

**Universidad Internacional de La Rioja**  
**Máster Universitario en Dirección de Operaciones y**  
**Calidad**

---

# Propuesta de mejora a las prácticas de Ingeniería de Software en SOFTNICHE Corp.

---

Proyecto fin de máster presentado por: Juan Gabriel Mejía Villamizar

Director/a: María del Socorro Sánchez Aparicio

Ciudad: Bogotá, D.C., Colombia.

Fecha: Enero de 2020

Firmado por: Juan Gabriel Mejía Villamizar

CATEGORÍA TESAURO: TFM, INGENIERÍA DE SOFTWARE, MEJORA DE  
PROCESOS

## RESUMEN

El presente trabajo versa sobre la mejora del proceso de ingeniería de software en Softniche Corp., empresa de desarrollo de software para un nicho específico del sector financiero.

Aunque la empresa es casi un monopolio regional en el sector, con un producto *legacy*, los retos actuales de la globalización y las últimas revoluciones industriales exigen una mayor optimización operacional que involucre calidad y oportunidad. Adicionalmente, acorde con la demanda del mercado de nuevas tecnologías, la empresa ha emprendido la construcción de una nueva versión de su producto.

Para dar una respuesta integral a estas coyunturas, se ha propuesto una mejora metodológica sobre el proceso de ingeniería de software, a partir de un diagnóstico inicial sobre el uso de las mejores prácticas de la industria. Lo anterior, siguiendo las líneas generales del Ciclo de Mejora de Procesos (CMP)<sup>1</sup>, y utilizando CMMI-DEV<sup>2</sup> y RUP<sup>3</sup> como marcos de mejores prácticas en ingeniería de software.

## PALABRAS CLAVE

CMP, Mejora de Procesos, CMMI-DEV, Ingeniería de Software, Calidad de Software.

---

<sup>1</sup> De acuerdo con Berenguer y Ramos-Yzquierdo (2004).

<sup>2</sup> Capability Maturity Model Integration (CMMI) for Development (DEV). Marco de referencia para la mejora de procesos de desarrollo y mantenimiento de Software, del Software Engineering Institute (SEI) de la Universidad de Carnegie Mellon.

<sup>3</sup> Rational Unified Process (RUP). Metodología integral de desarrollo de software con filosofía Orientada a Objetos (O-O), de IBM.

## **SUMMARY**

The present work deals with the improvement of the software engineering process at Softniche Corp., a software development company for a specific niche of the financial sector.

Although the company is almost a regional monopoly in the sector, with a legacy product, the current challenges of globalization and the latest industrial revolutions, require a greater operational optimization that involves quality and opportunity. Additionally, according to the market demand for new technologies, the company has undertaken the development of a new version of its product.

In order to give a comprehensive respond to these situations, a new methodological improvement on the software engineering process has been proposed, based on an initial diagnosis about the use of industry best practices. This has been accomplished following the general lines of the CMP (Process Improvement Cycle) and using CMMI-DEV and RUP as the best practice's frameworks for software engineering.

## **KEYWORDS**

CMP, Process Improvement, CMMI-DEV, Software Engineering, Software Quality.

## CONTENIDO

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>1</b>
1.1	EL CONTEXTO DE LA INDUSTRIA EN EL SIGLO XXI .....	1
1.2	EL DESARROLLO DE SOFTWARE COMO INDUSTRIA .....	2
1.3	SOFTNICHE CORP. EN LA INDUSTRIA DE SOFTWARE.....	3
1.4	JUSTIFICACIÓN DEL TFM .....	3
1.5	OBJETIVO DEL TRABAJO .....	4
<b>2</b>	<b>DESCRIPCIÓN DE LA SITUACIÓN DE PARTIDA Y PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>5</b>
2.1	SITUACIÓN ACTUAL .....	6
2.1.1	<i>Cifras de satisfacción, calidad y entrega en plazo.....</i>	<i>7</i>
2.1.2	<i>El precio de la calidad y la oportunidad .....</i>	<i>8</i>
2.1.3	<i>Las iniciativas de cambio.....</i>	<i>9</i>
2.2	FASE 1 – SENSIBILIZAR.....	9
2.2.1	<i>Compromiso con el BP y el seguimiento.....</i>	<i>12</i>
2.3	FASE 2 – IDENTIFICAR.....	13
2.4	FASE 3 – PLANIFICAR .....	15
2.5	FASE 4 – ANALIZAR.....	17
2.5.1	<i>El marco de Gestión de Proyectos .....</i>	<i>18</i>
2.5.2	<i>El marco de Gestión de Servicios .....</i>	<i>20</i>
2.5.3	<i>Diagnóstico del proceso actual de ingeniería de software (Assessment).....</i>	<i>21</i>
2.5.4	<i>Dashboard del proceso y medición del proceso AS-IS .....</i>	<i>29</i>
<b>3</b>	<b>MARCO TEÓRICO .....</b>	<b>31</b>
3.1	CMP – CICLO DE MEJORA DE PROCESOS DE NEGOCIO .....	31
3.2	RELACIÓN ENTRE CALIDAD DEL PRODUCTO Y DEL PROCESO .....	32
3.3	EL MODELO CMMI-DEV (CAPABILITY MATURITY MODEL INTEGRATION FOR DEVELOPMENT) .....	34
3.4	EL MARCO RUP (RATIONAL UNIFIED PROCESS) .....	37
<b>4</b>	<b>SOLUCIÓN PROPUESTA .....</b>	<b>39</b>
4.1	FASE 5 – DISEÑAR.....	40
4.1.1	<i>Alineación entre requerimientos y expectativas .....</i>	<i>41</i>
4.1.2	<i>Diseño de los subprocesos del proceso TO-BE.....</i>	<i>42</i>
4.1.3	<i>Integración con los marcos de Gestión de Servicios y de Proyectos.....</i>	<i>49</i>

4.2	FASE 6 – IMPLANTAR.....	50
4.3	FASE 7 – CONTROLAR.....	51
4.4	FASE 8 – CAPACITAR.....	52
<b>5</b>	<b>ANÁLISIS COSTO-BENEFICIO DE LA MEJORA.....</b>	<b>53</b>
<b>6</b>	<b>CONCLUSIONES.....</b>	<b>57</b>
<b>7</b>	<b>BIBLIOGRAFÍA.....</b>	<b>60</b>
<b>8</b>	<b>ANEXOS.....</b>	<b>63</b>
8.1	ANEXO 1 – GRAFICOS Y ANÁLISIS SOBRE EL CUMPLIMIENTO DE LAS PRÁCTICAS ESPECÍFICAS DE CMMI-DEV (CAPACIDAD DE AREAS DE PROCESO).....	63
8.2	ANEXO 2 – ROLES DE LAS PRÁCTICAS DE INGENIERÍA DE SOFTWARE.....	65
8.3	ANEXO 3 – EJEMPLOS DE INTEGRACIÓN DE ACTIVIDADES ENTRE MARCOS METODOLÓGICOS .....	69

## INDICE DE FIGURAS

FIGURA 1	REVOLUCIONES INDUSTRIALES.....	2
FIGURA 2	EL MODELO CMP® DE LA UNIVERSIDAD DE NAVARRA – CUATRO (4) PRIMERAS FASES.....	5
FIGURA 3	MAPA DE PROCESOS DE SOFTNICHE CORP. ....	14
FIGURA 4	SUBPROCESOS Y DUEÑOS DE PROCESO DE LOS PROCESOS DE GESTIÓN DE SERVICIOS Y GESTIÓN DE PROYECTOS .....	14
FIGURA 5	FASE DE PLANIFICACIÓN EN EL MARCO DE PROYECTOS: SUBPROCESO DE REQM (REQUIREMENT MANAGEMENT) .....	18
FIGURA 6	FASE DE PERSONALIZACIÓN EN EL MARCO DE PROYECTOS: SUBPROCESOS DE DESARROLLO (DVLP) Y PRUEBAS (TST).....	19
FIGURA 7	FASE DE UAT EN EL MARCO DE PROYECTOS: SUBPROCESO DE PRUEBAS DE UAT.....	19
FIGURA 8	SUBPROCESO DE MEJORAS.....	20
FIGURA 9	TEMAS DE MEJORA PARA OTRAS CAUSAS DE LA FALTA DE CALIDAD Y DE OPORTUNIDAD DE ENTREGA.....	24
FIGURA 10	DIAGRAMA DE PARETO, IDENTIFICANDO LAS OPORTUNIDADES DE MEJORA MENCIONADAS POR EL 80% DE LOS ENCUESTADOS. ....	25
FIGURA 11	EVALUACIÓN DE LAS ACTIVIDADES REALIZADAS POR LAS PERSONAS DE ACUERDO CON SUS ROLES, PARA LAS OPERACIONES DE PROYECTOS Y SERVICIOS.....	27
FIGURA 12	DESEMPEÑO HIPOTÉTICO VS. REAL DE LOS ROLES DE INGENIERÍA EN SOFTNICHE CORP. ....	28
FIGURA 13	CMP – CICLO DE MEJORA DE PROCESOS DE NEGOCIO .....	32
FIGURA 14	FOTO DE WATTS HUMPHREY, FUNDADOR DEL SEI (SOFTWARE ENGINEERING INSTITUTE).....	33
FIGURA 15	NIVELES DE MADUREZ Y DE CAPACIDAD EN CMMI. ....	35
FIGURA 16	REPRESENTACIÓN ESCALONADA Y CONTINUA DE CMMI.....	35
FIGURA 17	FOTOS: GRADY BOOCH, IVAR JACOBSON Y JAMES JACOBSON.....	37
FIGURA 18	RUP: ESQUEMA DE FASES Y DISCIPLINAS. ....	38

FIGURA 19 FIGURA 2 EL MODELO CMP® DE LA UNIVERSIDAD DE NAVARRA – CUATRO (4) ÚLTIMAS FASES.....	40
FIGURA 20 NUEVO MAPA DE PROCESOS DE SOFTNICHE CORP. ....	43
FIGURA 21 CICLO DE VIDA DE DESARROLLO DE SOFTWARE - MODELO EN CASCADA.....	44
FIGURA 22 SUBPROCESO DE ELABORACIÓN DE REQUERIMIENTOS.....	45
FIGURA 23 SUBPROCESO DE DISEÑO Y DESARROLLO .....	46
FIGURA 24 SUBPROCESO DE PRUEBAS.....	47
FIGURA 25 SUBPROCESO DE GESTIÓN DE ENTREGAS.....	48
FIGURA 26 MODELO DE RAMIFICACIONES CON GITFLOW .....	49
FIGURA 27 INTEGRACIÓN DE LOS MARCOS METODOLÓGICOS SOFTNICHE CORP.....	50
FIGURA 28 PRÁCTICAS DE INGENIERÍA VS. INDICADORES DE CONTROL.....	52
FIGURA 29 PORCENTAJE DE CUMPLIMIENTO DE LAS PRÁCTICAS ESPECÍFICAS DE CMMI-DEV EN MANTENIMIENTO DE MEJORAS....	63
FIGURA 30 PORCENTAJE DE CUMPLIMIENTO DE LAS PRÁCTICAS ESPECÍFICAS DE CMMI-DEV EN MANTENIMIENTO DE PROBLEMAS	64
FIGURA 31 PORCENTAJE DE CUMPLIMIENTO DE LAS PRÁCTICAS ESPECÍFICAS DE CMMI-DEV EN MANTENIMIENTO CORRECTIVO DE INCIDENTES .....	65

## INDICE DE TABLAS

TABLA 1 INDICADORES DE PROYECTOS 2018-2019 .....	7
TABLA 2 INDICADORES DE SERVICIOS DE MANTENIMIENTO – EVOLUTIVOS. 2018-2019 .....	8
TABLA 3 RESULTADOS DE LA ENCUESTA DE SATISFACCIÓN 2018-2019.....	8
TABLA 4 TABLA FODA – FORTALEZAS, OPORTUNIDADES, DEBILIDADES Y AMENAZAS .....	10
TABLA 5 PROJECT CHARTER PARA LA MEJORA DEL NUEVO PROCESO .....	15
TABLA 6 TOP-3 OPORTUNIDADES DE MEJORA EN SOFTNICHE CORP. (OTRAS CAUSAS). ....	25
TABLA 7 DESCRIPCIÓN DETALLADA DE LAS OPORTUNIDADES DE MEJORA POR TEMA .....	26
TABLA 8 INDICADORES Y MEDICIONES DEL PROCESO AS-IS. ....	29
TABLA 9 INDICADORES NO MEDIDOS DEL PROCESO AS-IS.....	30
TABLA 10 LAS ÁREAS DE PROCESO (PROCESS AREAS) DE CMMI. ....	36
TABLA 11 ALINEACIÓN REQUERIMIENTOS VS. EXPECTATIVAS.....	41
TABLA 12 ROLES DE INGENIERÍA DE SOFTNICHE CORP. VS. ROLES DE RUP.....	44
TABLA 13 ACTIVIDADES PARA LA IMPLANTACIÓN DE LAS PRÁCTICAS DE INGENIERÍA.....	50
TABLA 14 INDICADORES DE CONTROL DEL PROCESO TO-BE .....	51
TABLA 15 COSTES DE LA NO-CALIDAD 2018-2019 (VALORES EN DÓLARES) .....	54
TABLA 16 COSTES DE LA IMPLANTACIÓN DEL PROCESO DE MEJORA (VALORES EN DÓLARES) .....	55
TABLA 17 OBJETIVOS DE LA NO-CALIDAD PARA 2021 (CIFRAS EN DÓLARES) .....	56
TABLA 18 EVOLUCIÓN DE LOS COSTES POR AÑO (CIFRAS EN DÓLARES).....	56

# 1 INTRODUCCIÓN

El presente trabajo se enfoca en la mejora de los procesos y prácticas de ingeniería de software en una empresa dedicada a la producción y el mantenimiento de sistemas de información para un nicho particular del mercado, especializado y regulado.

Por razones de confidencialidad, el nombre de la empresa y algunos detalles han sido cambiados, sin embargo, se ha evitado distorsionar la objetividad de la propuesta de mejora.

La propuesta es principalmente metodológica y de procesos, sin excluir las sugerencias puntuales que en materia técnica convenga hacer especialmente sobre las herramientas de apoyo.

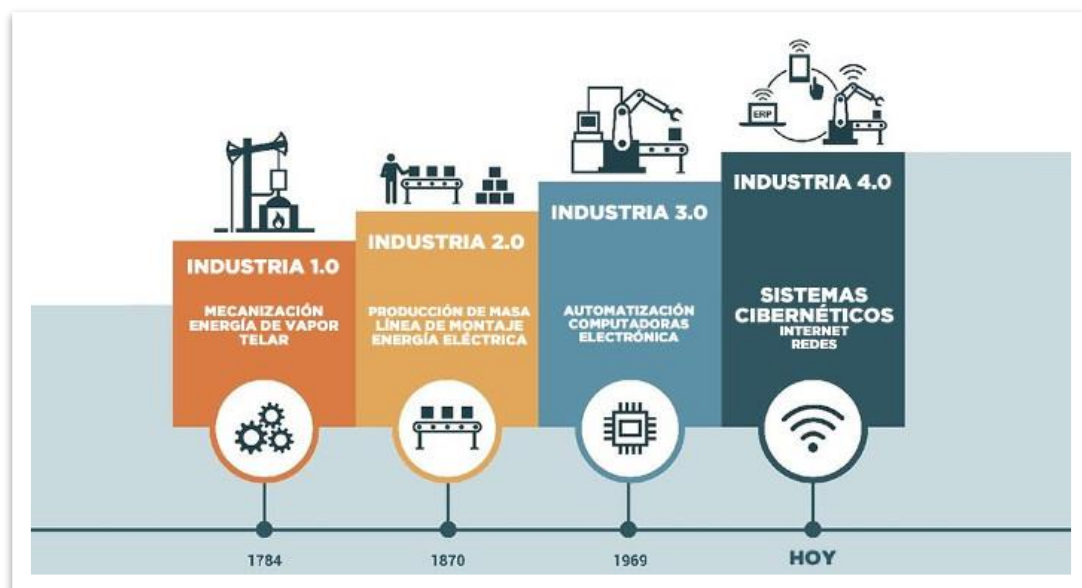
El proyecto de mejora aplica solamente sobre la unidad de Colombia de Softniche Corp.

## 1.1 EL CONTEXTO DE LA INDUSTRIA EN EL SIGLO XXI

La situación por resolver y la mejora propuesta se enmarcan en el contexto de la industria actual que, en su vertiginosa evolución, plantea enormes retos a las empresas hacia una mayor competitividad y por tanto sostenibilidad. Diversos eventos en la historia de la humanidad han sido determinantes en las transformaciones industriales y aquellos de índole tecnológico han apalancado grandes transformaciones en la manera de hacer las cosas.

Los principales impactos para la industria surgen desde fines del siglo XVIII en Gran Bretaña con la invención de la máquina de vapor y la automatización de los telares en la industria textil (primera revolución industrial). A inicios de 1900, el ingeniero y economista Frederick Winslow Taylor plantea “la gestión científica del trabajo” influenciando definitivamente la producción en masa liderada por Henry Ford, desde la primera década del siglo XX (segunda revolución industrial). La tercera revolución industrial, desde la visión de Jeremy Rifkin, viene impulsada por los avances científico-tecnológicos en materia de hardware y software que empujan hacia una posible desaparición del trabajo como lo conocemos actualmente y que, en consecuencia, estará basada en el conocimiento (Lastra Lastra, 2017). Finalmente, hay quienes hablan de una cuarta revolución industrial o Industria 4.0, que agrupa diversas tecnologías y metodologías de la revolución tecnológica como el *big data*, la inteligencia artificial, las nubes de datos, el *machine learning*, el agilismo, impresión 3D, etc., y que marchan agresivamente hacia una nueva configuración de la industria y de la sociedad.

Figura 1 Revoluciones Industriales



Fuente: <http://wonderwarenorthmex.com/industria-4-0/> (Sep./2019)

## 1.2 EL DESARROLLO DE SOFTWARE COMO INDUSTRIA

En el contexto descrito en el anterior numeral, la industria del software sufre su propia revolución. Si bien por una parte es posible hablar de una producción industrial de software, este concepto es bien distante del mismo concepto en relación con la producción de elementos tangibles. El software es un producto intangible, su materia prima son los requerimientos del cliente (ideas de sus necesidades) transformados en composiciones estructuradas en un lenguaje formal (lógico), lo cual se consigue a través de un proceso intelectual que realizan personas entrenadas en estos procesos abstractivos. Dadas estas características, es bastante complejo automatizar el proceso de construcción del software de principio a fin a partir de los requerimientos del usuario en lenguaje natural, por lo que se le ha considerado un proceso artesanal en el sentido de no ser intensivo en automatización, no obstante estar sustentado en bases teóricas, procesos y metodologías sólidas. Otra particularidad del software es que producir una unidad del intangible cuesta lo mismo que producir N unidades (sin importar lo grande que sea N), lo cual se traduce en que el coste marginal de la producción sea cero.

Los conceptos y tecnologías de las últimas dos revoluciones industriales también han contribuido a que los procesos de la industria del software sean más rápidos, estandarizados y confiables. En cuanto a la optimización y la calidad de los procesos, han sido propuestas diversas soluciones a través de paradigmas de desarrollo de software (Estructurado, Orientado a Objetos, Funcional), metodologías de desarrollo (RUP), metodologías de gestión



(SCRUM), marcos de mejora (CMMI), estándares de calidad (ISO/IEC 9001, ISO/IEC 25000<sup>4</sup>), estándares de seguridad (ISO/IEC 27000), entre otros. El conjunto de herramientas de apoyo al ciclo de vida del desarrollo del software también ha aumentado considerablemente: integradores y servidores de automatización (TFS, Jenkins), herramientas CASE<sup>5</sup> (*Enterprise Architect*), soporte a pruebas (*Testlink*, *Jmeter*, *Sonar*, *Veracode*, *Selenium*), IDEs<sup>6</sup> (Eclipse, *Visual Studio*), por mencionar sólo algunas.

### 1.3 SOFTNICHE CORP. EN LA INDUSTRIA DE SOFTWARE

Softniche Corp. es una empresa multinacional de desarrollo de software y de servicios de soluciones informáticas para un nicho de mercado particular. Cuenta con un único producto de software que soporta integralmente el core del negocio de sus clientes. No es un producto de “caja”, sino un producto complejo que requiere soporte, instalación, configuración y parametrización por parte de expertos.

Dentro del catálogo de servicios se ofrecen proyectos de instalación, personalización e implantación del sistema, migración de datos entre diversas plataformas de bases de datos, mantenimiento correctivo (solución de defectos) y mantenimiento evolutivo (desarrollos e implementaciones adicionales).

La versión original del software fue desarrollada en Visual Basic (VB), un lenguaje ya prácticamente obsoleto, y se encuentra instalada en todos los clientes. Actualmente la empresa está construyendo una nueva versión en lenguaje C#, un lenguaje actual, orientado a objetos y con nueva arquitectura de tecnología *web*, para reemplazar la versión Cliente/Servidor en lenguaje VB.

### 1.4 JUSTIFICACIÓN DEL TFM

Aunque la empresa es gestionada por procesos, certificada en ISO/IEC 9001, y los servicios se soportan en las mejores prácticas de ITIL y la ISO/IEC 20000, los clientes expresan insatisfacción en relación con la calidad del software y el incumplimiento en los plazos de entrega pactados. Hay frecuentes devoluciones (errores encontrados en la etapa de pruebas del Cliente), y retrasos en las fechas de entrega acordadas.

---

<sup>4</sup> En reemplazo y complementación de la ISO/IEC 9126.

<sup>5</sup> Computer Aided Software Engineering.

<sup>6</sup> Integrated Development Environment.

Debido a los cambios en la arquitectura del software y al aumento en la complejidad del mismo, se requieren mayores controles, más elaboradas definiciones de arquitectura, diseños funcionales y técnicos, así como un mayor esfuerzo para las pruebas internas. Por otra parte, las prácticas de ingeniería del ciclo de vida del software se encuentran desactualizadas frente a los nuevos paradigmas de la industria y diseminadas en dos de los procesos misionales de la organización: el proceso de Gestión de Servicios (Incidentes, Problemas, Mejoras) y el de Gestión de Proyectos.

Todo lo anterior justifica la propuesta de una mejora que apunte a definir un nuevo proceso unificado de ingeniería de software, que involucre las mejores prácticas de la industria para la ingeniería de software (CMMI, O-O<sup>7</sup>, RUP) y que se integre con los procesos existentes (Servicios y Proyectos).

## **1.5 OBJETIVO DEL TRABAJO**

### **Objetivo General**

Realizar una propuesta de mejora metodológica sobre el proceso de desarrollo e ingeniería de software en Softniche Corp., a partir de un diagnóstico inicial sobre el uso de las mejores prácticas en los proyectos y servicios de evolución y mantenimiento del producto de software. Lo anterior, siguiendo las líneas generales del Ciclo de Mejora de Procesos (CMP), y utilizando CMMI-DEV y RUP como marcos de mejores prácticas en ingeniería de software.

### **Objetivos específicos**

- Analizar la situación actual de la empresa y del proceso de ingeniería frente a la calidad.
- Proponer una mejora del proceso de ingeniería revisado contra las mejores prácticas de la industria.
- Proponer la manera de integrar el nuevo proceso de ingeniería con los procesos misionales de Servicios y Proyectos.
- Ofrecer un marco de medición y control inicial en relación con la calidad de los entregables de software y la oportunidad de entrega sobre los plazos convenidos.

### **Indicadores**

Los objetivos específicos señalan los indicadores que se deben medir para alcanzar de manera cuantificable el objetivo general:

---

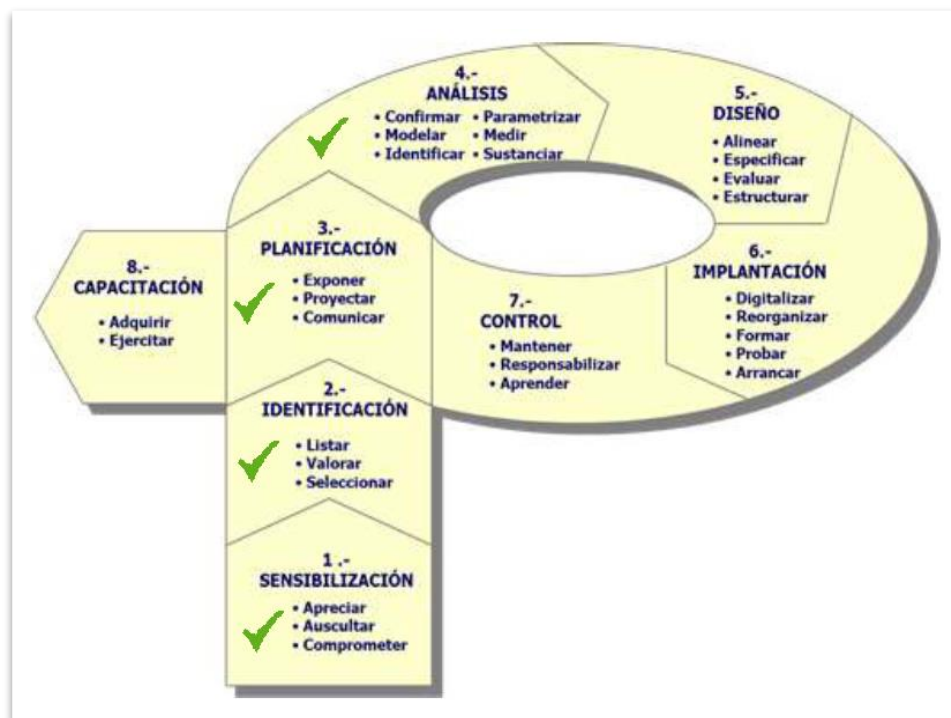
<sup>7</sup> Paradigma de programación orientada a objetos.

- Calidad del proceso: Adherencia al proceso de ingeniería en servicios y proyectos entregados.
- Calidad del software: una medición sobre los defectos y las devoluciones en relación con el tamaño del software.
- Entregas oportunas: cantidad de entregas a tiempo.

## 2 DESCRIPCIÓN DE LA SITUACIÓN DE PARTIDA Y PLANTEAMIENTO DEL PROBLEMA

En el primer capítulo se presentó de manera general la problemática y la justificación para este trabajo, en este capítulo se hará un análisis más profundo de la situación actual y se identificará el problema que se quiere resolver. Así mismo, se presentarán, a nivel general, los principales aspectos de las cuatro (4) primeras fases del modelo CMP, desde la Sensibilización hasta el Análisis, como se indica en la siguiente imagen con el signo ✓:

*Figura 2 El modelo CMP® de la Universidad de Navarra – Cuatro (4) primeras fases*



*Fuente: Elaboración propia con base en Berenguer J.M. y Ramos-Yzquierdo J.A. (2008, p.17)*

En el capítulo 4 se desarrollarán las últimas cuatro (4) fases del CMP.

## 2.1 SITUACIÓN ACTUAL

Softniche Corp. ha trabajado por casi medio siglo en el desarrollo y evolución de su producto Niche-V1<sup>8</sup>, realizando implementaciones personalizadas de su solución en múltiples empresas de su nicho financiero en toda la región. Se ha constituido como la empresa líder en su campo, con su producto que ha llegado a ser el software core del negocio para sus clientes.

Niche-V1, un producto nacido a inicios de los años 1980s, fue construido en un lenguaje popular en ese tiempo (Microsoft Visual Basic) y sobre una arquitectura novedosa para la época (Cliente/Servidor). En aquel momento la *web* no existía, la programación orientada a objetos era una tecnología que apenas estaba siendo desarrollada, las redes y las bases de datos distribuidas eran temas ya importantes, pero aún no populares en todos los ámbitos.

Impulsada por el gran crecimiento del sector y, al mismo tiempo, apalancando tecnológicamente a sus clientes, Sofniche Corp. siguió desplegando exitosamente su producto por varias décadas, ofreciendo una personalización de su software acorde con las necesidades particulares de cada cliente.

Hace algunos años la compañía emprendió la tarea de construir una nueva versión de su producto, Niche-V2<sup>9</sup>, lo que supone una fuerte actualización tecnológica, tanto internamente como frente a sus clientes. Con la nueva versión se pretende ofrecer el recambio de su producto *legacy* al nuevo producto de arquitectura *web*.

En relación con los procesos, hace pocos años Softniche Corp., en su unidad de Colombia, comenzó a organizar su operación por medio de la implementación de una filosofía de mejores prácticas de servicios conforme al marco ITIL v3. Poco después se certificó en ISO/IEC 9001 y año tras año, hasta el presente, ha seguido recibiendo las correspondientes auditorías. Sin embargo, los procesos de ingeniería de software no eran explícitos. Por una parte, existía una definición de un marco de Proyectos acorde con PMI y con prácticas de ingeniería fundadas en CMMI-DEV y, por otro, la definición de los procesos de servicios soportados en ITIL, con algunas prácticas de ingeniería.

Las prácticas de ingeniería de software no parecían ser las mejores. Las encuestas de satisfacción realizadas a los clientes arrojaban resultados no muy alentadores; los clientes se quejaban principalmente del gran volumen de defectos encontrados durante sus pruebas de

---

<sup>8</sup> Por razones de confidencialidad, el nombre de la empresa y del producto han sido cambiados.

<sup>9</sup> Actualmente sigue aún en desarrollo.

usuario una vez entregado el desarrollo<sup>10</sup>. Por otra parte, también manifestaban su insatisfacción debido a los frecuentes desvíos en los plazos de las entregas, especialmente en los proyectos. A tal punto ha llegado la situación que algunos clientes han tomado la decisión de no contar más con los servicios de Softniche Corp. en los siguientes meses.

### 2.1.1 Cifras de satisfacción, calidad y entrega en plazo

En los proyectos, se evidencia un porcentaje de adherencia a los procesos metodológicos por debajo de lo esperado, y al mismo tiempo porcentajes muy bajos de cumplimiento de los plazos de entrega final (Tabla 1). Desafortunadamente no existe actualmente una medición para los defectos encontrados durante las pruebas de los usuarios, que permitan tener una medida de la calidad<sup>11</sup>.

*Tabla 1 Indicadores de Proyectos 2018-2019*

Indicadores de Proyectos	2018	2019 Q1,Q2
% Adherencia a los procesos ( $\geq 85\%$ )	70%	79%
% De cumplimiento en la fecha de cierre ( $\geq 100\%$ )	34%	30%
* No se cuenta con una medición de la calidad para defectos del software en proyectos.		

*Fuente: Elaboración propia, 2019. A partir de datos de Softniche Corp.*

En cuanto a los servicios de mantenimiento, en particular de evolutivos, si bien el porcentaje de cumplimiento de las entregas y el porcentaje de calidad han estado dentro de los rangos esperados (Tabla 2), la apreciación del cliente no corresponde con los resultados de estas medidas. Para explicar esto se pueden dar varias razones; por una parte, es posible que algunos de los servicios más críticos hayan estado fuera de los rangos de las mediciones, y esto no es posible verlo en un promedio anual o semestral; por otra parte, en cuando a la calidad, en adelante convendría comenzar a disminuir el tope máximo del índice de calidad para exigir un mayor nivel, o bien resultaría útil aplicar una métrica más idónea como la calidad de defectos sobre el tamaño del software.

---

<sup>10</sup> A estos defectos se les llama “devoluciones” ya que el desarrollo debe ser devuelto a los desarrolladores para su corrección, pruebas internas y nuevo envío a pruebas de usuario.

<sup>11</sup> Sólo hasta el cuarto trimestre de 2019 se tiene planeado implementar indicadores de calidad para los proyectos.

Tabla 2 Indicadores de Servicios de Mantenimiento – Evolutivos. 2018-2019

Indicadores de Servicios de Mantenimiento Evolutivos	2018	2019 Q1,Q2
% De cumplimiento en la entrega ( $\geq 70\%$ )	91%	86%
% Índice de calidad ( $\leq 15\%$ ) (Horas de corrección de Devoluciones / Horas de Desarrollo total)	15%	7%

Fuente: Elaboración propia, 2019. A partir de datos de Softniche Corp.

A pesar de que los datos presentados no son concluyentes en relación con la calidad y la entrega a tiempo, es evidente que los clientes no están conformes con la calidad del producto y muestran una insatisfacción que se refleja en el porcentaje de satisfacción general (Tabla 3). Obsérvese igualmente que los rangos de tolerancia especificados actualmente son muy permisivos. Por ejemplo, se tolera (sin tener que tomar acciones de mejora), un porcentaje de satisfacción del cliente superior al 50%, como por ejemplo el 58% del año 2018. Lo mismo se puede decir para el porcentaje de percepción de calidad del producto, el cual se considera dentro de los límites aceptados si supera un 50%, como el 54% del año 2018, o el 66% del 2019.

Tabla 3 Resultados de la encuesta de satisfacción 2018-2019

ENCUESTA DE SATISFACCIÓN	2018	2019 Q1,Q2
% Percepción de Calidad del Producto ( $80\% \geq x \geq 50\%$ )	54%	66%
% Satisfacción ( $80\% \geq x \geq 50\%$ )	58%	65%

Fuente: Elaboración propia, 2019. A partir de datos de Softniche Corp.

## 2.1.2 El precio de la calidad y la oportunidad

Como se ha mencionado anteriormente, la insatisfacción de algunos clientes los ha llevado a buscar otros horizontes por fuera de Softniche Corp. Es así como en 2019 un par de clientes han notificado que no continuarán con los servicios de la compañía después de este año. Este hecho impactará los ingresos de la compañía en el futuro, reduciéndolos en un porcentaje aproximado entre el 5% y el 9%, conforme a su comportamiento respectivo en 2018 y 2019 Q1, Q2.

Algunos otros clientes han puesto en sobre aviso a la compañía en el sentido de que, si no se mejora la calidad del producto y los plazos en las entregas de los proyectos, habrá que incurrir en multas por incumplimientos de plazos o devoluciones de dinero por la no-calidad. A este

impacto hay que sumarle los efectos negativos que se producen cuando los clientes se comunican entre sí para quejarse de Softniche Corp., deteriorando la imagen y disminuyendo la confianza en la empresa.

El precio de la falta de calidad y oportunidad está costando cada vez más, especialmente cuando los riesgos se materializan. Además de las pérdidas por el abandono de algunos clientes, deben considerarse los costos de los reprocesos en los servicios de mantenimiento evolutivo, en particular, por corrección de las devoluciones, cuyo esfuerzo en 2018 fue del 15% (del esfuerzo total del mantenimiento), representando una menor utilidad en dichas operaciones; en 2019 el mismo indicador fue un poco más bajo, pero aún considerable, 7%.

Por otra parte, los retrasos en las entregas de los proyectos ponen en riesgo la viabilidad de estos, aumentan el riesgo de acciones contractuales que afecten a la compañía, desgastan la moral del equipo y minan la confianza del cliente. Es difícil cuantificar todos estos factores, pero se puede evidenciar a partir de datos tomados de Softniche Corp., que el impacto económico ha sido negativo y considerable, con un 15% de pérdida en 2018 y un 50% en lo que va corrido de 2019 (Q1, Q2), generando consecuentemente pérdidas (en 2018) y bajas utilidades (en 2019) para el conjunto de la Compañía.

### **2.1.3 Las iniciativas de cambio**

Los problemas que hay que atacar son variados y posiblemente habrá que afrontarlos simultáneamente. La empresa viene trabajando en varias iniciativas de mejora de procesos relacionadas con el gobierno del producto, la calidad de los servicios y el mejoramiento del proceso de desarrollo de software, entre otras. Todo lo anterior con miras a subsanar los problemas identificados y entregar al cliente un producto y servicios de calidad en los plazos convenidos.

El foco de este TFM estará puesto sobre la mejora al proceso de ingeniería de software, y en los siguientes numerales se describirá el ejercicio alrededor de la sensibilización, identificación, planificación y análisis de la mejora a través de la metodología CMP.

## **2.2 FASE 1 – SENSIBILIZAR**

Todos los años, entre el tercero y cuarto trimestre del año, la compañía realiza el ejercicio de revisión y preparación del *Business Plan* (BP)<sup>12</sup> para el siguiente año.

---

<sup>12</sup> En adelante, en este documento, se utilizará la abreviatura BP, para referirse al *Business Plan*.

Una vista del ejercicio para el 2019 se muestra en la siguiente matriz FODA:

*Tabla 4 Tabla FODA – Fortalezas, Oportunidades, Debilidades y Amenazas*

FORTALEZAS	DEBILIDADES
<ul style="list-style-type: none"> <li>• Alto conocimiento del negocio, y solución personalizada a cada cliente.</li> <li>• Se cuenta con políticas y programas de seguridad y bienestar a los empleados.</li> <li>• Se ha iniciado un proceso hacia la consecución de la madurez organizacional, gestión por procesos y enfoque de mejora continua.</li> <li>• Presencia regional.</li> <li>• Certificación regional ISO/IEC 9001.</li> <li>• Uso y/o conocimiento de marcos internacionales solicitados por los clientes: ITIL, PMI, ISO/IEC 20.000, CMMI-DEV.</li> </ul>	<ul style="list-style-type: none"> <li>• Nueva versión del software incompleta y sin pruebas.</li> <li>• Falta formación y de inversión en capacitación de los empleados a todo nivel para hacer frente a los retos actuales.</li> <li>• Dificultad de reaccionar y adaptarse a los cambios del mercado.</li> <li>• Pérdida de la confianza de parte de los clientes por la falta de calidad.</li> <li>• Resistencia al cambio frente a nuevos procesos y estrategias de mejora.</li> <li>• Falta de madurez en ingeniería de software y arquitectura.</li> </ul>
OPORTUNIDADES	AMENAZAS
<ul style="list-style-type: none"> <li>• Los clientes demandan productos con mayor calidad y menor costo, basados en nuevas tecnologías (SaaS, Análisis BI, Mobile, pagos P2P, Big Data) y enfoque en procesos.</li> <li>• Los clientes piden nuevos modelos que apunten a la mejora de la calidad, requieren servicios de soporte y líneas de servicio 7x24x365, necesitan nuevos de modelos operativos (fábrica de software) y uso de nuevas metodologías (ágiles).</li> <li>• Hay alta posibilidad de que los clientes adopten la nueva versión Niche-V2.</li> <li>• Las nuevas regulaciones legales pueden generar mayor demanda de evolutivos del sistema.</li> </ul>	<ul style="list-style-type: none"> <li>• Hay por lo menos tres empresas fuertes en el mismo mercado, con un producto que compite.</li> <li>• Aparición de pequeños productos, con nuevas tecnologías, fácil integración y configuración, que compiten en servicios con Niche-V1 y Niche-V2.</li> <li>• Dependencia en clientes dominantes, por razón del volumen y porque han apalancado cambios en tecnología, como Niche-V2.</li> <li>• Endurecimiento de condiciones financieras.</li> <li>• Condiciones económicas desfavorables para el nicho del mercado.</li> </ul>

*Fuente: Elaboración propia, 2019. A partir de datos de Softniche Corp.*

A partir de la información de la Tabla 4 se ha realizado un análisis, confrontando Fortalezas y Debilidades contra Oportunidades, y de forma semejante, Fortalezas y Debilidades contra Amenazas, como se presenta a continuación:



**Fortalezas vs. Oportunidades** (Pares de éxito<sup>13</sup>). Para responder a las solicitudes de los clientes en relación con modelos enfocados en procesos que apunten a la mejora de la calidad, es necesario apoyarse en las fortalezas relacionadas con la implantación de procesos hacia la madurez organizacional, a través de la gestión por procesos y el enfoque de mejora continua. La consolidación de un marco de ingeniería es una de las iniciativas en esta dirección.

**Debilidades vs. Oportunidades** (Pares de adaptación<sup>14</sup>). Se hace necesario revisar el estado de la nueva versión del software Niche-V2, con miras a definir acciones que permitan contar con un producto terminado y de buena calidad para los clientes que así lo demanden.

Se requiere inversión en formación de los empleados o contratación de nuevos recursos ya formados, con el objeto de responder a las exigencias de los clientes en materia de enfoque en procesos, nuevos modelos operativos, nuevas metodologías y las tecnologías de software de Niche-V2.

La implementación de un nuevo marco de ingeniería de software permitirá apoyar los esfuerzos de desarrollo y verificación de la nueva versión del producto, incluyendo las mejores prácticas para las nuevas tecnologías y arquitecturas, con los objetivos de mejor calidad y menor costo.

**Fortalezas vs. Amenazas** (Pares de reacción<sup>15</sup>). La presencia regional de la empresa, el alto conocimiento del negocio y la habilidad para personalizar soluciones para cada cliente, deben generar acciones que permitan a Softniche Corp. adquirir una posición de ventaja frente a sus competidores y frente a las nuevas ofertas de productos de la competencia.

Un nuevo proceso de ingeniería de software con las mejores prácticas de la industria permitirá a la empresa mantener su posición de líder en el mercado de soluciones de software para el nicho del sector.

**Debilidades vs. Amenazas** (Pares de riesgo<sup>16</sup>). Se evidencia el riesgo de no poder reaccionar de manera oportuna ni eficaz frente a las empresas y productos competidores, debido a las presentes debilidades de pérdida de confianza del cliente por la baja calidad, la incompletitud y la falta de pruebas de la nueva versión del software, la falta formación e inversión en capacitación y la falta de capacidades en las prácticas de ingeniería de software y arquitectura.

---

<sup>13</sup> Se utilizan Fortalezas para aprovechar Oportunidades constituyendo pares de éxito.

<sup>14</sup> Debe ponerse foco sobre algunas debilidades, para aprovechar Oportunidades, constituyendo pares de adaptación.

<sup>15</sup> Con algunas Fortalezas se pueden enfrentar y mitigar Amenazas, constituyendo pares de reacción.

<sup>16</sup> Las Debilidades junto con las Amenazas, constituyen pares de riesgos que deben ser gestionados.

La implantación de un marco de ingeniería de software integrado para las operaciones permitirá mitigar este riesgo al cubrir varias dimensiones: la calidad, las capacidades del proceso de ingeniería, las prácticas de arquitectura de software y las prácticas de pruebas de software.

### **Conclusiones del análisis FODA**

Tras el análisis presentado, se pone en evidencia la necesidad de aprovechar las ventajas de la empresa en este momento, para impulsar iniciativas que permitan afrontar los retos del mercado entregando valor a los clientes. Una de las iniciativas más importantes actualmente es la implementación de un nuevo marco de ingeniería de software que incluya las mejores prácticas de la industria, de manera que se refuercen todas las etapas del ciclo de vida del software, con miras a entregar al cliente un producto de alta calidad de manera oportuna, optimizando, al mismo tiempo, los costos del proceso. La relevancia de esta mejora se comprende mayormente a partir de los retos actuales del mercado, las exigencias de los clientes, la nueva versión de software que se ha desarrollado sobre nuevas tecnologías y las condiciones actuales de los recursos humanos y de procesos en la organización.

En los análisis preparatorios del BP de años anteriores se había lanzado como iniciativa, para apoyar el desarrollo de varias mejoras y mitigar los riesgos asociados a la calidad del producto, la contratación de un Líder de Control y Aseguramiento de la Calidad de Software. A la persona que ocupara este cargo, se le asignaría, entre otras iniciativas, la de ayudar a definir e implantar un proceso unificado de ingeniería de software. En el presente, este cargo lo desempeña el autor de este TFM.

#### **2.2.1 Compromiso con el BP y el seguimiento**

La organización ha tomado conciencia a través de este ejercicio y se ha comprometido a sacar adelante esta y otras iniciativas claves, como las de recuperar el gobierno del producto, gestionar el conocimiento de los analistas y expertos funcionales, adelantar un diagnóstico y cambio de la cultura organizacional, entre otros.

Es claro que la alta dirección está involucrada en impulsar, respaldar y llevar a término de manera exitosa todas las iniciativas del BP que cada unidad se ha propuesto. Ahora bien, en la unidad de Colombia, de manera particular y desde hace unos meses, la alta dirección ha querido hacer un seguimiento pormenorizado del avance de todas las iniciativas de su unidad, a través de la revisión por la dirección, ya no con una periodicidad trimestral, sino mensual.

Este hecho ha sido muy importante y necesario para impulsar también un cambio cultural en la organización, ya que, además de lo dicho en materia de atraso tecnológico, un gran

porcentaje de los empleados lleva más de 15 años trabajando en la empresa con el sistema *legacy* y metodologías de los años 1980s, y conocen muy poco de las nuevas tecnologías, de los marcos y mejores prácticas en relación con la ingeniería de software, por lo que hay mucha resistencia al cambio y se hace indispensable el apoyo cercano y reiterado de la alta dirección.

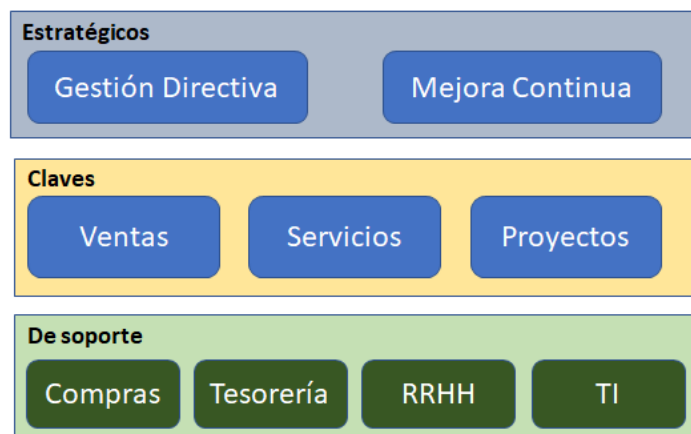
Finalmente, el compromiso para el cambio en la ingeniería de software se refleja en la contratación del nuevo Líder de Control y Aseguramiento de la Calidad de Software, a quién se le ha asignado la iniciativa de la mejora del proceso, y la alta dirección está empeñada en que se lleve a feliz término a través de seguimientos mensuales.

## **2.3 FASE 2 – IDENTIFICAR**

Hasta el momento se ha visto cómo a partir de un análisis FODA se ha hecho patente la conveniencia y relevancia de la mejora del proceso de ingeniería de software en la organización. No ha sido necesario hacer una selección, entre varios, de un proceso candidato a la mejora, ya que las iniciativas de mejora identificadas en el BP serán desarrolladas paralelamente por diversas áreas de la organización.

En el mapa de procesos de la Figura 3, en la línea de procesos claves o misionales, se puede ver cómo el flujo de valor inicia con las ventas, ya sea de licencias de software, de proyectos o de servicios; una vez concretada la venta se inicia un proyecto o un servicio, cuyo entregable será una versión del sistema Niche personalizada, o bien, una versión de Niche corregida o mejorada; el cliente finalmente recibe un producto de software de acuerdo con sus necesidades de personalización o de mejora entregado en el plazo y costo acordados. En la línea de procesos de soporte se encuentran compras, tesorería, recursos humanos y tecnologías de la información. Finalmente, la línea de procesos estratégicos comprende la gestión directiva y la mejora continua. Ahora bien, en cuanto a la ingeniería de software, el proceso de producción industrial no se hace explícito en el mapa de procesos vigente antes de la mejora, pues, como se ha mencionado, sus prácticas se encuentran difusas en varios procesos.

Figura 3 Mapa de procesos de Softniche Corp.



Fuente: Elaboración propia, 2019.

Los procesos que hacen parte directamente de la operación y que tienen relación con el desarrollo de software son los procesos clave de Gestión de Servicios y Gestión de Proyectos, según se describen en la Figura 4. Estos dos procesos, incluyen prácticas de ingeniería de software; pero no es posible inferir de ellos un conjunto unificado de prácticas ni de actividades ordenadas. Por esta razón, es necesario trabajar sobre ambos procesos cuando se levante el proceso AS-IS, tratando de delimitar el contexto y alcance de cada proceso de gestión, así como un marco común de ingeniería de software.

Figura 4 Subprocesos y dueños de proceso de los procesos de Gestión de Servicios y Gestión de Proyectos

Proceso de Gestión de <b>Servicios</b>	Proceso de Gestión <b>Proyectos</b>
<b>Subprocesos</b> <ul style="list-style-type: none"> <li>• Gestión de incidentes</li> <li>• Gestión de peticiones</li> <li>• Gestión de problemas</li> <li>• Gestión de mejoras</li> </ul> <b>Dueño del proceso</b> <ul style="list-style-type: none"> <li>• Gerente de Servicios.</li> </ul>	<b>Subprocesos</b> <ul style="list-style-type: none"> <li>• Proyectos de personalización del producto Niche.</li> <li>• Proyectos de nuevos desarrollos.</li> </ul> <b>Dueño del proceso</b> <ul style="list-style-type: none"> <li>• Gerente de Proyectos.</li> </ul>

Fuente: Elaboración propia, 2019.

Se concluye que no existe de manera explícita un proceso o un marco metodológico para la ingeniería de software en el mapa de procesos de la organización, y que las prácticas de desarrollo de software se encuentran incluidas dentro de las actividades de los subprocesos de los dos procesos de operaciones que se han identificado: Servicios y Proyectos (procesos claves).

## 2.4 FASE 3 – PLANIFICAR

A partir de la información identificada, el proyecto de mejora se describe a través del siguiente *Project Charter* (Tabla 5):

Tabla 5 *Project Charter para la mejora del nuevo proceso*

PROJECT CHARTER	
Definición del problema	<p>En los últimos años, la revolución digital, con la aparición de nuevas tecnologías y metodologías en la industria del software, ha ido transformando la visión y las necesidades de los clientes hacia productos entregados con mayor oportunidad y calidad, poniendo también de manifiesto el rezago tecnológico de Softniche Corp. Como consecuencia, el cliente ha perdido la confianza en la empresa, los niveles de calidad y oportunidad de entrega han caído, algunos clientes se están marchando y la rentabilidad de los desarrollos ha disminuido. Un análisis FODA demuestra la relevancia de una mejora en el proceso de construcción de software para mejorar la calidad y la oportunidad, y por ende la satisfacción del cliente, y así lograr la competitividad en el mercado.</p>
Declaración de objetivos	<p>Implementar un proceso de ingeniería de software, integrado a los procesos claves de Servicios y Proyectos, cuyos resultados, en un plazo de 2 años, sean:</p> <ul style="list-style-type: none"> <li>• <b>Adherencia:</b> 90% al proceso de ingeniería en servicios y proyectos.</li> <li>• <b>Calidad:</b> Esfuerzo en la atención de devoluciones inferior al 5%.</li> <li>• <b>Entregas oportunas:</b> Cumplimiento en la entrega o fecha de cierre superior al 80% en servicios y 100% en proyectos.</li> <li>• <b>Rentabilidad de las mejoras o proyectos:</b> Mantener entre el 15% y el 25%.</li> </ul>
Medidas o hechos que ilustran el problema	<p><b>Datos para los años 2018 y 2019:</b></p> <ul style="list-style-type: none"> <li>• Proyectos - Adherencia a los procesos: 70% y 79% (Mínimo 85%)</li> <li>• Proyectos - Cumplimiento de entregas: 34% y 30% (Mínimo 100%)</li> <li>• Mantenimientos - Cumplimiento de entregas: 86% y 91% (Mínimo 70%)</li> <li>• Mantenimientos - índice de calidad: 15% y 7% (Mínimo 15%)</li> <li>• Percepción de la calidad del producto: 54% y 66% (Mínimo 50%)</li> <li>• Utilidad en proyectos: -15% y -50% (Mínimo 15%)</li> </ul>

Recursos y limitaciones	<ul style="list-style-type: none"> <li>• Resistencia al cambio.</li> <li>• Las personas tienen un nivel bajo de conocimientos de ingeniería.</li> <li>• Restricciones de tiempo para que las personas participen en las tareas de implementación del nuevo proceso.</li> </ul>
Reglas de actuación del equipo	<p>El área responsable es el área de Calidad Técnica (QA/QC), debe coordinarse con:</p> <ul style="list-style-type: none"> <li>• Las áreas de calidad del Sistema de Gestión de Calidad (Para temas de documentación) y de calidad de Servicios (para integración con el marco de Gestión de Servicios).</li> <li>• El área de calidad corporativa para pedir validaciones y aprobación del proceso.</li> <li>• Las áreas de Gestión de Servicios y de Proyectos, para coordinar la implantación de la mejora.</li> <li>• La Gerencia de Operaciones, para reportar avance y hacer reuniones de seguimiento.</li> <li>• El área de arquitectura para obtener los lineamientos técnicos del proceso de ingeniería y solicitar revisiones.</li> </ul> <p>Debe participar en los seguimientos de la Alta Dirección para exponer el avance de la mejora.</p>
Miembros del equipo	Los participantes son todos los responsables de las diferentes áreas mencionadas en la fila anterior.
Riesgos	<ul style="list-style-type: none"> <li>• Resistencia al cambio por la cultura organizacional.</li> <li>• Dificultad de alineación de compromisos mutuos entre Softniche Corp. y cada cliente.</li> <li>• Falta de respaldo de las gerencias de Servicios y de Proyectos, por temor a ver afectadas sus operaciones con nuevos procesos.</li> <li>• Falta de presupuesto para capacitar a las personas en habilidades técnicas que exige un nuevo proceso de ingeniería de software.</li> </ul>

Plan preliminar del proyecto		<b>Quater</b>				
	<b>Hitos</b>	<b>Q1-2019</b>	<b>Q2-2019</b>	<b>Q3-2019</b>	<b>Q4-2019</b>	<b>Q1-2020</b>
	Identificar proceso existente	X				
	Diagnosticar proceso actual y analizar mejoras		X			
	Medir proceso o prácticas actuales		X			
	Diseñar nuevo proceso mejorado			X		
	Integrar el nuevo proceso en Gestión de Servicios y Proyectos			X		
	Implantar: Capacitación y acompañamiento, migración de repositorios.				X	
	Medir el proceso					X
	Controlar el desempeño					X
	...					

*Fuente: Elaboración propia, 2019.*

## 2.5 FASE 4 – ANALIZAR

En esta fase de análisis se presentará el modelo AS-IS del proceso de ingeniería de software, así como el diagnóstico sobre el mismo, identificando sus falencias y los requerimientos de mejora.

Un proceso de ingeniería estándar está conformado por las siguientes dimensiones:

- Identificación de las necesidades del usuario.
- Especificación de requerimientos (o diseño funcional).
- Arquitectura y diseño detallado (o técnico).
- Desarrollo de la solución.
- Pruebas.
- Entrega.

Como ya se ha mencionado a lo largo de este trabajo, las actividades de ingeniería de software en Softniche Corp., han estado tradicionalmente distribuidas en los dos procesos

claves de la organización, el proceso de Gestión de Proyectos y el de Gestión de Servicios, los cuales serán descritos a continuación.

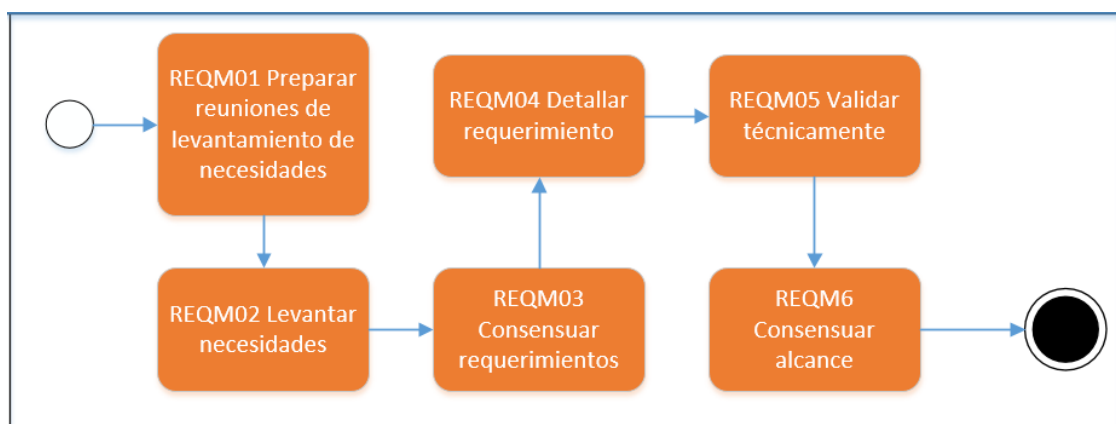
### 2.5.1 El marco de Gestión de Proyectos

Este marco fue uno de los primeros definidos e institucionalizados por la organización hace unos seis años. En él se intentaban recoger las mejores prácticas de ingeniería de software a partir de CMMI-DEV v1.2. Resulta un poco curioso notar que las actividades que definen el proceso de Gestión de Proyectos se basan más en un conjunto de prácticas de ingeniería de software que en un conjunto de prácticas del marco PMI<sup>17</sup>; esto es así ya que la definición del marco de Gestión de Proyectos la apoyó una empresa consultora en CMMI-DEV, contratada por Softniche Corp.

El marco de Proyectos está dividido en tres grandes etapas: Planificación, Personalización y UAT (*User Acceptance Tests*).

**Planificación:** En esta etapa se indaga sobre los requerimientos del cliente hasta clarificarlos, se analiza su viabilidad técnica y se llega a un acuerdo sobre los mismos, priorizándolos. Estas actividades se integran con actividades propias de la Gestión de Proyectos, como la labor de estimar el esfuerzo para el desarrollo. Las actividades de planificación se muestran en la Figura 5.

Figura 5 Fase de Planificación en el marco de Proyectos: Subproceso de REQM (Requirement Management)



Fuente: Elaboración propia, 2019

**Personalización:** Durante la personalización se detalla la especificación de requerimientos (REQM); se realizan las actividades de desarrollo de la solución o construcción del código

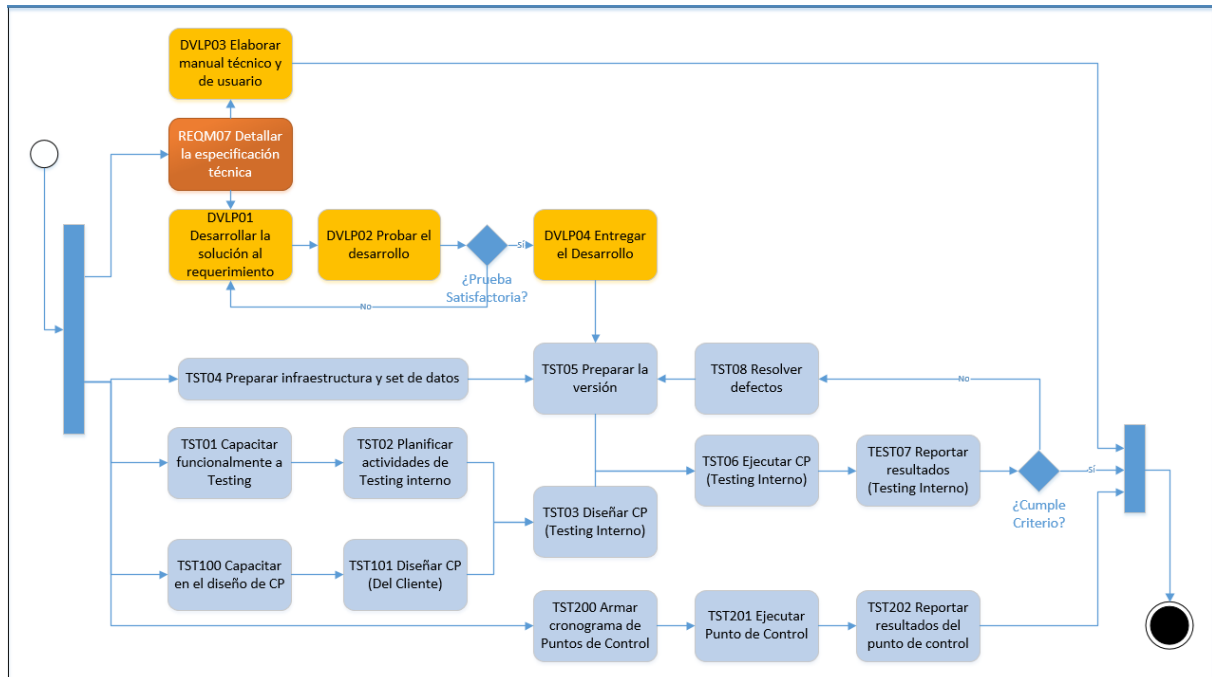
---

<sup>17</sup> Project Management Institute.



fuelle (DVLP); se llevan a cabo todas las actividades relacionadas con las pruebas internas del software y se adelantan las actividades de las pruebas con el cliente o pruebas UAT (TST). La Figura 6 presenta las actividades de esta fase.

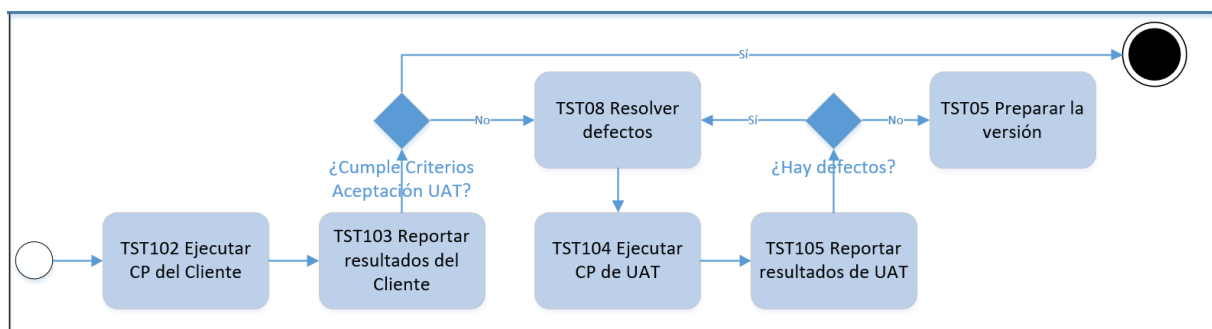
*Figura 6 Fase de Personalización en el marco de Proyectos:  
Subprocesos de Desarrollo (DVLP) y Pruebas (TST).*



*Fuente: Elaboración propia, 2019.*

**UAT (User Acceptance Tests):** En esta etapa final el cliente prueba el software desarrollado, se reportan los defectos encontrados y se corrigen hasta alcanzar el criterio de aceptación de calidad (actividades TST). En la Figura 7 se muestran dichas actividades.

Figura 7 Fase de UAT en el marco de Proyectos: Subproceso de pruebas de UAT.



*Fuente: Elaboración propia, 2019.*

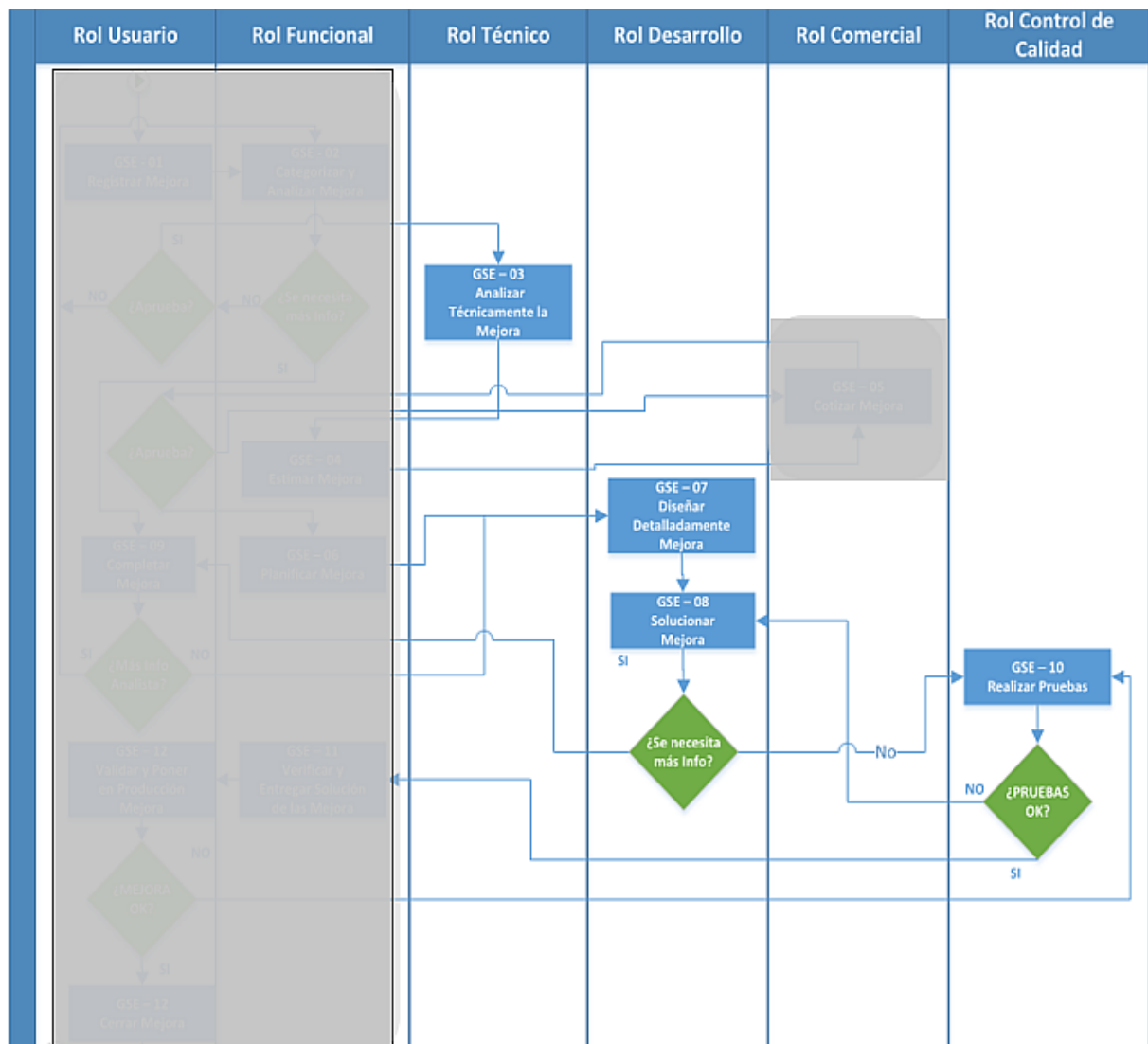
En general el marco de Proyectos incluye todas las actividades que debe involucrar un modelo en cascada de ingeniería de software, pero no involucra definiciones para nuevas tecnologías

y técnicas actuales del software: lenguajes orientados a objetos, arquitecturas complejas (web, servicios), diseño orientado a objetos, prácticas de versionamiento de código fuente, etc.

## 2.5.2 El marco de Gestión de Servicios

Este marco se compone esencialmente de tres subprocesos: Mejoras (evolutivos del software), Incidentes (corrección de defectos) y Problemas (corrección de causa raíz de incidentes recurrentes).

Figura 8 Subproceso de Mejoras.<sup>18</sup>



Fuente: Elaboración propia, 2019.

<sup>18</sup> Por razones de confidencialidad, se han ofuscado algunos sectores de la imagen.

**Subproceso de mejoras** (Figura 8): modela un servicio a través del cual el cliente solicita una modificación sobre el software. Durante el flujo se hace un análisis funcional y técnico del requerimiento, a continuación se estima y se cotiza y, si el cliente aprueba, se desarrolla la solución a través de actividades de desarrollo de software (análisis técnico, diseño detallado, solución, pruebas y entrega). La Figura 8 presenta el flujograma de este subproceso.

**Subproceso de incidentes:** su objetivo es formalizar el tratamiento de los incidentes del software reportados por el cliente.

**Subproceso de problemas:** incluye las actividades necesarias para diagnosticar la causa raíz de los incidentes recurrentes y darles solución.

Los diagramas de flujo de estos dos últimos subprocesos incluyen actividades de ingeniería de software similares a las del subproceso de mejoras, a través de las cuales se desarrolla el mantenimiento del software. Para el caso de los Problemas, se realiza el análisis técnico, el diseño detallado, la solución técnica, las pruebas y la entrega. Para el caso de Incidentes, solamente se realizan la solución técnica y la entrega.

El marco de Servicios, en cuanto a prácticas de ingeniería, presenta varias deficiencias. Por una parte, no incluye prácticas de diseño ni desarrollo para tecnologías *web* o lenguajes de programación orientados a objetos; por otra, no involucra prácticas para el versionado del código fuente; y, finalmente, no utiliza prácticas idóneas de pruebas de software, por cuanto sólo implementa una especie de pruebas internas que realiza el desarrollador (pruebas de desarrollador), eliminando las pruebas funcionales completas del software.

### 2.5.3 Diagnóstico del proceso actual de ingeniería de software (*Assessment*)

En el contexto de los marcos de solución de problemas y de mejora continua, un *Assessment* consiste en la aplicación de técnicas cuantitativas y cualitativas que permiten obtener un diagnóstico de la situación actual de los procesos desde varias perspectivas, con miras a identificar:

- Fortalezas, debilidades, actividades actuales de mejoramiento y áreas claves de mejoramiento.
- Estado actual del Proceso de Software y desarrollo de planes de acción para el mejoramiento.

- Nivel de capacidad de los procesos frente a un marco. En este caso se confronta contra el modelo **CMMI-DEV v1.3**<sup>19</sup> para las áreas de proceso de Ingeniería<sup>20</sup>.

#### 2.5.3.1 Metodología utilizada

Se aplicaron las siguientes técnicas para el levantamiento de la información del proceso:

1. **Entrevistas:** Se realizaron treinta (30) entrevistas con personas que desempeñan los roles propios del proceso de ingeniería: Arquitecto, Analistas Funcionales, Desarrolladores y Analistas de Pruebas.
2. **Aplicación de listas de chequeo:** Con el objeto de medir el cumplimiento de las Prácticas Específicas de CMMI-DEV en la operación, durante las entrevistas también se aplicó una lista de chequeo sobre las Prácticas Específicas de las seis (6) Áreas de Proceso de CMMI-DEV v1.3 asociadas con la ingeniería (REQM, RD, TI, PI, VAL, VER y CM); se obtuvo así una medición del porcentaje de cubrimiento de dichas prácticas. Esto da una idea del grado de capacidad de Softniche Corp. frente a estas áreas de proceso. Para un mayor entendimiento de estos resultados, léase el capítulo 3 MARCO TEÓRICO.
3. **Análisis de datos cualitativos** de información de las entrevistas.
4. Análisis de los roles del proceso de ingeniería a partir **de mediciones de tiempos imputados**.

#### 2.5.3.2 Capacidad frente a las áreas de procesos de ingeniería en CMMI-DEV

Como parte del *Assessment*, se quiso evaluar la capacidad de las Áreas de Proceso de ingeniería de CMMI-DEV, en el proceso que actualmente se ejecuta en las operaciones de Gestión Servicios. En CMMI-DEV, el primer nivel de capacidad para un Área de Proceso es *Performed* o Desarrollada (Nivel 1), y se llega a él cuando se evidencia el cumplimiento de todas las Prácticas Específicas; en caso de incumplimiento de alguna de las ellas, se dice que la capacidad del Área de Proceso es Incompleta (Nivel 0).

Son muchas las limitaciones para realizar una medición concluyente sobre un posible nivel de CMMI-DEV de la operación de Softniche Corp.; cualquier pretensión de este tipo sería irresponsable. Lo que sí se puede llegar a concluir es cuáles Áreas de Proceso están en un

---

<sup>19</sup> Para ampliar la información sobre el modelo CMMI-DEV véase el siguiente capítulo 3 sobre el marco Teórico.

<sup>20</sup> Las áreas de proceso de CMMI-DEV para ingeniería son: REQM (*Requirement Management*), RD (*Requirement Development*), PI (*Product Integration*), TS (*Technical Solution*), CM (*Configuration Management*), VAL (*Validation*) y VER (*Verification*).

nivel de capacidad cero (0) o Incompleta en el actual proceso de ingeniería de software. También hay que anotar que la evaluación de la totalidad de las Prácticas Específicas, sólo puede ser alcanzada para tecnología Niche-V3 para los subprocesos de Mejoras y Problemas, en la Gestión de Servicios, y así mismo para la Gestión de Proyectos, donde aplican todas ellas; esto es así ya que para Niche-V2 algunas prácticas pueden no aplicar (por ejemplo, arquitectura). Para el subproceso de incidencias, no aplican, por ejemplo, las prácticas relacionadas con diseño o pruebas funcionales.

De acuerdo con lo indicado, para un Área de Proceso de CMMI-DEV evaluada:

- Un cumplimiento del 100% significa que cumple todas las Prácticas Específicas que aplican en el proceso.
- Un cumplimiento menor al 100% significa que por lo menos alguna Práctica Específica que aplica en el proceso no se cumple, y por lo tanto la capacidad de dicha Área de Proceso es cero (0).

Los gráficos de resultados con su respectivo análisis se presentan en el **ANEXO 1 – GRAFICOS Y ANÁLISIS SOBRE EL CUMPLIMIENTO DE LAS PRÁCTICAS ESPECÍFICAS DE CMMI-DEV (CAPACIDAD DE AREAS DE PROCESO)**.

A partir de la evaluación de los resultados y teniendo en cuenta algunas consideraciones particulares de los subprocesos y de los clientes, se encontró que:

- Por lo general el nivel de cumplimiento de las prácticas de ingeniería evaluadas es deficiente (menor al 100%).
- Hay excepciones en las que el porcentaje de algún área de proceso ha llegado a 100%, para algún subproceso y/o cliente particular.
- Particularmente para el subproceso de incidentes hay puntajes del 100% en las Áreas de Proceso TS, VAL y VER. Es importante anotar que en este subproceso no todas las Prácticas Específicas son evaluables por lo que el resultado no es concluyente.

A partir de las mediciones y consideraciones, resulta que, ya sea por defecto del proceso de ingeniería usado actualmente, ya por defecto en la ejecución de los procesos de ingeniería en las operaciones, todas las Áreas de Proceso de CMMI-DEV (representación continua) para ingeniería, tienen una capacidad Incompleta (nivel 0).

En consecuencia, para subir el nivel de capacidad de las áreas de procesos de CMMI-DEV en las operaciones, es necesario mejorar el proceso de ingeniería e implantar el proceso mejorado en todos los clientes, tanto en servicios como en proyectos.

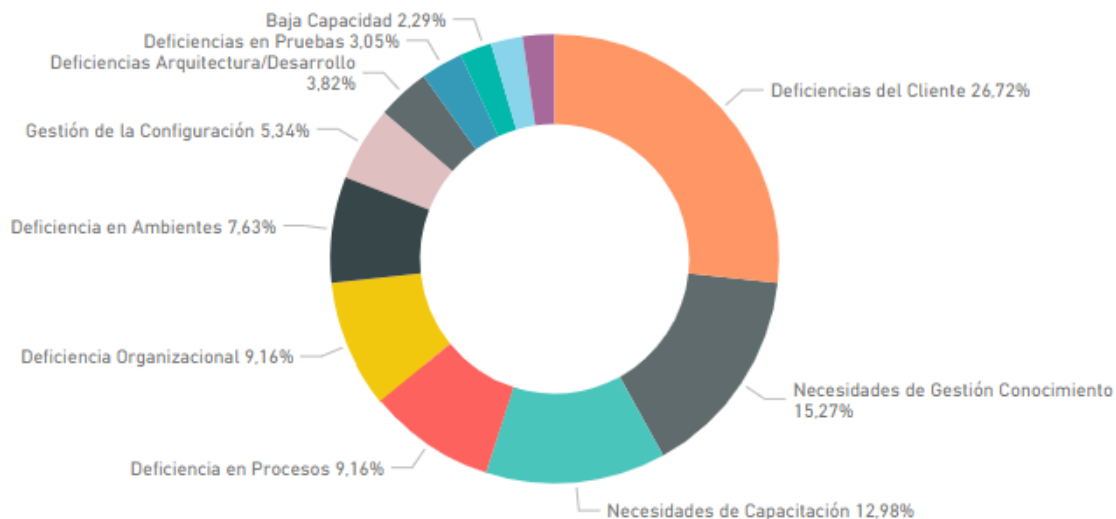
### 2.5.3.3 Resultados cualitativos a partir de las entrevistas

Con el propósito de ahondar un poco más en la identificación de otras causas que pudieran afectar el desempeño de los equipos, la calidad del producto y las entregas oportunas, los entrevistados fueron preguntados, al final de la entrevista, sobre temas adicionales que quisieran anotar, como por ejemplo:

- ¿Tiene algo más que decir?
- ¿Tiene alguna recomendación, alguna idea de mejora o propuesta?
- ¿Qué cuellos de botella identifica? ¿Qué soluciones ve viables?

Los siguientes datos se obtuvieron a partir de percepciones, sugerencias y recomendaciones, obtenidas de los entrevistados:

*Figura 9 Temas de mejora para otras causas de la falta de calidad y de oportunidad de entrega.*

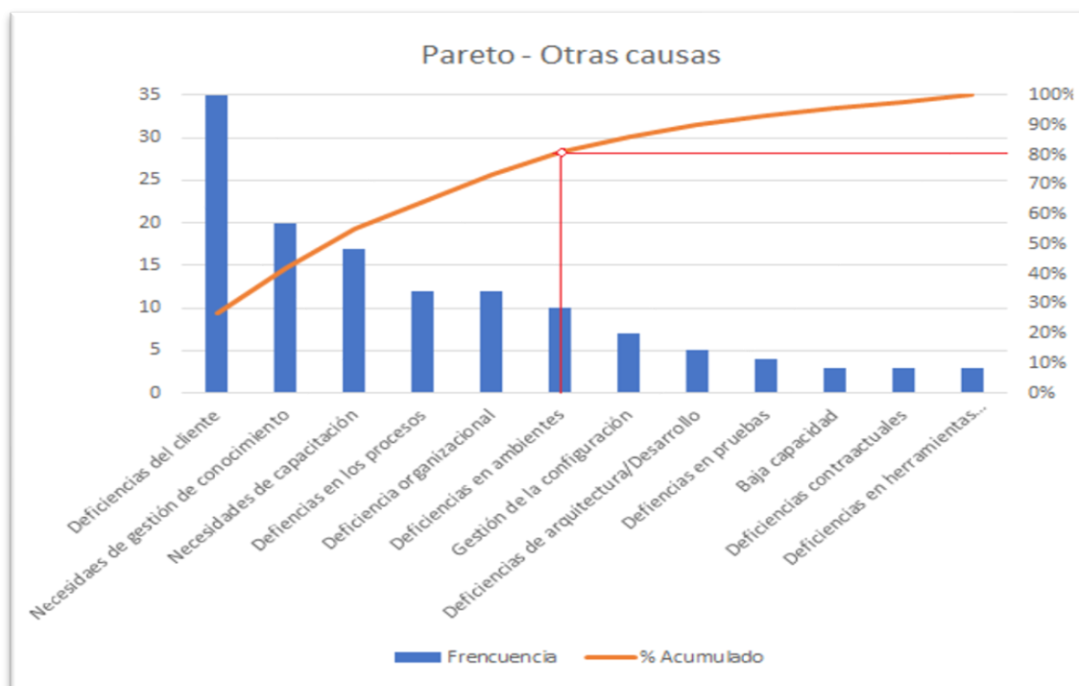


*Fuente: Elaboración propia, 2019.*

Por medio de un análisis de afinidad de las respuestas obtenidas, se obtuvo un grupo de temas que agrupan las oportunidades de mejora identificadas (Figura 9). Para cada tema se presenta su porcentaje—de acuerdo con la frecuencia de las oportunidades de mejora relacionadas con el tema—en relación con el total de oportunidades que conforman los temas.

A partir de esta información se realizó un diagrama de Pareto para identificar los principales temas—y sus correspondientes causas u oportunidades de mejora—en los que se debe poner foco.

Figura 10 Diagrama de Pareto, identificando las oportunidades de mejora mencionadas por el 80% de los encuestados.



Fuente: Elaboración propia, 2019.

El diagrama de Pareto (Figura 10) muestra una proporción 80%-50% entre la cantidad de respuestas recolectadas y las causas identificadas. Esto quiere decir que el 80% de los entrevistados coinciden en identificar las mismas seis (6) causas u oportunidades de mejora (el 50% del total de causas identificadas). Ahora bien, de estas seis (6) se han escogido las tres (3) primeras, las cuales se listan y describen en las Tablas Tabla 6 y Tabla 7:

Tabla 6 Top-3 oportunidades de mejora en Softniche Corp. (Otras causas).

	Causa	Frecuencia	Porcentaje	% Acumulado
1	<b>Deficiencias del cliente:</b> falta de alineación del cliente con el proceso.	35	26.7%	26.7%
2	<b>Necesidades de Gestión del Conocimiento:</b> conocimientos funcionales y técnicos.	20	15.3%	42.0%
3	<b>Necesidades de Capacitación:</b> en procesos, funciones e información funcional, técnica y metodológica.	17	13.0%	55.0%

Fuente: Elaboración propia, 2019.

*Tabla 7 Descripción detallada de las oportunidades de mejora por tema*

Tema	Oportunidad de mejora
<b>Deficiencias del Cliente</b>	Casos y ejecuciones de pruebas muy pobres.
	Demorado en responder y aprobar / O todo urgente.
	Despliegues y pruebas muy tarde – Con retrabajos de integración.
	Falsos positivos en Devoluciones (No se clasifican por causa).
	No tiene proceso / Hay desorden / Modifican BBDD Softniche Corp.
	Requerimientos pobres / Escenarios de prueba ausentes.
<b>Necesidades de Gestión del Conocimiento</b>	Crear una base de conocimientos de defectos, soluciones, mejores prácticas de BBDD.
	Crear una base de conocimientos funcionales.
	Deuda técnica: falta información funcional y técnica.
<b>Necesidades de Capacitación</b>	Falta capacitación / Falta guía de inducción al puesto.
	Falta capacitación en BBDD.
	Falta capacitación en ingeniería / UML y modelado BD.

*Fuente: Elaboración propia, 2019.*

Al enfocarse en estas tres primeras oportunidades de mejora mencionadas por los entrevistados, se pueden anotar los principales requisitos que el nuevo proceso de ingeniería podría solventar:

- En relación con las deficiencias del cliente:
  - Proveer medios para que el cliente aporte los criterios de aceptación o escenarios de pruebas.
  - Verificar la calidad de la documentación de las necesidades del cliente a través de listas de chequeo.
- En cuanto a las deficiencias en la Gestión del Conocimiento:
  - Documentar los diseños técnicos y funcionales, de manera que se haga explícito el conocimiento implícito de los expertos en las soluciones de software en el dominio específico del producto.
  - Contar con guías de desarrollo de software.
- Sobre las necesidades de capacitación:
  - La implantación del nuevo proceso debe incluir capacitaciones en procesos, ingeniería de software y metodologías.

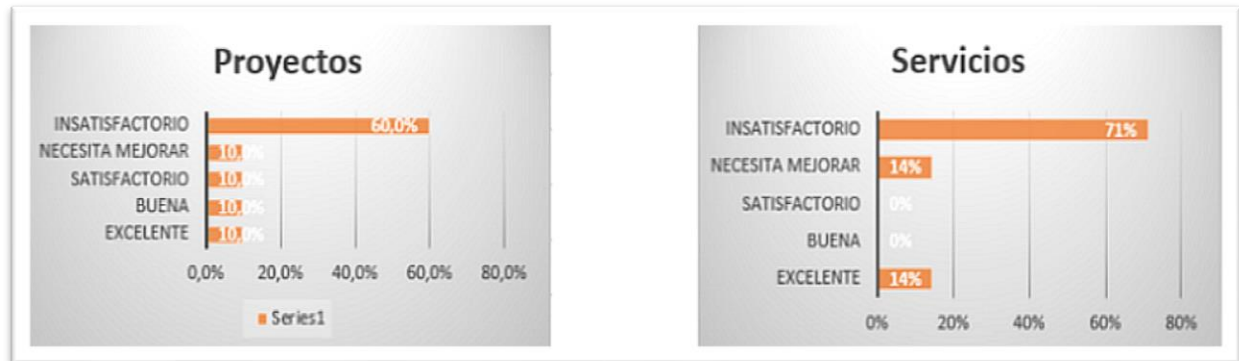
#### *2.5.3.4 Análisis de los roles del proceso de ingeniería*

De acuerdo con un análisis realizado en Q1-2019, a través de los datos de las horas imputadas por los equipos en diferentes actividades de ingeniería, se ha detectado que el 60% de los roles tienen un cumplimiento insatisfactorio en relación con las actividades que desempeñan versus las que deberían desempeñar de acuerdo con unos KPIs definidos, en



los que se han determinado rangos porcentuales de actividades que cada rol debe desempeñar.

*Figura 11 Evaluación de las actividades realizadas por las personas de acuerdo con sus roles, para las operaciones de Proyectos y Servicios.*



*Fuente: Elaboración propia, 2019.*

La Figura 11 muestra un porcentaje del 60% insatisfactorio para los roles participantes en la gestión proyectos y un 71% insatisfactorio para los roles que participan en la Gestión de Servicios. Un mayor detalle de los resultados muestra que sólo los roles Junior (*Tester* y *Analista Programador*) cumplen con las expectativas de realizar actividades en las que deben enfatizar, los demás roles no están enfocando sus esfuerzos sobre actividades propias de su rol y en cambio ejecutan muchas tareas propias de roles inferiores.

Con estos resultados se evidencia que la cadena de valor se está viendo fragmentada prácticamente desde el inicio del proceso, donde se detecta la ausencia de actividades fuertes de ingeniería (arquitectura, diseño técnico y funcional) y la dedicación casi exclusiva de los roles senior a desarrollo y a pruebas (cuando deberían dedicar gran parte de su tiempo a diseñar).

La siguiente figura (Figura 12) esquematiza, al lado izquierdo, el escenario ideal e hipotético de un proceso estándar de ingeniería de software; al lado derecho, el escenario real a partir de los datos obtenidos de las operaciones.

Figura 12 Desempeño hipotético vs. real de los roles de ingeniería en Softniche Corp.



Fuente: Elaboración propia, 2019.

En el escenario ideal, el cliente genera una necesidad, los roles senior (analista funcional, arquitecto, programador senior) producen los diseños necesarios y los documentan, creando los activos documentales necesarios para que los demás roles (programadores y testers) puedan construir el software (el muro, en la Figura 12) sin dificultades. Sin embargo, la realidad muestra que los roles altos, que deberían estar realizando labores de diseño y arquitectura, están bajando al nivel de los roles encargados de construir y probar el software, por lo que no se está haciendo diseño y, por tanto, han desaparecido los activos documentales, haciendo que las labores de cada cual se difuminen y se entrecrucen. En estas circunstancias, los analistas funcionales se desempeñan como testers, los arquitectos y diseñadores desarrollan software y los mismos desarrolladores realizan pruebas del sistema; de esta manera, el control se pierde y es imposible garantizar calidad u oportunidad en la entrega.

#### 2.5.3.5 Lista de los requisitos de mejora

A partir de toda la información y los análisis hasta aquí presentados es posible, en este momento, identificar con claridad los requisitos del nuevo proceso mejorado. El nuevo proceso, debe involucrar las mejores prácticas de la industria de software e intentar dar solución a las oportunidades de mejora identificadas por los entrevistados, ofreciendo, a la vez, soporte a las nuevas tecnologías (orientadas a objetos, arquitectura web) y a las versiones activas del producto (Niche V2 y Niche V3).

La siguiente lista presenta los requerimientos del nuevo proceso de ingeniería de software:

- Proveer un marco único y común para los subprocesos de la Gestión de Servicios y de Proyectos.
- Soportar las tecnologías de las dos versiones activas actualmente en Softniche Corp.: Niche V2 y Niche V3.
- Incluir las mejores prácticas en ingeniería de software en la industria.

- Definir sin ambigüedad los roles del proceso de ingeniería de software, identificando las actividades de las que son responsables y los artefactos de los que son dueños.
- Soportar tecnología orientada a objetos, integrando prácticas y técnicas orientadas a objetos.
- Obtener criterios de aceptación y verificar calidad de los requerimientos a partir de las necesidades del cliente.
- Aportar artefactos para documentar técnica y funcionalmente el producto desarrollado.
- Aportar guías de desarrollo de software.
- Incluir capacitaciones en procesos, ingeniería de software y metodologías, durante la implantación del nuevo proceso.

#### 2.5.4 **Dashboard** del proceso y medición del proceso AS-IS

A continuación, se definen los indicadores para el proceso de ingeniería de software. La siguiente tabla (Tabla 8) presenta el detalle de la definición de cada indicador, describiendo para cada uno los siguientes aspectos:

- El nombre del **indicador**.
- La métrica (**Metric**) o fórmula que calcula el indicador.
- La pregunta que resuelve (**Question**) dicho indicador.
- El objetivo o meta (**Goal**) establecida de cumplimiento del indicador. Adicionalmente, entre corchetes y otro color se presenta la [medición actual] del proceso AS-IS.
- Una descripción de **cómo se recibe la información** necesaria para realizar el cálculo.

Tabla 8 Indicadores y mediciones del proceso AS-IS.

Indicador	Metric	Question	Goal [Medición actual]	Cómo se recibe la información
Adherencia al proceso [Mensual]	Porcentaje de ítems de la lista de chequeo que se cumplen <u>frente</u> al total de ítems que apliquen. [Por Cliente]	¿En qué grado se están siguiendo las prácticas de ingeniería?	> 80% [Hoy 40%]	Resultados de las auditorías

Indicador de esfuerzo de no-calidad [Mensual]	Horas de solución de devoluciones <u>entre</u> horas total del proyecto o mantenimiento. [Por Cliente]	¿Qué porcentaje de esfuerzo se invierte en corregir la no-calidad, después de entregar al Cliente?	< 10% [Hoy < 15%]	Horas registradas en la herramienta de imputaciones.
Indicador de entrega oportuna [Mensual]	Cantidad de Proyectos o Mantenimientos entregados a tiempo <u>entre</u> el total de Proyectos o Mantenimientos. [Por Cliente]	¿Cuál es el porcentaje de entregas a tiempo?	> 80% [Hoy Proyectos aprox. 40%; meta 100% Mantenimientos Aprox. 80%; meta > 70%]	Herramienta de registro de casos y fechas de inicio y fin. ANS de terminación.
Indicador de roles de ingeniería [Mensual]	Cantidad de personas con calificación satisfactoria en las actividades ejecutadas de su rol <u>entre</u> el total de personas en el rol. [Por Cliente]	¿Las personas están realizando sus tareas especializadas correspondientes a su rol?	> 70% [Hoy < 40%]	Horas registradas en la herramienta de imputaciones. Tabla de porcentaje de horas de actividades por rol.

Fuente: Elaboración propia, 2019.

A continuación se describen otros indicadores que no se están midiendo actualmente en el proceso AS-IS, pero que es necesario considerar para tener una visión más amplia del desempeño (Tabla 9):

Tabla 9 Indicadores no medidos del proceso AS-IS.

Indicador	Metric	Question	Goal	Cómo se recibe la información
Indicador de calidad Interna [Mensual]	Cantidad de Bugs (detectados en pruebas internas) <u>entre</u> horas totales imputadas del proyecto o mantenimiento. [Por Cliente]	¿Cuál es la proporción de errores internos en relación con un estimado del “tamaño” <sup>21</sup> del software?	< 0,25	Cantidad de Bugs internos registrados en herramienta de registro de bugs. Horas registradas en la herramienta de imputaciones.

<sup>21</sup> Como estimado de tamaño, se usará el esfuerzo ejecutado de desarrollo.

Indicador de calidad Externa [Mensual]	Cantidad de Bugs (detectados por el Cliente) <u>entre</u> horas totales imputadas del proyecto o mantenimiento. [Por Cliente]	¿Cuál es la proporción de errores detectados por el Cliente en relación con un estimado del “tamaño” del software?	< 0,125	Cantidad de Bugs del Cliente registrados en herramienta de registro de bugs. Horas registradas en la herramienta de imputaciones.
Ineficiencia económica de los roles de ingeniería	(Costo teórico de los roles menos costo real de los roles) <u>entre</u> costo teórico. [Por Cliente]	¿Cuánto cuesta (+/- %) que las personas desempeñen actividades de otros roles que no son el suyo?	> -20%	Horas registradas en la herramienta de imputaciones. Tabla de porcentaje de horas de actividades por rol.

Fuente: Elaboración propia, 2019.

Habiendo llegado al final de este capítulo, el cual ha sido un poco extenso debido a la relevancia de los temas aquí presentados, es importante sintetizar lo que se ha desarrollado hasta el momento. Se ha hecho un análisis de la situación actual de la empresa, presentando cifras ilustrativas sobre el problema. Se han trabajado las cuatro (4) primeras fases del CMP: sensibilización, identificación, planificación y análisis, obteniendo importante documentación de diagnóstico del proceso actual: Análisis FODA, Mapa de Procesos, *Project Charter* con la propuesta de proyecto de mejora, el modelaje del proceso AS-IS, un *Assessment* sobre el mismo a la luz de marcos metodológicos de la industria (CMMI-DEV, RUP) y su respectivo *Dashboard* de indicadores.

En el siguiente capítulo se expondrá el marco teórico que respalda este TFM, y el capítulo 4 se continuará con el desarrollo de las últimas cuatro (4) fases del CMP, en donde se presentará el diseño la mejora y se abordará su implantación.

### 3 MARCO TEÓRICO

Para la elaboración de este TFM se han tomado como referencia varios marcos metodológicos relacionados con la calidad y la mejora de procesos, los cuales serán descritos a continuación.

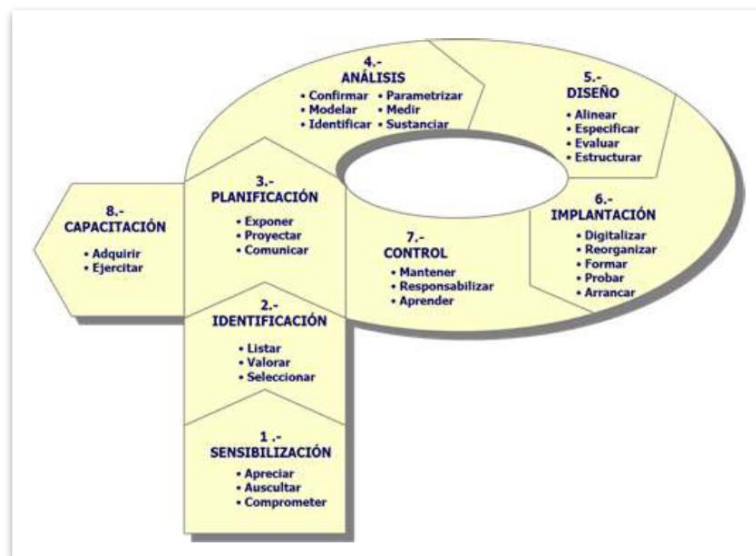
#### 3.1 CMP – CICLO DE MEJORA DE PROCESOS DE NEGOCIO

El CMP es una “metodología de análisis del ciclo de vida de una iniciativa de mejora de procesos de negocio” (Berenguer & Ramos-Yzquierdo, 2004, p.81).

Se trata de un modelo de resolución de problemas de ocho fases, que integra las perspectivas del negocio con las de los sistemas de información, fomentando, a su vez, la adquisición de una cultura de procesos.

La Figura 13 presenta el ciclo de las ocho (8) fases, cada una de las cuales se compone de actividades.

*Figura 13 CMP – Ciclo de Mejora de Procesos de Negocio*



*Fuente: Berenguer J.M. y Ramos-Yzquierdo J.A. (2008, p.17)*

Las primeras cuatro fases dicen relación con obtener el compromiso de la alta dirección, la identificación de un proceso candidato a la mejora, la formulación de un proyecto de mejora y el análisis del proceso AS-IS.

Las últimas cuatro fases están en correspondencia con la propuesta del diseño del nuevo proceso mejorado (proceso TO-BE), su implantación en la organización, el control del proceso por medio de mediciones y finalmente, la aplicación repetida de las prácticas mejoradas que crean una cultura de excelencia operacional.

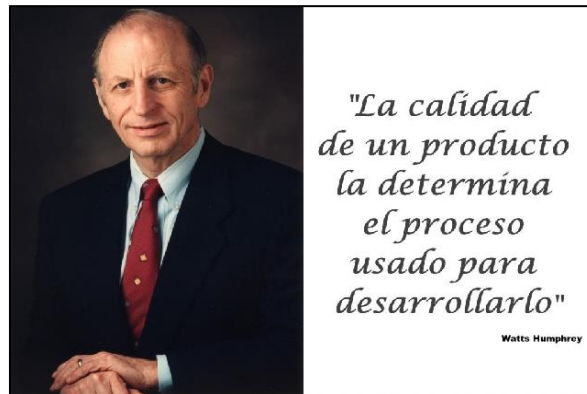
Este es el marco metodológico que se ha utilizado en este TFM para orientar la propuesta de mejora de procesos.

### **3.2 Relación entre calidad del producto y del proceso**

La mejora propuesta en este TFM se fundamenta en un principio sobre los procesos y la calidad, que formuló el llamado “padre de la calidad del software”:

Watts Humphrey (1988), al introducir el modelo CMM, afirmó que la premisa del CMM radica en la relación entre la calidad del proceso y la calidad del producto. La creencia es que la gestión efectiva del proceso será el factor clave para producir software de alta calidad. (Duggan & Reichgelt, 2006)

*Figura 14 Foto de Watts Humphrey, fundador del SEI (Software Engineering Institute).*



*Fuente: <https://alchetron.com/Watts-Humphrey> (Oct./2019)*

Humphrey fue vicepresidente de IBM y, en los años 1980, fundador y miembro de la junta directiva del Software Engineering Institute (SEI)<sup>22</sup>, creadores de CMMI.

Ahora bien, de acuerdo con Humphrey, para que un cambio o mejora en el proceso sea exitoso, se requiere respetar los siguientes cinco (5) principios:

1. Apoyo de la alta dirección. Los cambios en el proceso de software deben comenzar desde arriba.
2. Compromiso. Finalmente, todos los involucrados deben estar comprometidos.
3. Partir de un Diagnóstico para definir un Objetivo. Un cambio efectivo requiere un objetivo y el conocimiento del proceso actual.
4. Cambio continuo. Ya que no es un esfuerzo de una única vez, requiere esfuerzo consiente y refuerzo periódico.
5. El mejoramiento cuesta. Requiere una inversión financiera.

Esta orientación de mejora de procesos de Humphrey está en completa sintonía con la metodología CMP. En cualquier caso, será la filosofía de los procesos la que fundamentará la propuesta de mejora de este TFM.

---

<sup>22</sup> El Software Engineering Institute (SEI) es un instituto de la universidad de Carnegie Mellon auspiciado por el Departamento de Defensa de los Estados Unidos. En 1983, por concurso, se le adjudica la labor de diseñar un modelo de mejora para la industria del Software.



Ahora bien, en relación con la calidad, en la actualidad existen varios marcos de referencia relacionados con la calidad del software<sup>23</sup> y con las buenas prácticas, la madurez de las áreas y la capacidad de los procesos de desarrollo de software<sup>24</sup>.

En lo que sigue de este capítulo, se presentarán dos marcos metodológicos de madurez y capacidad (CMMI-DEV) y de buenas prácticas (CMMI y RUP) que fundamentarán el diseño del nuevo proceso de ingeniería de software para Softniche Corp.

### **3.3 El modelo CMMI-DEV (Capability Maturity Model Integration for Development)**

La sigla CMMI-DEV significa Modelo de Madurez y de Capacidad Integrado (CMMI) para Desarrollo de Software (DEV), fue desarrollado por el Software Engineering Institute (SEI), en colaboración entre miembros del Instituto, la industria y el gobierno de los Estados Unidos.

Consiste en una colección de buenas prácticas que ayudan a las organizaciones a mejorar sus procesos centrándose en las actividades para desarrollar productos y servicios de calidad con el fin de cumplir las necesidades de clientes y usuarios finales. Se ha convertido en un marco de referencia en organizaciones de desarrollo de software, siendo, por una parte, una base para la evaluación de capacidades y madurez, y, por otra, una guía para implementar una estrategia de mejora continua.

El modelo de mejoramiento provee un conjunto de prácticas reconocidas por la industria para la productividad, desempeño, costo y satisfacción del cliente en los procesos de ingeniería de sistemas y procesos de desarrollo de software. CMMI permite evaluar, tanto el desempeño o *performance* (los resultados alcanzados) como la capacidad o *capability* (resultados esperados) de un proceso.

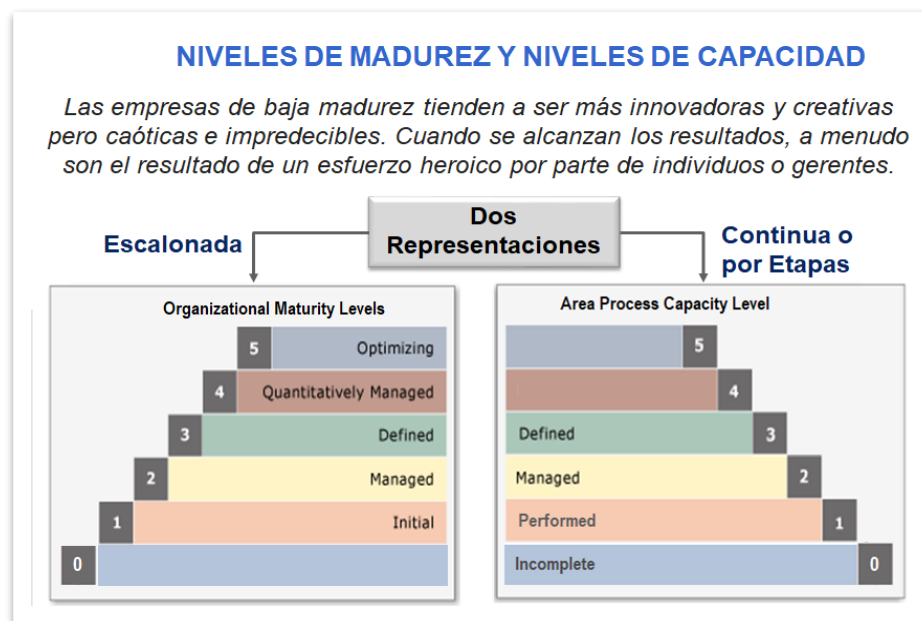
---

<sup>23</sup> Por ejemplo, la familia de normas ISO/IEC 25000, que reemplazó a la norma ISO/IEC 9126, define los atributos de la calidad de un producto de Software.

<sup>24</sup> Por ejemplo, la familia de normas ISO/IEC 33000, que reemplazó a la norma ISO/IEC 15504, constituye en marco para la evaluación de procesos. Así mismo, MMIS V.2., es el modelo de madurez fundamentado en la ISO/IEC 33000.



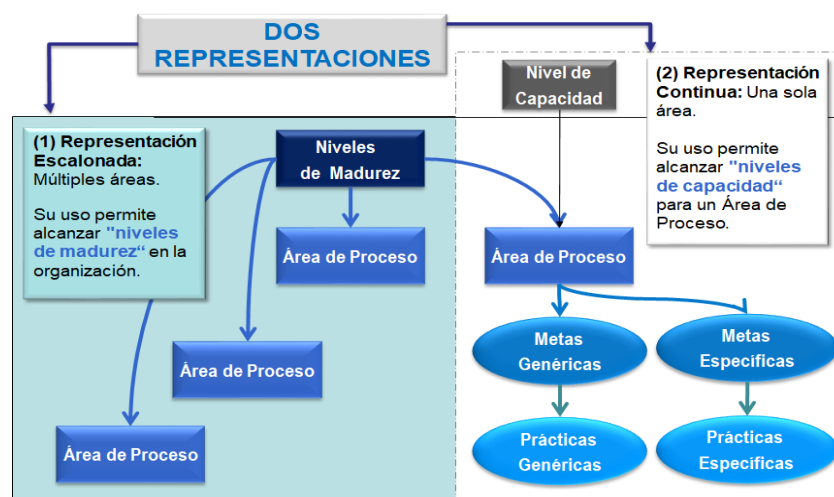
Figura 15 Niveles de madurez y de capacidad en CMMI.



Fuente: Elaboración propia, 2019.

CMMI se estructura en dos representaciones (Figura 15 y Figura 16), la Representación Escalonada, que permite evaluar el nivel de madurez de un equipo (fábrica, grupo, centro de desarrollo de software), y, la Representación Continua, que evalúa los niveles de capacidad o desempeño esperado de los procesos de la organización frente a una determinada Área de Proceso<sup>25</sup>.

Figura 16 Representación escalonada y continua de CMMI.



Fuente: Elaboración propia, 2019.

<sup>25</sup> Esta es la representación que se ha utilizado para hacer el Assessment al proceso actual de ingeniería de software en Softniche Corp., y que se ha presentado en el capítulo 2.

El modelo comprende veintidós (22) Áreas de Proceso<sup>26</sup>, que representan prácticas o actividades. También se tienen definidos unos objetivos o Metas Genéricas<sup>27</sup> (transversales a varias Áreas de Proceso) y Específicos<sup>28</sup> (para un Área de Proceso particular).

Los objetivos o Metas Genéricas se cumplen si se cumplen sus Prácticas Genéricas<sup>29</sup> asociadas. Los objetivos o Metas Específicas se cumplen si se cumplen sus Prácticas Específicas<sup>30</sup> asociadas.

Las veintidós Áreas de Proceso del modelo, se clasifican en varias categorías: gestión de procesos, Gestión de Proyectos, ingeniería y soporte. El nivel de madurez (representación escalonada) en la Tabla 10, permite identificar cuáles Áreas de Proceso son evaluadas en cada uno de los niveles de madurez de CMMI. Así, por ejemplo, para que un equipo o área de desarrollo de software de la organización pueda ser evaluada en nivel de madurez 5, además del cumplimiento de todas las Áreas de Proceso de los niveles anteriores, de debe verificarse el cumplimiento de las PA: “Gestión del rendimiento de la organización” y “Análisis causal y resolución”.

Tabla 10 Las Áreas de Proceso (Process Areas) de CMMI.

Área de Proceso	Sigla	Categoría	Nivel Madurez	Process Area
Definición de Procesos de la organización	OPD	Gestión De Procesos	3	Organizational Process Definition
Enfoque en Procesos de la organización	OPF	Gestión De Procesos	3	Organizational Process Focus
Formación en la organización	OT	Gestión De Procesos	3	Organizational Training
Rendimiento de Procesos de la organización	OPP	Gestión De Procesos	4	Organizational Process Performance
Gestión del rendimiento de la organización	OPM	Gestión De Procesos	5	Organizational Performance Management
Monitorización y control del Proyecto	PMC	Gestión De Proyectos	2	Project Monitoring and Control
Planificación del Proyecto	PP	Gestión De Proyectos	2	Project Planning
Gestión de acuerdos con Proveedores	SAM	Gestión De Proyectos	2	Supplier Agreement Management
Gestión integrada del Proyecto	IPM	Gestión De Proyectos	3	Integrated Project Management
Gestión de riesgos	RSKM	Gestión De Proyectos	3	Risk Management
Gestión cuantitativa del Proyecto	QPM	Gestión De Proyectos	4	Quantitative Project Management
Gestión de requisitos	REQM	Ingeniería	2	Requirement Management
Integración del Producto	PI	Ingeniería	3	Product Integration
Desarrollo de requisitos	RD	Ingeniería	3	Requirement Development
Solución técnica	TS	Ingeniería	3	Technical Solution
Validación	VAL	Ingeniería	3	Validation
Verificación	VER	Ingeniería	3	Verification
Gestión de configuración	CM	Soporte	2	Configuration Management
Medición y análisis	MA	Soporte	2	Measurement and Analysis
Aseguramiento de la calidad del Proceso y del Producto	PPQA	Soporte	2	Product and Process Quality Assurance
Análisis de decisiones y resolución	DAR	Soporte	3	Decision and Analysis Resolution
Análisis causal y resolución	CAR	Soporte	5	Causal Analysis and Resolution

Fuente: Elaboración propia, 2019.

<sup>26</sup> Process Area, en Inglés.

<sup>27</sup> Generic Goals, en Inglés.

<sup>28</sup> Specific Goal, en inglés.

<sup>29</sup> Generic Practices.

<sup>30</sup> Specific Practices.

La mejora propuesta en este TFM estará centrada en las PA de ingeniería que, de acuerdo con la tabla anterior (Tabla 10), son seis (6):

- **Gestión de requisitos (REQM):** levantamiento, depuración, análisis de viabilidad y priorización de necesidades, y registro de trazabilidad de las necesidades hasta las pruebas del usuario final.
- **Integración del Producto (PI):** interfaces, servicios externos, integración de varios módulos o componentes.
- **Desarrollo de requisitos (RD):** detallar los requisitos funcionales y no funcionales, diseño funcional y técnico, evaluaciones técnicas y de viabilidad.
- **Solución técnica (TS):** desarrollo de software, escritura de código fuente.
- **Validación (VAL):** pruebas funcionales, de integración, unitarias, etc.
- **Verificación (VER):** revisión de pares, inspecciones y revisiones formales, etc.

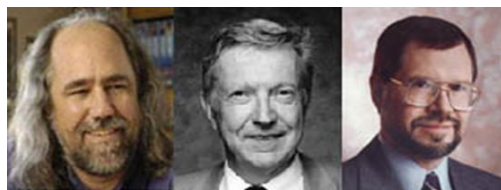
En este TFM se utilizarán las definiciones correspondientes a las mejores prácticas para las Áreas de Proceso de Ingeniería con el fin de mejorar el proceso de ingeniería de software de Softniche Corp., elevando la capacidad de los procesos.

### 3.4 El Marco RUP (Rational Unified Process)

RUP es una metodología de desarrollo de software creada por la empresa Rational Software, actualmente propiedad de IBM. Su objetivo es soportar el modelado y el ciclo de vida del Software en una filosofía orientada a objetos. Esto significa que es una metodología idónea para el desarrollo de software con lenguajes de programación orientados a objetos.

Fue creada en 1998 y hecha pública en 1999 por sus autores: Grady Booch (creador del método Booch), Ivar Jacobson y James Jacobson (Creador de la Técnica de Modelado de Objetos).

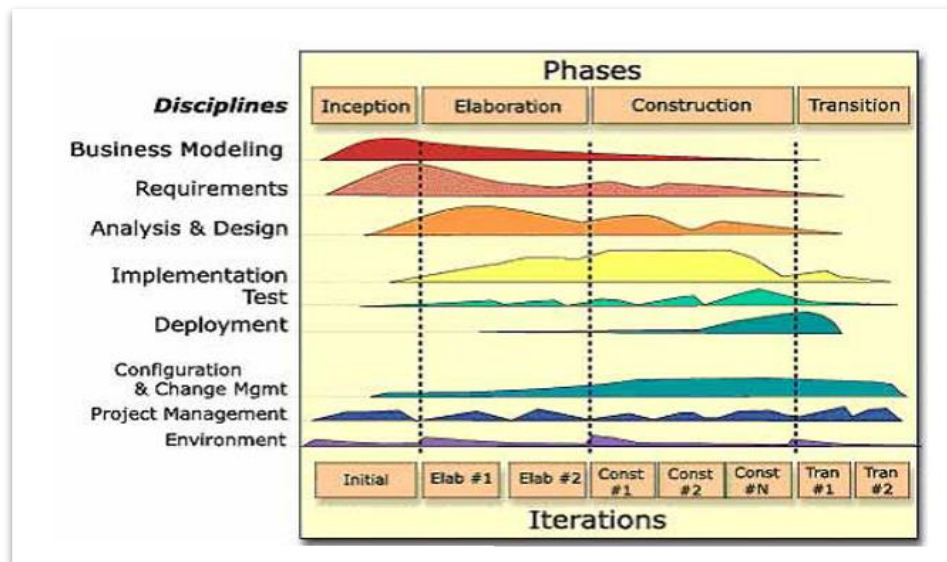
*Figura 17 Fotos: Grady Booch, Ivar Jacobson y James Jacobson.*



*Fuente: <http://rupequipo1.blogspot.com/2012/12/algo-de-historia.html> (Oct./2019).*

De acuerdo con este modelo, un proceso de desarrollo de software se divide en varias fases a través de las cuales se realizan varias dimensiones (modelado del negocio, requerimientos, análisis y diseño, implementación, pruebas, etc.).

Figura 18 RUP: Esquema de Fases y Disciplinas.



Fuente: <http://www.infforum.de/themen/anwendungsentwicklung/se-rup.htm> (Oct./2019)

De esta manera, la metodología es iterativa e incremental, es decir, cada fase puede involucrar varias iteraciones o repeticiones de una misma dimensión, por ejemplo, la implementación o las pruebas y, además, cada iteración incrementa el valor del producto de software obtenido. La iteración y el incremento vienen a contraponerse a la metodología típica en cascada, en la que, una vez terminada la realización de una dimensión, continúa la siguiente, y así sucesivamente hasta agotar todas las dimensiones.

Incluida la dinámica de las iteraciones, las siguientes son las características fundamentales de la metodología:

- Guiada/Manejada por casos de uso.
- Centrada en arquitectura.
- Iterativa e Incremental.
- Desarrollo basado en componentes.
- Utilización de un único lenguaje de modelado: UML (*Unified Modeling Language*)<sup>31</sup>.
- Proceso Integrado en todas las dimensiones.

---

<sup>31</sup> Se trata de un lenguaje de modelado orientado a objetos, soportado por varias técnicas, diagramas, descripciones y convenciones. Por ser orientado a objetos, es posible aplicar los paradigmas del re-uso, la herencia, la extensión, etc.

Los elementos esenciales del modelo son:

- **Roles:** Desempeñan un conjunto de actividades y son dueños de un conjunto de artefactos.
- **Actividades:** Una actividad de un *Worker* específico (alguien con un Rol) es una unidad de trabajo que se le pide hacer a un individuo.
- **Productos** (*Artifacts*): en términos OO<sup>32</sup> los artefactos son los parámetros de las actividades.
- **Flujos** (*Workflows*): una secuencia de actividades que produce un resultado de valor observable.

La metodología RUP es el fundamento de la filosofía orientada a objetos en la que se soporta la nueva versión del sistema Niche V3, y los nuevos desarrollos de tecnologías *web* y arquitecturas más complejas del software.

Con esto se concluye la exposición del marco teórico de este TFM, el cual ha incluido de manera general la descripción de los principales elementos de los marcos metodológicos para: CMP (Ciclo de Mejora de Procesos) de la UNAV, CMMI-DEV v1.3 (Capability Maturity Modelo Integration) del SEI y RUP (Rational Unified Process) de IBM.

## 4 SOLUCIÓN PROPUESTA

En el capítulo 2 se desarrollaron los principales aspectos de las cuatro (4) primeras fases del modelo CMP: Sensibilización, Identificación, Planificación y Análisis. En primer lugar, se describió la situación actual presentando cifras; a continuación, se analizaron las fortalezas, debilidades, oportunidades y amenazas, lo que permitió evidenciar la relevancia de implementar un nuevo marco de ingeniería de software (Sensibilización); posteriormente se identificó el mapa de procesos actuales, enfocándose en los procesos de Gestión de Servicios y de Proyectos, que son aquellos que contienen las prácticas de desarrollo de software (Identificación); acto seguido, se presentó el *Project Charter* para desarrollar la mejora de los procesos que involucran las prácticas de ingeniería de software (Planificación); finalmente, se presentó, a nivel general, el proceso AS-IS, los resultados de su diagnóstico (*Assessment*) y el *Dashboard* del proceso AS-IS (Análisis).

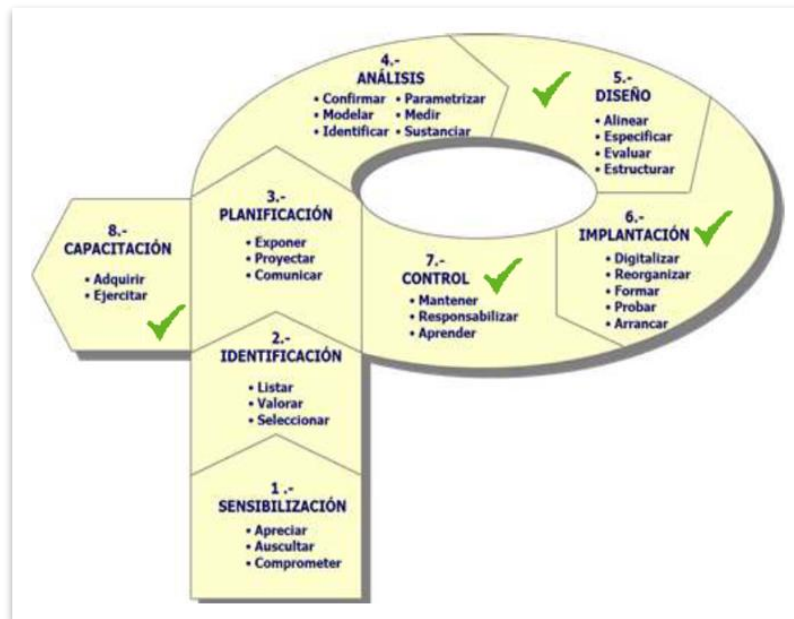
---

<sup>32</sup> OO: Orientado a Objetos.

Por otra parte, en el capítulo 3, se hizo una breve exposición de los conceptos y marcos metodológicos más importantes sobre los que se ha basado este TFM: CMP, Calidad de Producto vs. Calidad de Proceso, CMMI-DEV y RUP.

En el presente capítulo se desarrollarán, a nivel general, las principales actividades de las últimas cuatro (4) fases del CMP, identificadas en la siguiente imagen con el signo ✓:

Figura 19 Figura 2 El modelo CMP® de la Universidad de Navarra – Cuatro (4) últimas fases.



Fuente: Elaboración propia con base en Berenguer J.M. y Ramos-Yzquierdo J.A. (2008, p.17)

El principal resultado de este capítulo será la definición del proceso TO-BE, la formulación de las pautas para su implantación y los mecanismos de control y de mejora continua.

#### 4.1 FASE 5 – DISEÑAR

Durante esta fase se realiza la validación de la alineación de los requerimientos del proceso con las expectativas y adicionalmente se realiza el diseño de los subprocesos.

#### 4.1.1 Alineación entre requerimientos y expectativas

Al entrar en la fase de Diseño, el CMP recomienda hacer una comprobación de la alineación entre los requerimientos del proceso<sup>33</sup> y las expectativas<sup>34</sup> de los interesados tanto a nivel interno (la empresa) como a nivel externo (los clientes), con el fin de validar el grado de alineación del nuevo proceso con las expectativas. La Tabla 11 muestra el grado de alineación de los requerimientos del nuevo proceso contra las expectativas internas y externas; si un requerimiento muestra alineación con una expectativa, se indica con una “x”; los conteos de las “x” se totalizan en la última fila y última columna de la tabla.

Tabla 11 Alineación Requerimientos vs. Expectativas

REQUERIMIENTOS	EXPECTATIVAS									
	Externas					Internas				
	Calidad Percibida	Calidad medida	Entregas a tiempo	Adherencia proceso	Eficiencia Recursos	Rentabilidad	Total	%		
1 Proveer un marco único y común para los subprocesos de la gestión de Servicios y de Proyectos.	x	x		x	x	x	5	▲ 11%		ALTO
2 Soportar las tecnologías de las dos versiones activas actualmente en Softniche Corp.: Niche V2 y Niche V3.	x	x	x	x	x	x	6	▲ 13%		ALTO
3 Incluir las mejores prácticas en ingeniería de software en la industria.	x	x	x	x	x	x	6	▲ 13%		ALTO
4 Definir sin ambigüedad los roles del proceso de ingeniería de software, identificando las actividades de las que son responsables y los artefactos de los que son dueños.	x	x	x	x	x	x	6	▲ 13%		ALTO
5 Soportar tecnología orientada a objetos, integrando prácticas y técnicas orientadas a objetos.		x	x	x			3	▼ 6%		BAJO
6 Obtener criterios de aceptación y verificar calidad de los requerimientos a partir de las necesidades del cliente.	x	x		x	x	x	5	▲ 11%		ALTO
7 Aportar artefactos para documentar técnica y funcionalmente el producto desarrollado.	x	x		x	x	x	5	▲ 11%		ALTO
8 Aportar guías de desarrollo de software.	x	x	x	x	x	x	6	▲ 13%		ALTO
9 Incluir capacitaciones en procesos, ingeniería de software y metodologías, durante la implantación del nuevo proceso.	x	x		x	x	x	5	▲ 11%		ALTO
Total	8	9	5	9	8	8	47			
%	▲ 17%	▲ 19%	▼ 11%	▲ 19%	▲ 17%	▲ 17%				
	ALTO	ALTO	MEDIO	ALTO	ALTO	ALTO				

Fuente: Elaboración propia, 2019

Por un lado, analizando la última fila de totales, se puede ver que el nuevo proceso, para todos sus requerimientos tiene un grado de alineación **ALTO** ( $\geq 17\%$ ) con la mayoría de las expectativas, con excepción de la expectativa de “Entregas a tiempo” que tiene una alineación

<sup>33</sup> Los requerimientos del proceso se definieron en el numeral 2.5.3.5 Lista de los requisitos de mejora.

<sup>34</sup> Las expectativas se toman de la declaración de objetivos del *Project Charter* (numeral 2.4).



**MEDIO** (11%); lo anterior indica que la entrega oportuna no depende exclusivamente del proceso de ingeniería, sino de otros factores como la capacidad de los recursos, por ejemplo. Por el otro lado, analizando la última columna de totales, la mayoría de los requerimientos del nuevo proceso (todos menos el 5) se alinean en grado **ALTO** ( $\geq 11\%$ ) con todas las expectativas internas y externas; este hecho permite evidenciar que el requerimiento 5, por sí mismo, no contribuye directamente a la satisfacción de las expectativas, sino que se refiere a una característica tecnológica que debe estar acompañada, por ejemplo, de un buen diseño de la arquitectura de software<sup>35</sup>, lo cual afectará las expectativas de manera indirecta a través de la calidad del producto.

El análisis presentado permite asegurar que el nuevo proceso está alineado con las expectativas internas y externas.

#### **4.1.2 Diseño de los subprocesos del proceso TO-BE**

Una de las primeras decisiones importantes de diseño ha sido determinar que, más que la caracterización de un proceso que articule los diferentes subprocesos de ingeniería, lo que se requiere es un conjunto de buenas prácticas definidas a través de los subprocesos de un ciclo de vida típico (en cascada) de ingeniería de software.

Realmente, en las operaciones, cuando se solicita un mantenimiento o un proyecto, se instancia el proceso de Gestión de Servicios o el proceso de Gestión de Proyectos; por lo cual, las buenas prácticas de ingeniería deben integrarse dentro de las actividades de dichos procesos.

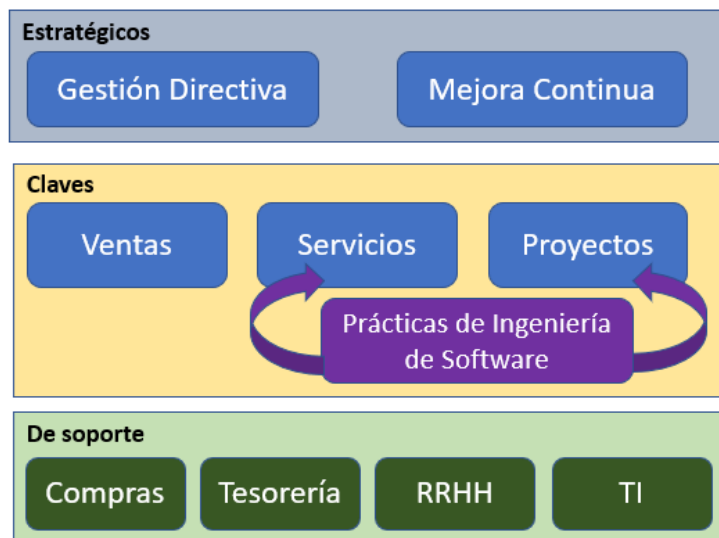
Bajo esta premisa, el nuevo mapa de procesos lucirá como el de la Figura 20.

---

<sup>35</sup> Este requerimiento coincide con la necesidad de alineación del nuevo proceso con las tecnologías usadas para el desarrollo del producto y con las políticas ya establecidas en materia de tecnología.



Figura 20 Nuevo Mapa de Procesos de Softniche Corp.



Fuente: Elaboración propia, 2019.

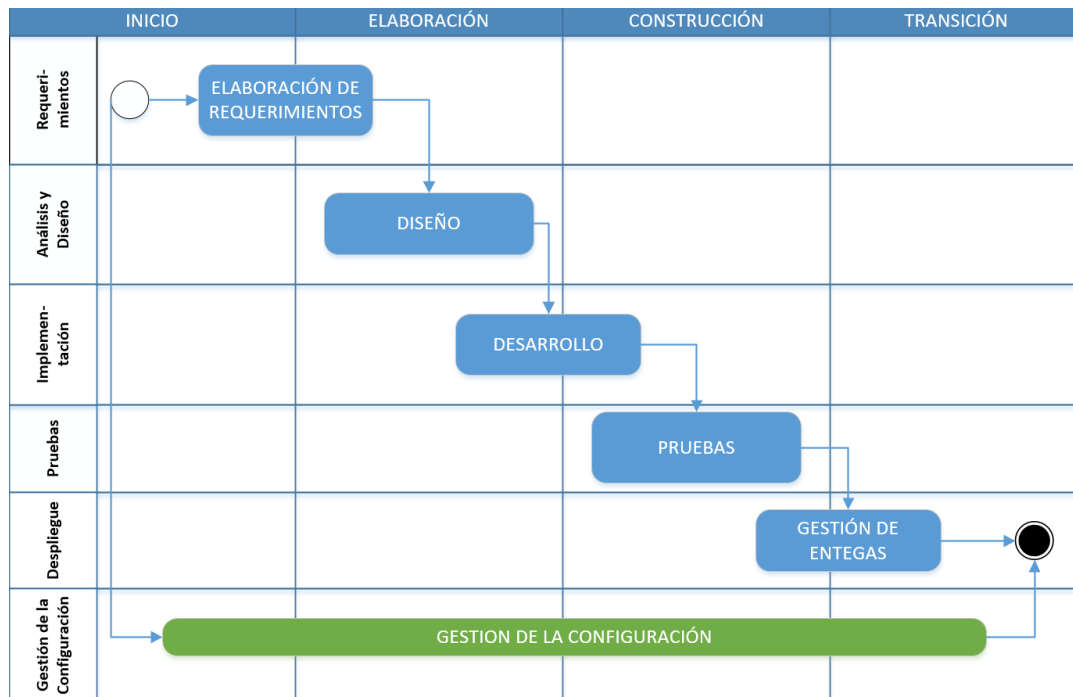
### Diagramas del proceso TO-BE

El ciclo de vida de desarrollo de software en cascada luce como el de la Figura 21<sup>36</sup>. Inicia con el levantamiento y elaboración de los requerimientos de los usuarios, generando un diseño funcional; a continuación, se realiza el diseño técnico que incluye el diseño de la arquitectura de software; tras el diseño técnico se hace la construcción o desarrollo del software; luego el software se debe probar generando una certificación de la calidad; y finalmente se generan y se organizan los entregables del software. Transversal a todo el ciclo de vida se realiza la gestión del código fuente en todas sus ramas de desarrollo, a lo que también se le llama la gestión de la configuración.

---

<sup>36</sup> Los diferentes subprocesos ocurren durante varias fases: la fase de inicio en la que se levantan las necesidades de los usuarios y se comienza la preparación del repositorio de código fuente; la segunda etapa en la que se terminan de depurar las necesidades de los usuarios y se elaboran los diseños funcionales y técnicos; en la tercera etapa de construcción se inicia el desarrollo de la solución, se prueba el sistema y se inicia la preparación de los entregables; finalmente, en la fase de transición, se termina la construcción de los entregables generando la línea base de producción.

Figura 21 Ciclo de Vida de Desarrollo de Software - Modelo en Cascada.



Fuente: Elaboración propia, 2019.

Siguiendo el mismo modelo, se han definido los siguientes cinco (5) subprocesos que conforman las Prácticas de ingeniería de software:

1. Subproceso de Elaboración de Requerimientos.
2. Subproceso de Diseño y Desarrollo.
3. Subproceso de Pruebas.
4. Subproceso de Gestión de Entregas.
5. Subproceso de Gestión de la Configuración.

#### 4.1.2.1 Los roles de las prácticas de ingeniería de software

Lo primero que se definió fueron los roles que actúan en los subprocesos. Para cada rol se hizo una comparación con el rol correspondiente en la metodología RUP, de manera que se pudieran revisar y complementar las responsabilidades y actividades que desempeña cada uno de los roles de las Prácticas de Ingeniería.

Tabla 12 Roles de Ingeniería de Softniche Corp. vs. roles de RUP.

Rol Ingeniería Softniche Corp.	Rol RUP
Arquitecto	Arquitecto
Técnico	Diseñador
Funcional o Analista Funcional	Analista de Sistemas o Especificador de Requerimientos

Especialista de Negocios	Analista de procesos del negocio
Líder de Pruebas	Líder de Pruebas
Analista de Pruebas	Analista de Pruebas
Tester	Tester
Desarrollador	Desarrollador
Gestor de entregables	Desarrollador
Responsable de entregables	Desarrollador
Líder de proyecto o de Servicio	No disponible. No es propiamente rol de ingeniería.

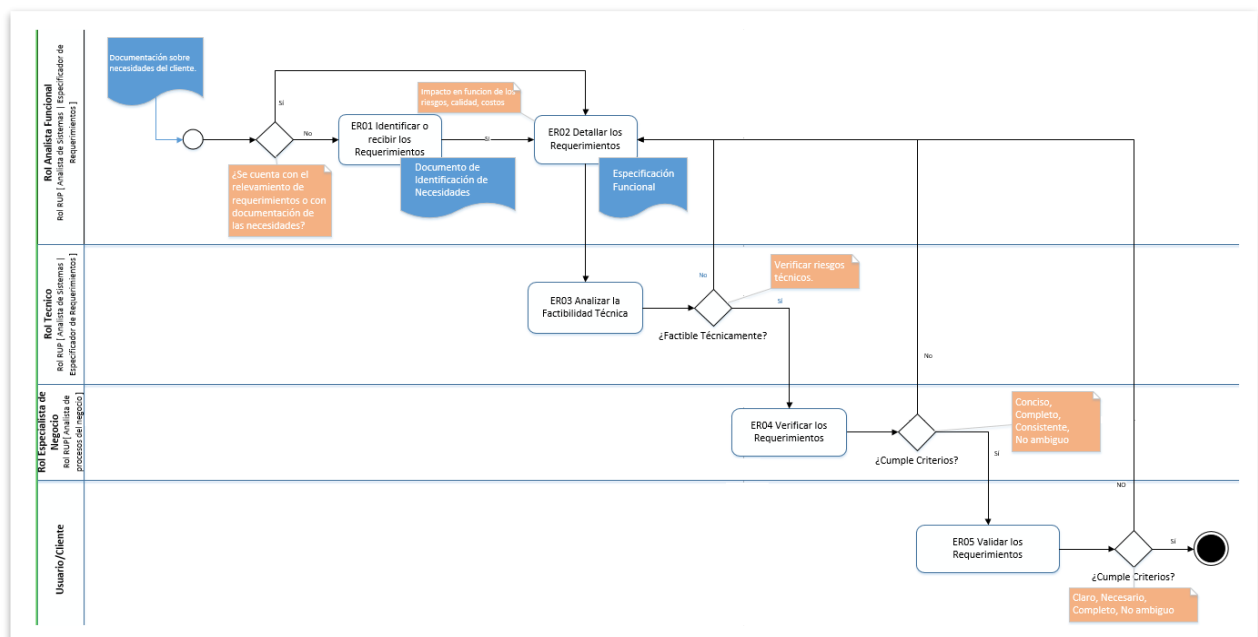
*Fuente: Elaboración propia, 2019.*

En el **ANEXO 1 – Roles de las prácticas de ingeniería de software**, se presenta la descripción de los diferentes roles aquí relacionados.

A continuación se presenta el diseño de cada uno de los diagramas de flujo de los nuevos subprocesos. En este documento no se incluirá el detalle de la caracterización de los subprocesos, en cambio, se listarán las mejores prácticas de la industria de software que han sido añadidas, complementadas o mejoradas en cada uno de los flujos.

#### 4.1.2.2 Subproceso de Elaboración de Requerimientos

*Figura 22 Subproceso de Elaboración de Requerimientos*



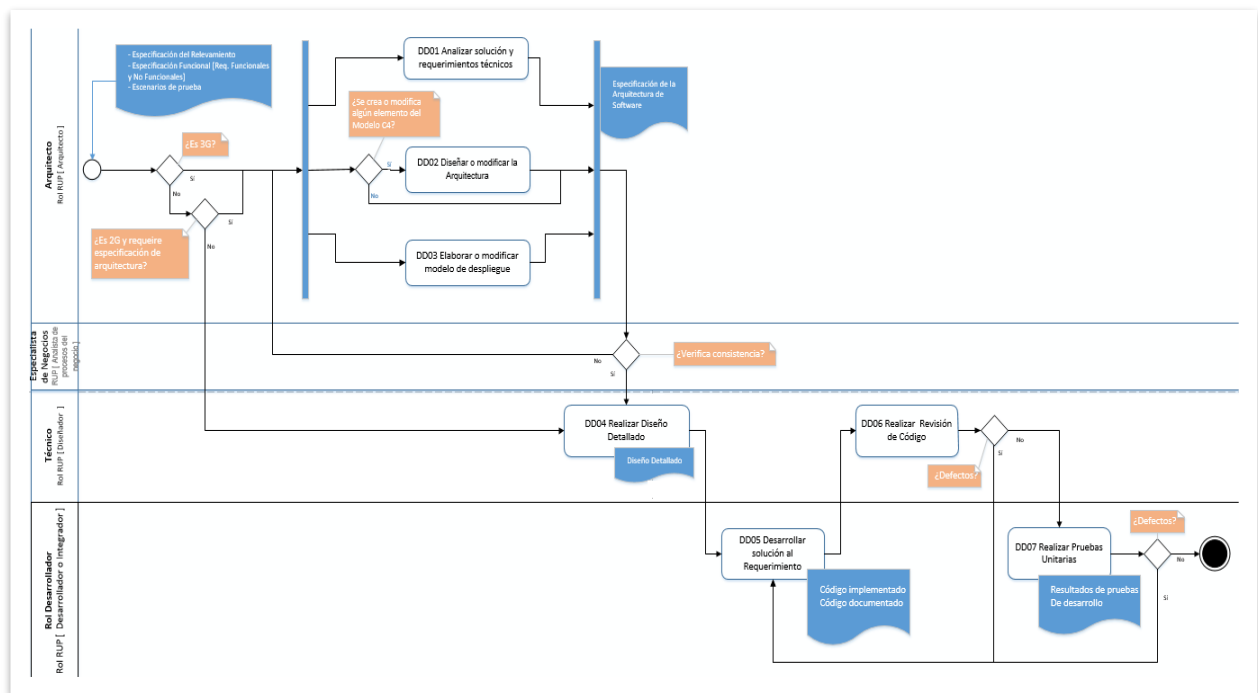
*Fuente: Elaboración propia, a partir de información de Sofniche Corp, 2019.*

En este subproceso se han incluido las siguiente mejores prácticas de la industria de software:

- Se incluye levantamiento de requerimientos no funcionales y de criterios de aceptación de los requerimientos (ER01).
- Chequeo de calidad de las necesidades formuladas por los usuarios (ER02).
- Chequeo de impacto transversal de los requerimientos (ER02).
- Casos de Uso y UML en Especificación Funcional (ER02).
- Análisis y chequeo de factibilidad técnica – Requerimientos no funcionales (ER03).
- Chequeo de calidad de los requerimientos (ER04).
- Validación de los requerimientos por parte del Cliente (ER05).

#### 4.1.2.3 Subproceso de Diseño y Desarrollo

Figura 23 Subproceso de Diseño y Desarrollo



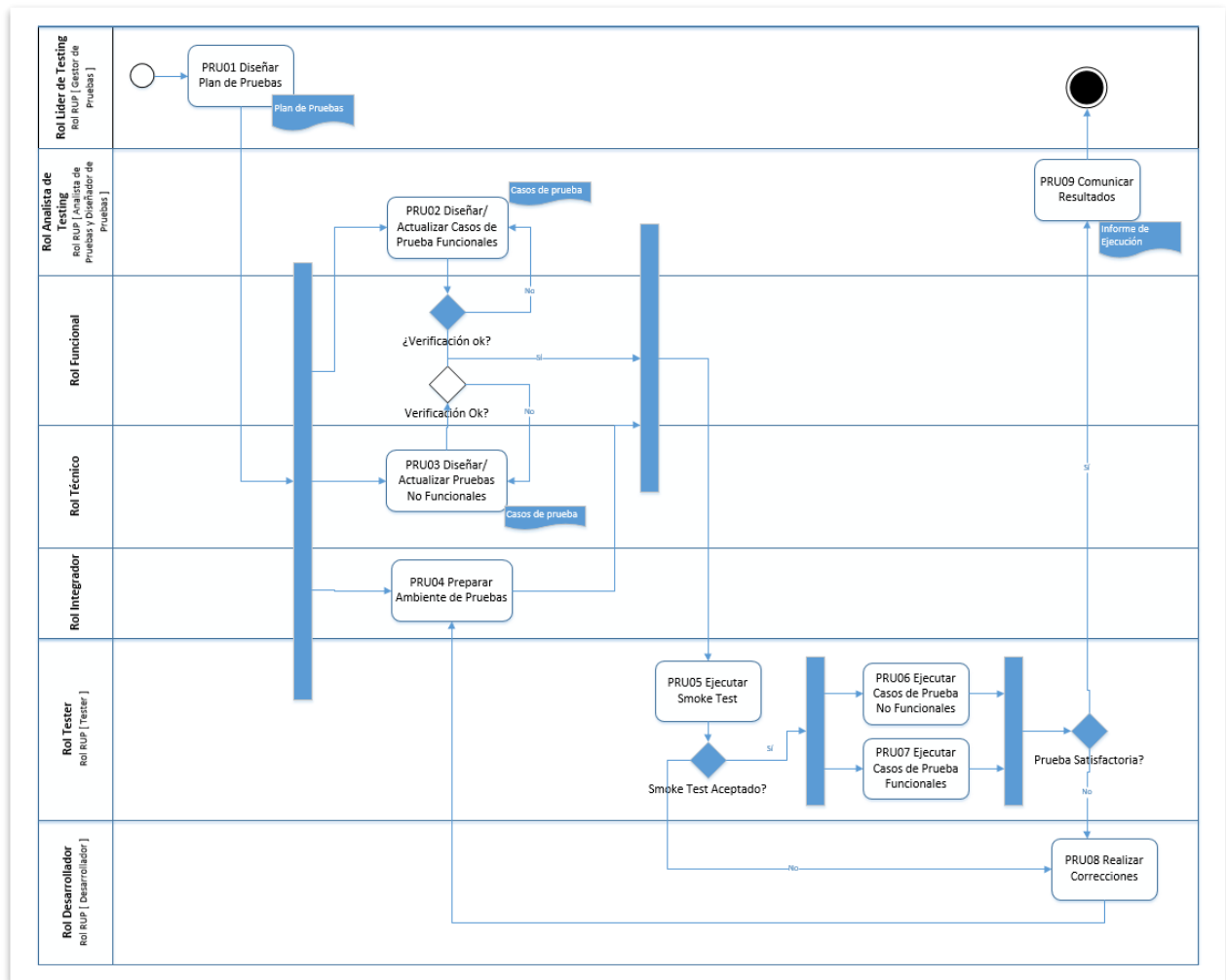
Fuente: Elaboración propia, a partir de información de Sofniche Corp, 2019.

Las siguientes mejores prácticas de la industria de software, han sido incluidas en este subproceso:

- Diseño o especificación de la arquitectura de software (DD01-DD03).
- Diseño detallado utilizando notación UML (DD04).
- Revisiones de código fuente o *code reviews* (DD06).
- Pruebas unitarias o de desarrollador sobre la línea base de desarrollo (DD07).

#### 4.1.2.4 Subproceso de Pruebas

Figura 24 Subproceso de Pruebas



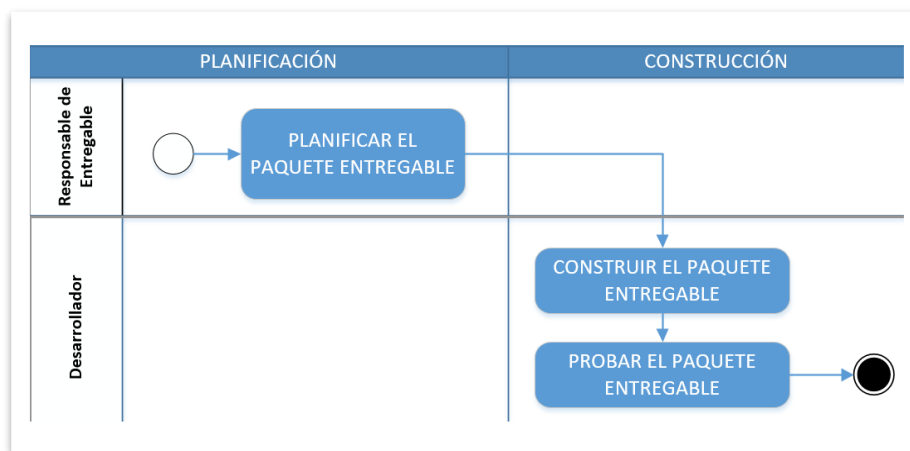
Fuente: Elaboración propia, a partir de información de Sofnische Corp, 2019.

Las siguientes son las mejores prácticas de la industria de software que se incluyeron en este subproceso:

- Elaboración de un Plan de Pruebas (PRU01).
- Diseño y ejecución por ciclos de pruebas funcionales, no funcionales y *Smoke Tests* (PRU02/03/05/06/07).
- Preparación de ambientes (PRU04).
- Corrección de defectos (PRU08).
- Comunicación de resultados (PRU09).

#### 4.1.2.5 Subproceso de Gestión de Entregas

Figura 25 Subproceso de Gestión de Entregas



Fuente: Elaboración propia, a partir de información de Sofniche Corp, 2019.

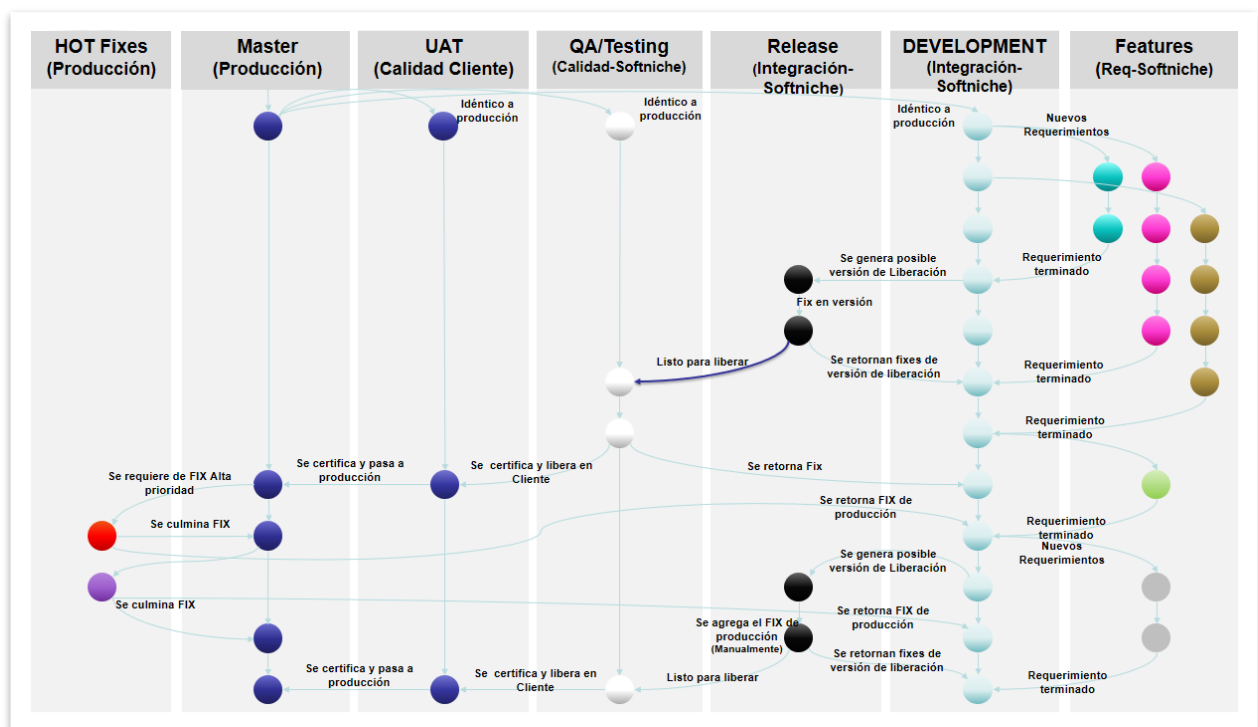
Este subproceso incluye como mejores prácticas de la industria de software, la planificación de los entregables, su construcción y sus pruebas de instalación y de *rollback* (reversión).

#### 4.1.2.6 Subproceso de Gestión de la Configuración

Este subproceso se fundamenta en **Gitflow**<sup>37</sup> como modelo de ramificaciones para la gestión de la configuración del código fuente del software. El flujo presentado define las diferentes posibilidades de ramificación del código fuente y la casuística para pasar de una rama a otra, manteniendo el gobierno del código fuente.

<sup>37</sup> Este popular modelo, propuesto originalmente por el gurú Vincent Driessen, ha llegado a ser de uso común en la industria del software.

Figura 26 Modelo de ramificaciones con Gitflow



Fuente: Elaboración propia, a partir de información de Sofniche Corp, 2019.

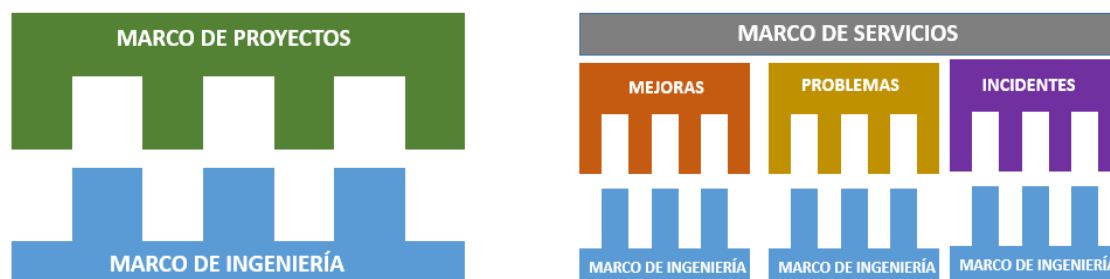
El subproceso incluye una guía de uso de la herramienta de versionado (GIT<sup>38</sup>), una guía de etiquetado del versionamiento y una presentación sobre Gitflow.

#### 4.1.3 Integración con los marcos de Gestión de Servicios y de Proyectos

Tanto para el Marco de Gestión de Proyectos como para el Marco de Gestión de Servicios, el Marco de las Mejores Prácticas de Ingeniería ha sido definido de manera independiente pero integrada a cada uno de ellos. La Figura 27 ilustra cómo los diferentes marcos metodológicos operan como un marco integral junto con el de ingeniería; no obstante, cada marco de procesos contiene sus propios subprocesos y actividades.

<sup>38</sup> GIT es el software de control de versiones diseñado por el gurú Linus Torvalds, quien inició el desarrollo del *kernel* de Linux.

Figura 27 Integración de los Marcos Metodológicos Softniche Corp.



Fuente: Elaboración propia, 2019.

Para un mayor entendimiento, en el **ANEXO 3 – EJEMPLOS DE INTEGRACIÓN DE ACTIVIDADES ENTRE MARCOS METODOLÓGICOS**, se presentan un par de ejemplos de cómo fueron integradas algunas actividades del subproceso de mejoras del marco de Gestión de Servicios, con las actividades propias del marco de ingeniería.

## 4.2 FASE 6 – IMPLANTAR

Para dar inicio con la implantación, se seleccionó uno de los clientes con mayor solidez en sus procesos de ingeniería, con el fin de no generar un impacto muy grande en la operación.

Por otra parte, se definieron las siguientes actividades de implantación para ser realizadas previamente al uso de los subprocesos en la operación.

Tabla 13 Actividades para la implantación de las Prácticas de Ingeniería

Dimensión	Actividad
<b>Procesos</b>	<ul style="list-style-type: none"> <li>Oficializar la caracterización de los subprocesos.</li> </ul>
<b>Tecnología</b>	<ul style="list-style-type: none"> <li>Levantar inventario de productos instalados en el cliente.</li> <li>Migrar repositorios de versionado a GIT.</li> </ul>
<b>Conocimientos</b>	<ul style="list-style-type: none"> <li>Producir documentación y realizar formación en temas de Versionado y Ambientes</li> <li>Realizar formaciones en: Marcos de Calidad, RUP, CMMI, UML, Casos de Uso, Marco de ingeniería de software y subprocesos de Servicios integrados con el marco de Ingeniería.</li> </ul>

Fuente: Elaboración propia, 2019.

De acuerdo con el plan inicial del *Project Charter*, estas actividades se planificaron para el último trimestre de 2019 y darán inicio con el primer cliente seleccionado, a manera de implantación piloto.



### 4.3 FASE 7 – CONTROLAR

Se nombraron como dueños de los nuevos subprocesos del Marco de Prácticas de Ingeniería a los dueños de los procesos de Gestión de Servicios y de Gestión de Proyectos, por estar integradas las prácticas de ingeniería con los marcos de gestión de la operación.

Los siguientes son los indicadores que serán medidos en el nuevo marco (Tabla 14). Para complementar la información véase también la Tabla 8 y Tabla 9 en el numeral 2.5.4 (*Dashboard* del proceso y medición del proceso AS-IS).

*Tabla 14 Indicadores de control del proceso TO-BE*

Indicador (Todos mensuales)	Descripción (Cálculo por cliente y total)	Objetivo por alcanzar
% Adherencia al proceso	Se aplica lista de chequeo con ítems a cumplir.	> 80%
Esfuerzo de no-calidad	Con base en las horas de solución de devoluciones.	< 10%
% de entrega oportuna	% de Proyectos o Mantenimientos entregados a tiempo.	> 80%
Indicador de roles de ingeniería	% de personas con calificación satisfactoria en las actividades ejecutadas de su rol.	> 70%
Indicador de calidad Interna	# Bugs internos (en pruebas internas) sobre horas totales del proyecto o mantenimiento.	< 0,25
Indicador de calidad Externa	# Bugs externos (en pruebas UAT) sobre horas totales del proyecto o mantenimiento.	< 0,125
Ineficiencia económica de los roles de ingeniería	(Costo teórico de los roles menos costo real de los roles) entre costo teórico.	> -20%

*Fuente: Elaboración propia, 2019.*

### COMPROBACIÓN DE MEJORES PRÁCTICAS VS. INDICADORES

Con el propósito de hacer una última comprobación, a continuación se presenta una matriz con el cruce de cada una de las mejores prácticas en cada subproceso (numeral 4.1.2 Diseño de los subprocesos del proceso TO-BE), con los diferentes indicadores definidos. De esta manera se puede saber si todos los indicadores están cubiertos por alguna práctica y por lo tanto si podrá ser medido. Así mismo se puede comprobar si cada práctica afecta por lo menos un indicador.

Figura 28 Prácticas de Ingeniería Vs. Indicadores de Control

PROCESO/ SUBPROCESO	PRÁCTICA	INDICADORES DE CONTROL						
		% Adherencia al proceso	Esfuerzo de no calidad	% de entrega oportuna	Indicador de roles de ingeniería	Indicador de calidad interna	Indicador de calidad externa	Ineficiencia económica de los roles
Gestión de Servicios y Gestión de Proyectos	Planificación, estimación control del tiempo		x					✓
Subproceso de Elaboración de Requerimientos	Requerimientos no funcionales y criterios de aceptación de los requerimientos (ER01).	x		x		x		✓
	Chequeo de calidad de las necesidades formuladas por los usuarios (ER02).	x		x		x		✓
	Chequeo de impacto transversal de los requerimientos (ER02).	x		x		x		✓
	Casos de Uso y UML en Especificación Funcional (ER02).	x		x		x		✓
	Análisis y chequeo de factibilidad técnica – Requerimientos no funcionales (ER03).	x		x		x		✓
	Chequeo de calidad de los requerimientos (ER04).	x		x		x		✓
	Validación de los requerimientos por parte del Cliente (ER05).	x		x		x		✓
Subproceso de Diseño y Desarrollo	Diseño o especificación de la arquitectura de software (DD01-DD03).	x		x		x		✓
	Diseño detallado utilizando notación UML (DD04).	x		x		x		✓
	Revisiones de código fuente o code reviews (DD06).	x		x		x		✓
	Pruebas unitarias o de desarrollador sobre la línea base de desarrollo (DD07).	x		x		x		✓
Subproceso de Pruebas	Elaboración de un Plan de Pruebas (PRU01).	x		x	x	x		✓
	Diseño y ejecución por ciclos de pruebas funcionales, no funcionales y Smoke Tests (PRU02/03/05/06/07).	x		x	x	x		✓
	Preparación de ambientes (PRU04).	x		x	x	x		✓
	Corrección de defectos (PRU08).	x	x	x	x	x	x	✓
	Comunicación de resultados (PRU09).	x		x	x	x		✓
Subproceso de Gestión de Entregables	Planificación de los entregables.	x		x		x		✓
	Construcción del entregable	x		x		x		✓
	Pruebas de instalación y de rollback (reversión).	x		x		x		✓
Subproceso de Gestión de la Configuración	Gobierno del código fuente	x		x		x		✓
		✓	✓	✓	✓	✓	✓	

Fuente: Elaboración propia, 2019.

En la Figura 28 se evidencia cómo todas las prácticas afectan algún indicador de control y cómo todos los indicadores de control son afectados por alguna práctica. Es importante anotar que para el indicador de **% de entrega oportuna**, las prácticas que contribuyen a su actualización pertenecen a los marcos dentro de los que fue integrado el marco de ingeniería; de esta manera, aunque el control del tiempo no se diseñó específicamente dentro de las prácticas de ingeniería, sí se gestiona a través de los otros marcos metodológicos de Servicios y Proyectos que constituyen una unidad con el marco de ingeniería.

#### 4.4 FASE 8 – CAPACITAR

Capacitar en este caso consiste en “habilitar a la organización para una eficiente gestión de sus procesos mejorados, así como para desempeñar futuras mejoras” (Berenguer & Ramos-Yzquierdo, 2004, p.225).

En relación con la madurez de las prácticas de ingeniería, en la medida en que la adherencia a los procesos crezca, se podrá pensar en realizar una nueva evaluación sobre la capacidad

de los procesos y, más aún, ampliar la evaluación a todas las demás Áreas de Proceso que conforman el marco CMMI-DEV, con el fin de seguir detectando oportunidades de mejora que puedan acercar a Sofnische Corp. a la posibilidad de enfrentar en el futuro un *Scampi* para la certificación CMMI-DEV en algún nivel superior o igual a tres (3).

Por otra parte, en cuanto a la redefinición del sistema de evaluación de desempeño de los empleados, se tiene previsto incluir dentro de los indicadores de desempeño de los recursos asignados a roles de ingeniería, así como de sus líderes, los siguientes dos indicadores:

- Indicador de roles de ingeniería, que mide que cada rol haga lo que tiene que hacer de acuerdo con sus conocimientos y habilidades.
- Indicador de calidad Interna, que evalúa la buena calidad del desarrollo.

Se llega así al final de este capítulo en el que se ha planteado la propuesta de solución. Antes de iniciar el diseño se ha hecho una verificación de la alineación entre los requerimientos y las expectativas; habiendo verificado la no existencia de desviaciones, se continuó con el diseño del nuevo proceso incluyendo sus diagramas y su integración con los procesos misionales de Gestión de Servicios y de Proyectos. Se continuó con los temas de la implantación, formulando los indicadores del nuevo proceso y se hizo una verificación final de los indicadores contra las prácticas incluidas en los subprocesos diseñados, de manera de detectar posibles desviaciones. Finalmente, se ha esbozado un camino para buscar la madurez del proceso de ingeniería y alinear las evaluaciones de desempeño de los empleados con los nuevos indicadores definidos. Se han alcanzado así los objetivos del itinerario planteado por las cuatro (4) fases finales del CMP.

En el próximo capítulo se analizarán los costes y los beneficios de la implantación de la mejora y, finalmente, el capítulo 6, concluirá este TFM con las principales enseñanzas y descubrimientos realizados.

## **5 ANÁLISIS COSTO-BENEFICIO DE LA MEJORA**

Después de un profundo análisis de la situación de Sofnische Corp., y de sus procesos de fabricación de software, y tras una elaborada propuesta de solución alcanzada en el capítulo anterior, al iniciar este capítulo se tiene ya claridad sobre los aspectos metodológicos de la propuesta de mejora y su implantación.

Conviene ahora abordar el análisis costo-beneficio de implantar dicha mejora. De acuerdo con esto, se presentarán los costos estimados de la implantación del proceso de mejora frente al

estimativo de ahorros en costos por efectos de la reducción de la no-calidad. Las cifras presentadas son tan sólo una estimación de algunos de los factores más relevantes, pero no constituyen un cálculo exhaustivo de todos los elementos<sup>39</sup>.

Lo importante en este capítulo será demostrar, como lo afirma Crosby (1987), que "la calidad no cuesta (es gratis), pero nadie va a saberlo si no existe algún tipo de sistema aprobado de medición" (p.100). De hecho, los costos de la implantación de la mejora propuesta son, bastante más inferiores al coste de la no-calidad; de esta manera se puede afirmar que lo que realmente cuesta, es la no-calidad, porque cuando las cosas se hacen bien a la primera, no se tendrán los costes adicionales que se general por todos los conceptos de la no-calidad.

Las cifras de la no-calidad han sido estimadas con base en los valores ya presentados sobre el comportamiento de la compañía en 2018 y 2019 (2.1.2 El precio de la calidad y la oportunidad). Las cifras del costo de la implantación se estimaron con base en las actividades previstas de la implantación relacionadas en el *Project Charter* (2.4 FASE 3 – PLANIFICAR) y en un estimado de los costos de los recursos empleados.

En primer lugar, se presentan los costos de la no-calidad asociados con los clientes que abandonan, las multas por incumplimientos en proyectos, los retrabajos en la solución de incidencias en servicios de mantenimiento y la utilidad negativa del ejercicio debido a sobrecostos de los proyectos. La Tabla 15 resume los costes de la no-calidad en el momento previo a la implantación de la mejora.

*Tabla 15 Costes de la no-calidad 2018-2019 (Valores en dólares)*

Concepto	2018	2019	Total	Justificación
<b>Clientes que abandonan</b>	\$ 42.008	\$ 88.964	\$ 130.971	5% (2018) y 9% (2019) de los ingresos por Mantenimiento <sup>40</sup> .
<b>Multas por incumplimientos</b>	\$ 90.000	\$ -	\$ 90.000	
<b>Retrabajos en solución de incidencias</b>	\$ 86.937	\$ 44.248	\$ 131.185	15% (2018) y 7% (2019) de los costos de servicios de Mantenimiento <sup>41</sup> .
<b>Utilidad negativa (Pérdida) en proyectos</b>	\$ 340.000	\$ 243.000	\$ 583.000	15% (2018) y 50% (2019)
	<b>\$ 558.944</b>	<b>\$ 376.212</b>	<b>\$ 935.156</b>	

*Fuente: Elaboración propia, 2019. A partir de datos de Softniche Corp.*

<sup>39</sup> Hay dos razones por las que no es posible contar con toda la información. Por una parte, los motivos de confidencialidad de la empresa en la que se implanta la mejora; por otra la parte, la falta de mecanismos de recolección de la información y la no disponibilidad de información histórica relevante sobre los temas de la calidad y la no-calidad.

<sup>40</sup> Los ingresos por servicios de Mantenimiento fueron de \$ 840.151 y \$ 988.485, para 2018 y 2019 respectivamente (Datos de Softniche Corp.).

<sup>41</sup> Los costos de servicios de Mantenimiento fueron de \$ 579.579 y \$ 632.121, para 2018 y 2019 respectivamente (Datos de Softniche Corp.).

En relación con los conceptos de la no-calidad, el total de costes en los que ha incurrido la empresa en los últimos dos años asciende a \$935.156.

En segundo lugar, se presenta la estimación de los costes de la implantación de la mejora, asumiendo que la implantación tendrá una duración de dos años a partir del 2020<sup>42</sup>.

La Tabla 16 muestra el detalle de la estimación de los costes de las actividades de implantación del proceso de mejora, incluyendo los tiempos de análisis, diseño, implantación (formación y acompañamiento) y control durante los primeros dos años.

*Tabla 16 Costes de la implantación del proceso de mejora (Valores en dólares)*

Concepto	Coste 2020	Coste 2021	Total	Justificación
Coste líder de procesos de Ingeniería	\$ 18.000	\$ 9.000	\$ 27.000	Esfuerzo 960 horas (2020) y 480 horas (2021)
Coste líder de procesos de Servicios	\$ 6.000	\$ 3.000	\$ 9.000	Esfuerzo 320 horas (2020) y 160 horas (2021)
Capacitación de equipos en Proceso de Ingeniería	\$ 30.000	\$ 30.000	\$ 60.000	Esfuerzo 8 horas al año x 200 personas (1600 horas)
Capacitación de equipos en versionado	\$ 12.000	\$ 6.000	\$ 18.000	Esfuerzo 80 horas (2020) y 40 horas (2021)
Coste Arquitectura (esfuerzo en versionado)	\$ 9.000	\$ 9.000	\$ 18.000	Esfuerzo 480 horas (2020) y 480 horas (2021)
Auditorías	\$ 2.250	\$ 2.250	\$ 4.500	Esfuerzo 120 horas (2020) y 120 horas (2021)
	<b>\$ 77.250</b>	<b>\$ 59.250</b>	<b>\$ 136.500</b>	

*Fuente: Elaboración propia, 2019. A partir de datos de Softniche Corp.*

Obsérvese que el coste total por concepto de la implantación de la mejora al final del año 2021 alcanza los \$136.500.

En un tercer momento se debe calcular el coste de la no-calidad durante los dos primeros años de la implantación, ya que no se puede pretender que en el momento cero de la implantación se tenga un coste cero de la no-calidad. Conviene comenzar estimado las metas de costes de la no-calidad propuestas para el final del período de dos años (tras la implantación); a este respecto la Tabla 17 muestra los costes y los objetivos perseguidos para el final del 2021:

---

<sup>42</sup> De acuerdo con el *Project Charter* las actividades de implantación se planificaron para el último trimestre de 2019 y han iniciado con un solo cliente (a manera de piloto). Por simplicidad se asume su inicio en 2020 (primeros días).

Tabla 17 Objetivos de la no-calidad para 2021 (Cifras en dólares)

Concepto	Meta	Coste 2021	Justificación
<b>Clientes que abandonan</b>	0%	\$ 0	Cero abandonos. Coste cero.
<b>Multas por incumplimientos</b>	0	\$ 0	Cero multas. Coste cero.
<b>Retrabajos en solución de incidencias</b>	5%	\$ 34.233	5% del total de costes de mantenimiento previstos para 202
<b>Utilidad negativa (Pérdida)</b>	0%	\$ 0	Cero pérdidas. Coste cero.
<b>Total</b>		<b>\$ 34.233</b>	

Fuente: Elaboración propia, 2019. A partir de datos de Softniche Corp.

Con estos objetivos planteados para fines de 2021, se puede hacer una proyección de la manera en que disminuirán los costes desde el inicio de la implantación de la mejora en 2020 hasta el final de plazo de la implantación en 2021, considerando al mismo tiempo los costes de la implantación en el período. A este respecto, la Tabla 18 muestra la evolución de los costes en el período de los dos años 2020 y 2021, y tiene la siguiente estructura:

- La primera fila muestra los costes de la no-calidad, los cuales inician con los costes acumulados de los dos años anteriores (2018 y 2019); al final del primer año se proyecta una disminución en un 48.2%<sup>43</sup> de estos costes de la no-calidad hasta llegar al objetivo mínimo al final de 2021 de \$34.233.
- La segunda fila describe el movimiento de los costes de la implantación que serán levemente inferiores en el segundo año.
- Las últimas dos filas muestran el total de costes por año y el porcentaje de ahorro anual.

Tabla 18 Evolución de los costes por año (Cifras en dólares)

Costes	2019	2020	2021	Total
<b>No-Calidad</b>	\$ 935.156	\$ 484.695	\$ 34.233	<b>\$ 1.454.084</b>
<b>Implantación</b>	\$ -	\$ 77.250	\$ 59.250	<b>\$ 136.500</b>
<b>Total Costes</b>	<b>\$ 935.156</b>	<b>\$ 561.945</b>	<b>\$ 93.483</b>	<b>\$ 1.590.584</b>
<b>Ahorro por año</b>	<b>0%</b>	<b>40%</b>	<b>90%</b>	En relación con el coste al final de 2019

Fuente: Elaboración propia, 2019. A partir de datos de Softniche Corp.

Nótese que el total de costes al final del período de los dos años alcanza \$1.590.584, que es un valor acumulado bastante significativo. Obsérvese no obstante que, de no haber implantado la mejora, el coste total acumulado hubiera seguido creciendo hasta alcanzar, por lo menos, el tripe del valor del inicio del período, es decir \$2.805.468. En relación con esta cifra acumulada, se estima un ahorro del 57% al final de los dos años. Así mismo, el ahorro

<sup>43</sup> Es decir, un ahorro de \$ 450.461, correspondiente a la diferencia entre los costes de 2109 y 2020.

efectivo por año alcanzará al final del 2021 el 90% de los costes antes de la implantación de la mejora; para el siguiente año, el ahorro superará el 96%.

Merece la pena subrayar, en relación con el retorno de la inversión, que el precio de la inversión (\$136.500 de implantación de la mejora) representa tan sólo el 14.6% del costo de la no-calidad antes de implantar el proceso (\$935.156). Ahora bien, si se asume que, por razones de resistencia al cambio, la mejora no dará sus resultados óptimos hasta pasados dos años y que los costos de la no-calidad irán disminuyendo paulatinamente durante esos dos años, se puede decir que, al final del primer año (2020) el costo se reducirá a la media de \$485.149 y al final del segundo año quedará reducido al mínimo de \$34.233. De esta manera el coste de la inversión estará más que cubierto por los ahorros que se obtengan.

Adicionalmente, el primer año habrá un pequeño costo adicional del 8.2% en relación con los costos de la no-calidad, que corresponde al valor de la implementación de la mejora \$77.250 del primer año. El siguiente año se espera una reducción de los costos hasta del 40%, para finalmente lograr una reducción de los costos del 90% al final del 2021 y del 96% en los primeros meses del 2022.

Hay que recalcar finalmente que, con fundamento en las estimaciones y proyecciones presentadas, se evidencia que la inversión está cubierta ya desde el año primero (segundo semestre) de la implantación debido a los ahorros que se esperan a partir la mejora del nuevo proceso.

## **6 CONCLUSIONES**

1. Este TFM es el resultado de la aplicación de diversos conocimientos adquiridos durante el Máster Universitario en Dirección de Operaciones y Calidad, y de la investigación sobre temas de procesos, calidad e ingeniería de software, entre otros. A través del CMP (Ciclo de Mejora de Procesos) se ha desarrollado con éxito un proyecto de mejora sobre las operaciones en una empresa multinacional de desarrollo de software especializado para un nicho del mercado. De esta manera se ha logrado el objetivo principal de ofrecer una solución de mejoramiento a las prácticas de ingeniería de software incluidas dentro de los procesos claves de la organización, considerando y analizando metodologías, tecnologías, objetivos estratégicos, expectativas internas y del cliente, así como aspectos sobre la calidad, los costos, la eficiencia y los resultados.

2. El estudio ha contemplado integralmente el contexto de la empresa y sus características particulares. En cuanto al contexto externo, la empresa se sitúa en un mercado actual muy competitivo y fuertemente orientado hacia la transformación digital. En cuanto a sus condiciones particulares, cuenta con un único producto de gran tamaño y complejidad, personalizado en sus clientes y que soporta dos tecnologías Cliente/Servidor en *Visual Basic* y *web* en C# (en cuanto a arquitecturas de software y lenguajes de programación). El negocio de la empresa consiste en vender licencias, personalizar el producto en cada cliente a través de proyectos y prestar los servicios de mantenimiento correctivo y evolutivo sobre el software.
3. Así mismo, se han alcanzado los objetivos específicos del trabajo. Se adelantó un detallado diagnóstico inicial, confrontando las prácticas en uso contra marcos de mejores prácticas en ingeniería de Software como son CMMI-DEV y RUP. Se logró articular de manera coherente la propuesta de mejora metodológica integrándola con los procesos misionales de Gestión de Proyectos y Servicios de Softniche Corp. Adicionalmente se ofreció un marco de medición y control en relación con la calidad de los entregables de software y la oportunidad de entrega sobre los plazos convenidos.
4. La realización del CMP ha implicado recorrer ocho (8) etapas, aunque no todas ellas puedan ofrecer resultados evaluables en el momento de entregar este trabajo, debido a la brevedad del tiempo:
  - **Sensibilizar** para obtener el respaldo de la alta dirección, justificando la mejora a partir del análisis FODA.
  - **Identificar** la necesidad de mejores prácticas de ingeniería de software, persiguiendo mayor calidad, entregas oportunas y reducción de costos.
  - **Planificar** la mejora de los procesos realizando un *Project Charter*.
  - **Analizar** la situación, a través de un detallado diagnóstico de las prácticas actuales (*Assessment*).
  - **Diseñar** y estandarizar los subprocesos de las prácticas de ingeniería de software, integrándolas en los marcos de operaciones (Gestión de Servicios y de Proyectos).
  - **Implantar**, a través de la caracterización y oficialización de los subprocesos, de la capacitación de los recursos y del uso de nuevas tecnologías.
  - **Controlar** a través de indicadores de desempeño del proceso y de los roles implicados.



- **Capacitar** a la organización en una cultura de madurez frente a las nuevas prácticas, alineando al mismo tiempo a las personas a través de la evaluación de desempeño.
5. Además del CMP como metodología, este TFM ha involucrado otros marcos metodológicos ampliamente reconocidos y probados en la industria del software, como son CMMI-DEV y RUP, los cuales han servido de fundamento y han sido determinantes para la realización del análisis del proceso AS-IS y el planteamiento del diseño del proceso TO-BE.
  6. Los resultados de la mejora se encuentran en camino. Hasta el momento en que se han escrito estas conclusiones, se ha podido medir la adherencia al proceso en uno de los clientes (con el cual se ha iniciado la implantación, a manera de piloto), obteniendo un porcentaje creciente en tres auditorías realizadas, hasta llegar al 80%. Por otra parte, el indicador de desempeño de los roles de ingeniería se mantiene en un nivel bajo (<40%) debido a restricciones en la capacidad de los equipos de desarrollo. Entre tanto se trabaja también en la estandarización de la recolección de la información para los indicadores de calidad referidos a los defectos del software.
  7. En relación con la implantación, aún hay varios aspectos que ajustar sobre temas de organización y compromiso de líderes y equipos, especialmente por la resistencia al cambio y la cultura organizacional; por lo cual, muy posiblemente se requieran algunos ajustes en el cronograma de la implantación, con el objeto de alcanzar los resultados previstos.
  8. El análisis Coste-Beneficio realizado ha podido esclarecer de manera concluyente, a partir de estimaciones de costes y ahorros esperados, la forma en que se está asegurando el retorno de la inversión por la implantación de la mejora, con base en una proyección a dos años.
  9. Ha sido grande el reto de formular un TFM completo sobre este tema muy especializado, con diversas aristas y complejidades, especialmente considerando la brevedad del tiempo para realizarlo. No obstante, el aprendizaje ha sido también enorme debido a la gran cantidad de dimensiones que hay que tener presentes y profundizar para plantear una solución coherente desde la perspectiva de la gestión de la calidad y de las operaciones. Temas como el liderazgo y la coordinación, la gestión *lean*, el diseño y desarrollo de productos y servicios, la generación de la demanda, la producción, la gestión y el control de costos, los diversos temas alrededor de la calidad y de los procesos, etc., son todos temas que en algún momento del trabajo influyeron en la construcción de las ideas y de los planteamientos finales del TFM.

10. No deja de ser indispensable para el futuro de este proyecto de mejora propuesto, abordar también temas organizacionales relacionados con la coordinación, la cultura y la gestión del conocimiento y, así mismo, los aspectos tan importantes del diseño del producto, la generación de la demanda y el estudio detallado del sistema de costes. Sin embargo, aunque no puedan ser objeto de este documento, seguirán siendo el reto de su autor frente a nuevas circunstancias que se presenten en el provenir.

## 7 BIBLIOGRAFÍA

- Agudelo Tobón, L. F. & Escobar Bolívar, J. (2010). *Gestión por procesos*. Medellín: ICONTEC.
- Báez, C.I. & Suárez, M.I. (2013). *Proceso de desarrollo de software: basado en la articulación de RUP y CMMI priorizando su calidad*. Tunja: Universidad de Boyacá.
- Beltrán Sanz, J., Carmona Calvo, M.A., Carrasco Pérez, R., Rivas Zapata M.A. & Tejedor Panchón, F. (2009). *Guía para una gestión basada en procesos*. Sevilla: Instituto Andaluz de Tecnología. Recuperado de [https://www.euskadi.eus/web01-s2ing/es/contenidos/informacion/bibl\\_digital/es\\_documento/adjuntos/Guia%20para%20una%20gestion-basada-procesos.pdf](https://www.euskadi.eus/web01-s2ing/es/contenidos/informacion/bibl_digital/es_documento/adjuntos/Guia%20para%20una%20gestion-basada-procesos.pdf)
- Berenguer, J.M. & Ramos-Yzquierdo, J.A. (2004). *Dirección de procesos digitales. Crear organizaciones más eficientes* (1ra ed.). Navarra, España: EUNSA.
- Berenguer, J.M. & Ramos-Yzquierdo, J. A. (2008). *Manual de técnicas del CMP - herramientas para la innovación de procesos* (1ra ed.). Navarra, España: EUNSA.
- Berenguer, J.M. (2011). *La fuerza del grupo. Coordinar los procesos, silos, tribus y otros puntos de fricción de las organizaciones*. (1ra ed.). Navarra, España: EUNSA.
- Camisón, C., Cruz, S., & González T. (2006). *Gestión de la Calidad: conceptos, enfoques, modelos y sistemas*. Madrid: Pearson Educación, S. A.
- Chrissis M.B., Konrad, M. & Shurum S. (2011). *CMMI for Development: Guidelines for Process Integration and Product Improvement*. (3ra Edición) - The SEI Series in Software Engineering). USA: Pearson Education Inc.
- Crosby, F.B. (1987) *La calidad no cuesta. El arte de cerciorarse de la calidad*. México: Compañía Editorial Continental, S.A.

- Domínguez Machuca, J.A., Álvarez, M.J., Domínguez Machuca, M.A., García González, S. & Ruiz Jiménez, A. (1995a). *Dirección de operaciones. Aspectos estratégicos en la producción y los servicios*. España: McGraw Hill.
- Domínguez Machuca, J.A., Álvarez, M.J., Domínguez Machuca, M.A., García González, S. & Ruiz Jiménez, A. (1995b). *Dirección de operaciones. Aspectos tácticos y operativos en la producción y los servicios*. España: McGraw Hill.
- Domínguez Machuca, J.A., González Zamora, M.M. & Aguilar Escobar B.G. (2007). Investigación en servicios en el ámbito de la dirección de operaciones. *Cuadernos de Economía y Dirección de la Empresa*. (30), 205-232. Recuperado de <https://www.sciencedirect.com/science/article/pii/S1138575807700798>
- Duggan, E. W. & Reichgelt, H. (2006). *Measuring information systems delivery quality*. USA: Idea Group Publishing.
- Fariña Gómez, B. & González González, Y. (1998). Gestión estratégica de la calidad. Herramientas: una aplicación en el campo sanitario. *Anales de estudios económicos y empresariales*, (13), 275-316. Recuperado de <http://dialnet.unirioja.es/servlet/oaiart?codigo=116409>
- García, M. L. & Bray, O. H. (1997). *Fundamentals of Technology Roadmapping*. Albuquerque: Strategic Business Development Department - Sandia National Laboratories. Recuperado de [https://digital.library.unt.edu/ark:/67531/metadc681205/m2/1/high\\_res\\_d/471364.pdf](https://digital.library.unt.edu/ark:/67531/metadc681205/m2/1/high_res_d/471364.pdf)
- Garmus, D. (2003). IT Metrics and Benchmarking. *Cutter IT Journal*, 16 (3), 2-42. Recuperado de <https://pdfs.semanticscholar.org/598c/941f6bcab10fc484575454c06a83a3eaccd4.pdf>
- Garzas J. (2017). *Peopleware y Equipos Ágiles. Con prácticas De Management 3.0*. Madrid: 233 Grados de TI.
- Hammer, M., (2007). La auditoría de proceso. *Harvard Business Review*, 85 (4), 92-104. Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=2305445>
- Heath C. & Heath D. (2011). *Cambia el chip: Cómo afrontar cambios que parecen imposibles*. (Versión Kindle). España: Gestión 2000.
- Laloux, F. (2017). *Reinventar las organizaciones (Guía práctica ilustrada): La guía práctica ilustrada del libro que ha revolucionado el management*. España: Arpa.

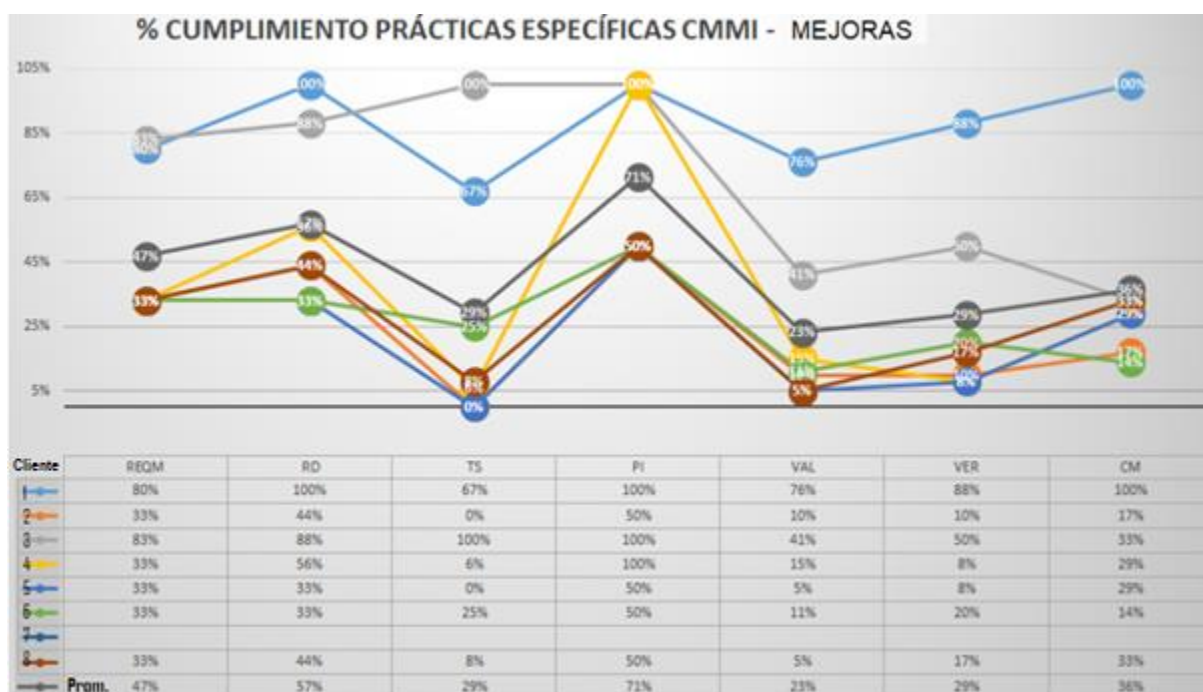
- Lastra Lastra, J. M. (2017). Rifkin, Jeremy, la tercera revolución industrial, trad. de Albino Santos Mosquera, España, Paidós, 2011, 397 pp. *Boletín Mexicano De Derecho Comparado*, 50(150), 1457-1462. doi:10.22201/ij.24484873e.2017.150.11847
- Liker, J. K. (2017). *Las claves del éxito de Toyota. 14 principios de gestión del fabricante más grande del mundo*. Bogotá: Editorial Planeta S.A.
- Montañés, M. A., Núñez, Montserrat, Pérez, Rosario & Tejada, Ángel. (1997). *Supuestos de Contabilidad de Costes*. Madrid: Universidad de Castilla la Mancha.
- Phillips-Donaldson, D. (2004). Gurus of quality. 100 years of Juran. *Quality Progress*, 37 (5), pp. 25-39. Recuperado de <https://docplayer.net/50970407-100-years-of-juran-in-50-words-or-less-gurus-of-quality-hard-work-from-an-early-age.html>
- Piattini Velthuis, M. G. & Garzas Parra, J. (2010). *FÁBRICAS DE SOFTWARE: Experiencias, Tecnologías y Organización*. (2da Edición). España: RA-MA.
- Ríos Ramos, F. (2006). La Dirección de Operaciones de Servicios (DOS). *Industrial Data*, 9(2), 84-94 Lima: Universidad Nacional Mayor de San Marcos. Recuperado de <https://www.redalyc.org/pdf/816/81690212.pdf>
- SEI (2010). *CMMI para desarrollo, versión 1.3*. USA: Software Engineering Institute.
- Senge, P.M. (2006). *The Fifth Discipline: The Art & Practice of the Learning Organization*. USA: Doubleday & Co.
- Slade, S. (2018). *Going Horizontal: Creating a Non-Hierarchical Organization, One Practice at a Time*. USA: Berrett-Koehler Publishers.
- Valencia, P. T. & Lozano, M. R. (2004). ¿Cómo diseñar un sistema de costes basado en las actividades ABC?: Un caso práctico. *Técnica Contable*, 56 (667), 4-24. Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=963804>

## 8 ANEXOS

### 8.1 ANEXO 1 – GRAFICOS Y ANÁLISIS SOBRE EL CUMPLIMIENTO DE LAS PRÁCTICAS ESPECÍFICAS DE CMMI-DEV (CAPACIDAD DE AREAS DE PROCESO).

A continuación se presentan los gráficos con los resultados de la aplicación de las listas de chequeo, para determinar el porcentaje de cumplimiento de las Prácticas Específicas de cada una de las Área de Proceso de Ingeniería, en diferentes clientes<sup>44</sup>.

Figura 29 Porcentaje de cumplimiento de las Prácticas Específicas de CMMI-DEV en mantenimiento de Mejoras



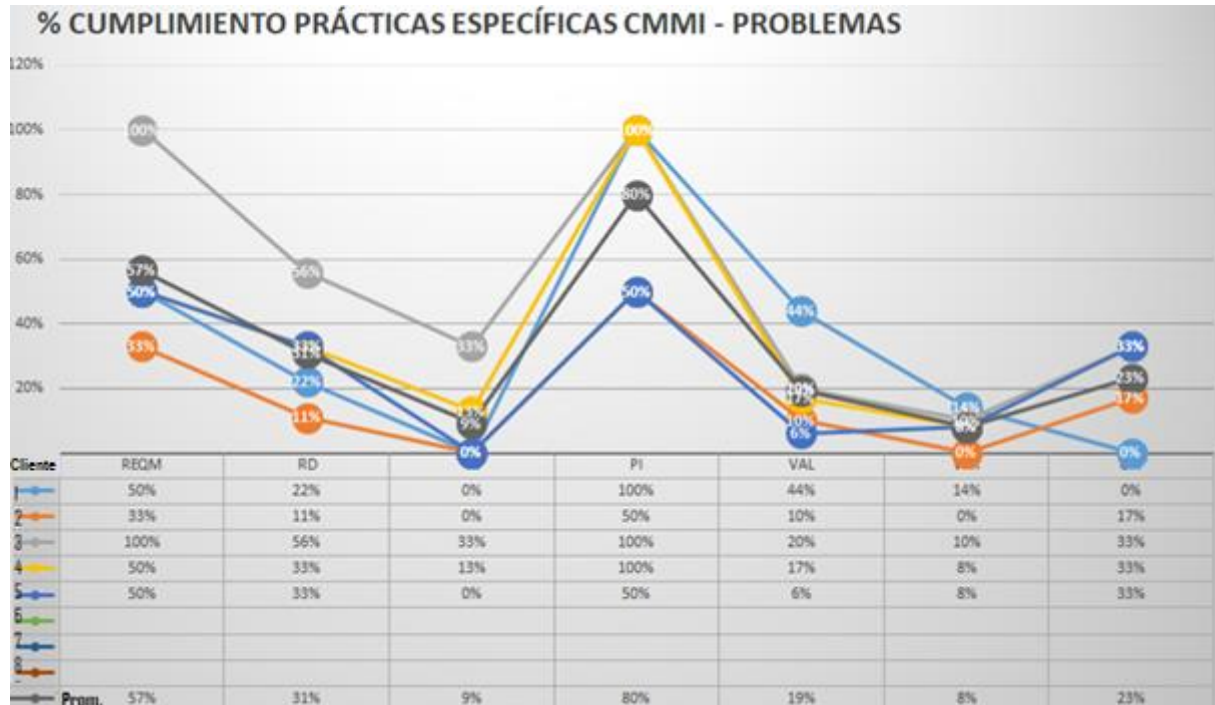
Fuente: Elaboración propia, 2019.

Para el subproceso de Mejoras (Figura 29), ninguna de las áreas de proceso demuestra capacidad, al no alcanzar un cumplimiento del 100% en el promedio general (fila Prom.). Solamente se dan casos particulares de capacidad, no representativos, para algunas áreas en algunos clientes, por ejemplo, el cliente-1 muestra capacidad (100%) en las áreas de proceso RD (*Requirement Development* o Desarrollo de Requerimientos) y CM (*Configuration*

<sup>44</sup> En general, los cumplimientos al 100%, para las Áreas de Proceso REQM, RD, CM y PI, son relativos, ya que, por simplificación de las entrevistas, las listas de chequeo no incluyeron exhaustivamente todas las Prácticas Específicas, dejando fuera un 10% de éstas, en particular para PI donde se excluyeron deliberadamente la mitad de ellas.

*Management* o Gestión de la Configuración) y el cliente-3 muestra capacidad en el área de proceso TS (*Technical Solution* o Solución Técnica).

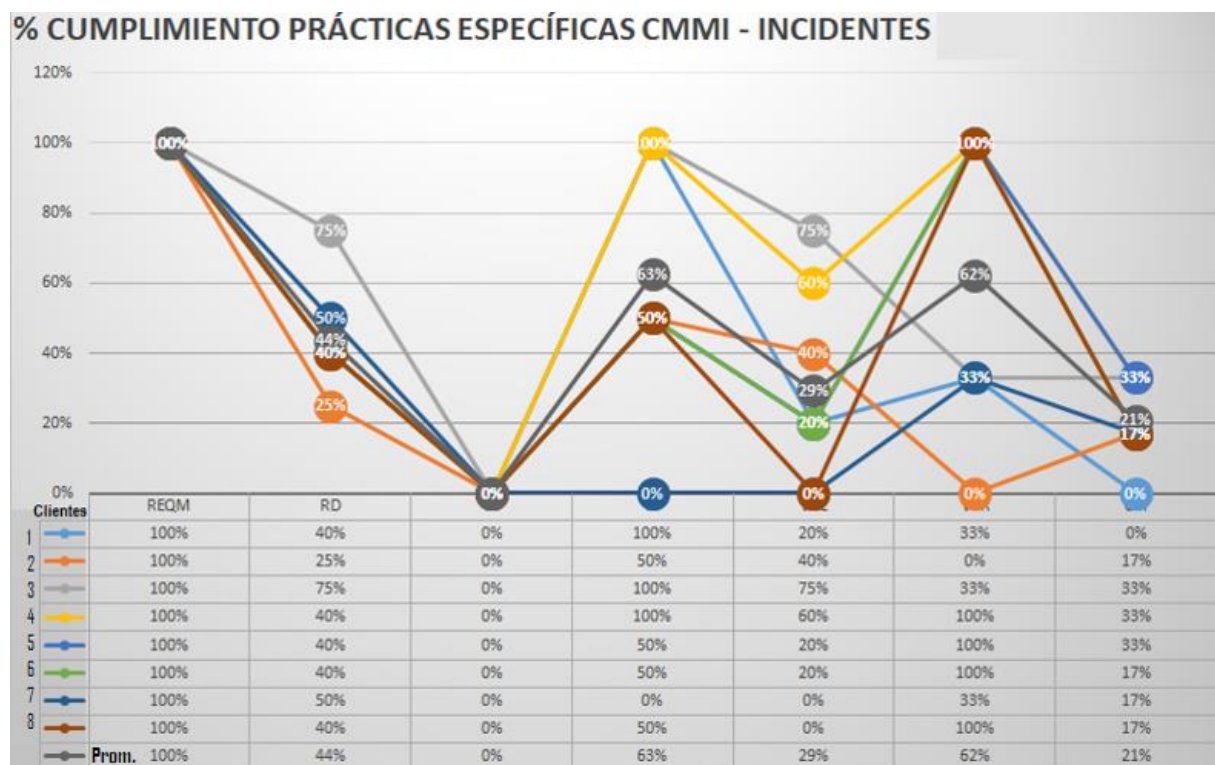
Figura 30 Porcentaje de cumplimiento de las Prácticas Específicas de CMMI-DEV en mantenimiento de Problemas



Fuente: Elaboración propia, 2019.

En esta segunda gráfica, para el subproceso de Problemas (Figura 30), ninguna de las áreas de proceso demuestra capacidad, al no alcanzar un cumplimiento del 100% en el promedio general (fila Prom.). De nuevo, sólo se dan casos no representativos de existencia de capacidad; por ejemplo, el cliente-3 muestra capacidad (100%) en el área de proceso RD (*Requirement Development* o Desarrollo de Requerimientos). Ahora bien, la aparente capacidad del área de proceso PI (*Product Integration* o Integración de Producto) para los clientes 1, 3 y 4, no es tan real si se considera que en PI se excluyeron deliberadamente la mitad de las Prácticas Específicas en las entrevistas.

Figura 31 Porcentaje de cumplimiento de las Prácticas Específicas de CMMI-DEV en mantenimiento correctivo de incidentes



Fuente: Elaboración propia, 2019.

En la tercera gráfica, para el subproceso de Incidentes (Figura 31), se repite el hecho de que ninguna de las áreas de proceso demuestra capacidad, al no alcanzar el 100% de cumplimiento en el promedio general (fila Prom.). Los casos de cumplimiento (100%), no representativos, en algunos clientes y para algunas áreas, se explica por la simplicidad del uso de las prácticas de ingeniería en este subproceso, en donde el desarrollo de requerimientos es mínimo, la solución técnica no requiere diseño técnico ni arquitectura, y aplican mínimas verificaciones.

## 8.2 ANEXO 2 – Roles de las prácticas de ingeniería de software

Este anexo presenta la descripción de las principales funciones de cada uno de los roles implicados en la propuesta de mejora de las prácticas de ingeniería de software.

### ARQUITECTO

- Vela por el cumplimiento de los lineamientos y directrices de la arquitectura.

- Analiza la documentación recibida para identificar el contexto y las restricciones, detectar tempranamente riesgos y posibles problemas que se puedan presentar, así como optimizaciones a la solución de software, e informa formalmente al Líder del Proyecto/Servicio.
- Identifica las restricciones y el contexto del problema.
- Diseña la estrategia de la solución: Decide sobre las tecnologías para la solución completa.
- Elabora los modelos de Componentes, Dominio, Servicios *web* y Despliegue.
- Elabora el documento de Arquitectura.
- Colabora en revisiones de pares para proyectos/servicios que lo soliciten.
- Colabora en revisiones de los aspectos técnicos de los requisitos.
- Revisa los Modelos de Datos
- Participa en la propuesta de mejora de procesos Técnicos.

#### **TÉCNICO**

- Vela por el cumplimiento de los lineamientos y directrices de la arquitectura.
- Analiza la documentación recibida para detectar tempranamente riesgos y posibles problemas que se puedan presentar, así como optimizaciones a la solución; lo valida con el Arquitecto y, si es necesario, informa formalmente al Líder del Proyecto/Servicio.
- Para casos de complejidad media o baja para los que no se requiera la intervención del Arquitecto, revisa y actualiza el documento los siguientes temas del documento de Arquitectura: restricciones y contexto del problema, modelo de Componentes, Modelo de Servicios, Modelo de Dominio y Modelo de Despliegue.
- Detalla el análisis y el diseño a partir de los requerimientos o casos de uso.
- Elabora el documento de Diseño detallado o técnico.
- Analiza la factibilidad técnica de los requerimientos.
- Realiza las revisiones de código.
- Colabora en revisiones de pares para proyectos/servicios que lo soliciten.
- Gestiona las tareas de su equipo en la herramienta disponible, generando reportes periódicos.
- Vela por la correcta aplicación de las metodologías y el correcto uso las herramientas seleccionadas.
- Coordina las revisiones entre pares, consolida los resultados y vela por que se ejecuten las correcciones a que haya lugar.
- Apoya técnicamente al equipo de desarrolladores.



- Vela por el correcto versionamiento y despliegue del software en los diferentes ambientes.
- Participa en las entrevistas con el cliente relacionadas con temas técnicos.
- Analiza la documentación de las necesidades no funcionales del cliente.
- Analiza los requerimientos no funcionales y propone alternativas y soluciones.
- Elabora la especificación de requisitos no funcionales, diseñando/actualizando pruebas no funcionales.
- Revisa los resultados del *Smoke test*.
- Participa en las entrevistas con el cliente relacionadas con temas técnicos y resuelve dudas al cliente sobre la especificación de los requisitos no funcionales.
- Participa en la propuesta de mejora de procesos.

#### **FUNCIONAL O ANALISTA FUNCIONAL**

- Realiza entrevistas al cliente y presentaciones del software actual (en caso de que exista).
- Resuelve dudas sobre los requisitos con el cliente.
- Analiza la documentación de las necesidades del cliente.
- Analiza requerimientos del cliente y propone alternativas y soluciones.
- Realiza el documento de especificación del relevamiento.
- Realiza las Especificaciones Funcionales, lo cual puede incluir realizar el modelo de casos de uso, identificar actores y detallar casos de uso.
- Diseña o valida los Escenarios de Prueba.
- Verifica los Casos de Prueba Funcionales y No funcionales.
- Participa en la propuesta de mejora de procesos

#### **ESPECIALISTA DEL NEGOCIO**

- Identifica objetos y actores del negocio.
- Verifica las Especificaciones Funcionales, de manera de asegurar los atributos de éstos.
- Valida la solución planteada y los requerimientos No funcionales.
- Valida el diagrama de componentes
- Participa en la propuesta de mejora de procesos

#### **LÍDER DE PRUEBAS**

- Diseña Plan de Pruebas.
- Realiza/Revisa estimaciones y cronograma del trabajo de pruebas.
- Coordina la operación del equipo de pruebas.

- Coordina la instalación en el ambiente de pruebas con el Integrador.
- Determina la continuidad de las pruebas a partir de los criterios de aceptación.
- Certifica la calidad del producto, como resultado de las pruebas.
- Consolida informes periódicos e informe de cierre.
- Analiza y evaluar el trabajo realizado con base en realimentación del cliente.
- Participa en la propuesta de mejora de procesos.

#### **ANALISTA DE PRUEBAS**

- Analiza la documentación recibida para detectar tempranamente riesgos y posibles problemas que se puedan presentar, y los valida con el Líder de Pruebas.
- Realiza estimaciones del trabajo de pruebas.
- Diseña/actualiza los casos de prueba funcionales para un conjunto de Requerimientos o Casos de Uso.
- Verifica la ejecución de los casos de prueba funcionales.
- Ejecuta casos de prueba funcionales y no funcionales.
- Comunica informes de resultados.
- Participa en la propuesta de mejora de procesos.

#### **TESTER**

- Ejecuta Casos de Prueba Funcionales.
- Registra defectos y evidencias de ejecución.
- Apoya la estimación del trabajo de ejecución de pruebas.
- Participa en la propuesta de mejora de procesos.

#### **DESARROLLADOR**

- Desarrolla el código de la solución, codificando un conjunto particular de clases o de operaciones de clase.
- Realiza pruebas unitarias o de componente.
- Resuelve defectos del software.
- Participa en la propuesta de mejora de procesos.

#### **GESTOR DE ENTREGABLES**

- Determina el alcance del Entregable.
- Define la estrategia.

- Planifica tareas y recursos
- Valida el Entregable.
- Participa en la propuesta de mejora de procesos.

#### **RESPONSABLE DE ENTREGABLES**

- Construye el Entregable.
- Documenta el Manual de Instalación y el procedimiento de remoción.
- Confecciona la Nota de entrega.
- Ejecuta los pasos de instalación y de remoción.
- Documenta los resultados y comunica al Rol Gestor de entregables.
- Participa en la propuesta de mejora de procesos.

#### **LÍDER DE PROYECTO/SERVICIO**

- Realiza asignaciones a su equipo de trabajo.
- Planifica las diferentes actividades.
- Garantiza el desarrollo eficaz de las actividades.
- Garantiza la capacidad de acuerdo con la demanda planificada.
- Monitoriza el desempeño y/o los acuerdos de nivel de servicio.

### **8.3 ANEXO 3 – EJEMPLOS DE INTEGRACIÓN DE ACTIVIDADES ENTRE MARCOS METODOLÓGICOS**

Este anexo muestra un par de ejemplos de la manera en que se integraron las prácticas de ingeniería con de los procesos claves de la organización. En particular los ejemplos presentan dos actividades del subproceso de Mejoras que forman parte del proceso de Gestión de Servicios.

**Ejemplo 1** - A continuación la integración de la actividad de **Analizar la Mejora** del subproceso de mejoras del marco de Gestión de Servicios, con las actividades propias del marco de Ingeniería. En **color azul** la porción de la caracterización de la actividad que corresponde a la integración con el **Marco de Ingeniería**.

### **Subproceso de MEJORAS**

**Actividad 02 - Analizar la Mejora.** El objetivo de esta actividad es analizar funcionalmente la mejora.

El Rol Funcional analiza, actualiza los atributos y pide más información al Rol Usuario de ser necesario.

Se debe utilizar el subproceso **Elaboracion de Requerimientos del Marco de Ingeniería** para obtener, analizar y detallar los requerimientos del cliente y del producto, realizando las actividades de: Identificar o recibir los Requerimientos (ER01), Detallar los requerimientos (ER02), dar inicio a Analizar la Factibilidad Técnica (ER03) y Verificar (ER04). Para la actividad Verificar los Requerimientos (ER04), un rol Funcional (desempeñando el papel de rol de Especialista del Negocio) debe realizar la verificación a través de una revisión de pares, para esto se recomienda utilizar la plantilla “Lista de Chequeo de Verificación del Requerimiento”.

Para medir el impacto de la solución se recomienda utilizar el formato “Matriz Control de Impacto”. Debe generar las salidas correspondientes que indica el subproceso (documento “Especificacion\_Funcional”). La Especificación Funcional deberá ser validada (ER05) por el rol Cliente en la actividad posterior en la que se valida la cotización (Actividad 5 del subproceso de Mejoras).

En esta instancia también se puede decidir la suspensión de la mejora que posteriormente podrá ser reactivada para su análisis. Ver la Guía de Suspensión de Casos.

Si se decide en consenso con el usuario cancelar la Mejora, se indica al Rol Líder de Cuenta para que pueda validar si el mismo es facturable o no, para su posterior cancelación.

**Ejemplo 2** - A continuación la integración de la actividad de **Solucionar la Mejora** del subproceso de mejoras del marco de Gestión de Servicios, con las actividades propias del Marco de Ingeniería. En **color azul** la porción de la caracterización de la actividad que corresponde a la integración con el **Marco de Ingeniería**.

### **Subproceso de MEJORAS**

**Actividad 08 - Solucionar la Mejora.** El objetivo de esta actividad es desarrollar, probar unitariamente y documentar la solución de la Mejora.

El rol Desarrollador recibe los documentos “Especificacion\_Funcional”, “Arquitectura\_De\_Software” y “Diseño\_Detallado” y desarrolla la mejora. Para esto debe llamar al **subproceso de Diseño y Desarrollo** del **Marco de Ingeniería**, en las actividades correspondientes al desarrollo, para desarrollar la solución (DD05), realizar las revisiones de código (DD06) y las pruebas unitarias del desarrollo (DD07). Para las revisiones es indispensable dejar evidencia de su realización y de las correcciones realizadas a partir de las observaciones hechas en la revisión (Ver actividad DD06). Así mismo, se debe dejar evidencia de las pruebas de desarrollador en el formulario “Prueba del Desarrollador” (Ver actividad DD07).

Para el caso de la versión Niche-V1 se considerará el Procedimiento de control de cambios en VB6.

Se puede requerir información adicional en el momento del desarrollo por lo cual asigna la Mejora al Rol Usuario para que lo complete y así luego el Rol Desarrollo puede continuar.

Por otro lado, debe instanciar al subproceso de **Gestión de Entregas** del **Marco de Ingeniería**, para comenzar el armado de los paquetes de entregables, según lo planificado (Actividades GE02 Construir Entregable y GE03 Probar Entregable). En este punto se inicia la construcción del documento “Informe de entrega”.

Por otra parte, en el caso de dependencias con otros componentes que actualmente están en pruebas el Rol Desarrollador puede suspender la Mejora, de manera consensuada con el Funcional. Ver la Guía de Suspensión de Casos.