



**INSTRUMENTO PARA LA GENERACIÓN DEL PROCESO
DE DESARROLLO DE REQUERIMIENTOS DE SOFTWARE PARA MICRO Y
PEQUEÑAS EMPRESAS**

PROYECTO DE GRADO

**ALEXANDER VARELA GALVIS
GUILLERMO EDUARDO ARANGO STERLING**

**Asesor
HUGO ARBOLEDA, PhD
Director**

**FACULTAD DE INGENIERÍA
DEPARTAMENTO ACADÉMICO DE TECNOLOGÍAS DE INFORMACIÓN Y
COMUNICACIONES
MAESTRÍA EN GESTIÓN DE INFORMÁTICA Y TELECOMUNICACIONES
SANTIAGO DE CALI
2012**

**INSTRUMENTO PARA LA GENERACIÓN DEL PROCESO DE DESARROLLO
DE REQUERIMIENTOS DE SOFTWARE PARA MICRO Y PEQUEÑAS
EMPRESAS**

**ALEXANDER VARELA GALVIS
GUILLERMO EDUARDO ARANGO STERLING**

**Trabajo de grado para optar al título de
Magister en Gestión de Informática y Telecomunicaciones**

**Asesor
HUGO ARBOLEDA, PhD
Director**



**FACULTAD DE INGENIERÍA
DEPARTAMENTO ACADÉMICO DE TECNOLOGÍAS DE INFORMACIÓN Y
COMUNICACIONES
MAESTRÍA EN GESTIÓN DE INFORMÁTICA Y TELECOMUNICACIONES
SANTIAGO DE CALI
2012**

Nota de aceptación

Firma del Presidente del Jurado

Firma del Jurado

Firma del Jurado

Santiago de Cali,

CONTENIDO

	pág.
RESUMEN	9
1. INTRODUCCIÓN	10
1.1 <i>PLANTEAMIENTO DEL PROBLEMA</i>	13
1.2 <i>OBJETIVO GENERAL</i>	13
1.3 <i>OBJETIVOS ESPECÍFICOS</i>	13
1.4 <i>RESUMEN DEL MODELO DE TRABAJO PROPUESTO</i>	14
1.5 <i>RESUMEN DE RESULTADOS OBTENIDOS</i>	15
1.6 <i>ORGANIZACIÓN DEL DOCUMENTO</i>	16
2. MARCO TEÓRICO	17
2.1 <i>CMMI Dev 1.3</i>	17
2.2 <i>Manifiesto Ágil</i>	18
2.3 <i>Enfoque de Turkey France (16)</i>	19
2.4 <i>Ingeniería de línea de productos y de procesos</i>	20
2.4.1 <i>Línea de productos de software</i>	20
2.4.2 <i>Línea de procesos de software</i>	22
2.3 <i>Caracterización del contexto para mejora de procesos</i>	25
2.3.1 <i>Criterios de SEI para la adaptación de procesos.</i>	25
2.3.2 <i>Características organizacionales para el modelamiento de procesos de desarrollo de software</i>	26
3. PROPUESTA DE MODELO DE ADAPTACION DEL PROCESO DE DESARROLLO DE REQUERIMIENTOS	29
3.1 <i>Identificación de características organizacionales</i>	30
3.1.1 <i>Enfoque de Calidad.</i>	30
3.1.2 <i>Estructura Organizacional.</i>	31
3.1.3 <i>Aceptación de Cambios.</i>	32

3.1.4	Comunicación organizacional.	33
3.1.5	Medidor del progreso en los proyectos.	34
3.2	<i>Definición del proceso base de desarrollo de requerimientos de software.</i>	35
3.3	<i>Definición de estrategia para relacionar características versus actividades.</i>	37
3.4	<i>Implementación del Instrumento para la generación automática del proceso de DR.</i>	41
3.4.1	Diagrama UML del Modelo de proceso base DR con su respectiva variabilidad	41
3.4.2	Modelo del Contexto en Pure Variants	43
3.4.3	Mapeo de modelos con Feature Mapper	44
3.5	<i>Prueba del instrumento</i>	46
3.5.1	Configuración del Modelo de Contexto para una organización hipotética.	46
3.5.2	Proceso de desarrollo de requerimientos generado por el instrumento	47
3.5.3	Lectura de las directrices sugeridas	48
4.	VALIDACIÓN DE LA PROPUESTA	49
4.1	<i>Descripción de la organización</i>	49
4.2	<i>Configuración del Modelo de Contexto con base en las características de la organización</i>	51
4.3	<i>Proceso de desarrollo de requerimientos sugerido</i>	52
4.4	<i>Análisis de proceso delta</i>	54
4.5	<i>Resultados de la Encuesta</i>	55
4.6	<i>Mediciones</i>	57
5.	RESULTADOS OBTENIDOS	58
6.	CONCLUSIONES Y FUTURO TRABAJO	61
7.	BIBLIOGRAFÍA	62
8.	ANEXO 1 - Descripción de la función directriz Ágil.	66
9.	ANEXO 2 - Codificación de las directrices ágiles	71
10.	ANEXO 3. Encuesta usada en la validación de la propuesta.	72

LISTA DE TABLAS

	pág.
Tabla 1 Población de empresas dedicadas al Desarrollo de Software	10
Tabla 2 Factores de éxito considerados por los gerentes de proyecto (5)	12
Tabla 3 Principios ágiles (17)	19
Tabla 4 Características de proyectos ágiles (8)	27
Tabla 5 Factores del contexto (29)	27
Tabla 6 Características del profile Básico (9)	28
Tabla 7 Modelo del contexto con base en características organizacionales	43
Tabla 8 Formación profesional y experiencia del panel - Caso Univalle	51
Tabla 9 Comparación entre actividades actuales y actividades sugeridas	55
Tabla 10 Resultados de la encuesta	56
Tabla 11 Índices de cumplimiento para atributos de calidad - validación	57

LISTA DE FIGURAS

pág.

Figura 1 Cuadro Standish Group sobre el éxito de proyectos de TI (5).....	11
Figura 2 Propuesta de modelo de adaptación.....	14
Figura 3 Artefacto para mapeo de característica vs práctica específica.....	15
Figura 4 Enfoque Simmons y Bastarrica (25).....	23
Figura 5 Estrategia para adaptar procesos (26).....	23
Figura 6 Marco de referencia-Línea de Procesos (27).....	24
Figura 7 Propuesta de modelo de adaptación.....	29
Figura 8 Modelo de proceso base de desarrollo de requerimientos.....	36
Figura 9 Ejemplo del artefacto usado para describir la directriz Ágil para cada característica del contexto.....	39
Figura 10 Ejemplo de codificación de las directrices ágiles propuestas.....	40
Figura 11 Variabilidad del proceso de desarrollo de requerimientos.....	42
Figura 12 Modelo del contexto en Pure Variants.....	44
Figura 13 Modelo de mapeo con Feature Mapper.....	45
Figura 14 Configuración del Modelo de Contexto - Caso de prueba.....	46
Figura 15 Modelo de proceso delta resultante - Caso de prueba.....	47
Figura 16 Comparación proceso base vs proceso delta - Caso de prueba.....	48
Figura 17 Configuración del Modelo del Contexto Caso Univalle.....	52
Figura 18 Modelo de proceso delta resultante - Caso Univalle.....	53
Figura 19 Modelo de procesos base vs delta Caso Univalle.....	54
Figura 20 Modelo de procesos base vs delta.....	58
Figura 21 Artefacto para mapeo de característica vs práctica específica.....	59
Figura 22 Artefacto para codificación de directrices ágiles del mapeo de característica vs práctica específica.....	59

LISTA DE ANEXOS

	pág.
ANEXO 1 - Descripción de la función directriz Ágil para cada característica del contexto.....	66
ANEXO 2 - Codificación de las directrices ágiles propuestas	71
ANEXO 3-Encuesta de Validación de la propuesta	72

RESUMEN

De acuerdo a estudios de la Asociación Colombiana de Ingenieros de Sistemas, en la industria del software cerca del 70% de los proyectos no son exitosos, siendo uno de los problemas los procesos relacionados con la ingeniería de requerimientos, entre sus causas más frecuentes está la falta de comprensión de los requerimientos iniciales y la inhabilidad para gestionar los cambios durante el desarrollo del mismo.

Para mejorar los procesos de desarrollo de software existen iniciativas de procesos como CMMI, SPICE y TSP, ampliamente aceptadas, orientadas para organizaciones grandes y robustas que disponen de recurso tanto financiero como humano. Sin embargo, estas iniciativas no son fáciles de aplicar en micro y pequeñas empresas (MiPyMEs) ya que son ajenas a características propias del contexto organizacional, lo que implica mayor complejidad, mayor cantidad de prácticas, rigidez y costos.

Por lo expresado en el párrafo anterior, se han realizado diversas propuestas que tratan de adaptar el proceso de desarrollo de requerimientos de software al contexto organizacional, autores como Hurtado y Bastarrica proponen implementar la adaptación del proceso por medio de un modelo de transformación cuyas entradas son el modelo de proceso organizacional, incluyendo su variabilidad, y un modelo del contexto del proyecto. Otro autor, Washizaki, propone una Arquitectura de Línea de Procesos para derivar a partir de ella procesos para proyectos específicos, modelando las características del contexto del proyecto por medio de un feature model.

En este trabajo de grado se presenta un instrumento que permite, ante unas características organizacionales, sugerir las actividades de desarrollo de requerimientos para la organización con base en el área de proceso de Desarrollo de Requerimientos de CMMI Dev 1.3. El instrumento se enfoca en Desarrollo de Requerimientos ya que los problemas relacionados con Ingeniería de Requerimientos son causas comunes del fracaso en los proyectos de Desarrollo de Software.

El aporte principal del trabajo se centra en relacionar características organizacionales, enfocadas en principios ágiles, con las prácticas específicas del área de proceso de Desarrollo de Requerimientos de CMMI Dev 1.3, mediante criterios de afinación de procesos propuestos por el SEI. La relación se expresa a través de directrices que permiten modificar las actividades para hacerlas más ágiles de acuerdo a las características de la organización.

1. INTRODUCCIÓN

Según Fedesoft (1) la industria del Software en Colombia creció en el año 2009 el 7.7% frente al 8.9% de Latinoamérica, generando 16 mil empleos directos en el país y exportaciones a 13 países, y se proyecta para el año 2013 un crecimiento de un 11% de organizaciones que desarrollan software (1). Los anteriores datos muestran la importancia del sector de la Industria del Software en Colombia, sector que mantiene su crecimiento y aporta a la economía del país.

De acuerdo a (1), en Colombia habían 645 organizaciones que desarrollaban software en el 2007, mostrando una proyección creciente hasta llegar a 740 en el 2012 y 750 proyectadas para el 2013, de las cuales el 89% está conformado por micro y pequeñas empresas (55% y 34% respectivamente) y según Merchán & Urrea en su estudio realizado en el 2007 (2), en el sur-occidente colombiano habían 306 organizaciones dedicadas al Desarrollo de Software con la distribución que se observa en la Tabla 1, siguiendo la proyección de (1) se estima que para el 2012 en el sur-occidente colombiano se cuenta aproximadamente con 351 organizaciones dedicadas al Desarrollo de Software (47% del total de organizaciones proyectadas para la nación).

Referente	Número total de empresas
Parquesoft Cali	183
Parquesoft Palmira	8
Parquesoft Tulúa	19
Parquesoft Popayán	34
Parquesoft Pereira	21
Parquesoft Manizales	27
Parquesoft Pasto	14
Total	306

Tabla 1 Población de empresas dedicadas al Desarrollo de Software Sur-occidente colombiano (1).

En (2) muestran que el 61,22% de las empresas encuestadas no siguen ningún modelo de calidad de software, si se tienen en cuenta que el sur-occidente Colombia es el cluster de desarrollo de software más grande y maduro del país (2), se puede estimar que la situación puede ser similar a nivel nacional.

Es también relevante mostrar que de acuerdo a estudio realizado en la industria del software (3) (4) en el 47% de los proyectos se consigue cumplir con la triple restricción de costo, tiempo y alcance. Como se puede observar en la Figura 1, Standish Group reporta que solamente el 30% de los proyectos de TI alcanzaron el éxito para el año 2009 (5).

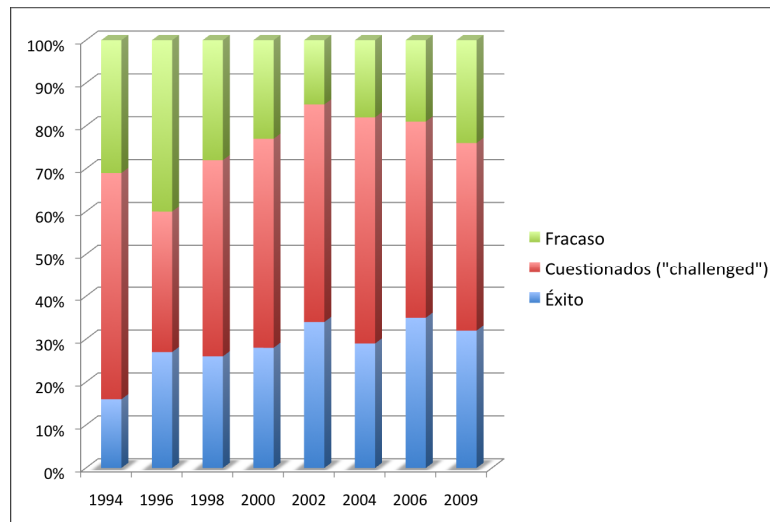


Figura 1 Cuadro Standish Group sobre el éxito de proyectos de TI (5)

Algunas de las causas por las cuales los proyectos no consiguen el éxito, no terminan satisfactoriamente o se desfasan en el cronograma, se presentan en los procesos relacionados con la ingeniería de requerimientos y las causas más frecuentes son la falta de comprensión de los requerimientos iniciales y la inhabilidad para gestionar los cambios durante el desarrollo (6).

La Tabla 2 muestra que la claridad en la definición de requerimientos es un factor importante para el éxito de los proyectos de software.

Factores de Éxito en Proyectos de Software	% de Respuestas
1. Involucramiento del usuario	15.9%
2. Soporte de administración ejecutiva	13.9%
3. Clara definición de requerimientos	13.0%
4. Planeación adecuada	9.6%
5. Expectativas realistas	8.2%
6. Pequeñas entregas de proyecto	7.7%
7. Personal competente	7.2%
8. Apropiación del proyecto	5.3%
9. Visión y objetivos claras	2.9%
10. Trabajo duro	2.4%
Otro	13.9%

Tabla 2 Factores de éxito considerados por los gerentes de proyecto (5)

Con el propósito de mejorar los procesos de desarrollo de software existen propuestas muy puntuales y para contextos bien definidos, **p.ej.** (7) (8) (2) (9) (10). Es posible también adaptar y usar iniciativas de procesos como CMMI, SPICE y TSP. Esto implicaría mayor complejidad, mayor cantidad de prácticas, rigidez y costos (7) (8) siendo estas iniciativas ajenas a las características del contexto y nada fáciles de aplicar en organizaciones pequeñas (2) (9).

De acuerdo al Ministerio de comercio, industria y turismo de Colombia, las microempresas tienen menos de 10 empleados y las pequeñas empresas menos de 50 empleados, por lo tanto y con base en los estudios de (1) (2) (7) (8), se puede afirmar que el 89% de las organizaciones dedicadas al desarrollo de software en Colombia cuenta con equipos de desarrollo pequeños y gran porcentaje de ellas no aplican ningún modelo de calidad en el proceso de desarrollo de software porque sus características organizacionales no les permiten afrontar los costos y la complejidad que implica la adopción de algún marco de referencia internacional.

Una tendencia en el desarrollo de software es la construcción de Líneas de Producto de Software (SPL por sus siglas en inglés). Una SPL se define como un conjunto de sistemas de software, que comparten un conjunto común de características (features), las cuales satisfacen las necesidades específicas de un

dominio o segmento particular de mercado, y que se desarrollan a partir de un sistema común de activos base (core assets) de una manera preestablecida.

Algunos autores hablan de líneas de procesos de software (Software Process Line) (11), lo cual se refiere a generar un conjunto de procesos partiendo de unas actividades base que se configuran de acuerdo con unas características particulares del contexto organizacional. El desarrollo de SPL requiere la definición y la identificación de las características variables y de artefactos reutilizables, de tal manera, que dada una selección de características se genere un producto/proceso adecuado para ellas. Bajo este enfoque de SPL, se pretende dar solución de mejora del proceso de desarrollo de requerimientos de software para contextos organizacionales heterogéneos.

1.1 PLANTEAMIENTO DEL PROBLEMA

Para realizar procesos de mejoramiento de Desarrollo de Software existen diferentes iniciativas aceptadas internacionalmente por ejemplo: CMMI, SPICE y TSP, sin embargo, la implementación de estas iniciativas en micro y pequeñas empresas implica mayor complejidad, mayor cantidad de prácticas, rigidez y costos (7) (8), y son ajenas a condiciones y/o características propias de la organización (2) (9). No existe un instrumento que sugiera un proceso de Desarrollo de Software con base en características de la organización y en buenas prácticas, sobre el cual la organización pueda iniciar su proceso de mejoramiento.

1.2 OBJETIVO GENERAL

Crear un instrumento para la generación del proceso de desarrollo de requerimientos, del tipo Software Process Line, mediante la integración de CMMI-Dev 1.3 y principios Ágiles, enfocado en micro y pequeñas empresas.

1.3 OBJETIVOS ESPECÍFICOS

1. Identificar características organizacionales relacionadas con principios ágiles y que tienen incidencia en el proceso de Desarrollo de Requerimientos.
2. Crear un proceso base de desarrollo de requerimientos partiendo de las especificaciones del área de proceso de Desarrollo de Requerimientos de CMMI Dev 1.3.

3. Definir una estrategia de mapeo entre características organizacionales y actividades del proceso base de Desarrollo de Requerimientos.
4. Construir un instrumento que permita, de manera asistida y automática, la generación del proceso de Desarrollo de Requerimientos con base en la selección de características organizacionales.
5. Validar la propuesta.

1.4 RESUMEN DEL MODELO DE TRABAJO PROPUESTO

La Figura 2 muestra el modelo propuesto y se basa en cinco procesos: dentro de la Ingeniería del Dominio se *identifican las características organizacional* es que se relacionan con los principios ágiles, se define *el modelo de proceso base de desarrollo de requerimientos* a partir de las prácticas específicas CMMI Dev 1.3, y se *definen directrices* que relacionan características organizacionales con actividades del proceso base, bajo criterios del SEI. Dentro de la Ingeniería de la aplicación se observan dos procesos, el primero es la *configuración manual del contexto*, que consiste en seleccionar las características organizacionales que identifican a una organización, y el segundo proceso es la *instanciación del modelo de procesos base* para generar un modelo de proceso de desarrollo de requerimientos *delta* ajustado ya a la organización.

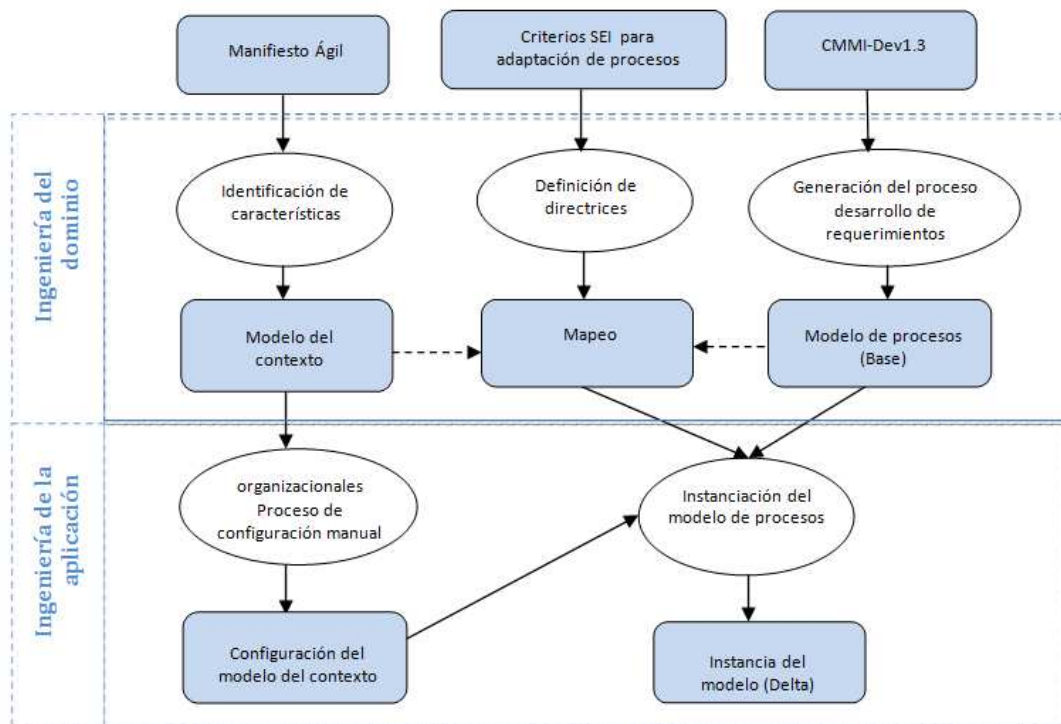


Figura 2 Propuesta de modelo de adaptación

1.5 RESUMEN DE RESULTADOS OBTENIDOS

1. El resultado más importante de la presente propuesta es la definición de un instrumento que integra Ágil y CMMI, y que permite a cualquier tipo de organización la posibilidad de adoptar un modelo semi robusto, principalmente para las micro y pequeñas empresas, las cuales presentan dificultades para implementar modelos estándares.
2. Artefacto que relaciona características basadas en Ágil vs prácticas específicas del Área de Desarrollo de Requerimientos. Incluye dos presentaciones, una donde se describen las directrices ágiles de manera explícita y otra donde solo se presenta el código asignado a cada directriz ágil, para efectos de legibilidad. La Figura 3 muestra parte del artefacto generado en su primera presentación y el cual se describe con mayor detalle en el apartado 3.3.

CARACTERÍSTICA	VALOR	SP1.1 EDUCIR LAS NECESIDADES DEL CLIENTE	SP1.2 TRANSFORMAR LAS NECESIDADES DE LAS PARTES INTERESADAS EN REQUISITOS DE CLIENTE
Medidor del progreso (MP)	Software funcionando (SF)		Formalidad: Priorizar grupos de requisitos por funcionalidad
	Documentación e informes parciales (DE)		Granularidad: Documento que describa: 1.Requisitos del cliente , 2. Restricciones de cliente para llevar a cabo la verificación. 3. Restricciones de cliente para llevar a cabo la validación
	Indicadores o métricas de avance del plan (IM)		Formalidad: Traducir la necesidades en requisitos documentados

Figura 3 Artefacto para mapeo de característica vs práctica específica

3. Artefacto que presenta un proceso de desarrollo de requerimientos con base en prácticas específicas del CMMI Dev 1.3. El artefacto se describe en el apartado 3.2.
4. Productividad académica, de tipo Artículo, donde se describe el presente trabajo.

1.6 ORGANIZACIÓN DEL DOCUMENTO

Las siguientes secciones del documento están organizadas de la siguiente manera:

2. Marco Teórico.

Esta sección describe todos los conceptos o áreas de conocimiento que se usan en el presente trabajo como base para generar el instrumento: CMMI Dev 1.3, procesos Ágiles, Ingeniería de línea de productos y procesos y caracterización del contexto para mejora de procesos.

También incluye la descripción de trabajos realizados orientados a resolver similares problemas.

3. Propuesta del modelo de adaptación del proceso de desarrollo de requerimientos.

En esta sección se describe en detalle cada uno de los elementos que intervienen en el modelo de adaptación propuesto: el modelo del contexto con base en características organizacionales identificadas, el modelo de procesos base de desarrollo de requerimientos, la estrategia para relacionar características con actividades del proceso, y las herramientas de automatización de eclipse utilizadas.

4. Adaptación de proceso de desarrollo de requerimientos.

En esta sección se describe el uso del instrumento a través de un caso de prueba.

5. Validación de la propuesta

Esta sección describe el proceso que se realizó para la validación del instrumento, el cual fue realizado en el Área de Desarrollo de software de la Universidad del Valle.

6. Resultados obtenidos

Esta sección describe los resultados que se obtuvieron del presente trabajo.

7. Conclusiones y futuros trabajos

Lista las conclusiones y los posibles trabajos que se pueden realizar sobre el enfoque planteado.

2. MARCO TEÓRICO

2.1 CMMI Dev 1.3

“El Modelo de Capacidad y Madurez (Capability Maturity Model Integration, CMMI), es un modelo de madurez de mejora de los procesos para el desarrollo de productos y de servicios. Consiste en las mejores prácticas que tratan las actividades de desarrollo y de mantenimiento que cubren el ciclo de vida del producto, desde la concepción a la entrega y el mantenimiento” (12).

Se inspira en el mejoramiento continuo y se basa en una escala de cinco niveles para la representación por etapas y seis niveles en la representación por capacidad, niveles que permiten clasificar a las organizaciones de desarrollo de software de acuerdo al dominio que evidencie para cada área de proceso involucradas en el nivel (13) y permiten trazar el camino evolutivo que debe recorrer una organización que desea mejorar sus procesos para desarrollar y mantener sus servicios.

CMMI Dev 1.3 está conformado por 22 áreas de proceso agrupadas dentro de 4 categorías: Gestión de Procesos, Ingeniería, Soporte y Gestión de Proyectos. (14). Cada área de proceso contiene metas específicas y metas genéricas cuya satisfacción se logra a través de actividades esperadas llamadas prácticas específicas y prácticas genéricas, éstas prácticas están soportadas por componentes informativos dentro de los cuales se encuentran las subprácticas específicas.

El área de proceso Desarrollo de Requerimientos (DR) pertenece a la categoría Ingeniería y está compuesta por tres metas específicas, la siguiente lista muestra las metas específicas del área DR y sus respectivas prácticas específicas:

1. Desarrollar los requisitos del cliente:
 - a. Educar las necesidades.
 - b. Transformar las necesidades de las partes interesadas en requisitos de cliente.
2. Desarrollar los requisitos del producto:
 - a. Establecer los requisitos de producto y de componente de producto.
 - b. Asignar los requisitos de componente de producto.
 - c. Identificar los requisitos de interfaz.
3. Analizar y validar los requisitos:
 - a. Establecer los conceptos y los escenarios de operación.
 - b. Establecer una definición de la funcionalidad y de los atributos de calidad requeridos.
 - c. Analizar los requisitos.
 - d. Analizar los requisitos para conseguir un equilibrio.
 - e. Validar los requisitos.

2.2 Manifiesto Ágil

El Manifiesto Ágil se centra en cuatro valores que son la base de doce principios. Valores y principios que, de acuerdo a sus autores, deberían facilitar el desarrollo de software de manera rápida y dando respuesta a los posibles cambios que se presentan durante un proyecto, ofreciendo una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas (15) (16).

Los métodos basados en Ágil presentan recetas (howto's) que funcionan bien en pequeños proyectos, mientras que CMMI proporciona las prácticas de Ingeniería de Sistemas que a menudo se requieren en proyectos de alto riesgo, proporciona también, la gestión de procesos y prácticas de apoyo (y principios) que ayudan a implementar y mejorar continuamente el despliegue de los métodos Ágiles en una organización independientemente de la organización o el tamaño del proyecto, en resumen, CMMI se centra en el **qué** hacer dentro de los proyectos y Ágil se enfoca en el **cómo**, por lo tanto, CMMI y Ágil pueden coexistir (12) (12).

La Tabla 3 muestra los principios ágiles, y la siguiente lista muestra los valores del Manifiesto Ágil:

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación extensiva.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta ante el cambio sobre seguir un plan.

Principios Ágiles
1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

- 9. La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- 10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- 11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- 12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia

Tabla 3 Principios ágiles (17)

2.3 Enfoque de Turk y France (16)

Plantea que los 12 principios del Manifiesto se relacionan con 7 grupos de supuestos que de manera subyacente se deberán abordar en cualquier proceso ágil, sea cual fuere la metodología seguida (XP, SCRUM, metodologías ágiles propias):

Supuestos subyacentes en los principios 1,3 y 7 del Manifiesto Ágil

- Supuesto de visibilidad: Para los clientes, es más fácil tener una idea de si el proyecto avanza en la dirección necesaria si realmente se puede ver la interfaz de usuario y ver realmente el software hacer las cosas que necesitan para hacer, en lugar de limitarse a confiar en los informes, especificaciones, y otras medidas
- Supuesto de Iteración: Un proyecto siempre se puede estructurar en iteraciones cortas de tiempo fijo.

Supuestos subyacentes en los principios 5, 11 y 12 del Manifiesto Ágil

- Supuesto de autoevaluación: Los equipos son capaces y están dispuestos a autoevaluarse.
- Supuesto de auto organización: Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- Supuesto de experiencia del equipo: Los desarrolladores tienen la experiencia necesaria para definir y adaptar sus procesos de manera adecuada

Supuestos subyacentes en los principios 8 y 9 del Manifiesto Ágil

- Supuesto de aseguramiento de la calidad: La evaluación de los artefactos de software (productos y procesos) se pueden restringir las frecuentes entrevistas informales opiniones y pruebas de código.
- Supuesto de rediseño: Los sistemas pueden ser continuamente rediseñados (refactorizados) y todavía mantener su integridad estructural y conceptual.

Supuestos subyacentes en el principio 6 del Manifiesto Ágil

- Supuesto de comunicación de equipo: los desarrolladores son localizados a la hora y en lugar, de tal manera que sean capaces de tener una comunicación frecuente, intensiva con cada otro.
- Supuesto de iteración con el cliente: Los equipos cliente están disponibles para la interacción frecuente cuando los necesitan los desarrolladores.

Supuestos subyacentes en el principio 4 del Manifiesto Ágil

- Supuesto de Face-to-Face: interacción cara a cara es el método más productivo de la comunicación con los clientes y entre los desarrolladores.
- Supuesto de Documentación: El desarrollo de modelos de software y documentación extensa (o relativamente completa) y consistente, es contraproducente.

Supuestos subyacentes en el principio 2 del Manifiesto Ágil

- Supuesto de requisitos cambiantes: los requisitos siempre evolucionan, debido a los cambios de la tecnología, las necesidades de los clientes, dominios de negocios o incluso la adquisición de nuevos clientes.
- Supuesto sobre el costo del cambio: el costo de cambio no aumenta dramáticamente con el tiempo.

Supuestos subyacentes en el principio 10 del Manifiesto Ágil

- Supuesto de desarrollo de aplicaciones específicas: la reutilización y generalidad no deben ser objetivos del desarrollo de software de aplicación específica

2.4 Ingeniería de línea de productos y de procesos

2.4.1 Línea de productos de software

La Ingeniería de Línea de Producto de Software (PLSE por sus siglas en inglés) es un paradigma de Ingeniería de Software emergente, que guía a la organización hacia el desarrollo de productos a partir de componentes base, en vez de desarrollar producto a producto desde cero (18), por lo tanto, se centra en la generación de Líneas de Productos de Software (SPL por sus siglas en inglés). Las PLSE han demostrado que las organizaciones pueden desarrollar nuevos productos con menos esfuerzo, costo y mayor calidad en comparación con el desarrollo de sistemas individuales (19).

Las SPL se pueden ver como un conjunto de productos de software que pertenecen a un dominio o negocio particular y por lo tanto comparten elementos comunes (20), los productos se pueden distinguir por sus elementos variables, representados por características que el cliente determina como importantes y que pueden generar la diferencia ante los demás productos.

Para desarrollar componentes base reusables, PLSE debe tener la habilidad tanto para explotar los aspectos comunes como para administrar la variabilidad, y esto se obtiene a partir del entendimiento del dominio, que proporciona una amplia perspectiva de ingeniería para la reusabilidad y la adaptabilidad en determinada línea de producto. Por lo tanto, el análisis del dominio, el cual identifica los aspectos comunes y la variabilidad a partir de una perspectiva del dominio, es un

requisito clave para el desarrollo de componentes base para línea de productos (18).

Los conceptos centrales para definir y documentar la variabilidad en una SPL son los puntos de variación y las variantes. Un punto de variación indica y especifica qué puede variar, una variante define la variación concreta (19).

En (21) se ilustran los dos procesos claves de PLSE dentro de un marco de referencia: Ingeniería de Dominio e Ingeniería de Aplicación:

- En la Ingeniería de Dominio los puntos en común y la variabilidad de la SPL se definen y se establecen los artefactos reusables (19) (21).
- Los procesos de la Ingeniería de Aplicación se encargan de generar aplicaciones de la línea de productos a partir de los artefactos reusables. Aprovecha la variabilidad de los artefactos reusables mediante la unión de la variabilidad de acuerdo a las necesidades específicas de la aplicación (19) (21).

Modelos de Variabilidad

Modelado de Características (Feature Modeling (FM)) es una de las técnicas de análisis de dominio más populares que analiza los aspectos comunes y la variabilidad en un dominio para desarrollar componentes base reusables en una línea de producto (18) (22). Es además una notación que contiene características opcionales, obligatorias, grupos de características y relaciones de inclusión y exclusión (23).

Según **Lee** (18), las características del producto pueden ser identificadas y clasificadas en términos de capacidades, tecnologías del dominio, técnicas de implementación y entornos operativos. Las características pueden ser obligatorias, opcionales o alternativas, y los diagramas de características que capturan la relación conceptual entre ellas pueden tener diferentes tipos de relaciones: de composición, de generalización/especialización y de implementación. De igual manera, las reglas de composición complementan el FM y son las que se utilizan para restringir las características opcionales o alternativas.

Czarnecki (22) presenta una relación o mapeo entre un FM y un diagrama de actividades, importante para brindarle significado semántico al FM. El diagrama de actividades contiene anotaciones de tipo PC (Condiciones de Presencia) y ME (Meta Expresiones) que están relacionados explícitamente con características del FM, buscan hacer algún tipo de acción en el diagrama de actividades. Por lo tanto, de acuerdo a la configuración de las características (instanciación), el

diagrama de actividades se modifica en el marco de una sintaxis propia que también propone el autor.

En la búsqueda de establecer relaciones entre FM y otros modelos, **Zschaler** (24) realiza un estudio importante para gestionar la variabilidad en una línea de producto, que consiste en presentar una familia de lenguajes que permite expresar explícitamente la relación de un Modelo de Características con cualquier otro artefacto de la línea de producto, como el Modelo de Requerimientos o Modelo de Arquitectura, con el propósito de mantener la consistencia de requerimientos en las diferentes fases del proceso de desarrollo, debido a la variabilidad involucrada en la fase inicial de la línea de producto.

2.4.2 Línea de procesos de software

Propuestas de trabajos en Línea de procesos de software

Según **Simmons y Bastarrica** (25), “línea de procesos de software son familias de procesos relacionados, construidos de una manera preestablecida a partir de un conjunto de elementos de procesos de software que son administrables”.

Simmonds y Bastarrica exploran la aplicabilidad del Modelado de Características para especificar la variabilidad de línea de procesos de software. El trabajo comprende utilizar una versión modificada de *The Software and Systems Process Engineering Meta-Model (SPEM)* (metamodelo que provee varios mecanismos de variabilidad para la configuración de procesos) con el propósito de separar la definición del proceso de la variabilidad del proceso, similar a como el Modelado de Características se utiliza para modelar la variabilidad del producto separadamente del resto de los modelos del producto. El estudio termina planteando la utilización de Modelado de Características para modelar la variabilidad de procesos **vSPEM**. La Figura 4 muestra el instrumento planteado.

Aunque el estudio no propone como modelar el contexto del proyecto, sí utiliza el Modelado de Características para brindar expresividad al modelamiento de la variabilidad de procesos y posibilitar el análisis y manejo de la consistencia entre configuraciones del Modelado de Características y su respectivo proceso base, aspecto importante para tener en cuenta en el momento de mapear dos modelos de dominios diferentes.

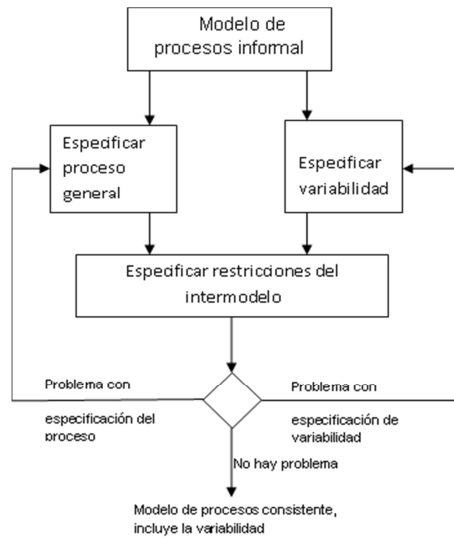


Figura 4 Enfoque Simmons y Bastarrica (25)

Hurtado y Bastarrica proponen un enfoque orientado al modelo para adaptar procesos de software y generar procesos específicos para un proyecto determinado, a partir del contexto del proyecto y del proceso organizacional (26). Los autores proponen un enfoque para adaptar procesos organizacionales al contexto del proyecto en particular, con base en técnicas de MDE (Model Driven Engineering). La adaptación es implementada por medio de un modelo de transformación cuyas entradas son el modelo de proceso organizacional, incluyendo la variabilidad, y un modelo del contexto del proyecto. La salida es el proceso adaptado al contexto. La Figura 5 muestra la estrategia para adaptar procesos.

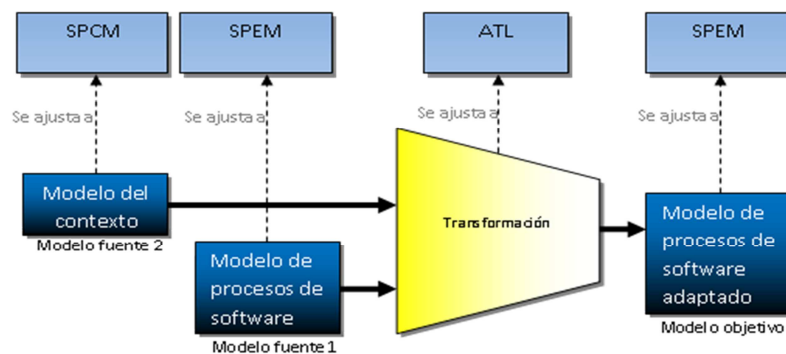


Figura 5 Estrategia para adaptar procesos (26)

El estudio propone modelar las características del contexto del proyecto, que bajo ciertas configuraciones del modelo, permiten obtener como resultado el modelo de proceso organizacional ajustado al contexto. Aunque la estrategia planteada es efectiva para lograr la adaptación de procesos, carece de usabilidad en el sentido de que cuando aparece una nueva configuración en el modelo del contexto, algo inevitable, se debe actualizar las reglas de la transformación. De igual manera, el enfoque no describe cómo memorizar las configuraciones pasadas, con el propósito de seguir mejorando el modelo del contexto y las decisiones de adaptación.

Washizaki (27) propone una Arquitectura de Línea de Procesos (Process Line Architecture-PLA) para derivar a partir de ella procesos para proyectos específicos: “PLA es una estructura de procesos que refleja los aspectos comunes y la variabilidad de un conjunto de procesos que componen una línea de proceso desde una perspectiva de optimización y consistencia global” (27). De acuerdo al estudio, PLA busca solucionar los problemas que se presentan en los dos tipos de enfoques establecidos hasta el momento para la adaptación de procesos: el enfoque basado en componentes, en el cual el proceso específico para el proyecto es construido a partir de partes de procesos ya existentes, y el enfoque generador, en el cual el proceso específico para el proyecto es determinado por un proceso referente típico o preestablecido. El problema, en el enfoque basado en componentes, es que las partes carecen de compatibilidad y consistencia global; y el problema en el enfoque generadores que no permite reusar fragmentos del proceso. A través de la semántica SPEM, las características del proyecto se asocian elementos del proceso base. La Figura 6 muestra el marco de referencia propuesto por los autores.

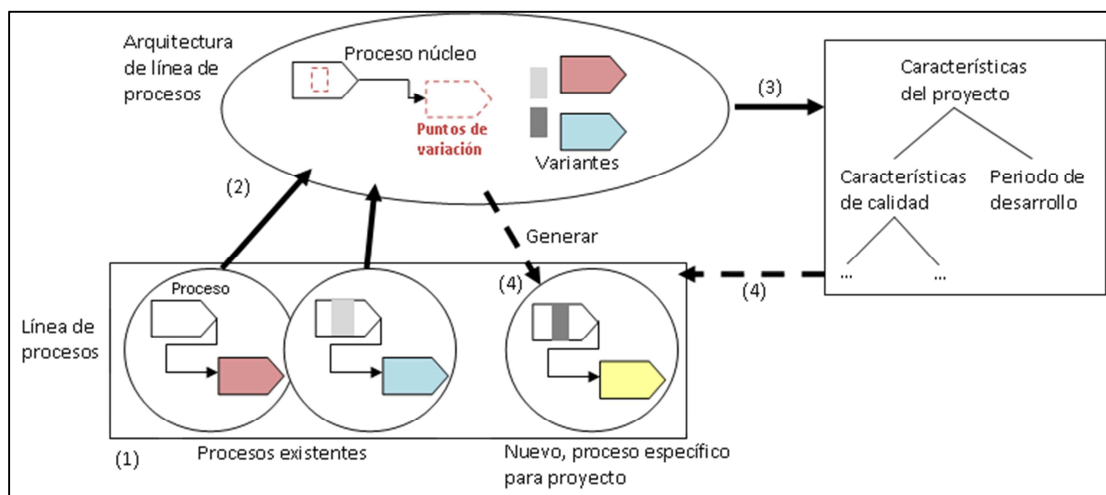


Figura 6 Marco de referencia-Línea de Procesos (27)

El estudio de Washizaki se desarrolla en la ingeniería del dominio, y aunque no presenta propuesta bajo la ingeniería de la aplicación, permite demostrar que la visualización de las relaciones entre una PLA y un modelo de características facilita de manera rápida la definición de un proceso nuevo y consistente, con base en varias características del proyecto.

2.3 Caracterización del contexto para mejora de procesos

2.3.1 Criterios de SEI para la adaptación de procesos.

El Software Engineering Institute (SEI) (28) propone que para usar las prácticas claves del SW-CMMI como requerimientos para la adaptación de procesos, la organización debe determinar las similitudes y las diferencias entre el contexto organizacional y el contexto expresado en los términos del SW-CMMI. Los resultados de este análisis son importantes para definir las actividades del proceso durante su adaptación.

Los criterios importantes para abordar este análisis, son los siguientes:

- *Estructura organizacional*: similitudes y diferencias entre la estructura organizacional implicada por el SW-CMMI y la estructura de la organización.
- *Relación con los clientes y usuarios finales*: Similitudes y diferencias en las relaciones con los clientes, tanto supuestas como reales.
- *Adaptación por afinación*: grado de formalidad, frecuencia, alcance y granularidad de los procesos de software estandarizados en la organización.
- *Objetivos del negocio*: Los objetivos de negocios específicos y necesarios para ser direccionados al implementar programas de mejoramiento de procesos.
- *Impacto del nivel de madurez durante el proceso de adaptación*: La capacidad del proceso actual de la organización

Adaptación de procesos por afinación es quizás la forma más común de realizar la adaptación de procesos. Adaptación por afinación significa que la intención de adaptar un elemento del proceso se satisface al solo aplicar leves cambios en su detalle. En el caso de la SW-CMM, los elementos para adaptación pueden ser actividades, productos de trabajo y artefactos de proceso que pueden necesitar ser modificados de alguna menor manera. El modo en que un elemento es alterado se especifica mediante sus atributos, y para el caso de la SW-CMM, diversos atributos para las actividades, productos de trabajo, y artefactos de

proceso se pueden definir para cada organización. Algunos de los atributos más comunes son:

- **Formalidad** - Los aspectos esenciales de una actividad puede realizarse con un grado de variación del detalle, o de atención a regla, procedimientos o normas formales. Por ejemplo, utilizando el procedimiento "gestión y control" (versión sencilla de control de cambios) es considerada informal, mientras que la gestión de la configuración completa como se describe en el KPA se considera muy formal (cambiar los mandos de control, informes de estado de configuración, etc).
- **Frecuencia** - Hay muchas actividades en el SW-CMM que se realizan de manera "periódica" o a causa de un evento disparador. La frecuencia de cada actividad tiene que ser interpretada a la luz de las necesidades de la organización y del proyecto.
- **Granularidad** - El nivel de detalle necesario en la definición del proceso puede variar. El SW-CMM a menudo dice: "Este documento incluye típicamente...". Una organización puede desear incluir mayor o menor detalle de lo que sugieren, en función de cómo sus artefactos de proceso están estructurados, de la consistencia deseada en el nivel de detalle con otros artefactos, etc.
- **Alcance** - No tiene sentido llevar a cabo ciertas actividades, debido a las limitaciones de la organización, entorno empresarial, etc. Un ejemplo es la "gestión subcontrato", si ninguna organización recurre a la subcontratación, no tiene por qué considerar esta KPA.

2.3.2 Características organizacionales para el modelamiento de procesos de desarrollo de software

Casallas y Arboleda proponen procesos de desarrollo de software para proyectos ágiles para un contexto específico (8), cuyas características se describen en la Tabla 4. Los autores plantean una configuración única para las características propuestas y determinan los procesos asociados para esa configuración.

Características		Nivel
Experiencia del equipo	En la tecnología de desarrollo	Baja/Media
	En la lógica de negocio de desarrollo	Baja/Media
	En investigación y desarrollo	Baja
	En procesos de desarrollo	Baja

Métricas de velocidad de desarrollo conocidas y existencia de datos históricos	Baja
Rotación del equipo	Alta
Condiciones apropiadas para la comunicación del equipo	Baja

Tabla 4 Características de proyectos ágiles (8)

XuyRamesh plantean un modelo de adaptación de procesos de software como respuesta a la necesidad de ajustar los procesos de software al contexto y a los objetivos de un proyecto (29). El modelo consta de cuatro componentes: configuración de los objetivos del proyecto, ajuste de los factores del contexto, evaluación de desafíos, y la adaptación de procesos. El estudio identifica los factores del contexto que soportan el modelo y también describe las guías para realizar la adaptación de los procesos. La Tabla 5 muestra los factores del contexto.

Categorías	Factores
Proyecto	Tamaño del proyecto, cronograma Presupuesto, Tipo Complejidad, Características de los requerimientos Controladores del proyecto
Equipo	Tamaño del equipo, Cohesión Rotación de personal, Experiencia Resistencia al cambio Disponibilidad del personal
Stakeholder externos	Número de stakeholders Experiencia de los stakeholders Acceso a los stakeholders
Organización	Política organizacional, Madurez de los procesos Compromiso de la alta dirección Soporte técnico

Tabla 5 Factores del contexto (29)

De acuerdo a **Laporte** (9), muchos estándares internacionales y modelos semejantes como ISO/IEC 12207 o CMMI no han sido diseñados para VSE (Very Small Entities) y por consiguiente se dificulta aplicarlos en tales entornos. El Grupo de Trabajo de ISO/IEC JTC1/SC7 se ha establecido para manejar esta dificultad, al proponer un estándar de ingeniería de software adaptado para las VSE.

El estándar ISO 29110-1, recientemente publicado, describe el *profile* Básico con las respectivas características del contexto de la organización y las actividades y los procesos de Desarrollo de Software (DS) y Gestión de Proyectos (GM) asociados a dicho *profile*. También plantea un Deployment Package del subproceso Análisis de Requerimientos, en su propósito de completar todo el Framework que promete el Grupo de trabajo. La Tabla 6 lista algunas de las características del *profile* Básico:

Grupo	Características
Características financieras y de recursos	Entre otras: <ul style="list-style-type: none"> • Pequeño número de Ingenieros (costo de nomina hasta 25 personas) • Problemas de liquidez a corto plazo • Proyectos de bajo presupuesto que dura pocos meses • El éxito del proyecto depende del cronograma y del presupuesto
Características de Interface Cliente	Entre otras: <ul style="list-style-type: none"> • Por lo general, un cliente por proyecto a la vez • Satisfacción del cliente depende de: • Cumplimiento de requerimientos específicos que puedan cambiar durante el proyecto.
Características internas del proceso de negocio	Entre otras: <ul style="list-style-type: none"> • El proceso principal está diseñado para desarrollar sistemas de software personalizadas hechos en casa sobre contrato • El producto de software se elabora progresivamente y tiene que ser compatible con necesidades de los clientes • Productos son desarrollados o mantenidos a través de proyectos a través de una sola línea de comunicación entre el grupo desarrollador y el cliente • Hay un pequeño número de ingenieros (por ejemplo, hasta 25 personas) en la organización, y por lo tanto la mayor parte de la comunicación, la toma de decisiones y resolución de problemas puede realizarse rápidamente, cara a cara.
Características de crecimiento y aprendizaje	Entre otras: <ul style="list-style-type: none"> • Conciencia de la importancia de estándares • Carencia de recursos humanos para participar en la estandarización • Carencia de información de estándares ISO/IEC • Falta de conocimiento de mejora de procesos de software y de evaluación de procesos

Tabla 6 Características del profile Básico (9)

3. PROPUESTA DE MODELO DE ADAPTACION DEL PROCESO DE DESARROLLO DE REQUERIMIENTOS

La adaptación de procesos requiere pasos repetibles con el objetivo de hacer consistente el proceso de adaptación dentro de una organización. La Figura 7 muestra el enfoque planteado en el presente trabajo.

La propuesta de modelo de adaptación se basa en cinco procesos: dentro de la Ingeniería del Dominio se *identifican las características organizacionales* que se relacionan con los principios ágiles, se define *el modelo de proceso base de desarrollo de requerimientos* a partir de las prácticas específicas CMMI Dev 1.3, y se *definen directrices* que relacionan características organizacionales con actividades del proceso base, bajo criterios del SEI. Dentro de la Ingeniería de la aplicación se observan dos procesos, el primero es la *configuración manual del contexto*, que consiste en seleccionar que características organizacionales identifican a una organización, y el segundo proceso es la *instanciación del modelo de procesos base* para generar un modelo de proceso de desarrollo de requerimientos delta ajustado a la organización.

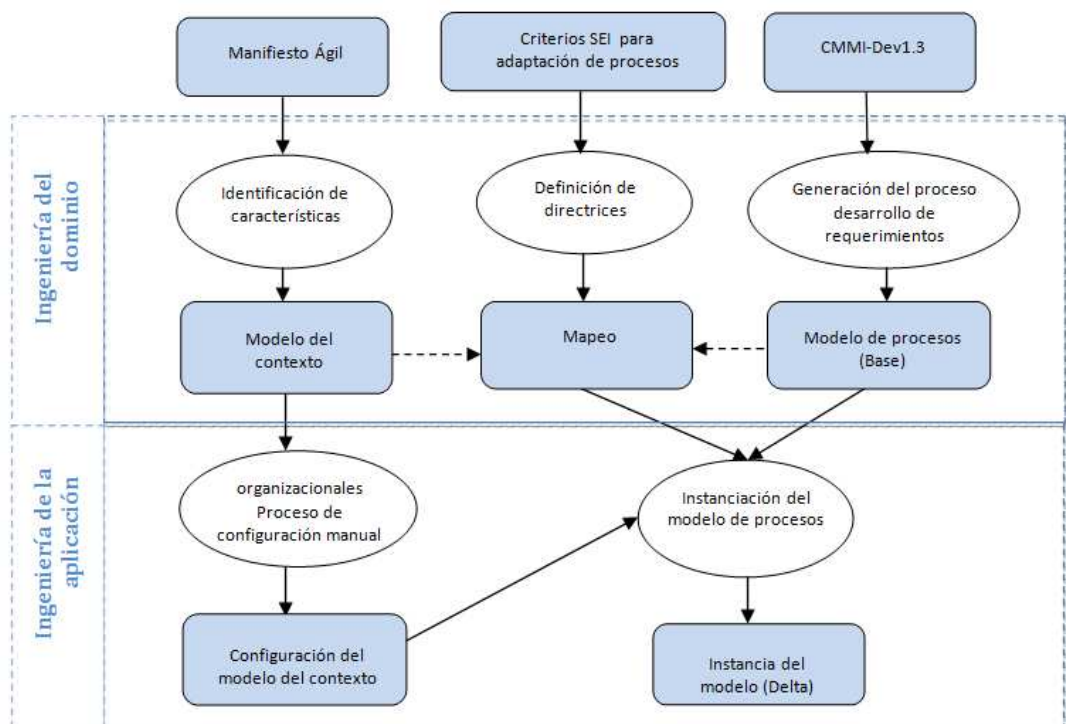


Figura 7 Propuesta de modelo de adaptación

3.1 Identificación de características organizacionales

De las investigaciones realizadas sobre caracterización de pymes y micro empresas, referenciadas en la sección 2.3.2, se realizó un proceso de compilación, análisis y clasificación de las características organizacionales, identificándose las siguientes características organizacionales para modelar el contexto en una organización pequeña: *enfoque de calidad establecido en la organización, estructura organizacional, aceptación de cambios de requerimientos, tipo de comunicación organizacional, y el medidor del progreso para proyectos.*

3.1.1 Enfoque de Calidad.

De acuerdo a lo referenciado en la sección 2.3.2, el factor llamado *madurez del proceso* se vuelve determinante para procesos de desarrollo de software (29), lo que implica que el enfoque de calidad establecido en la organización estará relacionado con la madurez de procesos en la medida en que este se aplique o se implemente en determinados procesos.

Desde otra perspectiva, Turk y France (16), plantean que el octavo principio del Manifiesto Ágil supone que la evaluación de artefactos de software, ya sean productos o procesos, pueden ser realizados al aprovechar las frecuentes revisiones y entrevistas informales de revisión de código, como lo dice el principio.

De esta manera, el Enfoque de Calidad, planteado como característica organizacional, está relacionado con principios ágiles por cuanto la evaluación de productos o procesos, se realiza en el marco de un enfoque de calidad establecido por la organización.

De acuerdo al modelo de calidad de ISO/IEC 9126-1 y a los tres enfoques de calidad de la teoría organizacional, el presente trabajo propone los siguientes tipos de enfoque de calidad:

- Orientado al proceso: una organización con un enfoque de calidad hacia el proceso, busca asegurar que se minimicen las desviaciones de sus procesos contra el modelo estándar que estiman adoptar. La premisa es que, de existir estas desviaciones en su proceso actual, se reduciría la calidad del producto desarrollado.
- Orientado al producto: una organización con un enfoque de calidad hacia el producto asume que la calidad está determinada por una variable precisa y medible, donde las diferencias en calidad están reflejadas por las diferencias en la cantidad de cierto ingrediente o atributo poseído por el producto.

- Orientada al cliente: una organización con un enfoque de calidad hacia el cliente, se basa en la premisa de que la calidad solo la determina el usuario o el cliente, al satisfacer sus preferencias, sus gustos y necesidades.

3.1.2 Estructura Organizacional.

De acuerdo a lo referenciado en la sección 2.3.2, factores como la *cohesión en los equipos de trabajo*, la *disponibilidad de personal*, la *toma de decisiones en equipos pequeños* y el *compromiso de alta gerencia*, son determinantes para procesos de desarrollo de software (29), (9). Desde otra perspectiva, Turk y France (16), plantean que el séptimo principio del Manifiesto Ágil supone que no solo las mejores arquitecturas, elicitaciones de requerimientos y diseños se producen a partir de equipos auto organizados, sino que también deben existir recursos para crear estos equipos y ser apoyados por la administración central.

De lo argumentado anteriormente, se deduce que el tipo de estructura organizacional establecida juega un papel importante para garantizar que se establezcan responsabilidades y autoridad en la jerarquía, y permitan que grupos de trabajo lideren, dispongan de sus recursos, tomen decisiones y se comprometan en procesos de desarrollo de software.

La estructura organizacional es la forma en que se dividen, agrupan y coordinan las actividades de la organización en cuanto a las relaciones entre los gerentes y los empleados, entre gerentes y gerentes, y entre empleados y empleados.

El presente trabajo propone los siguientes tipos de estructura organizacional:

- *Estructura funcional*: La organización por funciones reúne, en un departamento, a todos los que se dedican a una actividad o a varias relacionadas, que se denominan funciones.
- *Estructura por proyecto*: Los empleados de la organización se dividen en grupos y cada grupo se encarga de la producción de un producto específico, además cada grupo tiene un especialista para cada función y un gerente que es el responsable de supervisar el proceso que se lleva a cabo para la obtención del producto o servicio.
- *Estructura matricial*: La estructura matricial se denomina en ocasiones sistema de mandos múltiples. Una organización con una estructura matricial cuenta con dos tipos de estructura simultáneamente. Los empleados tienen, de hecho, dos jefes; es decir, trabajan con dos cadenas de mando. Una cadena de mando es la de funciones o divisiones, el tipo que se diagrama en forma vertical. El segundo es una disposición horizontal que combina al personal de diversas divisiones o departamentos funcionales para formar

un equipo de proyecto o negocio, encabezado por un gerente de proyecto o un grupo, que es experto en el campo de especialización asignado al equipo.

3.1.3 Aceptación de Cambios.

De acuerdo a lo referenciado en la sección 2.3.2, factores como las *características de los requerimientos (29)* y *el cumplimiento de requerimientos específicos que puedan cambiar durante el proyecto (9)*, son determinantes para procesos de desarrollo de software. Desde otra perspectiva, Turk y France (16), plantean que el segundo principio del Manifiesto Ágil supone que los requerimientos siempre evolucionan, debido a cambios de tecnología, necesidades del cliente o dominios del negocio.

De acuerdo al párrafo anterior, la aceptación de cambios se convierte en una característica organizacional, en la medida se establezca durante el desarrollo del proyecto, el momento en el cual se debe aceptar el cambio.

La aceptación de cambios en los requerimientos, está enmarcada, generalmente, en los tipos de proyectos que se requieran abordar. Para proyectos que la organización considere sean robustos y gran tamaño, y se tenga conocimiento del dominio del problema, la organización puede no ser tan flexible y ser suficiente con determinar los requerimientos al principio del proyecto. Para proyectos en los cuales el conocimiento del dominio del problema sea poco o vago, sería recomendable la aceptación de cambios frecuentemente.

El presente trabajo propone establecer los siguientes momentos para aceptar cambios:

- Al inicio del proyecto: la aceptación de cambios al inicio del proyecto puede ser la opción de una organización, debido a un requisito contractual con el cliente, o como resultado de las características propia del proyecto.
- Al final de cada iteración: para procesos de desarrollo de software, donde se elaboran entregables de manera incremental, es recomendable que el proceso de desarrollo de requerimientos también se ajuste a las iteraciones del proceso marco.
- En cualquier momento: la organización interactúa con el cliente frecuentemente para afinar y especificar los requerimientos en un área de dominio desconocida por el desarrollador, o también por acuerdo de las partes.

3.1.4 Comunicación organizacional.

De acuerdo a lo referenciado en la sección 2.3.2, factores como *condiciones apropiadas para la comunicación del equipo, acceso a stakeholders y características internas del proceso de negocio*, se vuelven determinantes para procesos de desarrollo de software, (8), (9), (29). Desde otra perspectiva, Turk y France (16), plantean que el sexto principio del Manifiesto Ágil supone que la interacción cara a cara es el método más productivo para utilizar con el cliente y entre los desarrolladores.

Con los argumentos anteriores, los factores mencionados y la interacción cara a cara que promulga el manifiesto ágil, son características inmersas en el tipo de comunicación organizacional que se adopte o se establezca en la organización.

La comunicación organizacional se define como el conjunto de actividades efectuadas por cualquier organización para la creación y mantenimiento de buenas relaciones con y entre sus miembros, a través del uso de diferentes medios de comunicación que los mantenga informados, integrados y motivados para contribuir con su trabajo al logro de los objetivos organizacionales.

Aunque las comunicaciones al interior de las organizaciones pueden ser, tanto de manera escrita (memorandos, circulares, cartas, folletos) como de tipo oral (conferencia, reuniones, entrevista, vía telefónica), existe una tendencia a utilizar un medio de comunicación más que el otro, debido al estilo de dirección organizacional establecida o su cultura organizacional.

El presente trabajo propone los siguientes tipos de comunicación:

- *Orientado a la escritura*: es preponderante la comunicación escrita, y se caracteriza por contar con mensajes claros, precisos, concretos y correctos, y podemos estar seguros que el mensaje que estamos transmitiendo como emisores, el receptor lo va a entender, tal cual.
- *Orientada a ser oral*: es preponderante la comunicación oral, y se presenta cuando se lleva a cabo una reunión de frente entre dos o más personas, y se puede presentar de manera formal o informal, así como puede estar sujeta a planeación o de forma espontánea.

3.1.5 Medidor del progreso en los proyectos.

De acuerdo a lo referenciado en la sección 2.3.2, factores como *Métricas de velocidad de desarrollo conocidas, controladores del proyecto, cronograma y presupuesto*, se vuelven determinantes para procesos de desarrollo de software, (8), (9), (29). Desde otra perspectiva, Turk y France (16), plantean que el séptimo principio del Manifiesto Ágil supone que la visibilidad o progreso del proyecto se logra solamente a través de la entrega de código funcionando.

De lo planteado anteriormente, se propone que el medidor de progreso en una tarea de desarrollo de software, puede ser a través del uso de métricas, indicadores, y código funcionando.

En la gestión de proyectos se habla de indicadores para hacer el seguimiento y control al presupuesto, al tiempo y al alcance del mismo, sirviendo estos también como base para formar métricas que muestren resultados de avance en general del proyecto. Otra manera de mostrar resultados en una organización, es presentar informes especializados en determinados momentos, sobre todo al final de una fase, para justificar y explicar el estado de avance de un proyecto. Con respecto a procesos de desarrollo de software, la presentación y uso de funcionalidades implementadas, es un indicador aceptado para conocer el avance de un proyecto.

El presente trabajo propone tres tipos de mecanismos que miden el progreso de un trabajo de desarrollo de software, y que puede disponer una organización:

- **Software funcionando:** el avance de un proyecto o de un trabajo de desarrollo de software se mide por las funcionalidades implementadas y usadas por el cliente.
- **Documentación e Informes:** el avance de un trabajo se presenta por la entrega de informes documentados al finalizar una fase del proceso de desarrollo, o al encontrarse con un hito del cronograma, que sea importante para las partes interesadas.
- **Métricas del plan:** el avance de un proyecto se muestra a través de métricas establecidas en el plan de trabajo, con el propósito de conocer el estado general del proyecto en cualquier momento, incluso, y no dejar escapar ninguna variable asociada al proyecto.

3.2 Definición del proceso base de desarrollo de requerimientos de software.

El proceso que se modela para el presente trabajo se basa en el Área de Proceso de Desarrollo de Requerimientos de CMMI Dev v1.3 y se usa como proceso base para el instrumento propuesto, sobre el cual se generarán los procesos delta dependiendo de las características organizacionales seleccionadas en el Modelo del Contexto. El modelo considera las siguientes restricciones y alcances:

- El CMMI está orientado a establecer objetivos específicos que deberán alcanzar una organización para mantener niveles de calidad y madurez en sus procesos, sin pensar en el cómo se deberán implementar las prácticas para llevarlos a cabo (12).

Siguiendo el planteamiento del párrafo anterior, el presente trabajo se enfoca al *qué* y no al *cómo*, en términos de las actividades que contendrá el proceso y lograr los objetivos específicos que promulga el CMMI. Por lo tanto, el modelo de procesos expuesto aquí hace énfasis en las actividades, más que en los roles, tareas y artefactos, con el propósito brindar flexibilidad a la organización objetivo en el momento de implementar las actividades resultantes.

Las Prácticas Específicas (PE o SP por sus siglas en inglés) se contemplan en el presente trabajo como **actividades obligatorias** y las Subprácticas Específicas conforman el universo de la variabilidad del proceso.

- El flujo de actividades planteado para el proceso propuesto recoge las guías subyacentes que establece el CMMI Dev v1.3 y la guía de desarrollo de requerimientos de la adquisición que propone Inteco (30).

La Figura 8 muestra el proceso base de desarrollo de requerimientos al nivel de actividades obligatorias o comunes. Cada uno de los cuadros azules representa una PE del área de proceso Desarrollo de Requerimientos y las flechas representan el flujo del proceso.

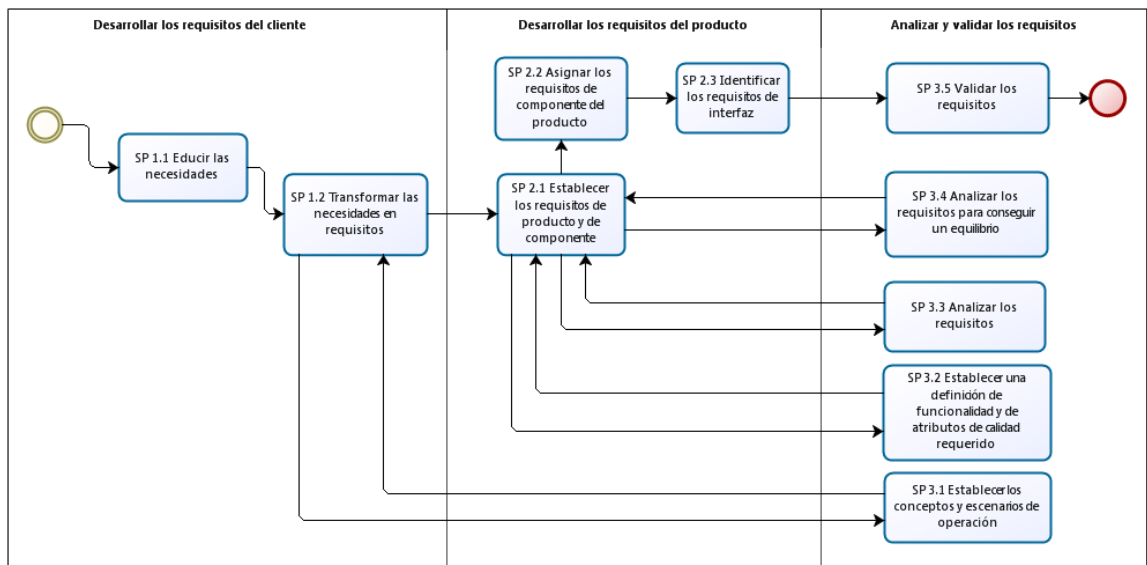


Figura 8 Modelo de proceso base de desarrollo de requerimientos.

A continuación se describen las relaciones identificadas entre las prácticas específicas (actividades obligatorias) del área de proceso Desarrollo de Requerimientos, se describen enmarcados en la meta específica a la que pertenecen:

Desarrollar los requisitos del cliente

El proceso de Desarrollo de Requerimientos inicia en con la actividad “Educir las necesidades” donde se recopilan e identifican las necesidades de las partes interesadas, posteriormente las necesidades son convertidas en requisitos de cliente priorizados en la actividad “Transformar las necesidades de las partes interesadas en requisitos del cliente”.

Desarrollar los requisitos de producto

La actividad “Establecer los requisitos de producto y de componente de producto” recibe los requisitos priorizados para ser expresados en términos técnicos denominados requisitos de producto y requisitos de componentes de producto, estos son recibidos por la actividad “Asignar los requisitos de componente de producto” para ser asignados a los componentes de producto (elementos que componen el producto final), una vez asignados se identifican las interfaces existentes entre funciones (o entre objetos u otras entidades lógicas) y se definen los requisitos de interfaz entre productos o componentes de productos a través de la actividad “Identificar los requisitos de interfaz”.

Analizar y validar los requisitos

Las actividades de esta meta específica brindan soporte a las actividades de las metas específicas “Desarrollar los requisitos del cliente” y “Desarrollar los requisitos de producto”.

Los requisitos del cliente son refinados en la actividad “Establecer conceptos y escenarios de operación” para ser descritos de forma que cualquiera sea capaz de entenderlos.

La actividad “Establecer una definición de la funcionalidad y de los atributos de calidad requeridos” recibe los requisitos de producto para establecer una definición de la funcionalidad requerida del producto, es decir, la descripción de lo que se pretende que haga, igualmente la actividad “Analizar los requisitos” recibe los requisitos de producto para asegurarse que son necesarios y suficientes y en “Analizar los requisitos para conseguir un equilibrio” se reciben los requisitos de producto para analizarlos y equilibrar las necesidades y las restricciones de las partes interesadas.

Por último, en “Validar los requisitos” se validan los requisitos para dar seguridad de que el producto resultante funcione según lo previsto en el entorno del usuario final.

3.3 Definición de estrategia para relacionar características versus actividades.

El presente trabajo propone una directriz ágil para una determinada actividad del proceso base, y con respecto a una característica organizacional, teniendo en cuenta dos criterios: el criterio de adaptación por afinación del SEI y el criterio de juicio de expertos.

Los criterios de SEI, *Frecuencia*, *Granularidad*, y *Formalidad*, permiten identificar el tipo de modificación que se planea efectuar a una actividad, a nivel de procesos. El juicio de expertos selecciona la actividad y determina directriz Ágil con base en subprácticas del CMMI Dev 1.3, que se sugiere para determinada actividad.

Para identificar las relaciones entre características organizacionales y actividades del proceso, susceptibles para establecer una directriz ágil, inicialmente se realiza una inspección entre las actividades y las características correspondientes, para reconocer palabras como verbos y sustantivos que hagan evidente una relación causal.

El proceso de elaboración del mapeo consiste básicamente en el siguiente algoritmo, formulado de la siguiente manera:

```
PARA x:= primera característica HASTA completar todas las características
  HACER
    PARA y := primera actividad HASTA completar todas las actividades bases
      HACER
        • Si se considera que x es una amenaza para implementar los principios ágiles,
          ENTONCES

          directrizAgil(x,y):=
          ¿Cómo adaptary para mitigar en lo posible la amenaza de x ?

        • SI POR EL CONTRARIO,
          ○ Si se considera que x favorece los principios ágiles, ENTONCES

          directrizAgil(x,y):=
          ¿Cómo adaptary para aprovechar la oportunidad que brinda x?

          ○ SI POR EL CONTRARIO,
            directrizAgil(x,y):= NULL.

      FIN HACER
    FIN HACER
```

A manera de ejemplo, si se selecciona la característica '*Estructura Organizacional*' en el FM, y se escoge el valor de '*Funcional*', entonces de acuerdo a la literatura y a las referencias bibliográficas, este tipo de organizaciones están orientadas a la burocracia y a seguir una formalidad en la cadena de autorizaciones. Esta configuración del modelo de características no está acorde con los principios ágiles, por lo tanto, se debe mitigar su impacto proponiendo mecanismos que eviten más la lentitud del proceso de desarrollo de requerimientos.

El presente trabajo propone como directriz ágil para esta configuración del modelo (Estructura Organizacional de tipo funcional) que las actividades del proceso de desarrollo de software en las cuales estén involucradas áreas funcionales como diseño, arquitectura, solución técnica, administración, se realicen de acuerdo a lineamientos de esa área funcional o en su defecto, las actividades se realicen comprometiendo a las áreas.

En la Figura 9 se muestra, parte del artefacto propuesto para establecer las relaciones entre cada característica organizacional y las actividades del proceso base. El artefacto se conforma de columnas, filas y su respectiva celda de intersección. Las filas están controladas por las características organizacionales, y

las columnas están controladas por las actividades del proceso base. Solo en la intersección entre las filas y columnas se puede establecer la directriz Ágil. Como resultado de aplicar la estrategia descrita al inicio de esta sección, se establecieron 52 directrices ágiles. El Anexo 1 incluye la totalidad del artefacto.

Por ejemplo: Para el valor “Software Funcionando (SF)” de la característica “Medidor de progreso (MP)” se observa que tiene relación con la práctica específica “SP1.2 Transformar las necesidades de las partes interesadas en requisitos de cliente”, y su directriz ágil a seguir es “Priorizar grupos de requisitos por funcionalidad”, bajo el criterio de Formalidad del SEI.

CARACTERÍSTICA	VALOR	SP1.1 EDUCIR LAS NECESIDADES DEL CLIENTE	SP1.2 TRANSFORMAR LAS NECESIDADES DE LAS PARTES INTERESADAS EN REQUISITOS DE CLIENTE
Medidor del progreso (MP)	Software funcionando (SF)		Formalidad: Priorizar grupos de requisitos por funcionalidad
	Documentación e informes parciales (DE)		Granularidad: Documento que describa: 1.Requisitos del cliente , 2. Restricciones de cliente para llevar a cabo la verificación. 3. Restricciones de cliente para llevar a cabo la validación
	Indicadores o métricas de avance del plan (IM)		Formalidad: Traducir la necesidades en requisitos documentados

Figura 9 Ejemplo del artefacto usado para describir la directriz Ágil para cada característica del contexto

Debido a que la directrices pueden ser demasiado largas, inclusive párrafos enteros, estas se codifican, para efectos de mejorar el manejo, visualización, presentación, y legibilidad en los modelos de procesos resultantes. La codificación se realiza de la siguiente manera: se inicia la codificación anteponiendo información abreviada de la Práctica Especifica CMMI (SP) a la que se hace referencia, seguido de información abreviada de la característica organizacional, seguido un guion, seguido información abreviada del valor de la característica organizacional, y termina con información entre corchetes del criterio SEI sobre el cual se enmarca.

Por ejemplo, **SP1.2 MP-SF{FOR}** significa: Una directriz de tipo formal (FOR) es propuesta para la actividad “Transformar las necesidades de las partes interesadas en requisitos del cliente” (SP1.2) cuando se selecciona el valor Software Funcionando (SF) de la característica Medidor de Progreso (MP). La Figura 10 muestra el resultado de la codificación para el ejemplo ilustrado en la Figura 9. El Anexo 2 incluye el artefacto de la Figura 10 en su versión completa.

CARACTERÍSTICA	VALOR	SP1.1 EDUCIR LAS NECESIDADES DEL CLIENTE	SP1.2 TRANSFORMAR LAS NECESIDADES DE LAS PARTES INTERESADAS EN REQUISITOS DE CLIENTE
Medidor del progreso (MP)	Software funcionando (SF)		SP1.2 MP-SF {FOR}
	Documentación e informes parciales (DE)		SP1.2 MP-DE {GRA}
	Indicadores o métricas de avance del plan (IM)		SP1.2 MP-IM {FOR}

Figura 10 Ejemplo de codificación de las directrices ágiles propuestas

La Figura 11, muestra el universo de la variabilidad del proceso base de desarrollo de requerimientos, resultado del proceso de mapeo y de la codificación de directrices. Cada variante se representa con un comentario en UML(en amarillo) y se refiere a la directriz ágil establecida en el mapeo para las cinco características organizacionales.

Por ejemplo en la Figura 11 se observa que en la práctica específica “SP1.2 Transformar las necesidades de las partes interesadas en requisitos de cliente” se relaciona con la característica “Medidor del progreso (MP)” en sus valores “Software funcionando (SF)”, “Documentación e informes (DE)” e “Indicadores o métricas de avance del plan (IM)”, y sus respectivas directrices son **SP1.2MP-SF {FOR}**, **SP1.2 MP-DE {GRA}** y **SP1.2 MP-IM {FOR}**. Si se observa la Figura 11, que muestra el universo de la variabilidad del proceso base, se observa que estas mismas directrices aparecen relacionadas a la práctica específica SP1.2 y además existen otras directrices, pero relacionadas a otras características. Son estas directrices (elementos del modelo resaltados en amarillo) las que pueden aparecer

o desaparecer del modelo de proceso delta resultante, dependiendo de la configuración de características seleccionadas en el modelo del contexto.

Si la directriz (elemento de color amarillo en el modelo) aparece en el modelo de proceso delta resultante, significa que se sugiere aplicar la directriz ágil que aparece codificada en el recuadro, sino aparece es porque no se sugieren cambios.

3.4 Implementación del Instrumento para la generación automática del proceso de DR.

3.4.1 Diagrama UML del Modelo de proceso base DR con su respectiva variabilidad

La figura 13 muestra el modelo de proceso base de desarrollo de requerimientos y su respectiva variabilidad. La variabilidad es el conjunto de todas las directrices propuestas en este trabajo para todas las posibles configuraciones de las características organizacionales. La variabilidad está representada en el modelo como *etiquetas* UML de color amarillo, asociadas a sus respectivas actividades de acuerdo a la estrategia de mapeo.

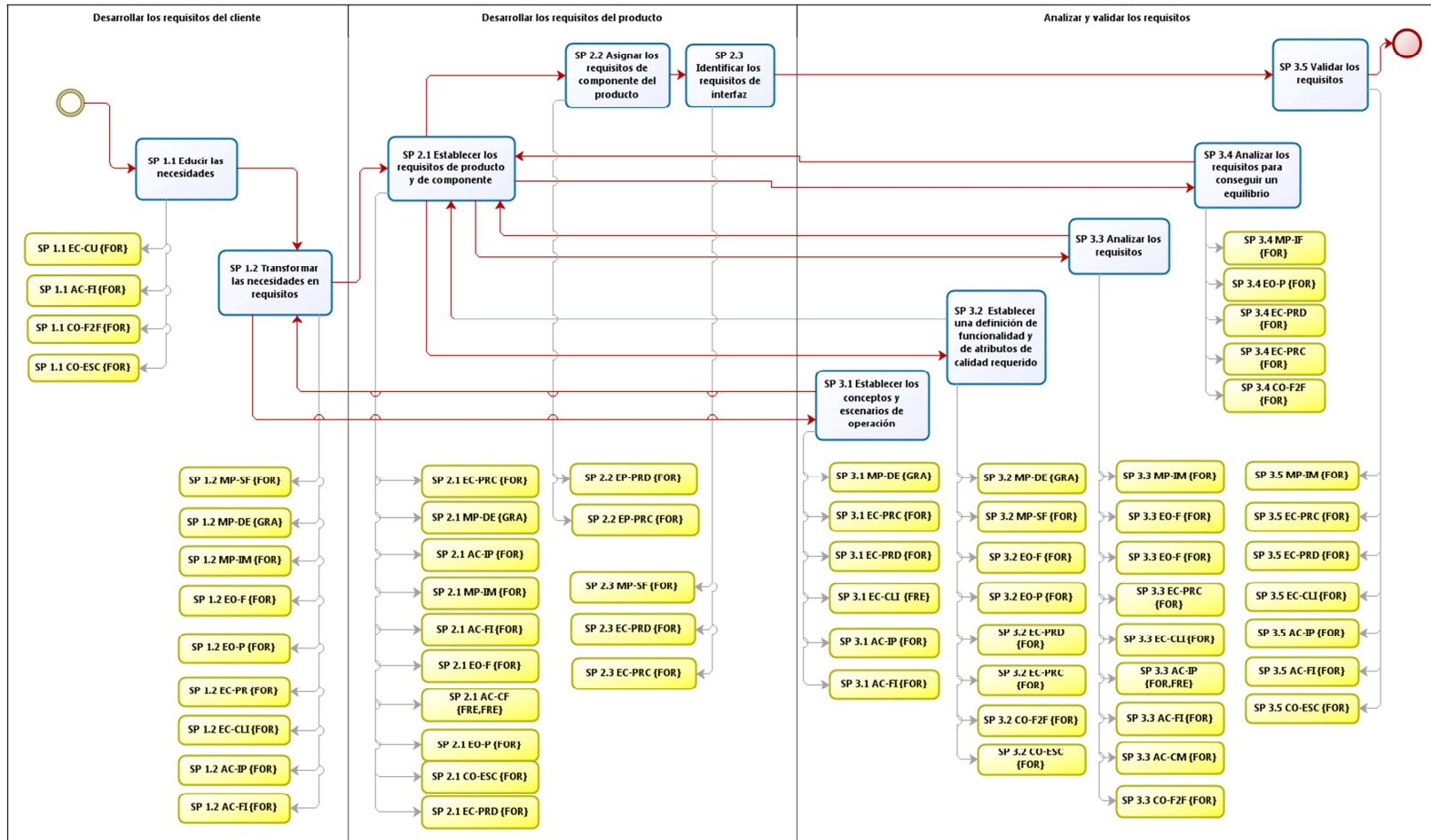


Figura 11 Variabilidad del proceso de desarrollo de requerimientos

3.4.2 Modelo del Contexto en Pure Variants

El modelo del contexto se elabora a partir de las características organizacionales utilizando la técnica de modelo de variabilidad llamada *feature model*, referenciada en la sección 2.2.1. La Tabla 7 muestra el modelo de contexto, donde los *puntos de variabilidad* son las características organizacionales y las *variantes* son las diferentes *especializaciones* que puede tomar la característica en un contexto organizacional determinado. En el presente trabajo, los puntos de variabilidad son de tipo *alternativo*, es decir, en un punto de variabilidad se puede escoger solo una variante.

Característica organizacional	Modelo del Contexto (feature model)	
	Punto de variabilidad (tipo alternativo)	Variantes
Enfoque de Calidad <ul style="list-style-type: none"> Orientado al producto Orientado al proceso Orientado al cliente 	EnfoqueCalidad	<ul style="list-style-type: none"> AIProducto AIProceso AICliente
Estructura organizacional <ul style="list-style-type: none"> Estructura funcional Estructura por proyecto Estructura matricial 	EstructuraOrganizacional	<ul style="list-style-type: none"> Funcional porProyecto Matricial
Aceptación de cambios <ul style="list-style-type: none"> Al inicio del proyecto Al final de cada iteración En cualquier momento 	AceptaciónDeCambios	<ul style="list-style-type: none"> AllnicioProyecto AlFinalCadaIteracion EnCualquierMomento
Comunicación organizacional <ul style="list-style-type: none"> Orientada la escritura Orientada a ser oral 	ComunicaciónOrganizacional OrientadaA	<ul style="list-style-type: none"> Escrito CaraACara
Medidor de progreso <ul style="list-style-type: none"> Software funcionando Documentación e informes Métricas del plan 	MedidorDelProgreso	<ul style="list-style-type: none"> SoftwareFuncionando DocumentaciónEinformes MetricasDelPlan

Tabla 7 Modelo del contexto con base en características organizacionales

El modelo del contexto se automatiza, utilizando la herramienta Pure Variants (31) con las anteriores características del contexto, todas de carácter obligatorio y sus valores de tipo alternativo (al menos una de las variantes debe ser seleccionada).

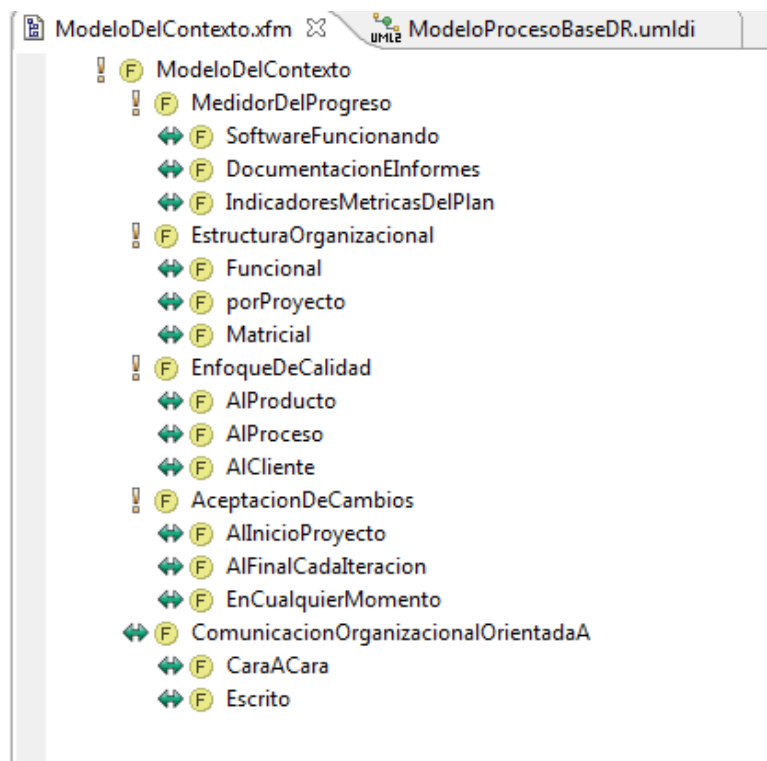


Figura 12 Modelo del contexto en Pure Variants

La Figura 12 muestra una imagen del modelo de contexto en el cual se observan las 5 características, cada una de ellas representa un punto de variación con sus respectivas variantes, por ejemplo: la característica Medidor de Progreso es el punto de variación e incluye las variantes SoftwareFuncionando, DocumentacionEInformes e IndicadoresMetricasDelPlan. Al hacer uso del modelo del contexto dentro de la presente propuesta se debe seleccionar una de las variantes en cada punto de variación para obtener una nueva configuración del contexto.

3.4.3 Mapeo de modelos con Feature Mapper

Se utiliza Feature Mapper (FMr) (32) como herramienta para mapear características del contexto organizacional contra el modelo del proceso base de desarrollo de requerimientos. Para usar FMr se ingresa tanto, el diagrama UML del modelo del proceso base de desarrollo de requerimientos, como el modelo del contexto de Pure Variant.

Se establecen manualmente las relaciones identificadas en el mapeo ilustrado en la Figura 9 y se genera automáticamente un modelo de procesos delta al recibir una configuración de modelo del contexto. Como se observa en la Figura 13, las directrices se han codificado y representado por etiquetas UML de color naranja, para mejorar la legibilidad y cobertura en la pantalla del Feature Mapper.

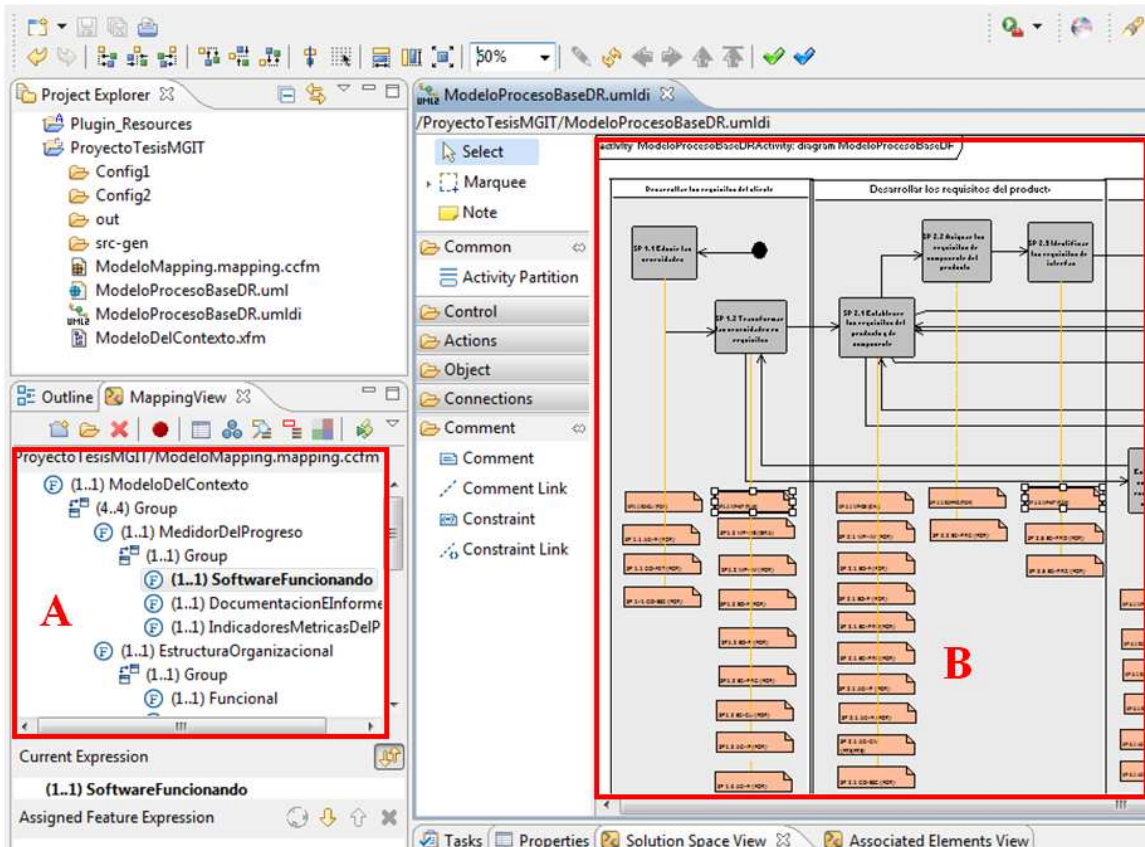


Figura 13 Modelo de mapeo con Feature Mapper

En la Figura 13 se observa una imagen del FMr en ejecución, en la parte inferior izquierda (recuadro A) se aprecia la Vista de Mapeo (Mapping View) en la cual se maneja el modelo de contexto (MC) y en la parte derecha (recuadro B) se observa parte del modelo de proceso base (MP) ilustrado en la Figura 13. La relación entre características del MC y elementos del MP, identificados previamente en el proceso de mapeo conceptual del apartado 3.3, se realiza a través de un proceso manual que involucra una serie ordenada de clics entre el recuadro A y el recuadro B. (Ver (32) para más detalles).

En esta sección se presenta el uso del modelo propuesto en la sección 3 aplicado al proceso de desarrollo de requerimientos de software.

3.5 Prueba del instrumento

3.5.1 Configuración del Modelo de Contexto para una organización hipotética.

La organización hipotética, es un área de desarrollo de software en una institución pública, y aseguran que el medidor de progreso que adoptan para conocer el avance en sus proyectos de desarrollo, es la presentación de funcionalidades del software, debido a que existen muchas quejas por la calidad del producto. Sin embargo, debido a que tienen otros tipos de actividades, solo reciben requerimientos al inicio, cuando se plantean necesidades de desarrollo, y las esporádicas interacciones con el cliente son de manera personal.

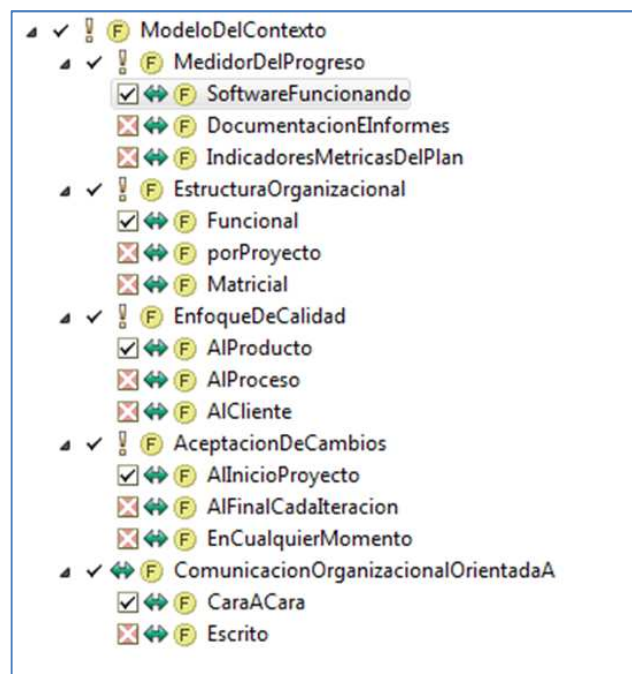


Figura 14 Configuración del Modelo de Contexto – Caso de prueba

En la Figura 14 se observa una configuración del FM en el cual se han seleccionado las siguientes características:

- | | | |
|---------------------------|----|------------------------|
| Medidor del Progreso | -> | Software Funcionando |
| Estructura Organizacional | -> | Funcional |
| Enfoque de Calidad | -> | Al producto |
| Aceptación de cambios | -> | Al inicio del proyecto |

Comunicación organizacional orientada a -> Cara a cara

3.5.2 Proceso de desarrollo de requerimientos generado por el instrumento

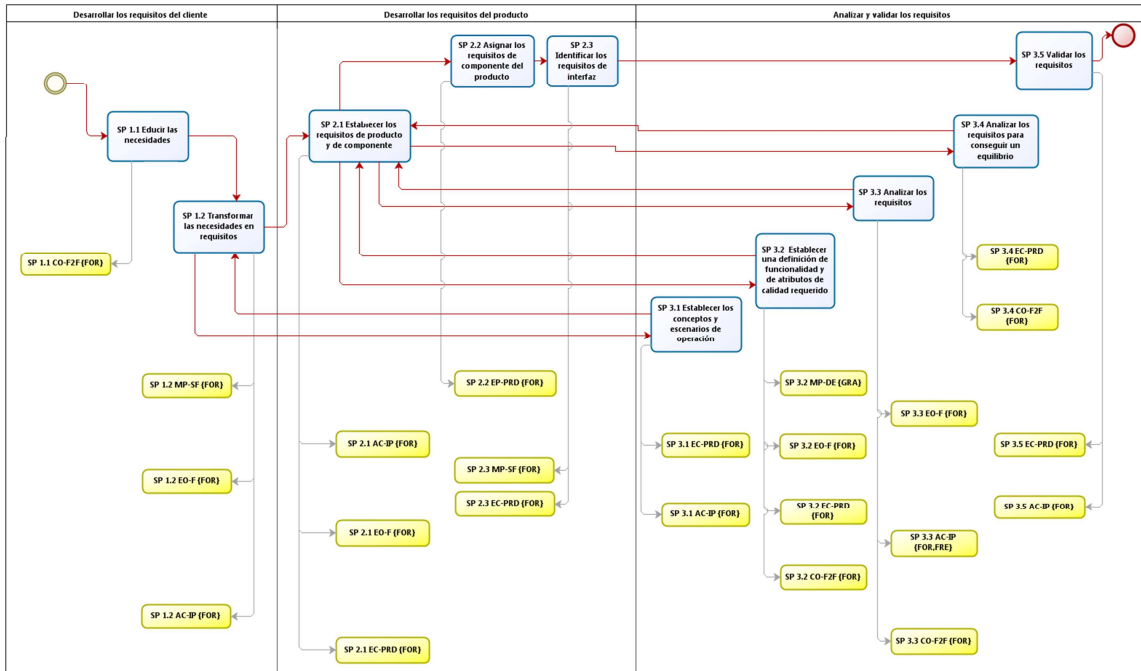


Figura 15 Modelo de proceso delta resultante - Caso de prueba.

El FMr genera un modelo de proceso delta, que contiene el modelo de proceso base (común) y las directrices codificadas (variable) que solo correspondan a esas características seleccionadas en el modelo de contexto. La herramienta, sin embargo, presenta directrices sin código marcadas con "Unresolved". Para efectos de mejorar la visualización, manualmente se reescribe el modelo delta eliminando las directrices marcadas como "Unresolved", para obtener el modelo resultante como se ilustra en la Figura 15. En la Figura 16 se observa la estructura del proceso, antes y después de hacer uso del instrumento. El proceso delta resultante sugiere realizar las modificaciones a la actividad "Educar los requisitos" en los términos de la directriz **SP1.1CO-F2F{FOR}**, y modificaciones a la actividad "Transformar las necesidades en requisitos" en los términos de las directrices **{SP1.2 MP-SF {FOR}, SP1.2 EO-F {FOR},SP1.2 AC-IP{FOR}**.

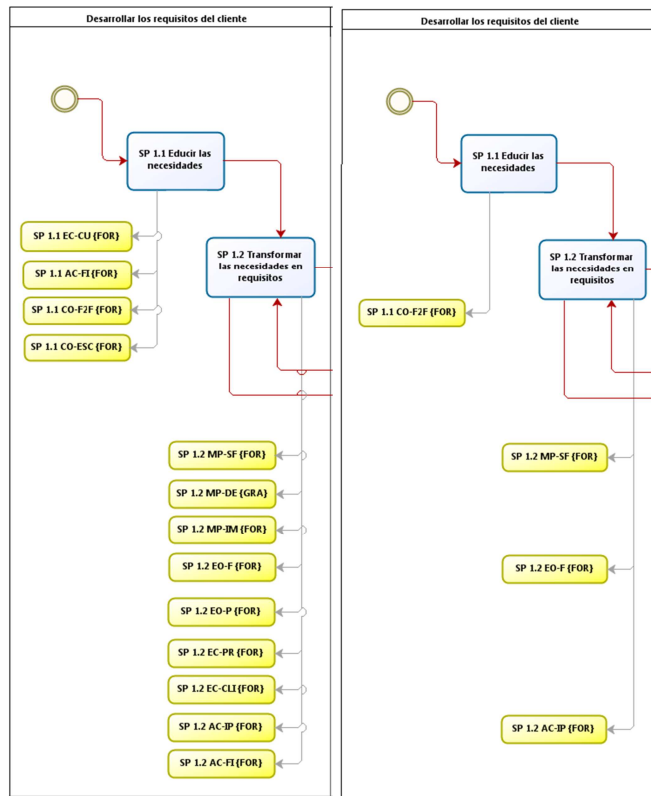


Figura 16 Comparación proceso base vs proceso delta – Caso de prueba

3.5.3 Lectura de las directrices sugeridas

Las directrices presentadas en la figura 19, sugieren que en la actividad de “educir necesidades”, se habiliten las técnicas de Lluvia de Ideas y Workshops, aprovechando que la interacción con el cliente es cara a cara (face to face). Las directrices también recomiendan a los analistas que tengan en cuenta o prioricen grupos de requerimientos por funcionalidad (relación con Software funcionando), documenten y validen con los demás líderes funcionales la entrega o recibimiento de su trabajo (relación con estructura de tipo funcional, y elicitar solo al inicio), para evitar excusas de entregas informales o no documentadas y por lo tanto perder tiempo, ya que se tendría que retomar el trabajo en un futuro inmediato implicando más tiempo dedicado a la misma tarea (Algo no deseado, no ágil).

4. VALIDACIÓN DE LA PROPUESTA

4.1 Descripción de la organización

La validación de la propuesta se realizó en el Área de Desarrollo de Software de la Universidad del Valle, adscrita a la Oficina de Informática y Telecomunicaciones (OITEL) de la Institución.

El Área de Desarrollo de Software inició su funcionamiento aproximadamente en el 2001 con la responsabilidad de construir los sistemas de información de la institución, inició con la creación del Sistema de Información para Registro Académico (SRA), posteriormente con la construcción del Sistema de Información para Recursos Humanos (SIRH) y paulatinamente fue desarrollando los demás sistemas de información institucional para lograr al 2012 un total de 31 sistemas construidos en casa.

Todos los sistemas han sido construidos con tecnologías libres como PHP, HTML, JavaScript, Postgres y MySQL, bajo el liderazgo de personal planta y colaboración de Ingenieros contratistas vinculados temporalmente de acuerdo a las necesidades a evacuar durante un periodo determinado, las soluciones a desarrollar responden a necesidades particulares de las diferentes dependencias de la Universidad, constituyéndose cada dependencia en los clientes del área.

Al ser un área interna que atiende necesidades propias de la institución, basa su estrategia de comunicación en el contacto personal con el cliente para obtener información de primera línea y directamente de los usuarios funcionales. La Universidad del Valle, al ser una organización estatal, debe responder ante las directrices del gobierno departamental y nacional, por lo tanto, el Área de Desarrollo acepta las solicitudes de cambio a realizar sobre los sistemas de información cuando la alta dirección o los líderes funcionales lo requieran.

El cliente determina el grado de atención y respuesta del área, de acuerdo a las funcionalidades de software puestas en producción, es decir, al ser miembro de la misma organización le es indiferente cualquier otro tipo de productos (documentación, informes, modelos y demás) diferentes a los productos de software que requiere para realizar sus trabajos académicos o administrativos. Y el interés principal en cuanto al software puesto en producción, es que este responda eficientemente en los procesos misionales y de alto impacto administrativo, principalmente donde se involucran procesamiento masivo, por ejemplo: admisiones y matrícula de estudiantes, vinculaciones y liquidación de nómina.

A pesar de pertenecer a una institución universitaria y de contar con el apoyo académico de la Escuela de Ingeniería de Sistemas de la institución, el Área no ha sido ajena a los problemas a los que se enfrenta una organización dedicada al

desarrollo de software, principalmente relacionados con la identificación, definición y modificación de requerimientos.

A partir del 2008 iniciaron la reestructuración de los procesos internos con el fin de adoptar buenas prácticas, la estrategia está implementada y se encuentra en maduración, pero a pesar de los esfuerzos realizados y los procesos establecidos aún se presentan deficiencias que han tenido impacto negativo antes la comunidad universitaria.

Desde el punto de vista del recurso humano, el equipo de desarrollo está conformado por un Coordinador de Desarrollo, seis Líderes de proyectos, un Administrador de Bases de Datos y dependiendo de las necesidades y recursos económicos, cuenta temporalmente con Ingenieros contratistas.

Las características del Área concuerdan con organizaciones calificadas como microempresas las cuales cuentan con menos de 10 empleados, tienen problemas comunes relacionados con el Desarrollo de Requerimientos y pertenecen a una organización académica que genera conocimiento relacionado con el área, por lo tanto, se consideró apta para realizar la validación.

Integración del panel de validación

Se estableció un panel de 6 integrantes ejerciendo los siguientes roles:

- Director de la OITEL.
- Coordinador de Desarrollo.
- Cinco Líderes de proyectos de desarrollo de software.

La oficina no cuenta en el momento con un Coordinador de Desarrollo, por lo tanto, el Director de la oficina asume los dos roles (Director y Coordinador) para efectos administrativos y consecuentemente para efectos de la validación.

Formación profesional del panel

La Tabla 8 describe la formación los integrantes a nivel de pregrado y postgrado, adicionalmente muestra los años de experiencia de cada individuo en lo referente a proyectos de Desarrollo de Software.

Rol	Formación Pregrado	Formación Postgrado	Experiencia en Desarrollo de Software (años)
Director de la OITEL	Ingeniero en Telecomunicaciones	Magister en Ingeniería Industrial Magister en Telecomunicaciones	10
Coordinador de Desarrollo	Ingeniero en Telecomunicaciones	Magister en Ingeniería Industrial Magister en Telecomunicaciones	10
Líder de proyectos 1	Ingeniero de Sistemas	Magister en Administración de Empresas	8
Líder de proyectos 2	Ingeniero de Sistemas	Magister en Administración de Empresas	8
Líder de proyectos 3	Ingeniero de Sistemas	Magister en Ingeniería de Sistemas	6
Líder de proyectos 4	Ingeniero de Sistemas		8
Líder de proyectos 5	Ingeniero de Sistemas		3

Tabla 8 Formación profesional y experiencia del panel - Caso Univalle

4.2 Configuración del Modelo de Contexto con base en las características de la organización

De acuerdo a la descripción del área y con base en lo sugerido por el panel propuesto para la validación, se seleccionaron en el Modelo de Contexto las siguientes características:

Medidor del progreso - Software Funcionando: Los clientes del área le dan mayor importancia a los productos de software puestos en producción que a cualquier otro tipo de producto (documentación, indicadores, modelos, etc).

Estructura organizacional – Funcional: El área se enfoca en realizar las funciones para la cual fue creada.

Enfoque de calidad - Orientado al producto: La calidad del producto se mide por el nivel de seguridad, rendimiento y confiabilidad del software.

Aceptación de cambios -en cualquier momento: El área responde a solicitudes y requerimientos de las diferentes dependencias de la Universidad, por lo tanto, debe admitir los cambios en cualquier momento de lo contrario puede obstruir la operación académico-administrativa de la institución.

Comunicación organizacional orientada a -Cara a cara: Los clientes del área son los mismos empleados de las diferentes dependencias, por lo tanto, el contacto es constante y directo.

La Figura 17 muestra la configuración del Modelo de Contexto que corresponde al Área de Desarrollo de la Universidad del Valle.

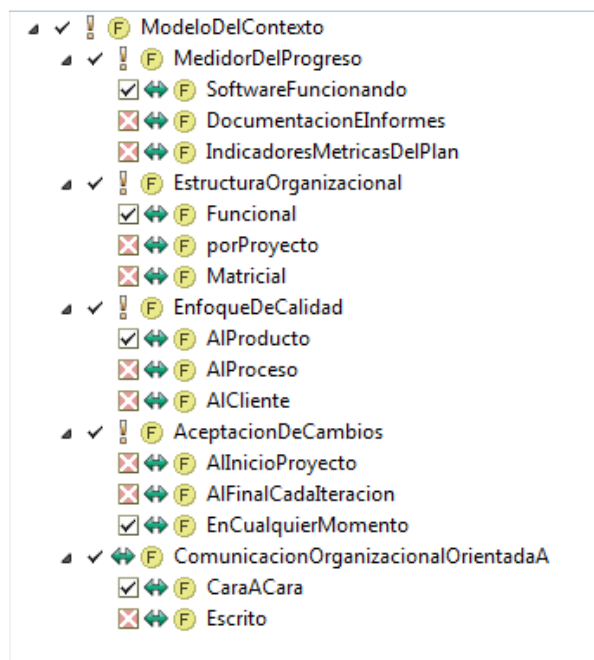


Figura 17 Configuración del Modelo del Contexto Caso Univalle
Área de Desarrollo

4.3 Proceso de desarrollo de requerimientos sugerido

De acuerdo a las características seleccionadas y con base en el mapeo descrito en el apartado 3.3, el instrumento sugiere el proceso delta ilustrado en la Figura 18.

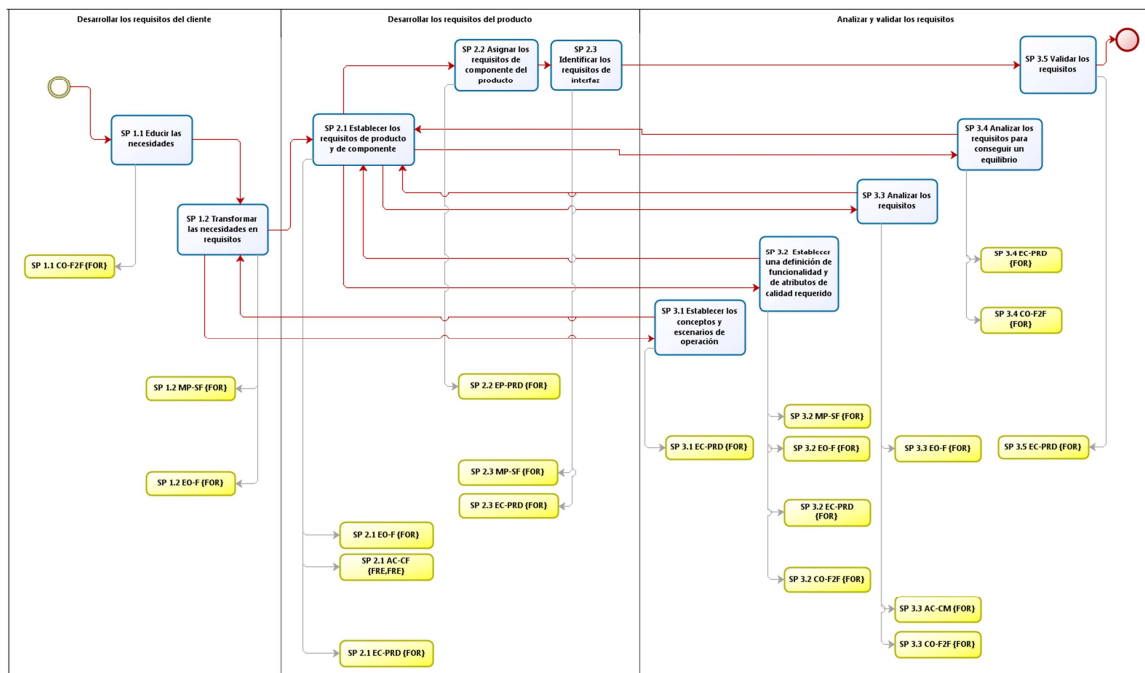


Figura 18 Modelo de proceso delta resultante - Caso Univalle

A continuación se describen las modificaciones sugeridas sobre las actividades de la meta específica “Desarrollo de los requisitos del cliente”.

En el bloque izquierdo de la Figura 19 se observa el modelo de procesos base para la meta específica “Desarrollo de los requisitos del cliente” y en el bloque derecho el proceso delta generado de acuerdo a las características seleccionadas en el modelo de contexto.

El proceso delta especifica las directrices ágiles **SP1.1CO-F2F{FOR}**, **SP1.2 MP-SF{FOR}** y **SP1.2 EO-F{FOR}**, lo cual indica que de acuerdo a las características seleccionadas el instrumento sugiere aplicar las modificaciones especificadas en dichas directrices, a saber:

SP1.1 CO-F2F{FOR}: Habilitar la educación de necesidades a partir técnicas como de Lluvia de Ideas y Workshops.

SP1.2 MP-SF{FOR}: Priorizar grupos de requisitos por funcionalidad.

SP1.2 EO-F{FOR}: Traducir las necesidades en requisitos documentados.
 - Definir las restricciones para la verificación y validación en coordinación con el grupo de verificación y validación.

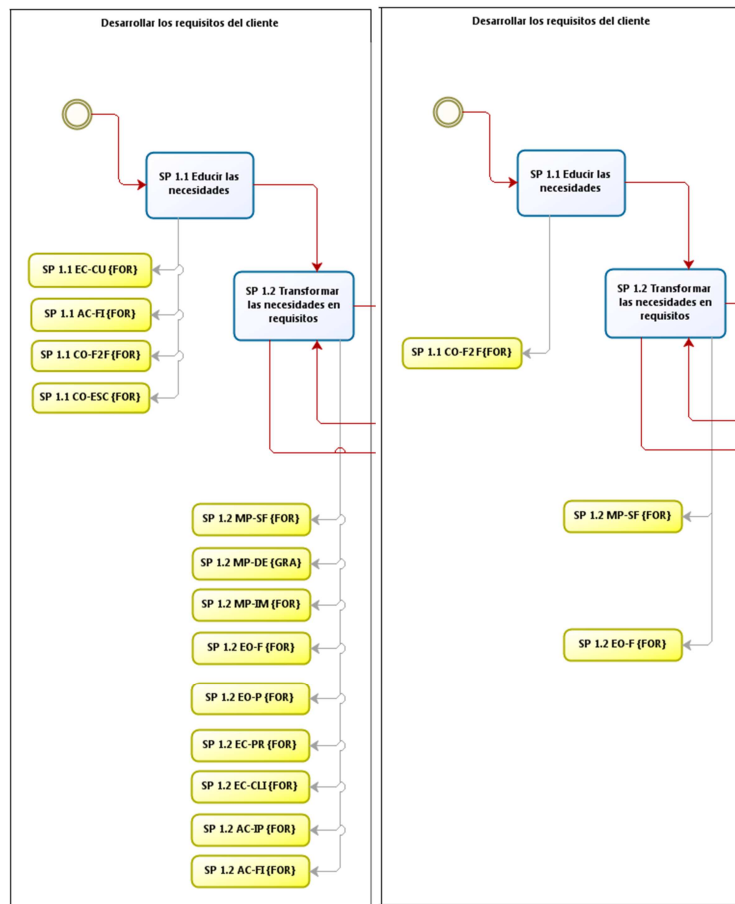


Figura 19 Modelo de procesos base vs delta Caso Univalle
Meta específica: Desarrollo de los requisitos del cliente

4.4 Análisis de proceso delta

En la Tabla 9 se observa una matriz que permite realizar la comparación del estado actual de las actividades en el Área de Desarrollo y las directrices sugeridas por el instrumento.

Características organizacionales	Actividad afectada	Estado actual de la actividad en la organización	Criterio de adaptación del SEI	Directriz ágil sugerida para la actividad	Conclusión
Comunicación organizacional orientada a - Cara a cara	Educir las necesidades del cliente	-Se hace cara a cara con el usuario funcional, en la gran mayoría de sesiones interviene solo un usuario funcional. -No se incluye a todos los stakeholders.	Formalidad	Habilitar la educación de necesidades a partir técnicas como de Lluvia de Ideas y Workshops	Al habilitar las técnicas sugeridas se asegura que se involucren en la educación de necesidades a la gran mayoría de stakeholders (los responsables del negocio y los desarrolladores)
Medidor del progreso - Software Funcionando	Transformar las necesidades de las partes interesadas en requisitos del cliente	- En muchas ocasiones Los requisitos de cliente entran a ser evacuados sin ser priorizados, es decir, de manera indiscriminada. - En ocasiones quien prioriza los requisitos a evacuar es el líder funcional.	Formalidad	Priorizar grupos de requisitos por funcionalidad	-Tener priorizados los requisitos de cliente ayuda a determinar el alcance del proyecto, de la iteración o del incremento. Esta priorización asegura que los requisitos funcionales y de los atributos de calidad críticos para el cliente y otras partes interesadas se tratan rápidamente - Al priorizar los requisitos por funcionalidad, se establecen relaciones de impacto con las actividades operativas de la organización, es decir, se puede determinar qué funcionalidades deben ser atendidas en primera instancia para no afectar la operación. Lo cual permite entregar software funcionando y que suma valor al software disponible en producción.
Estructura organizacional – Funcional	Transformar las necesidades de las partes interesadas en requisitos del cliente	-La gran mayoría de necesidades no se documenta pues no existe un artefacto formal para ello, las solicitudes iniciales quedan en un correo electrónico o carta, pero no su especificación. - Muchos requisitos de cliente no se registran en los artefactos aprobados para tal fin, los artefactos no se basan en herramientas automatizadas. - No se establecen criterios de aceptación, por lo tanto, al validar los requisitos hay incertidumbre en cuanto a lo que en realidad se debe entregar. En ocasiones es el líder de desarrollo quien define lo que se debe entregar.	Formalidad	- Traducir la necesidades En requisitos documentados. - Definir las restricciones para la verificación y validación en coordinación con el grupo de verificación y validación	-La documentación se se realiza en el artefacto que la organización defina. - En el grupo de validación y verificación se involucran stakeholders, quienes deben definir en común acuerdo con el equipo de desarrollo el con junto de restricciones para verificación y validación.

Tabla 9 Comparación entre actividades actuales y actividades sugeridas
Caso Univalle

4.5 Resultados de la Encuesta

Se realizó la encuesta del Anexo 3, con el propósito de conocer la opinión general del panel sobre el instrumento propuesto y determinar la efectividad del mismo.

De acuerdo a la tabulación de las respuestas del panel ilustrada en la Tabla 10 y de los aportes realizados por sus integrantes, se observa un entendimiento del instrumento y una aceptación generalizada del proceso delta resultante. Algunas observaciones y sugerencias recibidas se enfocan en:

- Automatizar el proceso de decodificación de las directrices ágiles pues resulta muy complejo para el usuario del instrumento.
- Enriquecer el contenido de las directrices ágiles.
- Involucrar otras MyPyMes en el proceso de definición de directrices ágiles.
- Aplicar la misma estrategia en otras Áreas de Proceso de CMMI.

Pregunta	Respondieron Si	Observaciones para respuestas (SI)	Observaciones para respuestas (NO)
1. ¿Considera que el propósito del instrumento es de interés para su organización y similares?	7	-CMMI es muy complejo para una PyME pero si se mapea con Agile puede brindar una ayuda significativa. -Se enfoca en las características de la organización. -Facilita la identificación de mejoras en los procesos. -Con el uso del instrumento la organización puede obtener un acercamiento a lo que puede implementar de CMMI. -Es muy importante para mejorar la calidad y efectividad de los procesos.	
2. ¿Considera que el enfoque planteado para el instrumento CMMI-Dev 1.3 – AGII, es pertinente para solucionar el problema planteado?	7	-Ofrece una estrategia para mapear Agile con otros modelos. -Es un enfoque claro, permite entender la sinergia entre las características y el modelo para poder sacar el mejor provecho.	
3- ¿Está de acuerdo con el modelo de procesos arrojado por el instrumento, con base en las características de su organización?	6	-Las directrices son acertadas, se debe definir cómo se debe proceder si algunas directrices están basadas en juicios de expertos. -Es muy objetivo el proceso sugerido.	Se requiere conocer más detalles de las directrices ágiles que arrojó la herramienta en el modelo delta.
4- ¿Considera que los argumentos teóricos son suficientes para soportar la propuesta?	7	-El argumento sustenta muy bien las bases utilizadas y el objetivo de la propuesta. -La propuesta está soportada con suficientes argumentos	
5- ¿Se deben realizar modificaciones al instrumento planteado?	1		Sería interesante que el conjunto de directrices se dividiera en 2. las que tengas sugerencias de los autores (subjetivas) y las básicas (propuestas por el SEI)
6- Plantee las observaciones y/o comentarios que considere pertinente		El área de desarrollo de Univalle tiene poca experiencia en Agile y CMMI. Es recomendable probar el instrumento en varias PyMES.	

Tabla 10 Resultados de la encuesta

4.6 Mediciones

El proceso sugerido se encuentra en implementación y seguimiento dentro del Área de Desarrollo, después de un mes de sugerir cambios en el proceso, se realiza una medición con dos conjuntos de requerimientos y con base en los atributos de calidad “requerimiento correcto” y “requerimiento no ambiguo”, la ambigüedad y correctitud del grupo de requerimientos se determinó en conjunto con los stakeholders involucrados.

Conjunto A: 10 requerimientos obtenidos a través de las prácticas habituales.

Conjunto B: 10 requerimientos obtenidos usando las modificaciones a la actividad sugeridas por el instrumento.

En la Tabla 11 se observan los índices de cumplimiento para los atributos de calidad evaluados.

Atributo de calidad	Cumplen Conjunto A	% de cumplimiento para Conjunto A	Cumplen Conjunto B	% de cumplimiento para Conjunto B	Diferencia
No ambigüo	5	50%	7	70%	20%
Correcto	7	70%	7	70%	0%

Tabla 11 Índices de cumplimiento para atributos de calidad - validación

- El índice de requerimientos no ambiguos del conjunto B es superior en un 20% respecto al conjunto A, esto se origina en la elicitación de necesidades y que se han involucrado a varios de los stakeholders en lugar de ejecutar la elicitación con un solo usuario funcional.
- No se observan cambios en lo referente al atributo de correctitud.
- No se realizan otras mediciones pues los cambios se encuentran en su etapa temprana de implementación.

5. RESULTADOS OBTENIDOS

El resultado más importante de la presente propuesta es la definición de un instrumento que integra elementos que parecían estar en caminos opuestos: Ágil y CMMI, y permite a cualquier tipo de organización el acercamiento a la adopción y/o adaptación de un modelo robusto, principalmente para las micro y pequeñas empresas las cuales presentan dificultades para hacerlo.

Se entrega la automatización del instrumento a través del uso de la herramienta Feature Mapper (32) con la cual se logra generar los modelos de procesos deltas de manera gráfica. En la parte izquierda de la Figura 20 se observa la el proceso base para la práctica específica “Educir las necesidad” y en el lado derecho el proceso delta generado después de hacer uso de la herramienta con base en unas características seleccionadas en el modelo de contexto o FM.

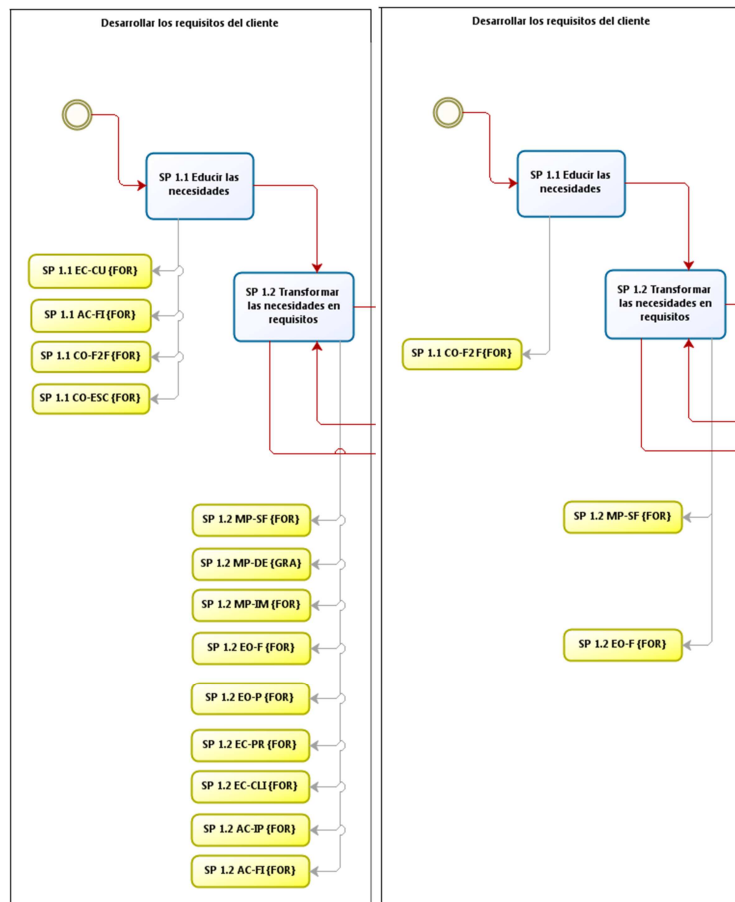


Figura 20 Modelo de procesos base vs delta

Con el objetivo de definir los elementos que establecen la relación entre Ágil y el CMMI, se creó un artefacto que permite registrar las relaciones entre

características organizacionales basadas en Ágil y practicas específicas de CMMI, para el presente estudio se usaron específicamente las del área de proceso Desarrollo de Requerimientos. El artefacto es en sí, una propuesta para relacionar características a las actividades de un modelo y puede ser usado para mapear características de cualquier tipo con otro tipo de modelos. Las Figuras 21 y 22 muestran partes del artefacto en sus dos versiones, para más detalle ir al apartado 3.3.

CARACTERÍSTICA	VALOR	SP1.1 EDUCIR LAS NECESIDADES DEL CLIENTE	SP1.2 TRANSFORMAR LAS NECESIDADES DE LAS PARTES INTERESADAS EN REQUISITOS DE CLIENTE
Medidor del progreso (MP)	Software funcionando (SF)		Formalidad: Priorizar grupos de requisitos por funcionalidad
	Documentación e informes parciales (DE)		Granularidad: Documento que describa: 1.Requisitos del cliente , 2. Restricciones de cliente para llevar a cabo la verificación. 3. Restricciones de cliente para llevar a cabo la validación
	Indicadores o métricas de avance del plan (IM)		Formalidad: Traducir la necesidades en requisitos documentados

Figura 21 Artefacto para mapeo de característica vs práctica específica

CARACTERÍSTICA	VALOR	SP1.1 EDUCIR LAS NECESIDADES DEL CLIENTE	SP1.2 TRANSFORMAR LAS NECESIDADES DE LAS PARTES INTERESADAS EN REQUISITOS DE CLIENTE
Medidor del progreso (MP)	Software funcionando (SF)		SP1.2 MP-SF {FOR}
	Documentación e informes parciales (DE)		SP1.2 MP-DE {GRA}
	Indicadores o métricas de avance del plan (IM)		SP1.2 MP-IM {FOR}

Figura 22 Artefacto para codificación de directrices ágiles del mapeo de característica vs práctica específica

La identificación y selección de características a ser consideradas en el modelo implica un esfuerzo significativo y adicionalmente la cantidad de características seleccionadas puede generar un crecimiento exponencial de relaciones característica-actividad en el mapeo, por lo tanto, se creó una artefacto basado en el enfoque de Turk y France (16) que permite establecer una estrategia para reducir el conjunto de características identificadas y relacionarlas con los principios ágiles, el artefacto permite cubrir la gran mayoría de principios Ágiles con un mínimo de características, lo cual permite reducir considerablemente el universo de relaciones entre características y prácticas específicas.

La metodología usada para proponer el instrumento se puede usar para relacionar características con las actividades de otras áreas de proceso de CMMI Dev y poder obtener procesos delta sugeridos partiendo del modelo base a implementar y con base en características organizacionales, por lo tanto, la metodología usada es en sí un producto significativo que puede ser usado en diferentes escenarios del proceso de Desarrollo de Software.

6. CONCLUSIONES Y FUTURO TRABAJO

Se ha presentado una propuesta que permite a cualquier organización dedicada al Desarrollo de Software, determinar qué actividades del área de proceso de Desarrollo de Requerimientos de CMMI Dev 1.3 puede ejecutar de acuerdo a sus características organizacionales. El enfoque puede ser usado por cualquier tipo de organización que se dedique al negocio pero está principalmente pensado para organizaciones pequeñas.

La inclusión de los principios ágiles hace que las actividades propuestos por CMMI se puedan implementar en organizaciones pequeñas ya que ágil propone la reducción del grado de complejidad de las actividades a realizar y se enfoca más en resultados y el cliente, más que en cualquier otro tipo de elementos.

Se ha propuesto una estrategia para relacionar características organizacionales con los doce principios ágiles, sin necesidad de generar un conjunto de características numeroso. Adicionalmente se ha proporcionado una herramienta que permite establecer conceptualmente las relaciones entre características y actividades (mapeo), las cuales posteriormente pueden ser automatizadas en una herramienta computacional para hacer más rápido el proceso de mapeo y generación de procesos delta.

La metodología usada para obtener el instrumento puede ser aplicada para relacionar características organizacionales con otras áreas de proceso de CMMI.

Es recomendable construir un aplicativo de software que permita obtener, del modelo de procesos delta generado, el detalle de la directriz ágil que está relacionado a cada elemento variable del modelo, pues el proceso de decodificar la directriz es manual y dispendioso, y puede introducir errores humanos al proceso.

7. BIBLIOGRAFÍA

1. **FEDESOFTE**. Fedesoft. [En línea] [Citado el: 14 de Febrero de 2012.] <http://www.fedesoft.org/content/hacia-donde-va-la-industria-del-software>.
2. *Definición de una metodología ágil e ingeniería de requerimientos para empresa emergentes de desarrollo de software del sur-occidente colombiano*. **Mercha, Luis, Urrea, Alba y Rebollar, Ruben**. 1, Vol. 6, págs. 37-50.
3. **ACIS**. IX encuentro de proyectos de TI. [En línea] 15 de Febrero de 2011. [Citado el: 6 de Febrero de 2012.] http://www.acis.org.co/fileadmin/Base_de_Conocimiento/IX_Jornada_Gerencia/ConferenciaAlbertoCuetoVigil.pdf.
4. —. VIII Encuesta de Proyectos de TI. [En línea] 23 de Febrero de 2010. [Citado el: 6 de Febrero de 2012.] http://www.acis.org.co/fileadmin/Base_de_Conocimiento/VIII_Jornada_Gerencia/VIIEncuestadeGerenciadeProyectosdeTIACV.pdf.
5. **GROUP, STANDISH**. Standish Newsroom CHAOS. [En línea] 23 de Abril de 2009. [Citado el: 11 de Noviembre de 2011.] https://secure.standishgroup.com/newsroom/chaos_2009.php.
6. —. TASSC. [En línea] 2011. [Citado el: 16 de Febrero de 2012.] <http://www.tassc-solutions.com/pages/factfile.htm>.
7. **Casallar, Ruby**. Revista 102 ACIS. [En línea] Octubre de 2007. [Citado el: 3 de Noviembre de 2011.] http://correo.acis.org.co/fileadmin/Revista_102/columnista.pdf.
8. **Arboleda, Hugo y Casallas, Ruby**. QualDev Process: Procesos Adaptables de Desarrollo de Software para Proyectos Ágiles. *ournal INGENIERIA Y COMPETITIVIDAD - Facultad de Ingeniería, Universidad del Valle*. [En línea]
9. *Contributions to Software Engineering and to the Development and Deployment of International Software Engineering Standards for Very small Entities*. **Laporte, Claude**.
10. *Mejora de procesos de software ágil SPI-*. **Pardo, Cesar, Hurtado, Julio Ariel y Collazos, Cesar**. 164, 2010, Dyna, Vol. 77, págs. 251-263.

11. **Ambrust, Ove y Yuko, Miyamoto.** Scoping Software Process Lines. [En línea] [Citado el: 20 de Junio de 2012.] http://www.ove-armbrust.de/downloads/Armbrust-Software_Process_Scoping_SPIP.pdf.
12. **Gillel, Glazel, y otros.** SEI - CMMI or Agile, Why not embrace both! [En línea] Noviembre de 2011. [Citado el: 14 de Febrero de 2012.] <http://www.sei.cmu.edu/reports/08tn003.pdf>.
13. *Análisis descriptivo del proceso de implementación del nivel 2 del modelo CMMI en una empresa regional de desarrollo de software.* **Picazo, Catherine, Villegas, Norma y Tamura, Gabriel.** Santiago de Cali : s.n., Diciembre de 2008, ICESI - Sistema y telemática, Vol. 6, págs. 89-109.
14. **SEI.** SEI - Software Engineering Institute - CMMI DEV 1.3. [En línea] Noviembre de 2010. [Citado el: 20 de Febero de 2012.] <http://www.sei.cmu.edu/library/assets/whitepapers/Spanish%20Technical%20Report%20CMMI%20V%201%203.pdf>.
15. *Metodologías Ágiles en el Desarrollo de Software.* **Letelier, P.** Citeseerks.
16. *Assumptions Underlying Agile Software Development.* **Turk, Daniel y France, Robert.** 4, 29 de Mayo de 2003, Journal of Database Management, Vol. 16, págs. 62-87.
17. **Beck, Kent, Beedle, Mike y Bennekum, Arie.** Principios el manifiesto Ágil. [En línea] 11 de Febrero de 2001. [Citado el: 20 de Agosto de 2012.] <http://agilemanifesto.org/iso/es/principles.html>.
18. *Concepts and Guidelines of Feature Modeling for Product Line Software Engineering.* **Lee, Kwanwoo, Kang, Kyo y Jaejoon, Lee.** 2002, SI Web of Science - Lecture Notes in Computer Science, Vol. 2319, págs. 62-67.
19. *SOFTWARE PRODUCT - Exploring principles and potential solutions.* **Klaus, Pohl y Metzger, Andreas.** 12, New York : ACM, Diciembre de 2006, Communications of the ACM - Software product line, Vol. 9, págs. 78-81.
20. **Clements, Paul y Northrop, Linda.** *Software Product Lines: Practices and Patterns.* 3. s.l. : Addison-Wesley Professional, 2001.

21. **Pohl, Claus y Metzger, Andreas.** *Variability Management in Software Product Line Engineering.* Shangai : ACM, 2006. págs. 1049-1050.
22. *Mapping Features to Models: A Template Approach Based on Superimposed Variants.* **Krzysztof, Czarnecki y Antkiewicz, Michal.** 2005 : s.n., Lecture Notes in Computer Science, Vol. 3676, págs. 422-437.
23. **Czarnecki, Krzysztof y Wasowski, Andrzej.** Feature Modeling and Model Templates - Feature Diagrams and Logics: There and Back Again. [En línea] 2007. [Citado el: 15 de Junio de 2012.] <http://gsd.uwaterloo.ca/featureModelingAndModelTemplates>.
24. *A Family of Languages for Variability Management in Software Product Lines.* **Zschaler, Steffen y Sánchez, Pablo.**
25. **Simmonds, Jocelyn y Bastarrica, María Cecilia.** Modeling Variability in Software Process Lines. [En línea] [Citado el: 11 de Junio de 2012.] http://swp.dcc.uchile.cl/TR/2011/TR_DCC-20110914-010.pdf.
26. *An MDE Approach to Software Process Tailoring.* **Hurtado, Julio, Bastarrica, Maria Cecilia y Quipe, Alcides.** ACM, págs. 43-52.
27. *Building Software Process Line Architectures.* **Hironori, Washizaki.** 2006, Lecture Notes in Computer Science, Vol. 4034, págs. 415-421.
28. **Ginsberg, Mark y Quinn, Lauren.** Software Engineering Institute. [En línea] Noviembre de 1995. [Citado el: 15 de Junio de 2012.] <http://www.sei.cmu.edu/library/abstracts/reports/94tr024.cfm>.
29. *Software Process Tailoring: An Empirical Investigation.* **Peng, Xu y Balasubramaniam, Ramesh.** 2007, Journal of Management Information Systems, Vol. 24, págs. 293-328.
30. **INTECO - Instituto Nacional de Tecnologías de la Comunicación.** INTECO - Instituto Nacional de Tecnologías de la Comunicación. [En línea] Noviembre de 2009. [Citado el: 20 de Agosto de 2012.] www.inteco.es/file/XaXZyrAaEYdn-ZA9X-rQJw.
31. Pure Variants. [En línea] [Citado el: 20 de Julio de 2012.] http://www.pure-systems.com/pure_variants.49.0.html.

32. **Florian, Heidenreich.** Feature Mapper. [En línea] 1 de Diciembre de 2008.
[Citado el: 15 de Julio de 2012.] <http://featuremapper.org/>.

8. ANEXO 1 - Descripción de la función directriz Ágil.

CARACTERÍSTICA	VALOR	SP1.1 EDUCIR LAS NECESIDADES DEL CLIENTE	SP1.2 TRANSFORMAR LAS NECESIDADES DE LAS PARTES INTERESADAS EN REQUISITOS DE CLIENTE	SP2.1 ESTABLECER LOS REQUISITOS DEL PRODUCTO Y COMPONENTES DEL PRODUCTO	SP2.2 ASIGNAR LOS REQUISITOS DE COMPONENTE DE PRODUCTO	SP2.3 IDENTIFICAR LOS REQUISITOS DE INTERFAZ	SP3.1 ESTABLECER LOS CONCEPTOS Y LOS ESCENARIOS DE OPERACIÓN	SP3.2 ESTABLECER UNA DEFINICIÓN DE FUNCIONALIDAD Y DE LOS ATRIBUTOS DE CALIDAD REQUERIDOS	SP3.3 ANALIZAR LOS REQUISITOS	SP3.4 ANALIZAR LOS REQUISITOS PARA CONSEGUIR UN EQUILIBRIO	SP3.5 VALIDAR LOS REQUISITOS
Medidor del progreso (MP)	Software funcionando (SF)		Formalidad: Priorizar grupos de requisitos por funcionalidad			Formalidad: priorizar la identificación de interfaces externas y humanas, del producto		Analizar y cuantificar la funcionalidad requerida por los usuarios finales. Analizar los requisitos para identificar las particiones lógicas o funcionales. Asignar los requisitos de cliente a las particiones funcionales, objetos, personal o elementos de soporte para dar apoyo a la síntesis de las soluciones			
	Documentación e informes parciales (DE)		Granularidad: Documento que describa: 1. Requisitos del cliente, 2. Restricciones de cliente para llevar a cabo la verificación. 3. Restricciones de cliente para llevar a cabo la validación	Granularidad: Documento de especificación de requisitos			Granularidad: Documento de conceptos de operación detallado	Granularidad: Artefacto de Arquitectura funcional			
	Indicadores o métricas de avance del plan (IM)		Formalidad: Traducir la necesidades en requisitos documentados	Formalidad: Desarrollar los requisitos en los términos técnicos necesarios para el diseño del producto y de los componentes de producto				Formalidad: Analizar los requisitos para asegurarse que son completos, factibles, realizables y verificables. Identificar las medidas técnicas de rendimiento que serán seguidas durante el esfuerzo de desarrollo	Evaluar el impacto de los requisitos de los atributos de calidad significativos en los costes y riesgos del desarrollo del producto		Formalidad: Explorar la adecuación y la completitud de los requisitos

CARACTERÍSTICA	VALOR	SP1.1 EDUCIR LAS NECESIDADES DEL CLIENTE	SP1.2 TRANSFORMAR LAS NECESIDADES DE LAS PARTES INTERESADAS EN REQUISITOS DE CLIENTE	SP2.1 ESTABLECER LOS REQUISITOS DEL PRODUCTO Y COMPONENTES DEL PRODUCTO	SP2.2 ASIGNAR LOS REQUISITOS DE COMPONENTE DE PRODUCTO	SP2.3 IDENTIFICAR LOS REQUISITOS DE INTERFAZ	SP3.1 ESTABLECER LOS CONCEPTOS Y LOS ESCENARIOS DE OPERACIÓN	SP3.2 ESTABLECER UNA DEFINICIÓN DE FUNCIONALIDAD Y DE LOS ATRIBUTOS DE CALIDAD REQUERIDOS	SP3.3 ANALIZAR LOS REQUISITOS	SP3.4 ANALIZAR LOS REQUISITOS PARA CONSEGUIR UN EQUILIBRIO	SP3.5 VALIDAR LOS REQUISITOS
Estructura organizacional (EO)	organización funcional (F)		<p>Formalidad: Traducir la necesidades en requisitos documentados.</p> <p>Formalidad: Definir las restricciones para la verificación y validación en coordinación con el grupo de verificación y validación</p>	<p>Formalidad: Desarrollar los requisitos en los términos técnicos necesarios dispuestos por el grupo diseñador</p>				<p>Formalidad: Determinar los atributos de calidad significativos para la arquitectura en base a Directrices Organizacionales sobre misión clave y los factores del negocio</p>	<p>Formalidad: Actividad de concertación con las partes interesadas para analizar sus necesidades, expectativas y restricciones, para resolver conflictos de requisitos</p>		
	organización por proyecto (P)		<p>formalidad: Mantener priorización de requisitos</p>	<p>las relaciones entre los requisitos para su consideración durante la gestión del cambio y la asignación de los requisitos.</p>				<p>Formalidad: Determinar la misión clave y los factores de negocio</p>		<p>impacto de los requisitos de los atributos de calidad significativos para la arquitectura en el producto y en los</p>	
	organización matricial (funcional, proyecto) (M)								<p>documento resultado de la concertación con las partes interesadas sobre las necesidades, expectativas y restricciones, para</p>		

CARACTERÍSTICA	VALOR	SP1.1 EDUCIR LAS NECESIDADES DEL CLIENTE	SP1.2 TRANSFORMAR LAS NECESIDADES DE LAS PARTES INTERESADAS EN REQUISITOS DE CLIENTE	SP2.1 ESTABLECER LOS REQUISITOS DEL PRODUCTO Y COMPONENTES DEL PRODUCTO	SP2.2 ASIGNAR LOS REQUISITOS DE COMPONENTE DE PRODUCTO	SP2.3 IDENTIFICAR LOS REQUISITOS DE INTERFAZ	SP3.1 ESTABLECER LOS CONCEPTOS Y LOS ESCENARIOS DE OPERACIÓN	SP3.2 ESTABLECER UNA DEFINICIÓN DE FUNCIONALIDAD Y DE LOS ATRIBUTOS DE CALIDAD REQUERIDOS	SP3.3 ANALIZAR LOS REQUISITOS	SP3.4 ANALIZAR LOS REQUISITOS PARA CONSEGUIR UN EQUILIBRIO	SP3.5 VALIDAR LOS REQUISITOS
Enfoque de Calidad (EC)	Al producto (PRD)			<p>Formalidad: Desarrollar los requisitos en los términos técnicos necesarios para el diseño del producto y de los componentes de producto. Inferir los requisitos resultantes de las decisiones de diseño. Desarrollar los requisitos de arquitectura capturando los atributos críticos de calidad y las medidas de atributos de calidad</p>	<p>Formalidad: Asignar los requisitos a las funciones, los componentes de producto y a la arquitectura. Asignar las restricciones de diseño a componentes de producto y a la arquitectura. Asignar requisitos a las entregas incrementales.</p>	<p>Formalidad: Identificar las interfaces tanto externas como internas al producto y Desarrollar sus requisitos</p>	<p>Formalidad: Desarrollar un concepto de operación detallado</p>	<p>Formalidad: Identificar la funcionalidad y los atributos de calidad deseados. Determinar los atributos de calidad significativos para la arquitectura en base a la misión clave y los factores del negocio. Analizar y cuantificar la funcionalidad requerida por los usuarios finales</p>		<p>Formalidad: 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</p>	<p>Formalidad: Analizar los requisitos para determinar el riesgo de que el producto resultante no funcione apropiadamente en el entorno de uso previsto. Explorar la adecuación y la completitud de los requisitos desarrollando representaciones del producto (p. ej., prototipos, simulaciones, modelos, escenarios, guiones gráficos)</p>
	Al proceso (PRO)		<p>Formalidad: Traducir la necesidades en requisitos documentados</p>	<p>Formalidad: Desarrollar los requisitos en los términos técnicos necesarios para el diseño del producto y de los componentes de producto. Establecer y mantener las relaciones entre los requisitos para su consideración durante la gestión del cambio y la asignación de los requisitos.</p>	<p>Asignar los requisitos a las funciones y a los componentes de producto y a la arquitectura.</p>	<p>Formalidad: Desarrollar los requisitos para las interfaces identificadas</p>	<p>Formalidad: Desarrollar los conceptos y los escenarios de operación</p>	<p>Formalidad: Determinar los atributos de calidad significativos para la arquitectura en base a la misión clave y los factores del negocio. Analizar y cuantificar la funcionalidad requerida por los usuarios finales</p>	<p>Formalidad: Analizar los requisitos para asegurarse que son completos, factibles, realizables y verificables. Identificar las medidas técnicas de rendimiento que serán seguidas durante el esfuerzo de desarrollo</p>	<p>Formalidad: 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</p>	<p>Formalidad: 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</p>
	Al Cliente (CLI)			<p>Formalidad: Definir las restricciones para la verificación y validación</p>				<p>Frecuencia: Revisar los conceptos y los escenarios de operación para refinar y descubrir requisitos.</p>	<p>Frecuencia: 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>Formalidad: 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</p>

CARACTERÍSTICA	VALOR	SP1.1 EDUCIR LAS NECESIDADES DEL CLIENTE	SP1.2 TRANSFORMAR LAS NECESIDADES DE LAS PARTES INTERESADAS EN REQUISITOS DE CLIENTE	SP2.1 ESTABLECER LOS REQUISITOS DEL PRODUCTO Y COMPONENTES DEL PRODUCTO	SP2.2 ASIGNAR LOS REQUISITOS DE COMPONENTE DE PRODUCTO	SP2.3 IDENTIFICAR LOS REQUISITOS DE INTERFAZ	SP3.1 ESTABLECER LOS CONCEPTOS Y LOS ESCENARIOS DE OPERACIÓN	SP3.2 ESTABLECER UNA DEFINICIÓN DE FUNCIONALIDAD Y DE LOS ATRIBUTOS DE CALIDAD REQUERIDOS	SP3.3 ANALIZAR LOS REQUISITOS	SP3.4 ANALIZAR LOS REQUISITOS PARA CONSEGUIR UN EQUILIBRIO	SP3.5 VALIDAR LOS REQUISITOS
Aceptacion de Cambios (AC)	Al inicio proyecto (P)		Formalidad: Traducir la necesidades en requisitos documentados y Definir las restricciones para la verificación y validación	Formalidad: Establecer y mantener las relaciones entre los requisitos para su consideración durante la gestión del cambio y la asignación de los requisitos.			Formalidad: Desarrollar los conceptos y los escenarios de operación. Definir el entorno en el que funcionará el producto o componente de producto. Desarrollar un concepto de operación detallado.		Formalidad: Analizar las necesidades, las expectativas, las restricciones y las interfaces externas de las partes interesadas para organizarlas en temas relacionados y eliminar conflictos. Frecuencia: Analizar los requisitos para determinar si satisfacen los objetivos de Los requisitos de nivel más alto. Frecuencia: Analizar los requisitos para asegurarse que son completos, factibles, realizables y verificables. Analizar los conceptos y los escenarios de operación para refinar las necesidades, las restricciones y las interfaces del cliente, y para inferir nuevos requisitos		Formalidad: 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
	Al final de cada iteración (FI)	Formalidad: Se socializa con el cliente los momentos establecidos para iterar	Formalidad: Los cambios a los requisitos se documentan al final de cada iteración.	Formalidad: Inferir los requisitos resultantes de las decisiones de diseño. Establecer y mantener las relaciones entre los requisitos para su consideración durante la gestión del cambio y la asignación de los requisitos, al final de cada iteración			Formalidad: Revisar los conceptos y los escenarios de operación para refinar y descubrir requisitos.		Formalidad: Analizar los requisitos para determinar si satisfacen los objetivos de Los requisitos de nivel más alto. Analizar los requisitos para asegurarse que son completos, factibles, realizables y verificables		Formalidad: 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
	En cualquier momento (CM)			Frecuencia: Inferir los requisitos resultantes de las decisiones de diseño. Frecuencia: Establecer y mantener las relaciones entre los requisitos para su consideración durante la gestión del cambio y la asignación de los requisitos					Formalidad: Analizar los requisitos para determinar si satisfacen los objetivos de Los requisitos de nivel más alto. Analizar los requisitos para asegurarse que son completos, factibles, realizables y verificables		

CARACTERÍSTICA	VALOR	SP1.1 EDUCIR LAS NECESIDADES DEL CLIENTE	SP1.2 TRANSFORMAR LAS NECESIDADES DE LAS PARTES INTERESADAS EN REQUISITOS DE CLIENTE	SP2.1 ESTABLECER LOS REQUISITOS DEL PRODUCTO Y COMPONENTES DEL PRODUCTO	SP2.2 ASIGNAR LOS REQUISITOS DE COMPONENTE DE PRODUCTO	SP2.3 IDENTIFICAR LOS REQUISITOS DE INTERFAZ	SP3.1 ESTABLECER LOS CONCEPTOS Y LOS ESCENARIOS DE OPERACIÓN	SP3.2 ESTABLECER UNA DEFINICIÓN DE FUNCIONALIDAD Y DE LOS ATRIBUTOS DE CALIDAD REQUERIDOS	SP3.3 ANALIZAR LOS REQUISITOS	SP3.4 ANALIZAR LOS REQUISITOS PARA CONSEGUIR UN EQUILIBRIO	SP3.5 VALIDAR LOS REQUISITOS
Comunicación orientada a ser (CO)	Oral (FaceToFace) (F2F)	Formalidad: Habilitar la educion de necesidades a partir tecnicas como de Lluvia de Ideas,Workshops						Analizar y cuantificar la funcionalidad requerida por los usuarios finales	Formalidad: Agendar reuniones para analizar las necesidades, las expectativas, las restricciones y las interfaces externas de las partes interesadas para organizarlas en temas relacionados y eliminar conflictos	Formalidad: Usar modelos, simulaciones y prototipos probados para analizar el equilibrio entre las necesidades y las restricciones de las partes interesadas.	
	Escrita (ESC)	Formalidad: establecer las técnicas para la educion de necesidades (casos de uso, prototipos, encuestas)		Formalidad: Inferir los requisitos resultantes de las decisiones de diseño con base en modelo de arquitectura o modelo de diseño				Formalidad: habilitar cuestionarios, entrevistas y escenarios para analizar y cuantificar la funcionalidad requerida por los usuarios finales			Explorar la adecuación y la completitud de los requisitos desa-rollando representaciones del producto (p. ej., prototipos, si-mulaciones, modelos, escenarios, guiones gráficos) y obteniendo realimentación sobre ellos de las partes interesadas relevantes.

9. ANEXO 2 - Codificación de las directrices ágiles

CARACTERÍSTICA	VALOR	SP1.1 EDUCIR LAS NECESIDADES DEL CLIENTE	SP1.2 TRANSFORMAR LAS NECESIDADES DE LAS PARTES INTERESADAS EN REQUISITOS DE CLIENTE	SP2.1 ESTABLECER LOS REQUISITOS DEL PRODUCTO Y COMPONENTES DEL PRODUCTO	SP2.2 ASIGNAR LOS REQUISITOS DE COMPONENTE DE PRODUCTO	SP2.3 IDENTIFICAR LOS REQUISITOS DE INTERFAZ	SP3.1 ESTABLECER LOS CONCEPTOS Y LOS ESCENARIOS DE OPERACIÓN	SP3.2 ESTABLECER UNA DEFINICIÓN DE FUNCIONALIDAD Y DE LOS ATRIBUTOS DE CALIDAD REQUERIDOS	SP3.3 ANALIZAR LOS REQUISITOS	SP3.4 ANALIZAR LOS REQUISITOS PARA CONSEGUIR UN EQUILIBRIO	SP3.5 VALIDAR LOS REQUISITOS
Medidor del progreso (MP)	Software funcionando (SF)		SP1.2 MP-SF {FOR}			SP 2.3 MP-SF {FOR}		SP 3.2 MP-DE {GRA}			
	Documentación e informes parciales (DE)		SP1.2 MP-DE {GRA}	SP 2.1 MP-DE {GRA}			SP 3.1 MP-DE {GRA}	SP 3.2 MP-SF {FOR}			
	Indicadores o métricas de avance del plan (IM)		SP1.2 MP-IM {FOR}	SP 2.1 MP-IM {FOR}					SP 3.3 MP-IM {FOR}	SP 3.4 MP-IM {FOR}	SP 3.5 MP-IM {FOR}
Estructura organizacional (EO)	organización funcional (F)		SP1.2 EO-F {FOR}	SP 2.1 EO-F {FOR}				SP 3.2 EO-F {FOR}	SP 3.3 EO-F {FOR}		
	organización por proyecto (P)		SP1.2 EO-P {FOR}	SP 2.1 EO-P {FOR}				SP 3.2 EO-P {FOR}		SP 3.4 EO-P {FOR}	
	organización matricial (funcional, proyecto) (M)							SP 3.3 EO-M {FOR}			
Enfoque de Calidad (EC)	Al producto (PRD)			SP 2.1 EC-PRD {FOR}	SP 2.2 EC-PRD {FOR}	SP 2.3 EC-PRD {FOR}	SP 3.1 EC-PRC {FOR}	SP 3.2 EC-PRD {FOR}		SP 3.4 EC-PRD {FOR}	SP 3.5 EC-PRC {FOR}
	Al proceso (PRC)		SP1.2 EC-PRC {FOR}	SP 2.1 EC-PRC {FOR}	SP 2.2 EC-PRC {FOR}	SP 2.3 EC-PRC {FOR}	SP 3.1 EC-PRD {FOR}	SP 3.2 EC-PRC {FOR}	SP 3.3 EC-PRC {FOR}	SP 3.4 EC-PRC {FOR}	SP 3.5 EC-PRD {FOR}
	Al Cliente (CLI)	SP1.1 EC-CLI {FOR}	SP1.2 EC-CLI {FOR}				SP 3.1 EC-CLI {FRE}	SP 3.2 EC-PRC {FOR}	SP 3.3 EC-CLI {FRE}		SP 3.5 EC-CLI {FOR}
Aceptación de Cambios (AC)	Al inicio proyecto (IP)		SP1.2 AC-IP {FOR}	SP 2.1 AC-IP {FOR}			SP 3.1 AC-IP {FOR}		SP 3.3 AC-IP {FOR-FRE}		SP 3.5 AC-IP {FOR}
	Al final de cada iteración (FI)	SP 1.1 AC-FI {FOR}	SP 1.2 AC-FI {FOR}	SP 2.1 AC-FI {FOR}			SP 3.1 AC-FI {FOR}		SP 3.3 AC-FI {FOR}		SP 3.5 AC-FI {FOR}
	En cualquier momento (CM)			SP 2.1 AC-CM {FRE,FRE}					SP 3.3 AC-CM {FOR}		
Comunicación orientada a ser (CO)	Oral (FaceToFace) (F2F)	SP 1.1 CO-F2T {FOR}						SP 3.2 CO-F2F {FOR}	SP 3.3 CO-F2F {FOR}	SP 3.4 CO-F2F {FOR}	
	Escrita (ESC)	SP 1-1 CO-ESC {FOR}		SP 2.1 CO-ESC {FOR}				SP 3.2 CO-ESC {FOR}			SP 3.5 CO-ESC {FOR}

10. ANEXO 3. Encuesta usada en la validación de la propuesta.

**INSTRUMENTO PARA LA GENERACIÓN DEL PROCESO DE DESARROLLO
DE REQUERIMIENTOS DE SOFTWARE PARA MICRO Y PEQUEÑAS
EMPRESAS.**

ROL DEL INTEGRANTE _____

- 1- ¿Considera que el propósito del instrumento es de interés para su organización y similares? Si _____ No _____

Indique las razones: _____

- 2- ¿Considera que el enfoque planteado para el instrumento CMMI-Dev 1.3 – AGIL, es pertinente para solucionar el problema planteado?

Si _____ No _____

Indique las razones: _____

- 3- ¿Está de acuerdo con el modelo de procesos arrojado por el instrumento, con base en las características de su organización?

Si _____ No _____

Indique las razones: _____

4- ¿Considera que los argumentos teóricos son suficientes para soportar la propuesta?

Si____ No____

Indique las razones:_____

5- ¿Se deben realizar modificaciones al instrumento planteado? Si____ No____

Si su respuesta es sí, por favor indique sus sugerencias:_____

6- Plantee las observaciones y/o comentarios que considere pertinente:_____
