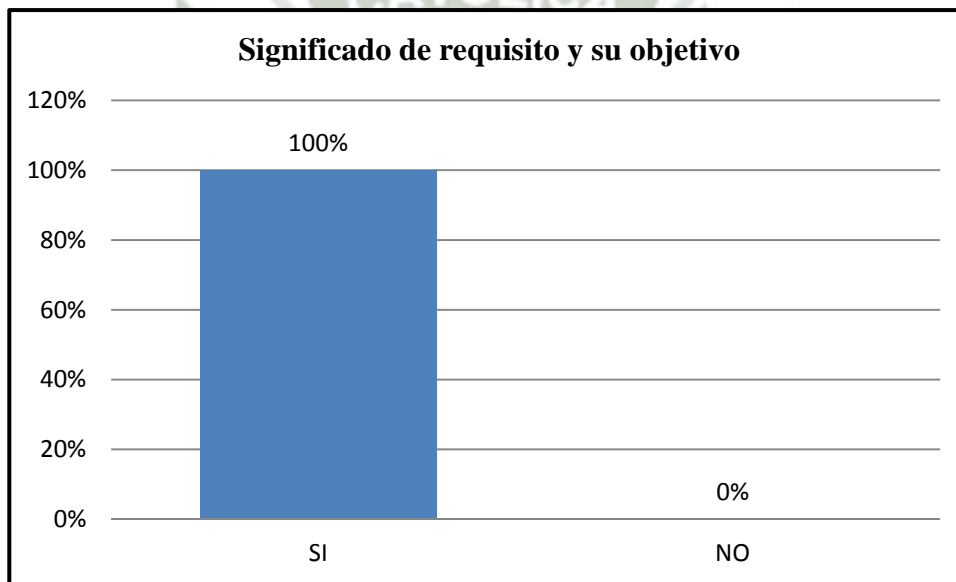


Figura 13. Significado de requisito y su objetivo
Fuente: [Elaboración propia]

Por otro lado, mientras que el 12% de los encuestados siempre tienen claro el resultado que esperan al finalizar la implementación de un requisito y el 71% lo saben casi siempre (ver figura 14), sólo el 13% conoce siempre la información que se necesita para obtener el resultado esperado del requisito (ver figura 15), 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 reproceso.

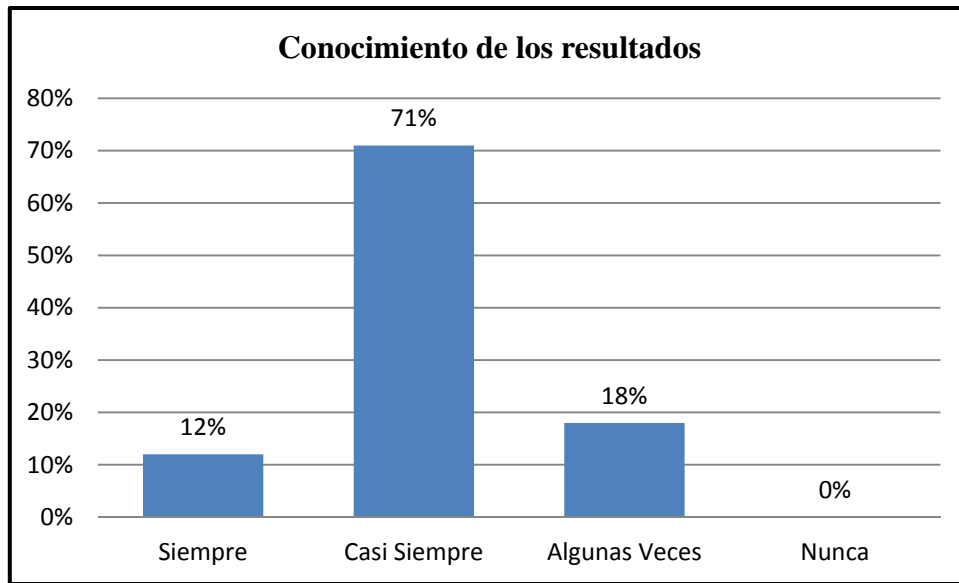


Figura 14. Conocimiento de los resultados
Fuente: [Elaboración propia]

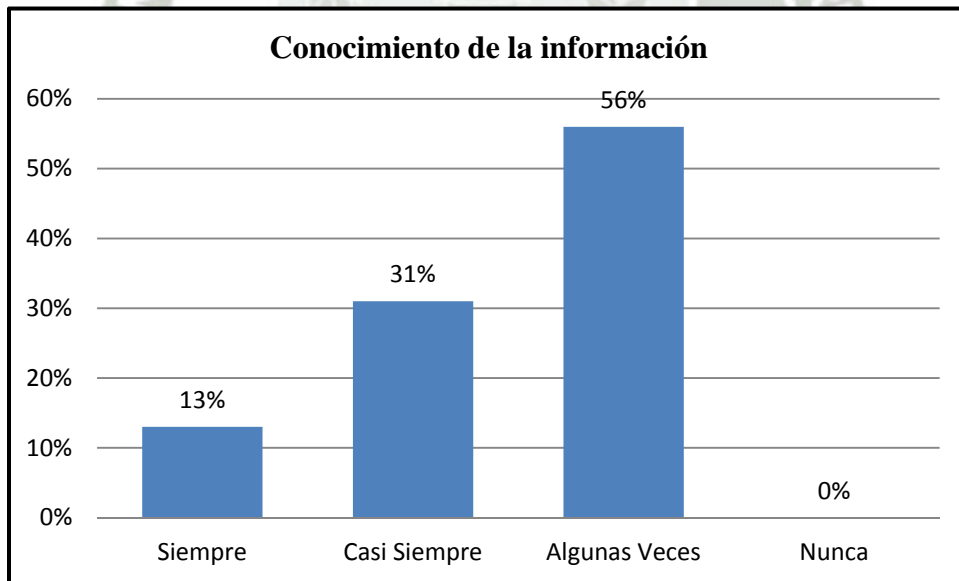


Figura 15. Conocimiento de la información
Fuente: [Elaboración propia]

En base a la información expuesta, se puede concluir que generalmente se sabe el resultado final que se desea obtener, sin embargo, no siempre se conoce la información necesaria que se requiere para lograr dicho resultado.

Grado de articulación al contexto de la organización con que se define los requisitos

Con respecto a que si los requisitos definidos tienen en cuenta los lineamientos establecidos por los entes de control y por la organización, se procedió a analizar las respuestas de las preguntas 4, 5 y 6 del instrumento (Ver Anexo 1) y se identificó que sólo el 29% 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. Ver figura 16.

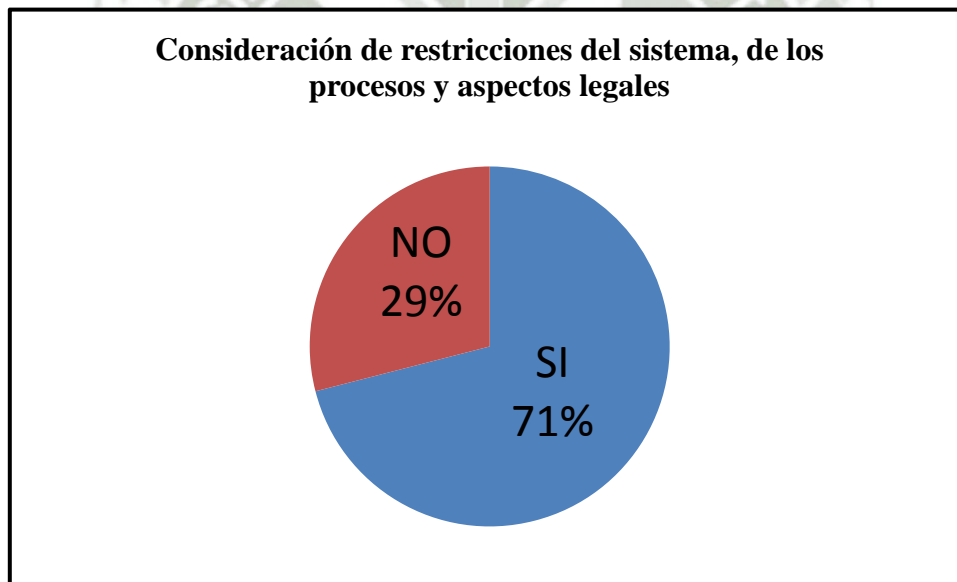


Figura 16. Consideración de restricciones del sistema, procesos y aspectos legales
Fuente: [Elaboración propia]

También se pudo establecer que casi siempre conocen los procesos de la organización que serán afectados y las políticas definidas por la organización al momento de definir nuevos requisitos, no obstante otro porcentaje de los encuestados considera que no tienen un buen conocimiento sobre estos aspectos, como se puede ver en las figuras 17 y 18.

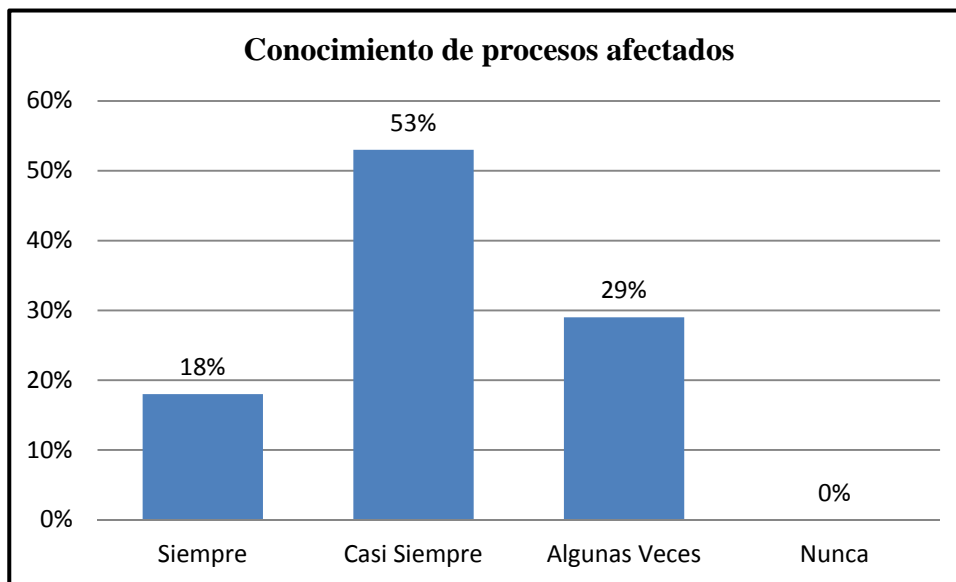


Figura 17. Conocimiento de procesos afectados
Fuente: [Elaboración propia]

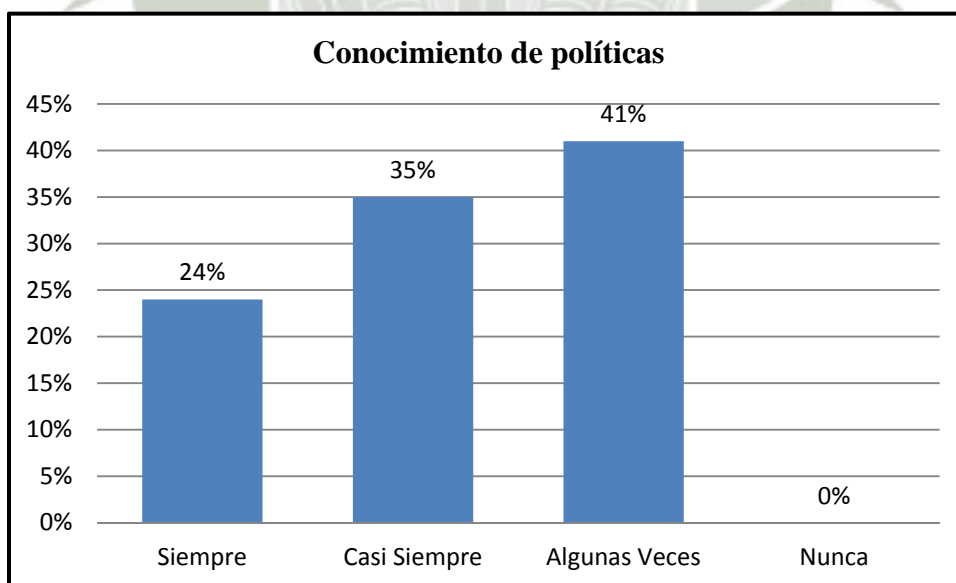


Figura 18. Conocimiento de políticas
Fuente: [Elaboración propia]

En conclusión, se puede afirmar que el grado de articulación de los requisitos con el contexto de la organización es aceptable, debido a que más de la mitad de los encuestados tienen en cuenta los procesos y políticas establecidos por la organización.

Grado de importancia con el que se reconoce al proceso de requisitos

De otro lado, se analizaron las preguntas 7, 8, 9 y 10 del instrumento (Ver Anexo 1) con el fin de determinar el grado de importancia que tiene el proceso de requisitos y se puede observar que la mayoría está de acuerdo en que una mala definición de un requisito puede afectar considerablemente el desarrollo del proyecto (Ver figura 19).

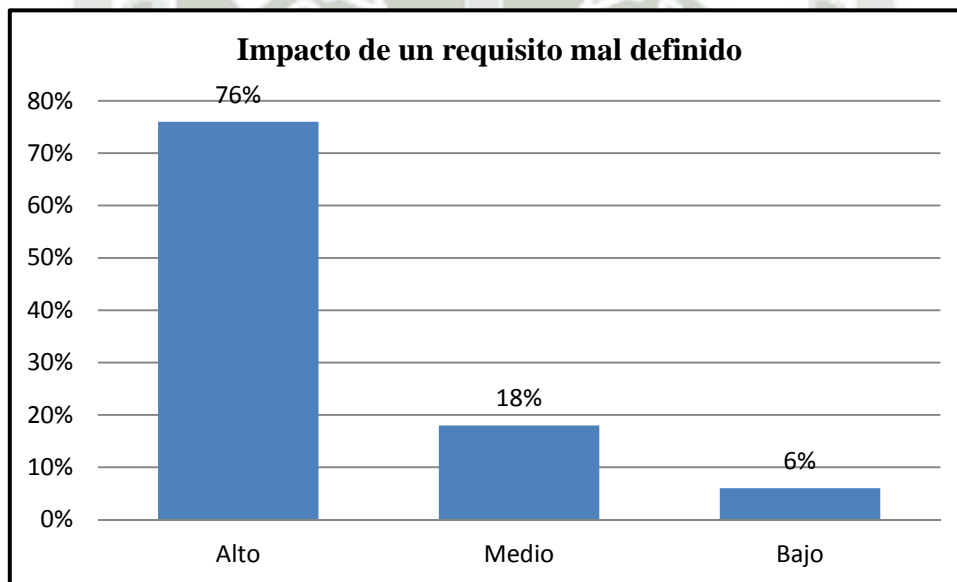


Figura 19. Impacto de un requisito mal definido
Fuente: [Elaboración propia]

Debido a esto, el 81% de los encuestados son conscientes de los recursos que se requieren y conocen el impacto que puede generar una mala definición de requisitos en el proyecto (Ver figura 20).

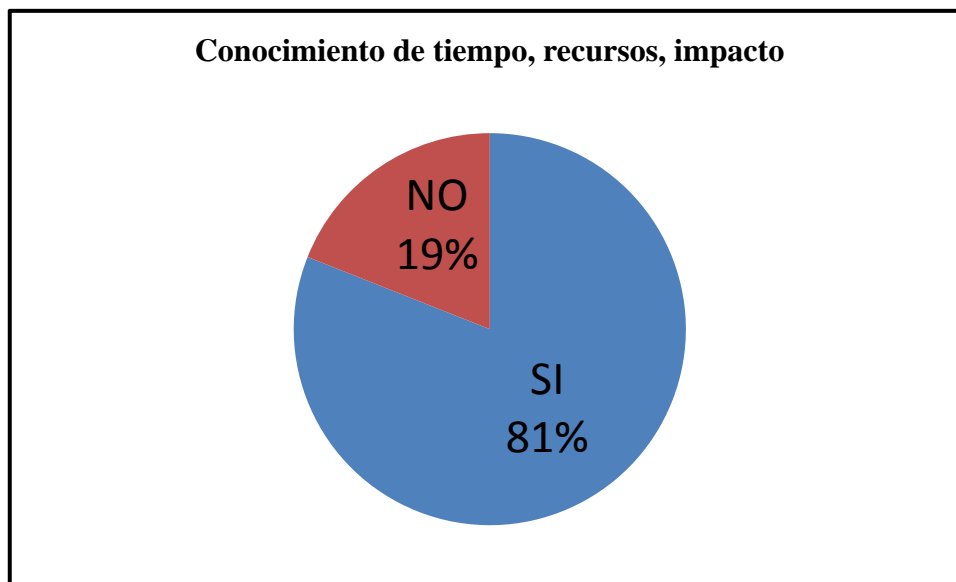


Figura 20. Conocimiento de tiempo, recursos, impacto
Fuente: [Elaboración propia]

Por otra parte el 65% especifican claramente las validaciones que se deben realizar al momento de definir los requisitos (Ver figura 21), sin embargo, el 12% indica que no se priorizan los requisitos (Ver figura 22).

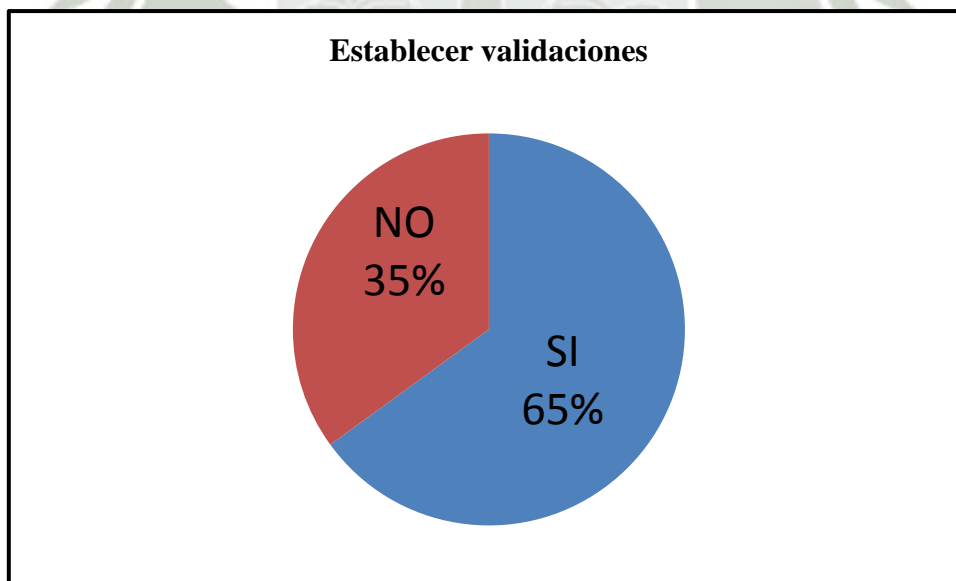


Figura 21. Establecer validaciones
Fuente: [Elaboración propia]

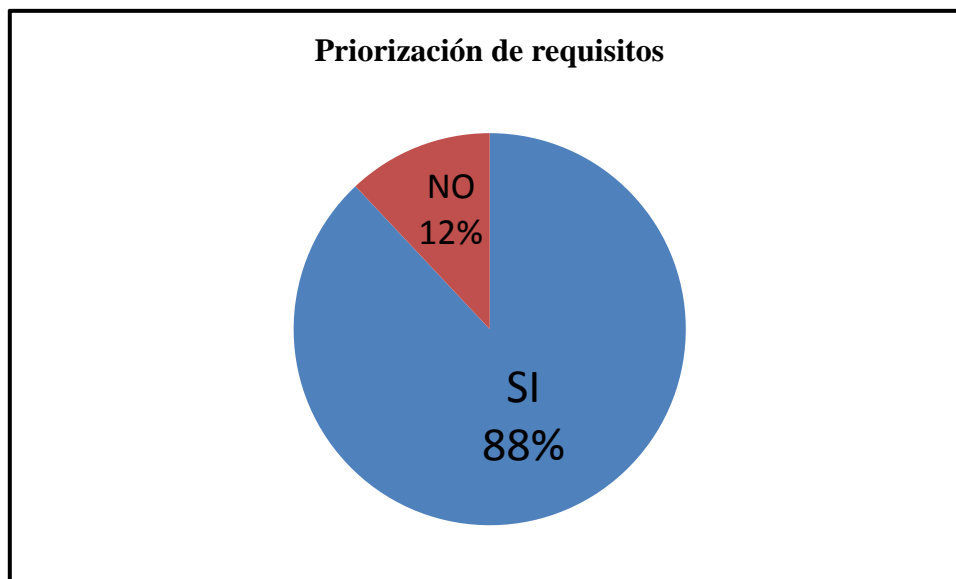


Figura 22. Priorización de requisitos

Fuente: [Elaboración propia]

En base a los datos mostrados, se puede concluir que los encuestados consideran que el proceso de requisitos es importante para el éxito de un proyecto, aunque un pequeño porcentaje no prioricen los requisitos.

Grado de responsabilidad que se tiene con respecto al proceso de requisitos

Las preguntas 11 y 12 del instrumento (Ver Anexo 1) están asociadas a la responsabilidad que se tiene con respecto al proceso de requisitos, el 82% considera que los responsables de definir los requisitos deben ser personas expertas en el proceso (Ver figura 23).

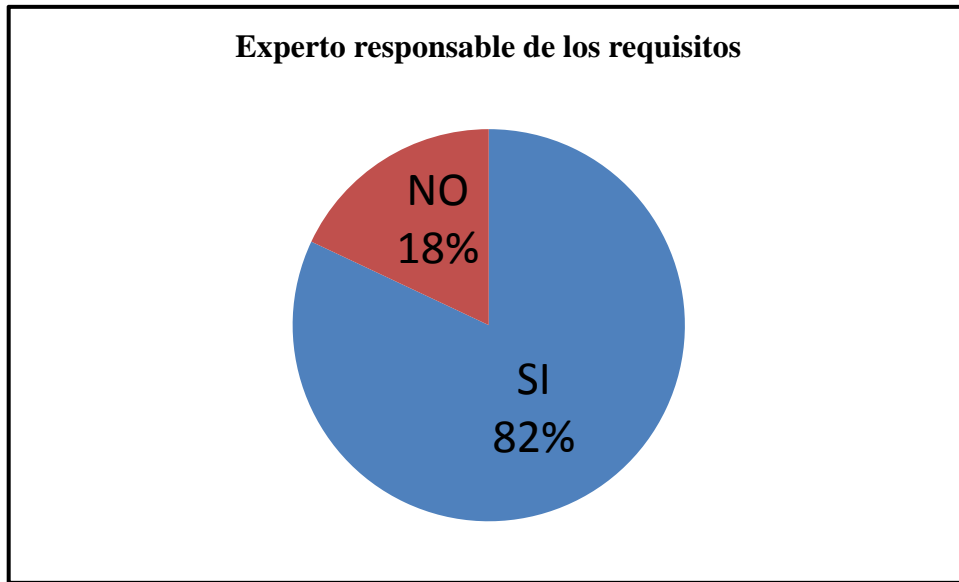


Figura 23. Experto responsable de los requisitos
Fuente: [Elaboración propia]

Por otra parte el 47% considera muy bueno el nivel de equipamiento de la organización al momento de definir los requisitos (Ver figura 24).

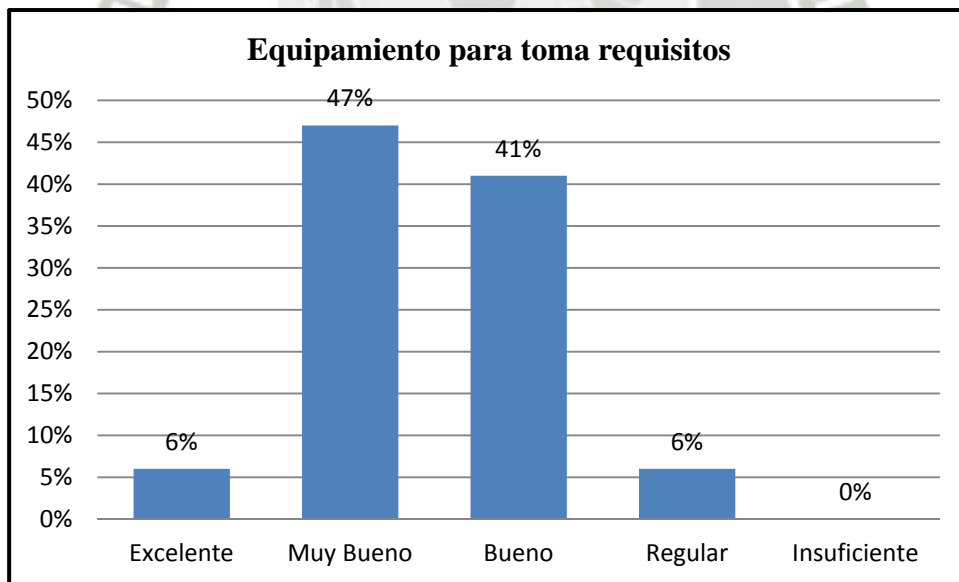


Figura 24. Equipamiento para toma requisitos
Fuente: [Elaboración propia]

Oportunidades de mejora en el proceso de requisitos

Para finalizar, la pregunta 13 del instrumento (Ver Anexo 1) está asociada a las oportunidades de mejora en el proceso de requisitos, luego del análisis de misma, se logró establecer que todos los encuestados consideran que se deben hacer mejoras a las prácticas actuales que se ejecutan dentro de la organización. Ver figura 25.

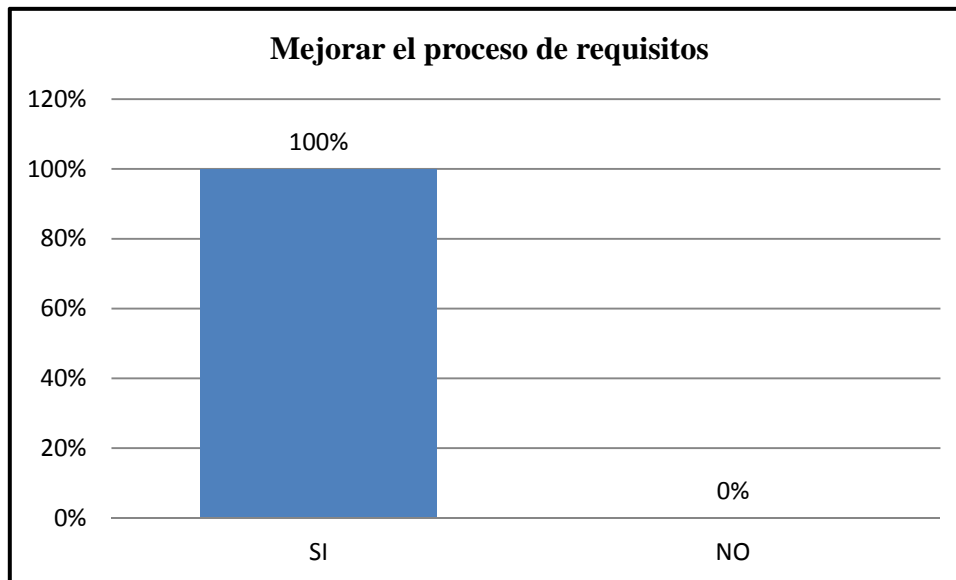


Figura 25. Mejorar el proceso de requisitos
Fuente: [Elaboración propia]

En base a lo expuesto se llega a la conclusión que existe la necesidad de mejorar los procesos de requisitos, para lo cual se detectaron los problemas que actualmente tiene la organización.

Los problemas que se detallan a continuación se tomaron en cuenta sobre la base de las áreas de proceso de Desarrollo de requisitos (RD) y Gestión de requisitos (REQM) del modelo CMMI-DEV.

Problemas Detectados

En el proceso de Desarrollo de requisitos (RD):

- Se suele especificar requisitos que lleva a la confusión entre el personal y los clientes.
- Al interpretar los requisitos de diseño, implementación y prueba, estos algunas veces suelen ser incompatibles.
- Se tarda un tiempo excesivo para obtener un acuerdo sobre el diseño del producto.
- Existe un mayor potencial de aumento de los costos para satisfacer las expectativas de los clientes.
- Algunos clientes no dedican tiempo a trabajar con los analistas funcionales sobre los requisitos.
- Muchas veces el equipo no puede interactuar directamente con los usuarios representativos para entender sus necesidades.
- Los desarrolladores identifican ambigüedades y falta de información al momento del desarrollo del software, por lo tanto algunas veces suponen la solución.
- Algunas veces los clientes solicitan cierta funcionalidad, esta se desarrolla, pero luego el cliente no la pone en funcionamiento.

En el proceso de Gestión de Requisitos (REQM)

- Los requisitos son aceptados por el personal de cualquier fuente que considere que son autorizados.
- En ocasiones existe la incapacidad para demostrar que el producto cumple con los requisitos aprobados.

- La falta de trazabilidad de los requisitos a menudo da lugar a pruebas de productos incompletas o incorrectas.
- La especificación de requisitos muchas veces no es clara.
- Los clientes afirman que todos los requisitos son críticos y por lo tanto no establecen prioridades.
- Los clientes aprueban los requisitos y luego los cambian continuamente.
- Los clientes no aprueban los requisitos.



CAPITULO V: VALIDACIÓN

5.1.DESARROLLO DE LA VALIDACIÓN DEL MODELO

A continuación se detalla el proceso de mejora realizado en la organización objeto de estudio con el objetivo de implementar y mejorar los procesos de desarrollo y gestión de requisitos.

5.1.1. Fase Inicio

La iniciativa para el mejoramiento de los procesos de requisitos en la organización fue propuesta a raíz de la realización de este trabajo de tesis. Además, tal como se detalla al final del Capítulo IV, se llega a la conclusión que existe la necesidad de mejorar los procesos de requisitos en la organización.

Una vez identificado el estímulo para el cambio, se procedió a definir la meta del programa:

- Analizar el proceso de requisitos actual con el propósito de mejorar el proceso de desarrollo y gestión de requisitos en el contexto del desarrollo de proyectos de software.

También se definió los posibles beneficios que se puede obtener al implementar el proceso de requisitos, estos pueden ser:

- Incrementar la calidad del proceso de requisitos.
- Formalización de procesos de requisitos en la organización.
- Reducir los costos y retrasos del proyecto.
- Mejorar la comunicación entre el cliente y el equipo del proyecto.
- Satisfacción del cliente.

Para lograr el respaldo para la ejecución de este proyecto, se realizó una presentación en base a los puntos anteriores. Después de hacer la presentación y discusión de los argumentos, se decidió apoyar el proyecto de mejora.

Posteriormente se continuó con la definición del equipo de trabajo, la tabla 29 muestra las responsabilidades de los roles involucrados en el esquema de trabajo.

Tabla 29. Roles y responsabilidades y habilidades de cada grupo del proyecto

Rol	Responsabilidades
Comité directivo	<ul style="list-style-type: none"> ✓ Alinear el proyecto de mejoramiento con la visión, misión y objetivos de la compañía. ✓ Definir las prioridades con respecto a las prácticas a ser adoptadas. ✓ Dar soporte a la implementación de las recomendaciones aprobadas, apoyando el uso permanente de dichas recomendaciones. ✓ Dar soluciones a las dificultades identificadas. ✓ Monitorear periódicamente el progreso del programa de mejoramiento.
Rol	Responsabilidades
Grupo de Ingeniería de procesos	<ul style="list-style-type: none"> ✓ Proveer la capacitación necesaria en los procesos organizacionales así como métodos y tecnologías relacionadas con las áreas de proceso de requisitos. ✓ Liderar los ciclos de mejoramiento de procesos incluyendo las actividades de diagnóstico, evaluaciones y certificaciones de los procesos de la organización. ✓ Monitorear, controlar y comunicar periódicamente el estado de avance del proyecto de mejoramiento. ✓ Establecer y mantener una librería de activos de procesos. ✓ Apoyar a los equipos de proyectos en el uso de los procesos estándar de la compañía. ✓ Guiar y apoyar la planeación, desarrollo e implantación de herramientas de software de apoyo a los procesos.
Rol	Responsabilidades
Equipo de definición de procesos	<ul style="list-style-type: none"> ✓ Desarrollar un plan de implementación por cada área de proceso: Desarrollo de requisitos (RD) y Gestión de requisitos (REQM). ✓ Aportar ideas de mejoras a los procesos organizacionales actuales. ✓ Definir los procesos, contribuyendo cada miembro del equipo con su punto de vista de acuerdo con la labor que desempeña en la organización. ✓ Revisar e integrar los procesos que se definan con los otros procesos ya definidos de la organización. ✓ Ajustar los procesos definidos de acuerdo con la retroalimentación que se reciba de las diversas fuentes. ✓ Preparar y realizar la capacitación necesaria al resto de la organización. ✓ Reportar el estado de sus actividades.

Rol	Responsabilidades
Gerente de proyecto	<ul style="list-style-type: none"> ✓ Responsable de institucionalizar las prácticas definidas dentro de la organización. ✓ Asegurar el debido involucramiento de cada profesional en el logro de los objetivos de la empresa. ✓ Ayudar al correcto entendimiento y uso del proceso definido. ✓ Coordinar eventuales ajustes o mejoras con el Grupo de ingeniería de procesos.
Rol	Responsabilidades
Equipo evaluador	<ul style="list-style-type: none"> ✓ Documentar y mantener los insumos del diagnóstico. ✓ Comprometerse con el Acuerdo de Confidencialidad y de no Atribución. ✓ Estar familiarizados con la estructura de la organización. ✓ Asegurarse que el diagnóstico sea conducido de acuerdo con lo que se formalizó en el documento de planeación. ✓ Difundir la Metodología del proceso de diagnóstico a los Miembros del Equipo Evaluador. ✓ Preparar a las personas a entrevistar o las personas encargadas de suministrar información.

Fuente: [Elaboración Propia]

5.1.2. Fase Diagnostico

En esta fase se procedió a evaluar el estado actual de las prácticas, el objetivo de la evaluación (Ver Anexo 2) fue establecer la brecha entre las prácticas actuales que maneja la organización y las prácticas recomendadas por el modelo CMMI-DEV.

Este instrumento está totalmente estructurado, dado que las preguntas fueron planeadas, analizadas y evaluadas antes de la ejecución, la encuesta adoptó conceptos de las prácticas recomendadas por CMMI-DEV, para lo cual se decidió involucrar los siguientes roles: Jefes de Desarrollo, Analistas funcionales y Analistas técnicos.

La encuesta está conformada por 40 preguntas con una muestra de 17 participantes y con ellas se desea responder los siguientes puntos:

- Nivel de adopción de las prácticas sugeridas por CMMI-DEV para el Desarrollo de Requisitos dentro de las actividades de desarrollo de software definido por la organización.
- Nivel de adopción de las prácticas sugeridas por CMMI-DEV para la Gestión de Requisitos dentro de las actividades de desarrollo de software definido por la organización.
- Prácticas menos utilizadas por la organización en el proceso de Desarrollo y Gestión de Requisitos sugeridas por CMMI-DEV.

La tabla 30 muestra la relación entre las preguntas de la encuesta y las prácticas recomendadas por CMMI-DEV.

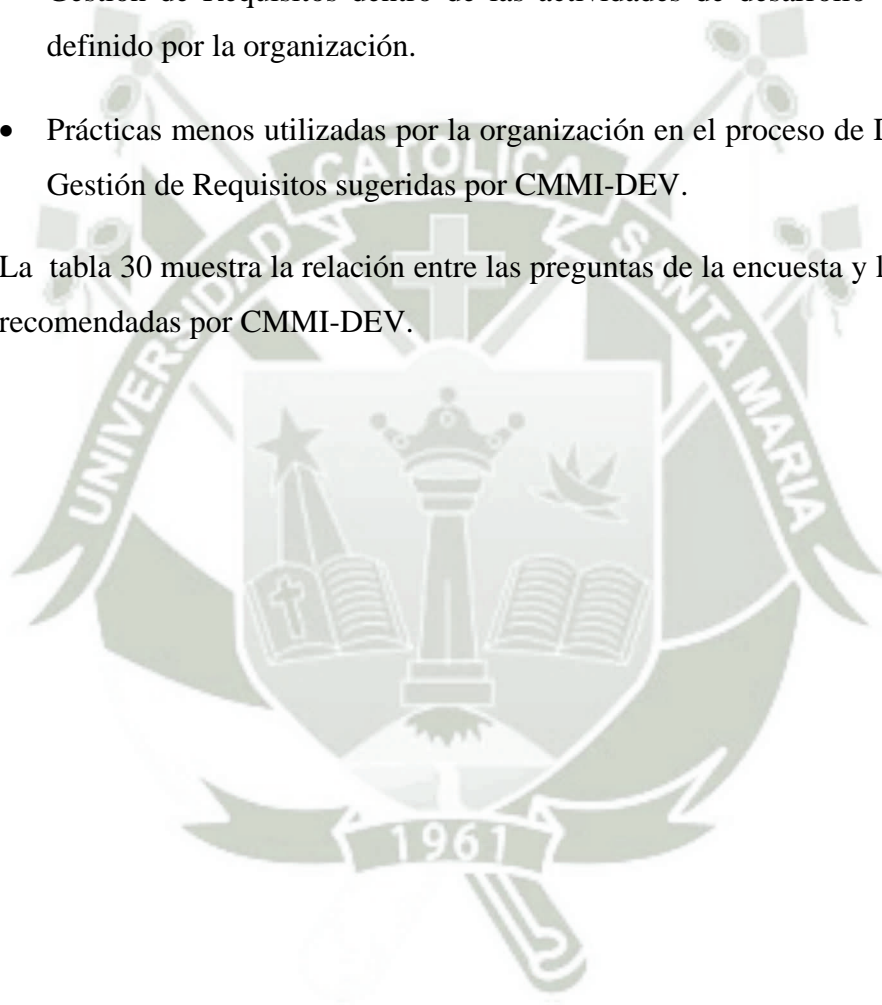


Tabla 30. Relación entre prácticas recomendadas y preguntas de la encuesta

Área de Proceso	Metas específicas (SG)	Prácticas específicas (SP)	# Pregunta
Desarrollo de Requisitos (RD)	1. Desarrollar los requisitos del cliente.	1.1. Educar las necesidades.	1
		1.2. Transformar las necesidades de las partes interesadas en requisitos de cliente.	2, 3, 4, 5
	2. Desarrollar los requisitos de producto.	2.1. Establecer los requisitos de producto y de componente de producto.	6, 7, 8
		2.2. Asignar los requisitos de componente de producto.	9, 10, 11
		2.3. Identificar los requisitos de interfaz.	12, 13, 14
	3. Analizar y validar los requisitos.	3.1. Establecer los conceptos y los escenarios de operación.	15, 16, 17
		3.2. Establecer una definición de la funcionalidad y de los atributos de calidad requeridos.	18, 19, 20, 21, 22
		3.3. Analizar los requisitos.	23, 24, 25, 26
		3.4. Analizar los requisitos para conseguir un equilibrio.	27, 28
		3.5. Validar los requisitos.	29
Gestión de Requisitos (REQM)	1. Gestionar los requisitos.	1.1. Comprender los requisitos.	30, 31, 32
		1.2. Obtener el compromiso sobre los requisitos.	33, 34
		1.3. Gestionar los cambios a los requisitos.	35, 36, 37, 38
		1.4. Mantener la trazabilidad bidireccional de los requisitos.	39
		1.5. Asegurar el alineamiento entre el trabajo del proyecto y los requisitos.	40

Fuente: [Elaboración Propia]

Una vez finalizada la encuesta, a continuación se presentan los resultados obtenidos luego del análisis correspondiente, a través de estos resultados se logró resolver las interrogantes planteadas como base para la generación del cuestionario.

Presentación de resultados

Nivel de adopción de prácticas CMMI-DEV para Desarrollo de Requisitos

De acuerdo a la primera interrogante y a la información suministrada por los encuestados, se pudo identificar el nivel de ejecución de las prácticas específicas, el nivel de cumplimiento de las metas específicas, y finalmente concluir con el nivel de adopción del área de proceso de Desarrollo de Requisitos.

En la figura 26, se puede observar las prácticas que tienen mayor o menor porcentaje de ejecución, según la opinión de los expertos.



Figura 26. Cumplimiento de Prácticas específicas – Desarrollo de Requisitos

Fuente: [Elaboración Propia]

Respecto a las metas específicas sugeridas para el área de Desarrollo de Requisitos, se presentan porcentajes medios de cumplimiento tal como se aprecia en la figura 27, donde la meta “SG 3 Analizar y validar los requisitos” destaca sobre las demás metas por tener un porcentaje de cumplimiento bajo.

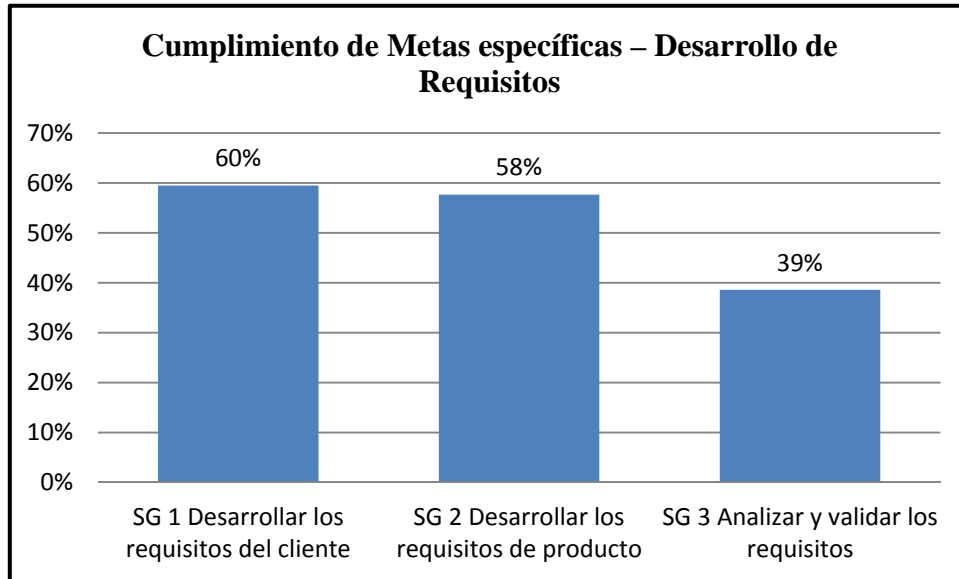


Figura 27. Cumplimiento de Metas específicas – Desarrollo de Requisitos
Fuente: [Elaboración Propia]

Finalmente, a partir de los datos presentados se puede dar respuesta a la primera interrogante planteada, concluyendo a partir del porcentaje de cumplimiento de las metas específicas que la organización solo ejecuta en un 52% las prácticas recomendadas por CMMI-DEV en el Área de Desarrollo de Requisitos, lo cual es coherente con el estado actual y problemáticas identificadas. Ver figura 28.

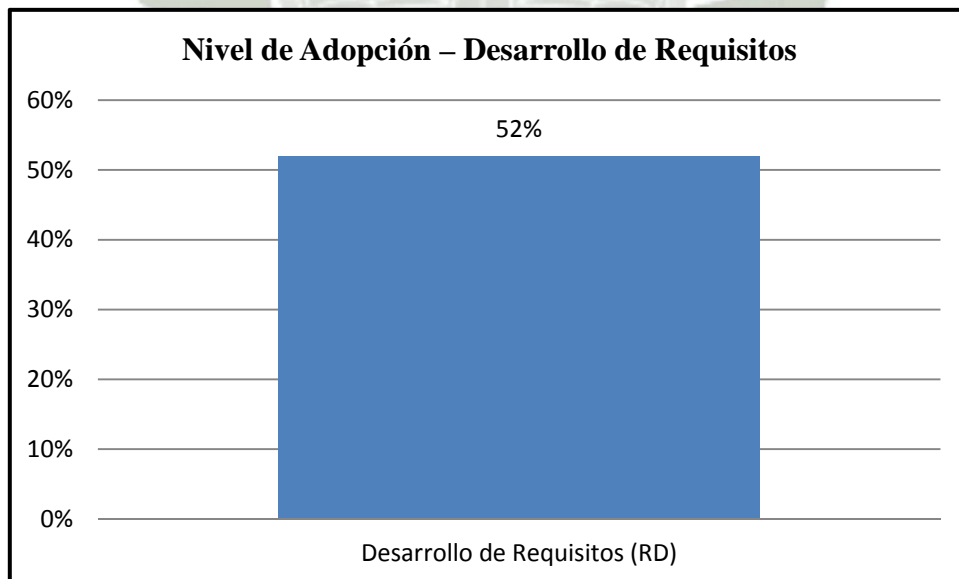


Figura 28. Nivel de Adopción – Desarrollo de Requisitos
Fuente: [Elaboración Propia]

Nivel de adopción de prácticas CMMI-DEV para Gestión de Requisitos

Para dar respuesta a la segunda interrogante fue necesario evaluar las respuestas de varias preguntas, esto con el fin de conocer el nivel de ejecución de las prácticas específicas 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 de Gestión de Requisitos.

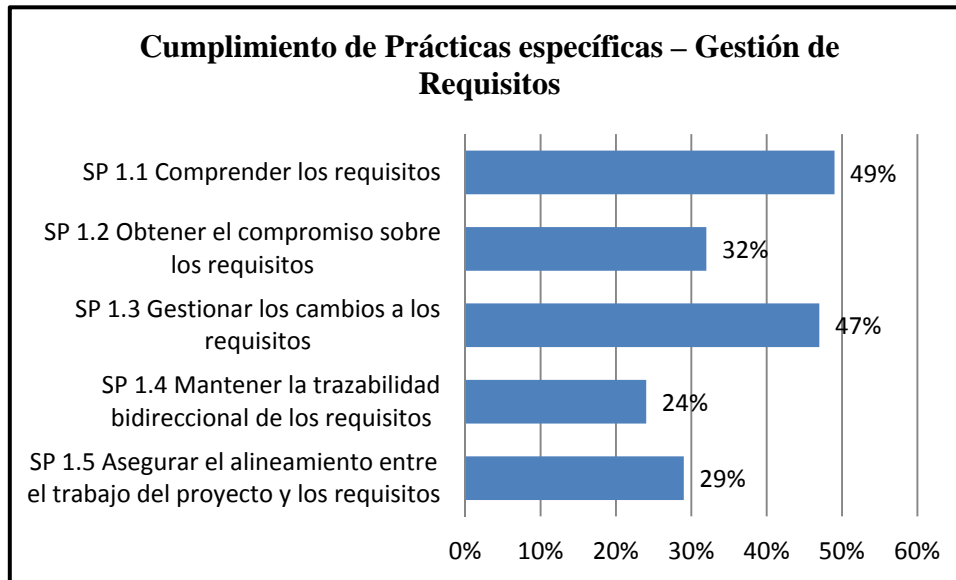


Figura 29. Cumplimiento de Prácticas específicas – Gestión de Requisitos

Fuente: [Elaboración Propia]

En la figura 29 se puede observar que la práctica específica más adoptada es: “SP 1.1 Comprender los requisitos”, pero uno de los aspectos más alarmante de este análisis, es que las prácticas específicas menos utilizadas son: “SP 1.2 Obtener el compromiso sobre los requisitos” y “SP 1.4 Mantener la trazabilidad bidireccional de los requisitos”, lo cual es concordante con las problemáticas expuestas 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 36%. Ver figura 30.

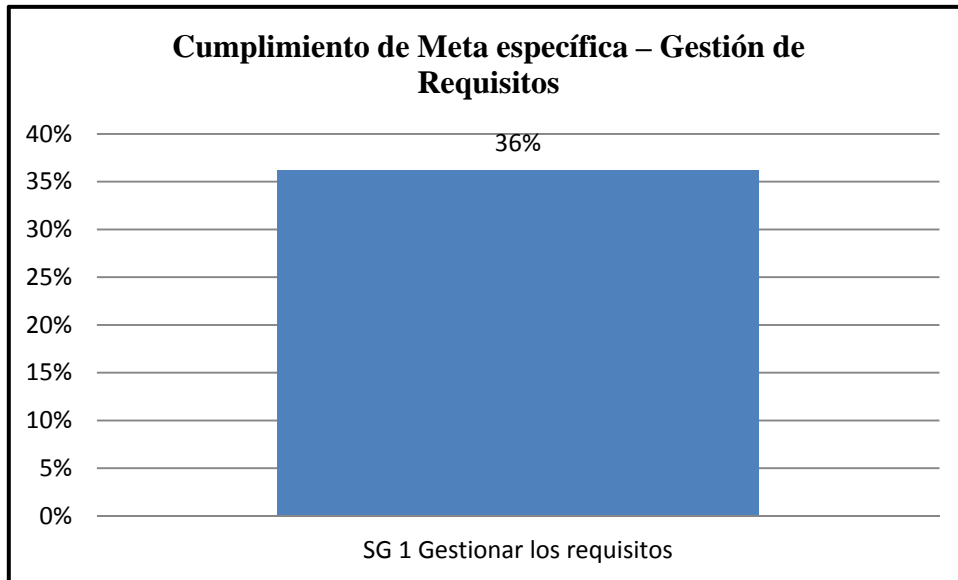


Figura 30. Cumplimiento de Meta específica – Gestión de Requisitos
Fuente: [Elaboración Propia]

Por lo tanto, a partir de los datos presentados se puede dar respuesta a la segunda interrogante planteada, concluyendo que el porcentaje de adopción de las prácticas establecidas por CMMI-DEV en el área de proceso Gestión de Requisitos, es tan sólo del 36%.

Prácticas menos utilizadas en el proceso de Desarrollo y Gestión de Requisitos sugeridas por CMMI-DEV

Finalmente en base a los datos ya expuestos se da respuesta a la última interrogante, para lo cual se identificaron las cinco (5) prácticas menos utilizadas por la organización en el Proceso de Requisitos sugeridas por CMMI-DEV. La tabla 31 muestra los resultados:

Tabla 31. Prácticas menos utilizadas en Proceso de Requisitos

Área de proceso	Práctica específica	%
RD	SP 3.3 Analizar los requisitos	21%
RD	SP 3.4 Analizar los requisitos para conseguir un equilibrio	32%
REQM	SP 1.2 Obtener el compromiso sobre los requisitos	32%
REQM	SP 1.4 Mantener la trazabilidad bidireccional de los requisitos	24%
REQM	SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos	29%

Fuente: [Elaboración Propia]

Recomendaciones de prácticas

Después de realizar un análisis de la información recolectada y basándose en el primer componente del modelo propuesto, se procedió a seleccionar las prácticas fundamentales necesarias con el propósito de construir un conjunto de recomendaciones que permitan lograr la meta establecida para el programa de mejora.

1. Los analistas deben ser los responsables de educir los requisitos en compañía de las personas expertas en los procesos.
2. Los analistas deben hacer uso de más técnicas para educir las necesidades, por ejemplo: Lluvia de ideas, escenarios, prototipos, observación, historias de usuario, etc.

3. Los analistas deben ser los responsables de elaborar el documento de la definición del sistema (a veces conocido como documento de requisitos de usuario).
4. Los analistas deben identificar si el requisito es funcional o no funcional (atributos de calidad).
5. Los analistas deben clasificar los requisitos por prioridad con el fin de resolver los conflictos y planificar las entregas, por ejemplo: Obligatorio, muy deseable, deseable u opcional.
6. Los analistas deben elaborar el documento de especificación de requisitos de software, el cual debe estar expresado en los términos técnicos necesarios para el diseño del producto.
7. Los analistas deben establecer y mantener las relaciones entre los requisitos para su consideración durante la gestión del cambio y la asignación de los requisitos.
8. Los analistas deben desarrollar los requisitos de interfaz del producto.
9. Los analistas deben desarrollar escenarios y elaborar modelos conceptuales para entender la situación en la que se produce el problema y representar una solución. Ej. Diagramas de casos de uso, etc.
10. Los analistas deben establecer y mantener una definición de requisitos funcionales y de los requisitos no funcionales (atributos de calidad).
11. Los analistas deben expresar los requisitos claramente e inequívocamente como sea posible y en su caso, en términos cuantitativos.
12. El jefe de proyecto debe asignar un grupo que lleve a cabo la revisión del documento de especificación de requisitos de software, al menos un representante del cliente debe ser incluido.

13. El grupo que lleve a cabo la revisión debe analizar los requisitos para asegurarse que son completos, factibles, realizables y verificables.
14. El grupo que lleve a cabo la revisión debe evaluar el impacto de los requisitos en los costes y riesgos del desarrollo del producto.
15. El grupo que lleve a cabo la revisión del documento de requisitos debe verificar que este conforme a los estándares de la compañía y que sea: comprensible, coherente y completo.
16. El jefe del proyecto debe establecer criterios para designar las fuentes oficiales desde las que se reciben los requisitos. Ej.
 - El cargo que tiene en la empresa.
 - El conocimiento que tiene sobre la problemática presentada.
 - El tiempo que lleva en la empresa.
 - El grado en que empleará el producto a desarrollar.
 - Su trabajo no se verá perjudicado por el producto.
17. Los analistas deben alcanzar una comprensión de los requisitos con los proveedores de requisitos para que los participantes del proyecto puedan comprometerse con ellos.
18. El equipo del proyecto debe evaluar el impacto sobre los participantes del proyecto cuando cambian los requisitos o cuando se especifica un nuevo requisito.
19. Los analistas deben negociar antes de que los participantes del proyecto se comprometan con un nuevo requisito.
20. Los analistas deben obtener el compromiso de los participantes del proyecto sobre los requisitos.

21. Los analistas deben documentar todos los requisitos y los cambios a los requisitos que se reciben o generan por el proyecto.
22. Los analistas deben mantener una historia de cambios de los requisitos.
23. Los analistas deben desarrollar y actualizar la matriz de trazabilidad de requisitos durante todo el ciclo de vida del producto.

5.1.3. Fase Establecimiento

Siguiendo las recomendaciones, se procedió a impartir capacitación al personal con el fin de acelerar el entendimiento de las prácticas y nivelar el conocimiento de todo el personal.

Posteriormente, debido al alto costo de implementar todas las recomendaciones al mismo tiempo, se procedió a establecer prioridades acorde con las necesidades y recomendaciones que surgieron en la fase Diagnóstico. Para esto se decidió agrupar estas recomendaciones en metas, considerando a cada meta una unidad de trabajo, teniendo en cuenta el dominio a la que pertenecen en base a las áreas de proceso de Desarrollo y Gestión de Requisitos. Ver tabla 32.

Tabla 32. Priorización de las recomendaciones

Metas específicas	Recomendaciones
1. Desarrollar los requisitos del cliente.	<p>Los analistas deben ser los responsables de educir los requisitos en compañía de las personas expertas en los procesos.</p> <p>Los analistas deben hacer uso de más técnicas para educir las necesidades, por ejemplo: Lluvia de ideas, escenarios, prototipos, observación, historias de usuario, etc.</p> <p>Los analistas deben ser los responsables de elaborar el documento de la definición del sistema (a veces conocido como documento de requisitos de usuario).</p> <p>Los analistas deben identificar si el requisito es funcional o no funcional (atributos de calidad).</p> <p>Los analistas deben clasificar los requisitos por prioridad con el fin de resolver los conflictos y planificar las entregas, por ejemplo: Obligatorio, muy deseable, deseable u opcional.</p>
2. Desarrollar los requisitos de producto.	<p>Los analistas deben elaborar el documento de especificación de requisitos de software, el cual debe estar expresado en los términos técnicos necesarios para el diseño del producto.</p> <p>Los analistas deben establecer y mantener las relaciones entre los requisitos para su consideración durante la gestión del cambio y la asignación de los requisitos.</p> <p>Los analistas deben desarrollar los requisitos de interfaz del producto.</p>
3. Analizar y validar los requisitos.	<p>Los analistas deben desarrollar escenarios y elaborar modelos conceptuales para entender la situación en la que se produce el problema y representar una solución. Ej: Diagramas de casos de uso, etc.</p> <p>Los analistas deben establecer y mantener una definición de requisitos funcionales y de los requisitos no funcionales (atributos de calidad).</p> <p>Los analistas deben expresar los requisitos claramente e inequívocamente como sea posible y en su caso, en términos cuantitativos.</p> <p>El jefe de proyecto debe asignar un grupo que lleve a cabo la revisión del documento de especificación de requisitos de software, al menos un representante del cliente debe ser incluido.</p> <p>El grupo que lleve a cabo la revisión debe analizar los requisitos para asegurarse que son completos, factibles, realizables y verificables.</p> <p>El grupo que lleve a cabo la revisión debe evaluar el impacto de los requisitos en los costes y riesgos del desarrollo del producto.</p> <p>El grupo que lleve a cabo la revisión del documento de requisitos debe verificar que este conforme a los estándares de la compañía y que sea: comprensible, coherente y completo.</p>

Metas específicas	Recomendaciones
<p>4. Gestionar los requisitos.</p>	<p>El jefe del proyecto debe establecer criterios para designar las fuentes oficiales desde las que se reciben los requisitos. Ej.</p> <ul style="list-style-type: none"> ✓ El cargo que tiene en la empresa. ✓ El conocimiento que tiene sobre la problemática presentada. ✓ El tiempo que lleva en la empresa. ✓ El grado en que empleará el producto a desarrollar. ✓ Su trabajo no se verá perjudicado por el producto. <p>Los analistas deben alcanzar una comprensión de los requisitos con los proveedores de requisitos para que los participantes del proyecto puedan comprometerse con ellos.</p> <p>El equipo del proyecto debe evaluar el impacto sobre los participantes del proyecto cuando cambian los requisitos o cuando se especifica un nuevo requisito.</p> <p>Los analistas deben negociar antes de que los participantes del proyecto se comprometan con un nuevo requisito.</p> <p>Los analistas deben obtener el compromiso de los participantes del proyecto sobre los requisitos.</p> <p>Los analistas deben documentar todos los requisitos y los cambios a los requisitos que se reciben o generan por el proyecto.</p> <p>Los analistas deben mantener una historia de cambios de los requisitos.</p> <p>Los analistas deben desarrollar y actualizar la matriz de trazabilidad de requisitos durante todo el ciclo de vida del producto.</p>

Fuente: [Elaboración Propia]

Después de establecer la prioridad, se procedió a definir para cada meta específica, los recursos necesarios para su ejecución, es decir, establecer las personas que harán parte del equipo de trabajo y la infraestructura requerida para el éxito del mismo.

5.1.4. Fase Acción

En esta fase se evaluó los proyectos de software en curso, esto con el fin de establecer que proyectos serían los pilotos. Al principio se contempló la idea de esperar un nuevo proyecto y desde su inicio comenzar con la prueba de los nuevos procesos, idea que luego fue descartada.

Luego, por el contrario se decidió ir poniendo a prueba las recomendaciones en proyectos que se encontraban en curso, teniendo en cuenta que el proyecto elegido no represente un problema y de este modo disminuir el riesgo de deserción o resistencia al cambio por parte del equipo de trabajo o del cliente.

Antes de que se iniciara la puesta en marcha del proyecto, se realizaba una reunión donde se indicaba claramente las prácticas que se iban a adoptar y como se debían de llevar a cabo, simulando las prácticas que se iban a emplear en el transcurso del proyecto.

Después de dar inicio al proyecto piloto, se utilizó la técnica de observación la cual permitía dar un seguimiento al trabajo realizado, esto permitió recopilar información para realizar mejoras.

Durante el proceso de implementación se presentaron algunas barreras que significaron un riesgo para la implementación, la principal barrera se relaciona con la resistencia al cambio. Sin embargo esto fue cambiando progresivamente.

Logros alcanzados

- Se logró educir los requisitos en compañía de las persona expertas en los procesos reduciendo el tiempo al momento de obtener el acuerdo sobre el diseño del producto.
- Se logró utilizar técnicas (entrevistas, reuniones y observación) para educir los requisitos del cliente.
- Se logró clasificar los requisitos por los siguientes conceptos: Tipo de Evento, Sistema y Severidad (Impide Operar, No impide operar, Sugerencia, Normativo).
- Se logró elaborar el documento de especificación de requisitos basado en plantillas institucionales.
- Se logró especificar los requisitos en términos técnicos necesarios para su desarrollo, reduciendo así las ambigüedades y falta de información al momento del desarrollo del software.
- Se logró poner a disposición del proyecto mediante un Sistema Web de Gestión de Proyectos toda la información de los requisitos.
- Se logró especificar los requisitos de interfaz: diseños de pantallas, formularios con diseños de reportes, los cuales se adjuntan en el documento de especificación de requisitos.
- En compañía del cliente se llevaba a cabo la revisión del documento de especificación de requisitos, minimizando de esta manera la confusión que existía entre el personal y los clientes al momento de especificar los requisitos.
- Se logró diseñar una matriz de validación la cual se adjunta al documento de especificación de requisitos para su posterior verificación y homologación reduciendo así la incompatibilidad que existía.

- Se logró indicar el impacto del nuevo requisito en otras funcionalidades del sistema dentro del documento de especificación de requisitos.
- Los cambios a los requisitos se volvían a estimar, generando controles de cambio.
- Se negocian los requisitos nuevos o cambios antes de ser aprobados, reduciendo los costos para satisfacer las expectativas de los clientes.
- Los requisitos son gestionados mediante un Sistema Web de Gestión de Proyectos, siendo el Jefe del proyecto el encargado de obtener el compromiso de los participantes del proyecto.
- Los requisitos nuevos son documentados y sus cambios también en el documento de especificación de requisitos y en la herramienta de Sistema Web de Gestión de Proyectos.
- Se logró elaborar y mantener una matriz de trazabilidad.

Problemas identificados

- Desconocimiento de los requisitos no funcionales por parte del cliente.
- Algunas veces los clientes afirman que todos los requisitos son críticos y por lo tanto no establecen prioridades.
- Por cuestiones de tiempo y conocimiento no se desarrollan escenarios o modelos conceptuales.
- Algunas veces las personas expertas en los procesos no dedicaban tiempo a trabajar con los analistas, por lo que los requisitos se obtienen de otras fuentes.
- Algunas veces no se estima correctamente el total de horas de estimación, dando lugar a problemas de entrega del producto.

5.1.5. Fase de Aprendizaje

En esta fase se analizaron los logros alcanzados, las ocurrencias y percances que fueron detallados a manera de lecciones aprendidas, con el fin de que pueda ser de utilidad como guía base en los próximos ciclos de mejora.



5.2.EVALUACIÓN DEL MODELO

Después de haber aplicado el cuestionario (Ver anexo 2) con el objetivo de reflejar la realidad de las prácticas que maneja la organización y las prácticas recomendadas por el modelo y luego de haber presentado el modelo en la organización objeto de estudio y habiendo tomado conocimiento y compromiso de aplicar las prácticas a partir de ese momento con el compromiso que en un futuro próximo mínimo de 2 meses sería nuevamente evaluado.

Se presenta a continuación la evaluación final del modelo propuesto, donde se aplicó un cuestionario (Ver anexo 3) del universo de 17 personas con una muestra de 12 expertos, las mismas que fueron tabuladas y cuyos resultados se muestran a continuación. (Ver tablas 33 y 34)

Tabla 33. Relación entre Variables, Indicadores e Ítems del cuestionario

Variable	Indicadores	Ítem del cuestionario
Variable Independiente Modelo de integración para desarrollar y gestionar requisitos de software	Claridad	Pregunta N° 2
	Mejora del desarrollo de requisitos	Pregunta N° 3
		Pregunta N° 4
		Pregunta N° 5
	Mejora de la gestión de requisitos	Pregunta N° 6
	Comprensión	Pregunta N° 8
	Satisfacción de expectativas	Pregunta N° 9
Variable Dependiente Desarrollo de proyectos de software	Desarrollo y gestión de requisitos en el desarrollo de proyectos de software	Pregunta N° 7

Fuente: [Elaboración Propia]

Tabla 34. Tabulación de respuestas

N°	PREGUNTA	RESPUESTA	CANT.	%
1	¿Qué opinión tiene usted del modelo propuesto?	Muy Bueno	8	67%
		Bueno	4	33%
		Malo	0	0%
		Pésimo	0	0%
2	¿Cómo calificaría usted la claridad del modelo propuesto?	Muy Bueno	5	42%
		Bueno	7	58%
		Malo	0	0%
		Pésimo	0	0%
3	¿Considera usted que la aplicación del modelo permite mejorar el desarrollo de los requisitos del cliente?	Bastante	10	83%
		Si	2	17%
		Muy poco	0	0%
		No	0	0%
4	¿Considera usted que la aplicación del modelo permite mejorar el desarrollo de los requisitos de producto?	Bastante	1	8%
		Si	11	92%
		Muy poco	0	0%
		No	0	0%
5	¿Considera usted que la aplicación del modelo permite mejorar el análisis y validación de los requisitos?	Bastante	3	25%
		Si	9	75%
		Muy poco	0	0%
		No	0	0%
6	¿Considera usted que la aplicación del modelo permite mejorar la gestión de los requisitos?	Bastante	4	33%
		Si	8	67%
		Muy poco	0	0%
		No	0	0%
7	¿Cree usted que es necesario emplear este tipo de modelo para llevar a cabo el desarrollo y gestión de requisitos en el desarrollo de proyectos de software?	Si es necesario	9	75%
		Algo necesario	3	25%
		Muy poco necesario	0	0%
		No es necesario	0	0%
8	¿El modelo propuesto es fácil de entender y comprender?	Muy fácil	4	33%
		Fácil	7	58%
		Difícil	1	8%
		Muy difícil	0	0%
9	¿El modelo propuesto satisface sus expectativas?	Satisface mucho	5	42%
		Satisface	7	58%
		Satisface poco	0	0%
		No satisface	0	0%
10	¿Considera usted necesario institucionalizar el modelo propuesto en la empresa?	Si es necesario	10	83%
		Algo necesario	2	17%
		Muy poco necesario	0	0%
		No es necesario	0	0%

Fuente: [Elaboración Propia]

Interpretación de resultados

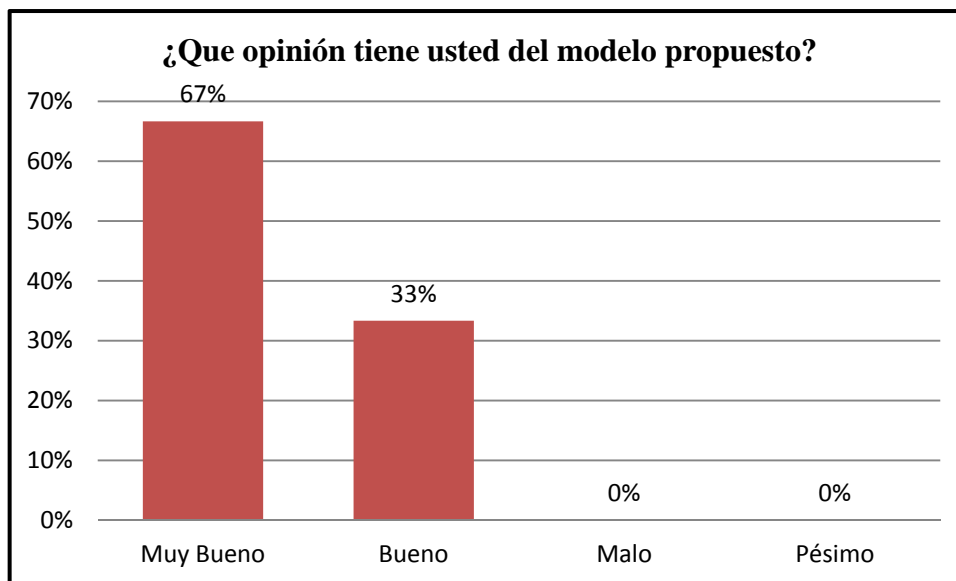


Figura 31. Interpretación de resultados – Pregunta 1
Fuente: [Elaboración Propia]

En la primera pregunta (Ver figura 31) se consulto acerca de la opinión del modelo propuesto y la mayoría de los encuestados consideran al modelo propuesto como muy bueno.

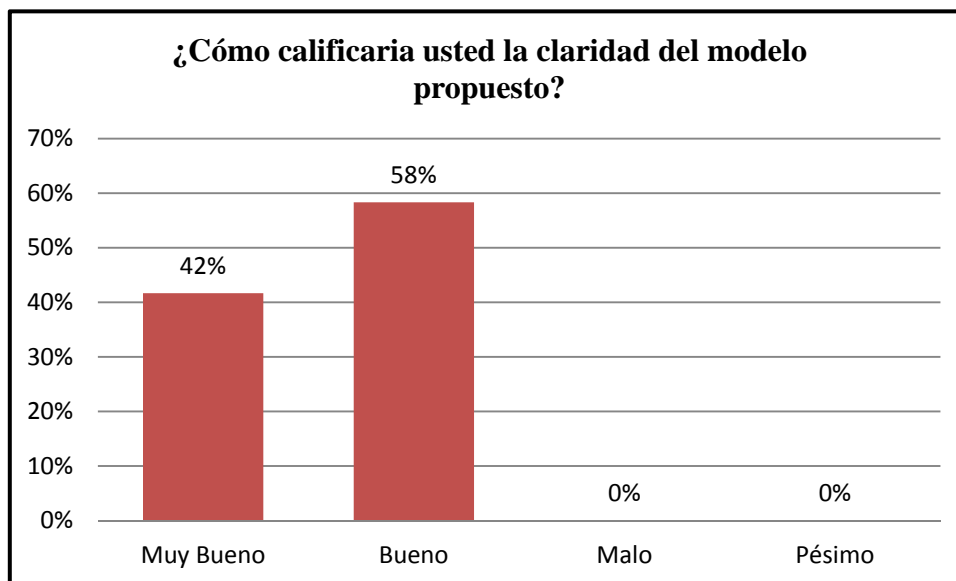


Figura 32. Interpretación de resultados – Pregunta 2
Fuente: [Elaboración Propia]

En la segunda pregunta (Ver figura 32) se consulto acerca de la claridad del modelo propuesto y la mayoría de los encuestados consideran que el modelo es claro, que se encuentra bien estructurado y es de fácil entendimiento.

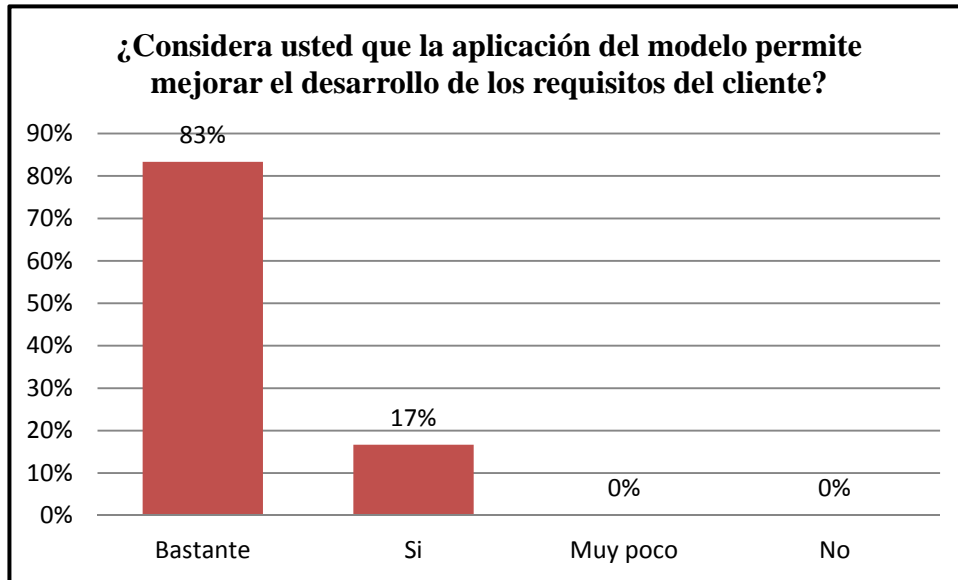


Figura 33. Interpretación de resultados – Pregunta 3
Fuente: [Elaboración Propia]

En la tercera pregunta (Ver figura 33) se consultó si considera que la aplicación del modelo permite mejorar el desarrollo de los requisitos del cliente y casi la mayoría de los encuestados coinciden en que el modelo es de bastante ayuda al momento de educir las necesidades, expectativas de las partes interesadas y la vez es de gran utilidad al momento de transformar estas necesidades en requisitos del cliente permitiendo de esta manera mejorar el proceso.

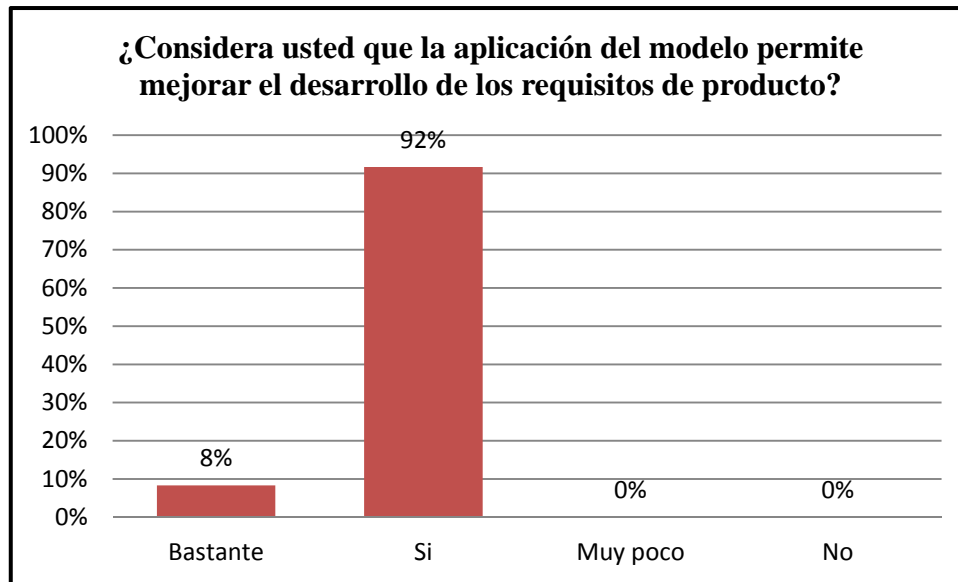


Figura 34. Interpretación de resultados – Pregunta 4
Fuente: [Elaboración Propia]

En la cuarta pregunta (Ver figura 34) se consultó si considera que la aplicación del modelo permite mejorar el desarrollo de los requisitos de producto y según la gran mayoría de los encuestados consideran que el modelo es de gran ayuda al momento de elaborar los requisitos en términos técnicos necesarios para el diseño del producto permitiendo mejorar este proceso.

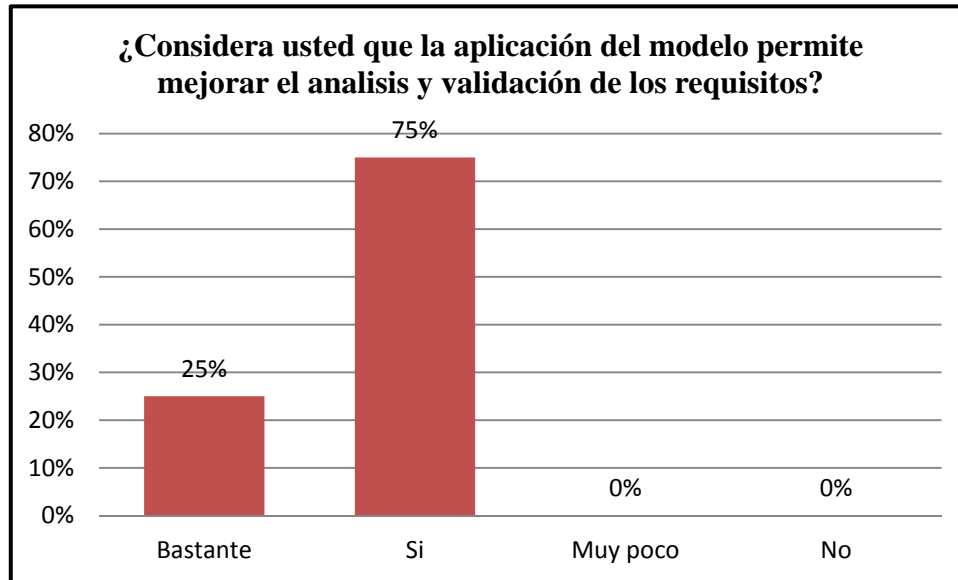


Figura 35. Interpretación de resultados – Pregunta 5
Fuente: [Elaboración Propia]

En la quinta pregunta (Ver figura 35) se consultó si considera que la aplicación del modelo permite mejorar el análisis y validación de los requisitos y según los encuestados, gran parte de ellos considera que el modelo propuesto aporta una serie de buenas prácticas que permiten mejorar el análisis y validación de requisitos.

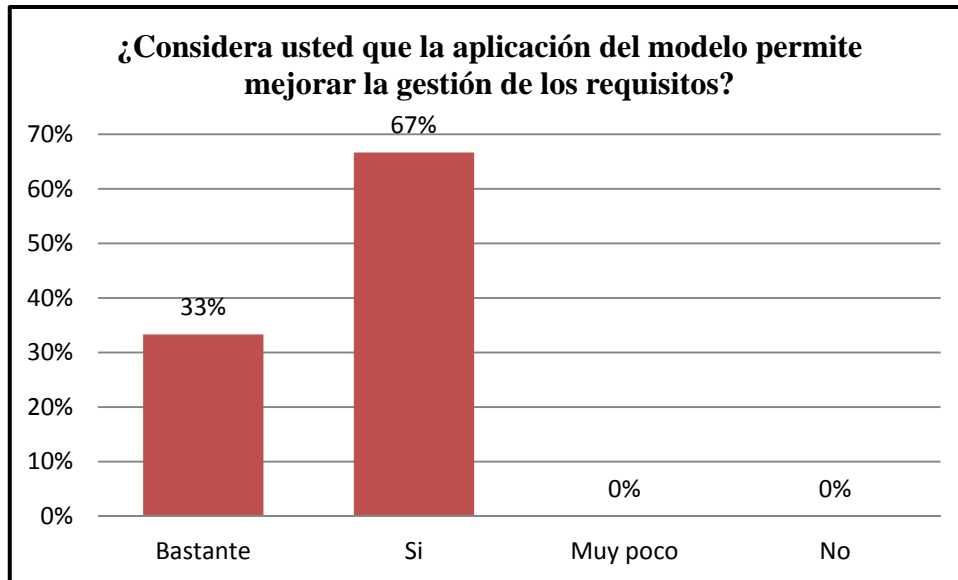


Figura 36. Interpretación de resultados – Pregunta 6
Fuente: [Elaboración Propia]

En la sexta pregunta (Ver figura 36) se consultó si considera que la aplicación del modelo permite mejorar la gestión de requisitos y según los encuestados considera que el modelo mejora el proceso de gestionar los cambios a los requisitos a medida que evolucionan durante el proyecto.

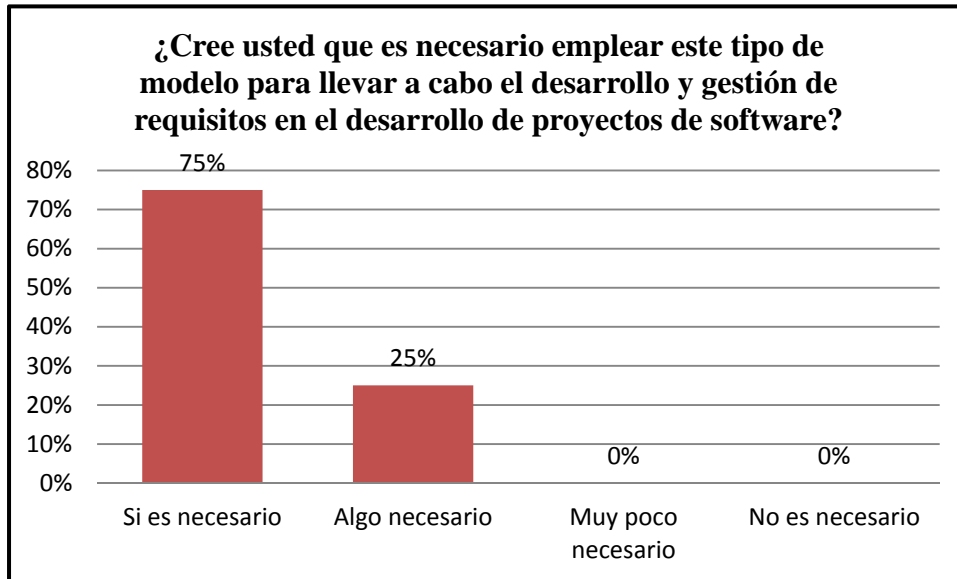


Figura 37. Interpretación de resultados – Pregunta 7
Fuente: [Elaboración Propia]

En la séptima pregunta (Ver figura 37) se consultó si considera necesario emplear este tipo de modelo para llevar a cabo el desarrollo y gestión de requisitos en el desarrollo de proyectos de software y gran parte de los encuestados considera que si es necesario emplear el modelo propuesto, mientras que una pequeña parte lo considera algo necesario.

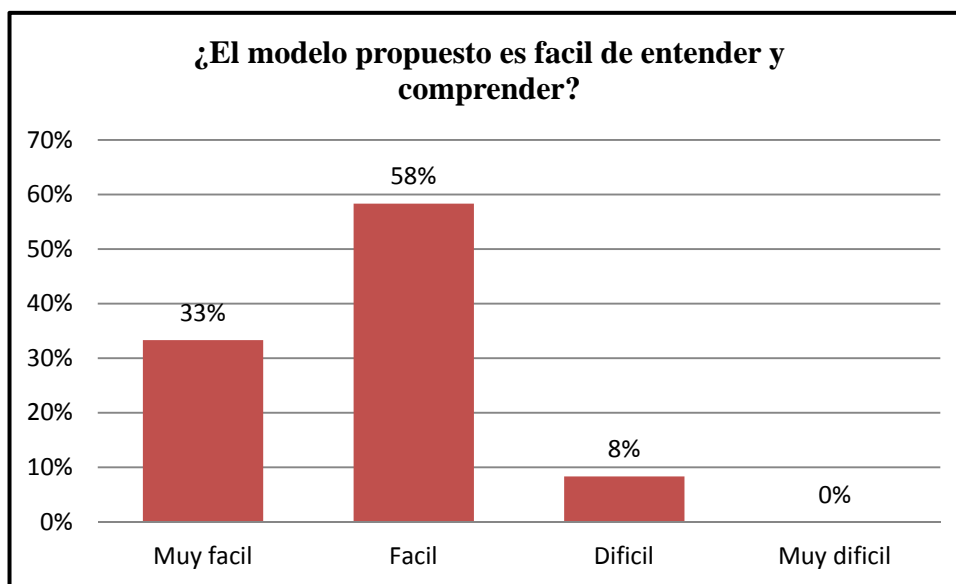


Figura 38. Interpretación de resultados – Pregunta 8

Fuente: [Elaboración Propia]

En la octava pregunta (Ver figura 38) se consultó si el modelo propuesto es fácil de entender y comprender y según los encuestados, consideran que el modelo está bien estructurado y es de fácil entendimiento, por otra parte un pequeño porcentaje lo considera difícil.

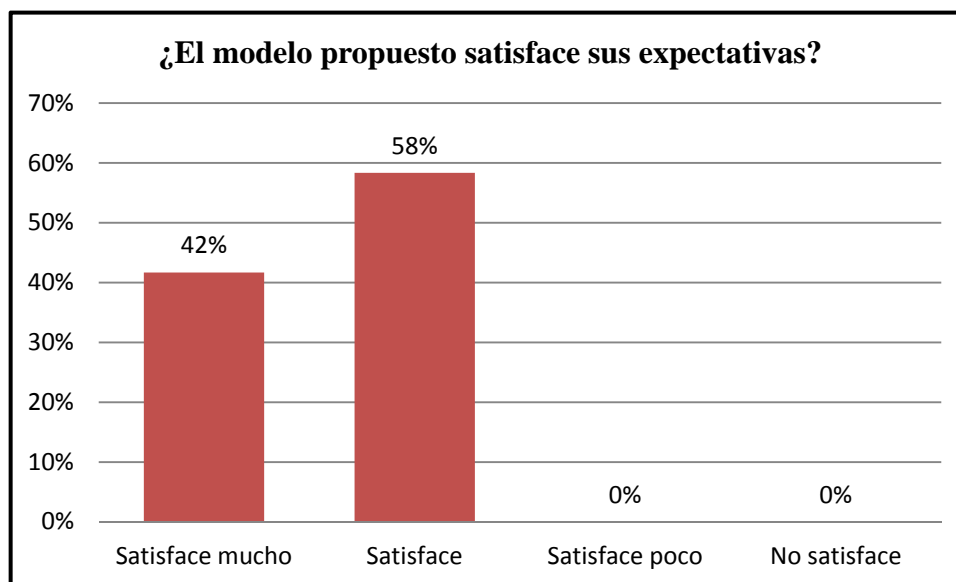


Figura 39. Interpretación de resultados – Pregunta 9
Fuente: [Elaboración Propia]

En la novena pregunta (Ver figura 39) se consultó si el modelo propuesto satisface sus expectativas y según la gran mayoría de los encuestados consideran que el modelo satisface y cumple con lo que ellos esperaban.

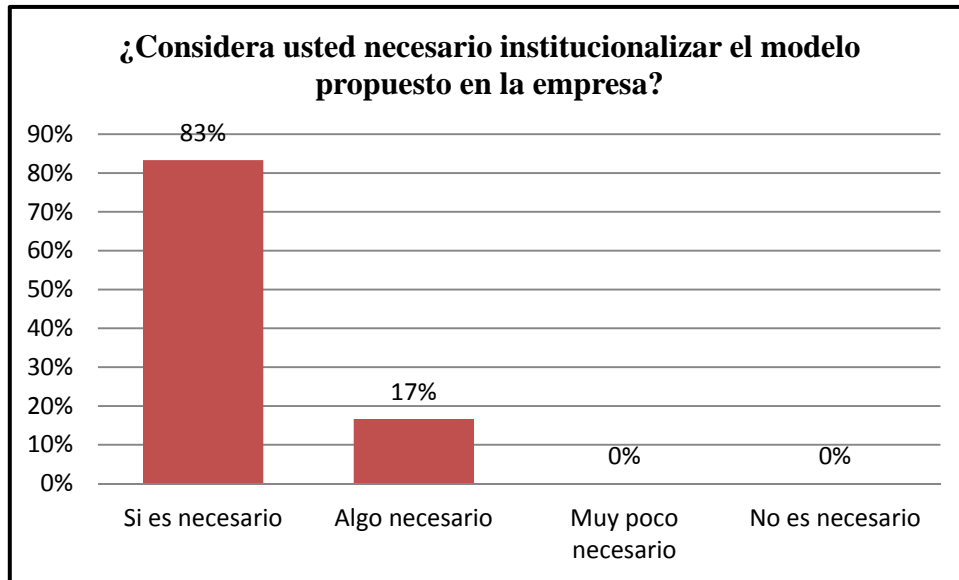


Figura 40. Interpretación de resultados – Pregunta 10
Fuente: [Elaboración Propia]

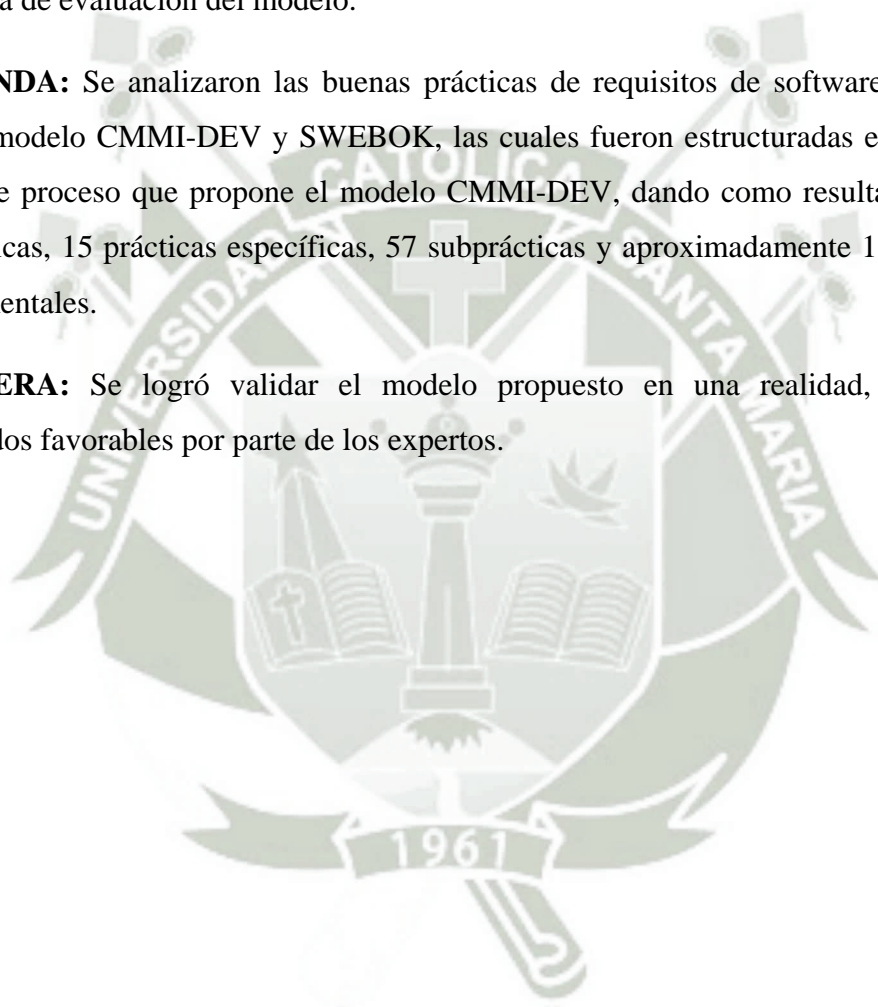
Para finalizar, en la última pregunta (Ver figura 40) se consultó si considera necesario institucionalizar el modelo propuesto en la empresa y casi todos los encuestados consideran que debe existir una política que indique que se debe de poner en práctica el modelo en toda la organización.

CONCLUSIONES

PRIMERA: A través del presente trabajo, se logró proponer un modelo de integración para desarrollar y gestionar requisitos en el desarrollo de proyectos de software basado en el estándar CMMI-DEV y SWEBOK que logró mejorar el proceso de requisitos en el desarrollo de proyectos de software, lo que se demuestra por los resultados de la encuesta de evaluación del modelo.

SEGUNDA: Se analizaron las buenas prácticas de requisitos de software propuestas por el modelo CMMI-DEV y SWEBOK, las cuales fueron estructuradas en base a las áreas de proceso que propone el modelo CMMI-DEV, dando como resultado 4 metas específicas, 15 prácticas específicas, 57 subprácticas y aproximadamente 130 prácticas fundamentales.

TERCERA: Se logró validar el modelo propuesto en una realidad, obteniendo resultados favorables por parte de los expertos.



SUGERENCIAS

Se sugiere como trabajo futuro elaborar una herramienta automatizada que pueda facilitar el uso del modelo propuesto.

Se sugiere a la organización que aplique el presente modelo, pueda aplicar el ciclo de mejora esperando en la siguiente ejecución resultados más precisos y efectivos.



BIBLIOGRAFÍA

- **Arias y Mejía 2006** Arias, J.; Mejía, Y. *Metodología para diagnosticar el estado de las Organizaciones de Software con relación al modelo CMMI*. Medellín: Universidad EAFIT, 2006.
- **Boehm 1975** Boehm, B. *Software design and structuring*. 1975.
- **Brooks 1987** Brooks, Fred. *No Silver Bullet: Essence and Accidents of Software Engineering*. 1987.
- **CMMI Institute 2014** Disponible en la Web:
<https://sas.cmmiinstitute.com/pars/pars.aspx>
- **DeMarco 1984** DeMarco, T. *An algorithm for sizing software products*. 1984.
- **DLYA 2012** Disponible en la Web:
<http://www.bantotal.com/compania/perfil.asp>
- **Doran 1981** George, Doran. *There's a S.M.A.R.T Way to Write Management's Goals and Objectives*. 1981.
- **García 2005** García, Carlos. *Qué es un modelo y su uso*. Puebla México: Instituto Tecnológico de Puebla, 2005.
- **Guillen 2011** Guillen, Jorge. *Propuesta de un modelo de asociación en el gerenciamiento de riesgos de proyectos informáticos basado en el estándar PMBOK Y CMMI-SW*. Arequipa: Universidad Católica de Santa María, 2011.
- **IEEE 1990** Instituto de Ingenieros Eléctricos y Electrónicos. *IEEE Std 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology*. 1990.
- **IEEE 2013** Instituto de Ingenieros Eléctricos y Electrónicos. *SWEBOK v3.0 Guide to the software engineering body of knowledge*. 2013.
- **Sommerville 2002** Sommerville, Ian. *Ingeniería de software*. 2002.
- **Morales 2013** Morales, Irma. *Implementando CMMI: ¿Por dónde empezar?* 2013. Disponible en la Web:
<http://www.teamsoft.com.pe/blog/implementando-cmmi-por-donde-empezar/>
- **Mutafelija y Stromberg 2005** Mutafelija, B; Stromberg, H. *Architecting Standard Processes with SWEBOK and CMMI*. 2005.

- **PMI 2001** Project Management Institute. *Practice Standard for Work Breakdown Structures – Second Edition*. 2001.
- **PMI 2013** Project Management Institute. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) Fifth Edition*. 2013.
- **RAE 2001** Real Academia Española. *Diccionario de la lengua española*. 2001.
- **Raninen 2014** Raninen, Anu. *Practical process improvement: how to initiate software process improvement in a small company*. 2014.
- **SEI 1996** Software Engineering Institute. *IDEAL: A User's Guide for Software Process Improvement*. 1996.
- **StandishGroup 2013** The Standish Group International. *Chaos Manifesto 2013: Think Big, Act Small*. 2013.
- **StandishGroup 1995** The Standish Group International. *The Chaos Report*. 1995.
- **SEI 2010** Software Engineering Institute. *CMMI para Desarrollo, Versión 1.3*. 2010.
- **Strauss y Corbin 1998** Strauss, A.; Corbin, J. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. 1998.
- **Souza, Sanches y Vaccare 2009** Souza, A.; Sanches, R.; Vaccare, R. *PRO-REQ: a facilitator guide to implement CMMI-Dev requirements engineering and management areas*. São Paulo: Universidad de São Paulo, 2009.
- **Torres y Arbeláez 2008** Torres, C.; Arbeláez, D. *Guía de implantación de CMMI en la empresa de software Colombiana*. Medellín: Universidad EAFIT, 2008.
- **Wieggers y Beatty 2013** Wieggers, K.; Beatty, J. *Software Requirements*. 2013.

ANEXO 1

Encuesta - Proceso de Requerimiento

La presente encuesta es anónima (No hay respuestas erradas o correctas) y tiene como único fin recopilar datos sobre el proceso de requerimientos. Datos que serán utilizados para una investigación. Para ello se solicita responder la siguiente encuesta marcando con una X la alternativa que mejor te parezca, seleccionando solo una alternativa por cada pregunta.

N°	Pregunta
1.	¿Sabe usted que es un requerimiento y cuál es su principal objetivo? Sí:..... No:.....
2.	En la definición de los requerimientos, ¿Conoce claramente la información que se requiere para obtener el resultado esperado del requerimiento? Siempre:..... Casi Siempre:..... Algunas Veces:..... Nunca:.....
3.	En la definición de los requerimientos, ¿Conoce claramente el resultado que se debe obtener al finalizar el requerimiento? Siempre:..... Casi Siempre:..... Algunas Veces:..... Nunca:.....
4.	¿Se tiene en cuenta las restricciones del sistema, de los procesos y aspectos legales para la definición de los requerimientos? Sí:..... No:.....
5.	En la definición de los requerimientos, ¿Conoce los procesos de la Empresa que serán afectados? Siempre:..... Casi Siempre:..... Algunas Veces:..... Nunca:.....
6.	En la definición de los requerimientos, ¿Se tiene en cuenta las políticas definidas por la Empresa para esta actividad? Siempre:..... Casi Siempre:..... Algunas Veces:..... Nunca:.....
7.	¿Conoce usted qué impacto tiene un requerimiento mal definido en el desarrollo de un proyecto? Bajo:..... Medio:..... Alto:.....
8.	Al momento de definir los requerimientos, ¿Se tiene en cuenta las estimaciones de tiempo, recursos e impactos que estos conllevan para la Empresa? Sí:..... No:.....
9.	En la definición de los requerimientos, ¿Se especifica claramente las validaciones que se deben de realizar para garantizar su objetivo? Sí:..... No:.....
10.	¿Se prioriza los requerimientos solicitados? Sí:..... No:.....
11.	¿Considera usted que los expertos en los procesos de toma de requerimientos deberían ser los encargados de definir los requerimientos? Sí:..... No:.....
12.	¿Cómo considera el nivel de equipamiento de la Empresa para la toma de requerimientos? (Herramientas, manuales, capacitación, etc.) En una escala del 1 al 5, dónde 5 es “Excelente” y 1 es “Insuficiente”. 1:..... 2:..... 3:..... 4:..... 5:.....
13.	¿Cree usted que la manera como se vienen realizando la toma de requerimientos en la Empresa debe mejorar? Sí:..... No:.....

ANEXO 2

Encuesta - Diagnóstico del proceso de requisitos

Desarrollo de Requisitos	
RD	Preguntas
Metas específicas (SG) y Practicas específicas (SP)	
SG 1	Desarrollar los requisitos de cliente
SP 1.1	Educir las necesidades
	1. ¿Se emplean técnicas para identificar los requisitos de software? (Ejemplos de técnicas: Grupos de trabajo, Tormenta de ideas, Cuestionarios o entrevistas.) Sí:..... No:.....
SP 1.2	Transformar las necesidades de las partes interesadas en requisitos de cliente
	2. ¿Se documentan los requisitos del cliente? Sí:..... No:.....
	3. ¿Se obtienen los requisitos funcionales del cliente? Sí:..... No:.....
	4. ¿Se obtienen los requisitos no funcionales del cliente? Sí:..... No:.....
	5. ¿Se definen restricciones para una posterior verificación y validación? Sí:..... No:.....
SG 2	Desarrollar los requisitos de producto
SP 2.1	Establecer los requisitos de producto y de componente de producto
	6. ¿Los requisitos del cliente se analizan y se expresan luego en términos técnicos? Sí:..... No:.....
	7. ¿Después de realizar un análisis técnico de los requisitos, se puede llegar a identificar nuevos requisitos derivados de las decisiones del diseño? Sí:..... No:.....
	8. ¿Se establecen relaciones entre los requisitos? Sí:..... No:.....
SP 2.2	Asignar los requisitos de componente de producto
	9. ¿Se asignan los requisitos a los componentes del producto? Sí:..... No:.....
	10. ¿Se asignan restricciones de diseño a los componentes del producto? Sí:..... No:.....
	11. ¿Se documenta las relaciones entre los requisitos asignados? Sí:..... No:.....
SP 2.3	Identificar los requisitos de interfaz
	12. ¿Se identifican las interfaces externas al producto? Sí:..... No:.....
	13. ¿Se identifican las interfaces internas al producto? Sí:..... No:.....
	14. ¿Se especifican los requisitos para las interfaces identificadas? Sí:..... No:.....
SG 3	Analizar y validar requisitos
SP 3.1	Establecer los conceptos y los escenarios de operación
	15. ¿Se desarrollan conceptos y escenarios de operación? Sí:..... No:.....
	16. ¿Se define el entorno en el que funcionará el producto? Sí:..... No:.....

	17. ¿Se revisan y refinan los conceptos y los escenarios de operación para descubrir nuevos requisitos? Sí:..... No:.....
SP 3.2	Establecer una definición de la funcionalidad y de los atributos de calidad requeridos
	18. ¿Se determina la misión clave y los factores de negocio? Sí:..... No:.....
	19. ¿Se identifica la funcionalidad y los atributos de calidad? (Ejemplo de medidas de los atributos de calidad: Respuesta dentro de 1 segundo, Disponibilidad del sistema en un 99% del tiempo.) Sí:..... No:.....
	20. ¿Se analiza y cuantifica la funcionalidad requerida por el usuario final? Sí:..... No:.....
	21. ¿Se dividen los requisitos en grupos, en base a criterios para facilitar el análisis de requisitos? (Ej. Funcionalidad similar, requisitos similares de atributos de calidad, acoplamiento) Sí:..... No:.....
	22. ¿Se asignan los requisitos de cliente al personal o elementos de soporte para dar apoyo a la síntesis de las soluciones? Sí:..... No:.....
SP 3.3	Analizar los requisitos
	23. ¿Se analizan los requisitos para determinar si satisfacen los objetivos de los requisitos de nivel más alto? Sí:..... No:.....
	24. ¿Se analizan que los requisitos sean completos, factibles, realizables y verificables? Sí:..... No:.....
	25. ¿Se identifican los requisitos clave que tienen una fuerte influencia en el coste, el calendario, el rendimiento o el riesgo? Sí:..... No:.....
	26. ¿Se identifican medidas técnicas de rendimiento? Sí:..... No:.....
SP 3.4	Analizar los requisitos para conseguir un equilibrio
	27. ¿Se utilizan modelos, simulaciones o prototipos probados para analizar el equilibrio entre las necesidades y las restricciones de las partes interesadas? Sí:..... No:.....
	28. ¿Se evalúa el impacto de los requisitos no funcionales en los costes y riesgos del desarrollo del producto? Sí:..... No:.....
SP 3.5	Validar los requisitos
	29. ¿Se utilizan técnicas para validar los requisitos? (Ejemplos de técnicas: simulaciones, prototipos, demostraciones.) Sí:..... No:.....
Gestión de Requisitos	
REQM	Preguntas
Metas específicas (SG) y Practicas específicas (SP)	
SG 1	Gestionar los requisitos
SP 1.1	Comprender los requisitos
	30. ¿Están establecidos criterios para distinguir a los proveedores apropiados de requisitos? Sí:..... No:.....
	31. ¿Están establecidos criterios objetivos para la evaluación y aceptación de los requisitos? (Ejemplos de criterios de evaluación y de aceptación: Claramente y correctamente establecidos, completos, apropiados para implementar o que se pueden probar.)

	Sí:..... No:.....
	32. ¿Se llega a una comprensión de los requisitos con los proveedores de requisitos? Sí:..... No:.....
SP 1.2	Obtener el compromiso sobre los requisitos
	33. ¿Se evalúa el impacto de los requisitos sobre los compromisos existentes? Sí:..... No:.....
	34. ¿Se negocia y registran los compromisos? Sí:..... No:.....
SP 1.3	Gestionar los cambios a los requisitos
	35. ¿Se documentan todos los requisitos y los cambios que se reciben? Sí:..... No:.....
	36. ¿Se mantiene una historia de cambios de los requisitos? Sí:..... No:.....
	37. ¿Se evalúa el impacto de los cambios de requisitos desde el punto de vista de las partes interesadas relevantes? Sí:..... No:.....
	38. ¿Se pone a disposición del proyecto los requisitos y los datos de los cambios? Sí:..... No:.....
SP 1.4	Mantener la trazabilidad bidireccional de los requisitos
	39. ¿Se maneja una matriz de trazabilidad de los requisitos? Sí:..... No:.....
SP 1.5	Asegurar el alineamiento entre el trabajo del proyecto y los requisitos
	40. ¿Se documentan las inconsistencias entre los requisitos y los planes del proyecto y los productos de trabajo? Sí:..... No:.....

ANEXO 3

Encuesta - Evaluación del Modelo

Se solicita respuestas la siguiente encuesta marcando con una X la alternativa que mejor te parezca, seleccionando solo una alternativa por cada pregunta.

Nº	Pregunta
1.	¿Qué opinión tiene usted del modelo propuesto? Muy Bueno () Bueno () Malo () Pésimo ()
2.	¿Cómo calificaría usted la claridad del modelo propuesto? Muy Bueno () Bueno () Malo () Pésimo ()
3.	¿Considera usted que la aplicación del modelo permite mejorar el desarrollo de los requisitos del cliente? Bastante () Si () Muy poco () No ()
4.	¿Considera usted que la aplicación del modelo permite mejorar el desarrollo de los requisitos de producto? Bastante () Si () Muy poco () No ()
5.	¿Considera usted que la aplicación del modelo permite mejorar el análisis y validación de los requisitos? Bastante () Si () Muy poco () No ()
6.	¿Considera usted que la aplicación del modelo permite mejorar la gestión de los requisitos? Bastante () Si () Muy poco () No ()
7.	¿Cree usted que es necesario emplear este tipo de modelo para llevar a cabo el desarrollo y gestión de requisitos en el desarrollo de proyectos de software? Si es necesario () Algo necesario () Muy poco necesario () No es necesario ()
8.	¿El modelo propuesto es fácil de entender y comprender? Muy fácil () Fácil () Difícil () Muy difícil ()
9.	¿El modelo propuesto satisface sus expectativas? Satisface mucho () Satisface () Satisface poco () No satisface ()
10.	¿Considera usted necesario institucionalizar el modelo propuesto en la empresa? Si es necesario () Algo necesario () Muy poco necesario () No es necesario ()