

Usar en computadora con ayuda del ufficio de modalidad.

La modalidad de tesis es única para los programas.

Nivel:	
Licenciatura	
Maestría	X
Doctorado	

Modalidad: Tesis

Año: 2021

Marcar con una X

Poner el número de año p.e. 2015

**Información sobre Obtención de Grado Académico:**

Nombre	<u>Joel Taja Flores</u>
NUA	<u>146984</u>
Programa	<u>Maestría en Administración de Tecnologías.</u>

Para modalidades con Jurado completar la siguiente información

**Lugar, hora y fecha de la presentación**

Lugar		Firma y sello de autorización de reservación de lugar.
Hora	<u>14:00</u>	
Fecha	<u>26 de febrero del 2021</u>	

Título del trabajo	<u>Propuesta metodológica para la implementación de la norma ISO/IEC 27001 en el Centro de Desarrollo de Software del Instituto Tecnológico del Sur de Coahuila (ITSC), México</u>
--------------------	--

**Jurado**

	Nombre con grado académico completo: p.e. Doctor en Informática Industrial Nombre Apellido Paterno Apellido Materno	Firma de autorización para realización de examen de grado o titulación.
Presidente	<u>Doctora en Economía María de Lourdes Cobano Salis</u>	
Secretario	<u>Doctor en Ciencias Juan Antonio Bautista</u>	
Vocal (1)	<u>Doctor en Ingeniería Industrial Roberto Brea Serrato</u>	
Vocal 2 (Doctorado)		
Vocal 3 (Doctorado)		

**Asesoría**

Director del trabajo	<u>Dr. Juan Antonio Bautista</u>
Codirector	<u>M. I. D. Fernando José Martínez López</u>

(No llenar para uso exclusivo de la Coordinación.)

Valida (nombre y firma): \_\_\_\_\_

Una vez terminado de llenar imprimir en dos tantos (uno para entregar al iniciar el trámite de autorización del examen de grado o titulación y otro para firma de recibido).

YUCA de a 11 de Enero 2011

M. en I. HERIBERTO GUTÉRREZ MARTÍN  
JEFE DE LA UNIDAD DE ADMINISTRACIÓN ESCOLAR  
PRESENTE.

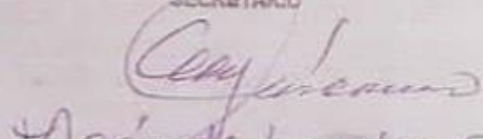
Por medio de la presente, se otorga autorización para producir el tipo de impresos  
compuesto de tesis y titulación al alumno Joel Tapia Flores  
del Programa de Maestría en Administración de las Tecnologías y cuyo número  
de NUA es: 146984 del cual soy director. El título de la tesis es: Investigación tecnológica  
para la implementación de la Norma ISO 9001:2010 en el Centro de  
Defensa de Software del Instituto Tecnológico Superior del Estado de Yucatán

Hago constar que he revisado dicho trabajo y he tenido comunicación con los estudiantes asignados  
para la revisión de la tesis, por lo que no hay impedimento alguno para fijar la fecha de examen de titulación

ATENTAMENTE

  
JUAN ANTONIO BENÍTEZ  
NOMBRE Y FIRMA  
DIRECTOR DE TESIS  
SECRETARIO

  
JUAN ANTONIO BENÍTEZ  
NOMBRE Y FIRMA  
DIRECTOR DE TESIS

  
María de Lourdes Cárcama Solís  
NOMBRE Y FIRMA  
PRESIDENTE

  
Roberto Bacia Serrato  
NOMBRE Y FIRMA  
LOCAL



**UNIVERSIDAD DE GUANAJUATO**

---

**CAMPUS IRAPUATO-SALAMANCA  
DIVISIÓN DE INGENIERÍAS  
DEPARTAMENTO DE ESTUDIOS MULTIDISCIPLINARIOS  
YURIRIA, GUANAJUATO**

**PROPUESTA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE  
LA NORMA ISO/IEC 29110 EN EL CENTRO DE DESARROLLO DE  
SOFTWARE DEL INSTITUTO TECNOLÓGICO SUPERIOR DEL  
SUR DE GUANAJUATO (ITSUR), MEXICO**

**TESIS**

**PARA OBTENER EL TÍTULO DE MAESTRO EN  
ADMINISTRACIÓN DE TECNOLOGÍAS**

**PRESENTA  
ING. JOEL TAPIA FLORES**

**ASESOR  
DR. JUAN ANTONIO BAUTISTA.**

**COASESOR  
M.I.S. FERNANDO JOSÉ MARTÍNEZ LÓPEZ**

**YURIRIA, GUANAJUATO 2020**

## Contenido

Resumen .....	10
Introducción .....	12
Antecedentes.....	14
Planteamiento del Problema .....	19
Justificación .....	22
Objetivos.....	24
Marco teórico .....	25
2.1 Gestión de Procesos .....	25
2.2 Proceso .....	26
2.3 Software.....	27
2.4 Ingeniería de Software .....	27
2.5 Proceso de software .....	28
2.6 Metodología .....	34
2.7 Metodologías Agiles .....	34
2.8. Metodología SCRUM .....	35
2.9. Metodología Extreme Programming (XP) .....	38
2.10 Calidad .....	39
2.11 CMMI .....	40
2.12 Estándar ISO/IEC 29110 .....	41
Metodología .....	43
Realizar un diagnóstico técnico del CDS.....	¡Error! Marcador no definido.
Analizar la norma ISO .....	46
Elaborar la propuesta metodológica .....	¡Error! Marcador no definido.
Resultados .....	52
Análisis del centro de desarrollo de software del ITSUR .....	126
Revisar esquema completo y verificar su lógica e integridad.....	52
Etapa 1. Administrar proyecto .....	53
Elaborar Propuesta .....	53
Levantar y Analizar requerimientos. ....	55
Estimar Proyecto. ....	55
Elaborar/Ajustar propuesta.....	55
Revisar técnicamente la propuesta .....	55

Elaborar propuesta económica .....	56
Presentar propuesta al Cliente.....	56
Desarrollar acuerdo formal y presentación de Kick-off .....	57
Realizar Kick-off con clientes y firmas de contrato .....	58
Planear Proyecto .....	58
<b>Planear Proyecto .....</b>	<b>58</b>
<b>Analizar ciclo de vida aplicable al proyecto .....</b>	<b>60</b>
<b>Realizar ajustes de procesos.....</b>	<b>61</b>
<b>Planear Auditorías.....</b>	<b>61</b>
<b>Verificar ajuste de procesos .....</b>	<b>61</b>
<b>Evaluar reutilización de activos y/o componentes .....</b>	<b>62</b>
<b>Establecer y configurar ambiente de trabajo .....</b>	<b>62</b>
<b>Desarrollar/Ajustar calendario y presupuesto del proyecto .....</b>	<b>62</b>
<b>Identificar productos a verificar y validar .....</b>	<b>63</b>
<b>Desarrollar Planes y Estrategia del Proyecto .....</b>	<b>64</b>
<b>Administrar Riesgos.....</b>	<b>65</b>
<b>Revisar internamente planes de trabajo.....</b>	<b>65</b>
<b>Ejecutar revisión entre Colegas .....</b>	<b>66</b>
<b>Validar calendario detallado con cliente.....</b>	<b>66</b>
<b>Monitorear Planes y Proyecto.....</b>	<b>66</b>
<b>Cerrar Proyecto.....</b>	<b>66</b>
<b>Estimar proyecto .....</b>	<b>66</b>
<b>Estructurar Alcance del proyecto. ....</b>	<b>67</b>
<b>Identificar fases y ciclo de vida aplicables en el proyecto. ....</b>	<b>68</b>
<b>Estimar esfuerzo y duración. ....</b>	<b>68</b>
<b>Revisar estimación de esfuerzo y duración.....</b>	<b>69</b>
<b>Realizar estimación de costos .....</b>	<b>69</b>
<b>Arrancar CM.....</b>	<b>69</b>
<b>Identificar elementos de configuración .....</b>	<b>71</b>
<b>Implementar repositorios.....</b>	<b>71</b>
<b>Identificar líneas Base (LB) .....</b>	<b>72</b>
<b>Realizar Auditorías y revisiones de configuración .....</b>	<b>72</b>
<b>Crear y Liberar LB.....</b>	<b>72</b>

<b>Monitorear Planes y Proyectos</b> .....	73
<b>Administrar Riesgos</b> .....	73
<b>Establecer estrategia de riesgos</b> .....	74
<b>Identificar y categorizar los riesgos</b> .....	74
<b>Establecer plan de administración de riesgos</b> .....	75
<b>Dar seguimiento al plan de administración de riesgos</b> .....	76
<b>Monitorear Planes y Proyecto</b> .....	76
<b>Ejecutar revisiones de avance internas</b> .....	78
<b>Presentar estatus al cliente y Gerente del CDS</b> .....	80
<b>Monitorear costos y presupuestos del proyecto</b> .....	80
<b>Atender asuntos y/o planes de riesgos.</b> .....	80
<b>Administrar Riesgos</b> .....	81
<b>Analizar y evaluar resultados.</b> .....	81
<b>Ejecutar revisiones de hito</b> .....	81
<b>Cambiar requerimientos</b> .....	81
<b>Cerrar Proyecto</b> .....	81
<b>Cerrar Proyecto</b> .....	81
<b>Elaborar carta de cierre</b> .....	83
<b>Ejecutar reunión <i>post-mortem</i></b> .....	84
<b>Compartir información a la organización</b> .....	85
<b>Respaldar Proyecto</b> .....	85
<b>Liberar Recursos</b> .....	86
<b>ETAPA 2. Administrar Requerimientos</b> .....	86
<b>Levantar y Analizar Requerimientos</b> .....	86
<b>Comprender las necesidades del cliente (actividad a realizar en preventa/pre-análisis)</b> .....	88
<b>Factibilidad técnica (actividad a realizar en preventa/preanálisis)</b> .....	89
<b>Identificar proveedores de requerimientos</b> .....	92
<b>Elaborar prototipos/pruebas de concepto (actividad a realizar en fase de análisis)</b> .....	92
<b>Detallar Requerimientos (actividad a realizar en fase de análisis)</b> .....	92
<b>Ejecutar Revisión entre pares</b> .....	94
<b>Diseñar Solución</b> .....	94
<b>Cambiar requerimientos</b> .....	94
<b>Diseñar Solución</b> .....	94

Identificar criterios y alternativas de solución .....	97
Análisis y Resolución de Decisiones .....	98
Presentar y evaluar alternativas de solución .....	98
Realizar diseño de alto nivel.....	98
Realizar diseño de bajo nivel.....	99
Diseñar los casos de prueba .....	100
Diseñar Plan de Integración.....	101
Diseñar los manuales.....	102
Construir e integrar .....	102
<b>Construir e integrar .....</b>	<b>102</b>
<b>Construir la solución e integrar en ambiente de desarrollo. ....</b>	<b>104</b>
Configurar ambiente beta .....	105
Ejecutar Revisión entre Colegas .....	106
Desarrollar plan de implementación.....	106
Probar y estabilizar .....	106
<b>Probar y Estabilizar .....</b>	<b>106</b>
Ejecutar Casos de Prueba .....	108
Registrar defectos.....	108
Asignar <i>Tickets</i> .....	109
Corregir defectos .....	109
Reintegrar y ejecutar pruebas de regresión .....	109
Actualizar registro.....	109
Analizar Reportes de Verificación y Validación.....	109
Monitorear Planes y Proyecto .....	110
Liberar a producción.....	110
<b>Liberar a Producción.....</b>	<b>110</b>
<b>Empaquetar producto .....</b>	<b>111</b>
<b>Configurar ambiente de producción .....</b>	<b>112</b>
<b>Ejecutar validación .....</b>	<b>112</b>
Cambiar requerimientos.....	112
Probar y estabilizar .....	112
Cerrar Proyecto.....	112
<b>Etapa 3. Procesos continuos y de soporte .....</b>	<b>114</b>

Crear y liberar LB .....	114
Obtener autorización de la Línea Base.....	115
Identificar elementos de configuración a incluirse en línea base.....	116
Publicar en Línea base.....	116
Realizar Auditorías y Revisiones de Configuración .....	116
Realizar Auditorías.....	117
Registra inconsistencias.....	117
Informar resultados de las revisiones .....	118
Generar y Analizar indicadores .....	118
Recolectar datos .....	120
Generar los indicadores.....	120
Revisar integridad de los datos .....	120
Analizar los datos.....	121
Monitorear Planes y Proyecto .....	121
Ejecutar Revisiones entre Colegas.....	121
Solicitar PR .....	123
Revisar el Producto.....	123
Atender defectos .....	123
Analizar resultados .....	124
Revisar Producto.....	124
revisión Formal.....	124
Ejecutar Sesión de PR .....	124
Registrar Defectos .....	125
Analizar Resultados .....	125
Identificar los principales problemas multifuncionales con respecto a la situación en cuestión. .....	¡Error! Marcador no definido.
Formular el problema central.....	¡Error! Marcador no definido.
Conocer los efectos provocados por el problema central.....	129
Análisis de la norma ISO.....	132
Proceso de gestión de proyectos (PM).....	133
Objetivo del proceso de gestión de proyectos.....	133
Requisitos del proceso de gestión de proyectos.....	133
Flujo de información del proceso de gestión de proyectos .....	135



Tareas del proceso de gestión de proyectos .....	135
Proceso de implementación del software (SI) .....	143
Objetivo del proceso de implementación del software .....	143
Requisitos del proceso de implementación del software .....	143
Flujo de información del proceso de implementación del Software .....	145
Tareas del proceso de implementación del software .....	146
Mapeo de las partes 4-1 y 5-1-2 de la Norma ISO/IEC 29110 .....	160
Análisis de los modelos .....	160
Diseño del mapeo .....	¡Error! Marcador no definido.
Identificar las funciones de cumplimiento y no cumplimiento .....	166
Evaluación del ISO/IEC 29110-4-1 .....	166
Definición de indicadores de cumplimiento de requisitos .....	178
Elaborar la propuesta metodológica de implementación de la ISO/IEC 29110 en el CDS. ....	¡Error! Marcador no definido.
Conclusiones .....	200
Bibliografía .....	202

# Capítulo I

## Resumen

En el presente documento de investigación se abordarán 4 puntos claves que darán la pauta para la generación de una metodología que permita al Centro de Desarrollo de Software (CDS) del Instituto Tecnológico Superior del Sur de Guanajuato (ITSUR) a lograr la certificación del ISO/IEC 29110e que permita a la institución generar software de calidad y una única forma de trabajo.

El primer punto que se aborda es el análisis de la certificación CMMI, ahí desglosamos cuáles son sus diagramas de procesos, sus actividades, sus documentos de entrada y salida que contienen cada diagrama. Cada proceso está formado para ser realizado por cada uno de los actores que forman parte de esta certificación, de estos puntos y más se abordan en la presente investigación.

El segundo punto que se aborda es la forma de trabajar en el CDS, se analizan cuáles son los problemas a los que se enfrentan al estar trabajando con bajo capital y recursos humanos limitados, ya que estos fungen con más de una actividad fuera del CDS, la pérdida de la certificación hizo que se trabajará de formas diferentes e independientes para cada uno de los miembros del equipo del CDS. De estos y más problemas se abordarán en este documento.

La tercera parte habla de los procesos y características que forman el ISO/IEC 29110e, desglosamos cuáles son los diagramas de actividades que contienen, que actores que participan en la ISO y que documentos tanto de entrada como de salida se necesitan para lograr de forma cabal dicha certificación, los roles de los participantes pueden llegar a ser de los más flexibles y a diferencia de la certificación CMMI los requisitos que se deben de cumplir son más fáciles de desarrollar, tanto administrativamente como de recursos ya sean económicos como humanos.

En la cuarta parte de este documento se abarca el mapeo entre la Certificación CMMI y el ISO/IEC 29110, en las cuales nos damos cuenta cuáles son sus similitudes que ayudan a facilitar la certificación de este ISO y cuales debilidades puedan tener la una de la otra al momento de realizar el proceso de certificación.

Ahora que conocemos a grandes rasgos los puntos que se trata en este documento de investigación, empezaremos con la introducción de la investigación y ampliamos los puntos acordados para lograr los resultados de esta investigación.



## Introducción

Las empresas que se desempeñan en el mundo del desarrollo de software deben de tener estrategias y métodos claros para adentrarse en este mundo, y así competir con las grandes organizaciones del desarrollo.

El Desarrollo del Software (DS), contiene dos etapas importantes para su realización, la gestión del proyecto y el desarrollo del mismo, por esto las empresas buscan estrategias que les permitan mejorar el desarrollo de estas etapas, para así competir en el mercado actual.

Las empresas que llevan ventaja en este desarrollo son las que cuentan con un sinfín de recursos que les permiten sacar ventaja a gran nivel, dichas empresas se apoyan de certificaciones y normas que las ayudan a consolidarse en lo más alto de la cadena del DS, tal es el caso de las grandes corporaciones que cuentan con certificación como el CMMI en sus diferentes niveles, la cual ayuda a las empresas a encaminar de mejor forma su desarrollo, tanto en planificación, como en gestión y desarrollo del mismo proyecto.

Para que una empresa obtenga este tipo de certificaciones se necesitan que dispongan de todos los recursos con los que cuenta la empresa, llámese Recursos Humanos (RH), Recursos Económicos (RE), tiempo, planificación, entre otros. Es por eso por lo que las pequeñas y medianas empresas se les dificulta obtener este tipo de certificación o normas que les permita competir al tú por tú con las grandes esferas globales.

Por esto, los entes encargados de regular el DS, formaron una asamblea para crear un estándar nuevo que ayude a las muy pequeñas empresas, a desarrollar de mejor manera las dos etapas principales del DS y así poder competir en el mercado actual del Desarrollo

Estamos hablando del estándar ISO/IEC 29110, el cual contiene una serie norma que ayuda a las muy pequeñas empresas (no más de 25 personas) a obtener mejor resultados en la planificación y desarrollo del proyecto.

El actual documento cuenta con una metodología que está dividida en cuatro parte, la cual tiene el fin de sembrar un camino que más adelante ayude al CDS a conseguir la certificación ISO/IEC 29110. En la primera parte se analiza de forma detalla la certificación CMMI, que actores participan y cuáles son sus funciones en las diferentes etapas de dicha certificación. En la segunda parte de esta metodología se hace un análisis acerca de la forma de trabajo del CDS, en este análisis se descubre

cuales son sus problemas que actualmente tiene, de donde se derivaron y la solución a estos problemas. La parte número 3 de la metodología habla acerca de la norma ISO/IEC 29110, en esta parte descubrimos todo lo referente a ella, quienes son sus actores, que documentos se generan y como podemos implementarla. La cuarta y ultima parte de la metodología hace un análisis cruzado entre ambas certificaciones, el cual nos arroja los resultados acerca de lo que debe de ser el CDS del ITSUR para conseguir la certificación ISO/IEC 29110. Sin más que agregar, empecemos con esta investigación.

## Antecedentes

En 2005 el subcomité de ISO/IEC junto con el comité técnico, inicio un nuevo proyecto con el objetivo de crear un estándar internacional dirigido a los perfiles de ciclo de vida y a las pautas usadas en las muy pequeñas empresas (Very Small Entities, por sus siglas en ingles VSE). Para ello, se creó el WG 24 a cargo de un nuevo proyecto bajo la coordinación de Tanin Uthayanaka de Tailandia. Una vez identificados los problemas que afectan a las VSE, el WG24 definió un conjunto de objetivos a lograr mediante la implementación del nuevo proyecto de ingeniería de software estándar ISO/IEC 29110. La primera reunión del WG24 tuvo lugar en Tailandia en 2006 y contó con la presencia de Estados Unidos, India, Irlanda, Bélgica, Finlandia, Luxemburgo, Canadá, Nueva Zelanda, Corea del Sur y México. Durante la reunión, el grupo tomó la decisión de tomar el estándar mexicano como base para su trabajo (Morales , 2013).

A partir de las prácticas realizadas con cuatro empresas del valle del Cauca, Colombia se propone un modelo para la implementación de la norma ISO 29110. El modelo está fundamentado en las buenas prácticas que se deben llevar a cabo en cada uno de los elementos que componen esta norma. Esto permitió a las 4 empresas organizar mejor su forma de trabajo, mantener el orden de su información en todo momento, tener trazabilidad de los proyectos y apropiar herramientas de control de calidad que reducen el reproceso. Además, mejoró la forma en cómo recopilan la información garantizando un mejor resultado en los compromisos que pueden ser asumidos no desde la suposición sino desde datos reales basados en estimación y priorización. Por otra parte, la implementación de las retrospectivas hace que los equipos tengan la posibilidad de entender que hacen bien y mal para poder generar una transformación real de las compañías (Ordoñez, 2019).

La dirección de Desarrollo Tecnológico e informático de la Universidad Técnica del Norte de Ibarra, Ecuador, contaba con un proceso de desarrollo ágil de manera empírica basado en la metodología del Proceso Unificado Rational (RUP), la cual no se ajustaba, pues los proyectos CRM no contaban con la formalidad adecuada, por lo cual, no se puede llevar un control de los requerimientos implementados y por ende disminuye el control en las pruebas del producto; pero con la fusión de la metodología Scrum y el estándar ISO/IEC 29110 se logró incluir la formalidad a los proyectos distribuyendo las actividades de generación de documentos en cada uno de los sprint; así pues la documentación se genera a la par del proyecto y al finalizar cada sprint tenemos entregables tanto formalmente como en producto final (Madruñero, 2018).

En Perú, las empresas de desarrollo de software han mostrado una importante actividad económica. En los últimos años, han tenido una tasa de crecimiento del 15 por ciento anual (2003-2011), con cifras de ventas para el mismo período que aumentaron de \$85 millones a \$239 millones. De todas las empresas que forman parte de esta industria, el 63 por ciento son microempresas y el 27 por ciento son pequeñas empresas (PromPeru,2013).

Como se puede ver, las micro y pequeñas empresas son la base de la economía del Perú. Sin embargo, dentro de este sector, la gran mayoría de empresas carecen de estándares y prácticas comprobadas que puedan ayudarlas a incrementar su competitividad. Un estudio del Grupo de trabajo 24 de ISO sobre el uso de estándares en pequeñas organizaciones de desarrollo de Software encontró que más del 67 por ciento de las empresas consideran importante ser reconocidas o certificadas; sin embargo, una gran cantidad de empresas encuestadas también indicaron que los estándares deben ser prácticos, fáciles de entender y tener plantillas y ejemplos que se puedan implementar rápidamente y a bajo costo (Laporte,2018).

Un estudio similar en Dublín sobre empresas emergentes y pequeñas organizaciones software formadas recientemente (en 24 meses) encontró que existe una percepción generalizada de que los procesos descritos en los estándares de software son demasiados complicados, no tienen guías para una implementación fácil, no son fáciles de integrar y requieren recursos adicionales que generalmente no están disponibles en los negocios (Laporte,2014).

La competitividad de las empresas de desarrollo de Software depende de su capacidad para ofrecer productos de Software con atributos de calidad con un presupuesto y un cronograma no aprobados. La mayoría de las VSE que desarrollan Software no ven los beneficios de implementar estándares de software. En consecuencia, limitan su potencial para ser reconocidos como entidades de desarrollo de Software de calidad. Los resultados obtenidos a través de la aplicación de la ingeniería de software empírica en un experimento en el que se utilizó la ISO/IECTR29110-5-1-2 “ingeniería de software – perfiles de ciclo de vida 1-2. La guía incluye dos procesos: el proceso de Gestión de Proyectos (PM) y el proceso de implementación de Software (SI) (Castillo,2020).

El objetivo del proyecto fue el desarrollo de un producto de Software para la programación de citas médicas para el Centro de Bienestar Estudiantil de la Universidad del Ecuador. Se involucran catorce equipos de pregrado. Dos de ellos (equipos controlados) implementaron un subconjunto del proceso de SI, mientras que los otros dos (equipos no controlados) tuvieron libertad para elegir las actividades de desarrollo del estándar. Todos los equipos desarrollaron el producto de software

utilizando el marco de trabajo de SCRUM dentro del mismo tiempo. La ejecución del experimento encontró varias dificultades. Todos los equipos experimentaron esta dificultad, especialmente cuando tuvieron que construir y probar los componentes del Software. En general, los equipos que utilizaron la guía ISO/IECTR29110-5-1-2 obtuvieron mejores puntuaciones en la evaluación de la calidad de sus procesos (Castillo, 2020).

Para enfrentar estos desafíos, se ha desarrollado un conjunto de estándares y pautas para satisfacer las necesidades de las entidades muy pequeñas (VSE) involucradas en el desarrollo de software, la norma ISO/IEC 29110. La cual se implementó en una VSE de desarrollo de Software de cuatro personas en Perú. El proyecto fue liderado por la Universidad Peruana de Ciencias Aplicadas (Universidad Peruana de Ciencias Aplicadas) (UPC). Tres proyectos de estudiantes de posgrado de la UPC contribuyeron a la exitosa implementación del estándar. VSE utilizó métodos y prácticas ágiles para sus proyectos (por ejemplo, Scrum, desarrollo impulsado por pruebas, integración continua). La norma ISO se adaptó al uso de estas prácticas ágiles. Los resultados de esta adaptación (es decir, procesos, productos de trabajo y herramientas de apoyo) se aplicaron a un proyecto real para uno de los clientes de la VSE: la segunda aseguradora más grande del Perú. No solo se mejoraron las habilidades de la VSE, sino que la VSE solo había desperdiciado esfuerzo (es decir, reelaboración) del 18 por ciento de las tareas presupuestadas (García, 2015).

Aqp-Alfa inicia un proceso de adopción de buenas prácticas en ingeniería de software a través del proyecto COMPETISOFT-Perú en el año 2010 en la que se realizó la adopción del MoProSoft. En dicha oportunidad el modelo se implementó en 8 meses, obteniendo la certificación en mayo del 2011 al nivel de madurez organizacional. Con este antecedente y la decisión de la empresa de incursionar en el mercado internacional, llevaron a que en enero del 2013 inicien la mejora de sus procesos de desarrollo de software para cumplir con lo establecido en el perfil Básico del estándar internacional ISO/IEC 29110. Por tanto, Aqp-Alfa realizó diversas actividades en favor de sus colaboradores de modo que mejoren sus competencias profesionales en temas relacionados a la calidad de software (Abarca, 2015).

En la universidad de Lima Perú, se propuso crear un nuevo sitio web en el cual las pequeñas empresas de software puedan realizar una autoevaluación en el conjunto de estándares de calidad ISO/IEC 29110, permitiéndoles la detección de deficiencias en sus prácticas de desarrollo. Adicionalmente, el sitio web propuesto incluirá contenido educativo sobre el estándar. Se listarán las herramientas de autoevaluación existentes para ISO/IEC 29110; la propuesta de implementación



del sitio Web en sí, y cómo estos ayudan no solo a mejorar la calidad de sus productos sino también a aumentar su competitividad en el mercado internacional. El nombre de esta nueva web propuesta es Sistema de autoevaluación para pequeñas organizaciones (SAPPO) (Marroquín, 2018).

En la empresa de estudio HOLINSYS se desarrolló un modelo de mejora para el ciclo de vida del desarrollo de Software, la cual pretende certificarse en la norma internacional ISO/IEC 29110. Para el desarrollo del modelo de mejora, se revisaron los proyectos de desarrollo de software GALI, RFID y GUARDIAN para evaluar el estado actual del ciclo de vida del desarrollo de software de la empresa de estudio utilizando la herramienta de evaluación *Activity Assessment*, que sirve de apoyo para la evaluación de los procesos: Gestión del Proyecto e Implementación del Software según lo establecido en la ISO 29110 e identificar sus brechas y oportunidades. Teniendo como resultado recomendaciones para las actividades y procesos desarrollados, como también la documentación de las funciones de cada rol y plantillas para los productos de trabajo desarrolladas en el ciclo de vida del desarrollo de software (Egusquiza, 2016).

El estandar ISO/IEC 29110, como se ha mencionado se ha hecho muy recuente en las universidades pero para determinar cómo se usa el ISO/IEC29110 en el entrenamiento en procesos de desarrollo y de administración del desarrollo en estudiantes universitarios, se pidió a un grupo de estudiantes inscritos en la materia de Proceso de Software, de la carrera de Ingeniería de Software de la Universidad Autónoma de Zacatecas, que desarrollarán un sistema con un cliente real, los resultados fueron el desarrollo de la funcionalidad relacionada con el registro al servicio así como como el módulo para añadir y consultar información sobre la normatividad del servicio. Para el desarrollo, los estudiantes se organizaron en equipos con los roles de Líder de equipo, responsable técnico, responsable de calidad y responsable de colaboración (de León, 2020).

Se descubrió que los estudiantes encontraron complicado relacionar las responsabilidades de los roles particulares con la ejecución de las tareas del PM y SR debido a que en el ISO/IEC 29110 Perfil de Entrada se establecen los roles generales de Adquiridor, Administrador de Proyecto y Equipo de Trabajo, para el PM. Dicho lo anterior, para que el entrenamiento resulte más organizado y que pueda ser utilizado en el desarrollo de productos reales y que los estudiantes obtengan mas experiencia, es necesario que se hagan adaptaciones que permitan que las tareas de PM y SR sean llevadas a cabo en su totalidad y además en las mejores condiciones que permitan un entrenamiento que les resulte útil para su vida profesional (de León, 2020).

En este contexto el gobierno del estado de Zacatecas, México, ha desarrollado una estrategia para incrementar la competitividad de las empresas que esté alineada a la agenda nacional definiendo dos actividades para lograr tal fin:

1. Lograr que las empresas muy pequeñas incrementen el nivel de capacidad de sus procesos a nivel nacional: que hace referencia al incremento de la capacidad de los servicios/productos de las Tecnologías de la Información (TI) mediante la adopción de modelos y estándares que sirvan como referencia para la adopción de buenas prácticas de ingeniería de software que apoyen a las organizaciones a incrementar la productividad y calidad de las organizaciones.
2. Incrementar la calidad y cantidad de talento relacionado con el desarrollo de software y servicios de TI: esto hace referencia a disponibilidad de capital humano mejor preparado, que pueda aprovechar el crecimiento del sector TI tanto a nivel local, como a mercados internacionales (Muñoz, 2018).

De mismo modo se implementó el estándar ISO/IEC 29110 en centros de desarrollo de Software (CDS) de universidades del Estado de Zacatecas, México para reforzar el proceso de desarrollo de software. La finalidad de la implementación en las universidades es lograr que los estudiantes tengan experiencia en el uso de las buenas prácticas en la realización de proyectos reales. Esta acción cubre los dos retos principales a los que se enfrentan los recién graduados al integrarse a la industria del software: 1) tener el conocimiento para trabajar con roles en el desarrollo de un proyecto y 2) tener la experiencia de trabajar con modelos y estándares de calidad (Muñoz, 2018).

También, en la universidad Tecnológica del Estado de Zacatecas (UTZAC) se establecen vínculos entre empresas para que los estudiantes puedan desarrollar proyectos de Tecnologías de la información (TI). Sin embargo, para llevar a cabo estos proyectos no se había utilizado metodologías estandarizadas para lograr la calidad de los productos. Derivado de reuniones con otros grupos de investigación, surgió la idea de unificar los procesos bajo la norma ISO/IEC 29110, la cual ofrece la oportunidad de trabajar en un enfoque de mejora continua de procesos en el desarrollo de software, tanto desde el punto de vista de la administración como del enfoque de implementación del software. Esto da el valor de contar con un producto completamente documentado y estructurado en cada etapa del desarrollo de software en términos de monitoreo de las tareas de implementación (Barrios, 2018).

## Planteamiento del Problema

Actualmente la industria del software representa una actividad económica importante en el mundo, al contribuir al producto interno bruto, en Europa representa el 8% del PIB, mientras que en México representa 0,7% (Pino Correa, 2018) y se ha convertido en una actividad especializada, que ha originado un desplazamiento de estas con respecto a su tamaño (Anaya, 2006). Esta industria está formada por Grandes, medianas, pequeñas y microempresas (conocidas como PYMES) desarrolladoras de software.

Las Grandes empresas utilizan estándares de calidad como el *Software Engineering Institute (SEI)* y *la international Organization for Standardization (ISO)*, que incluye certificaciones y normas de calidad como como: *Capability Maturity Model Integration (CMMI-DEV)*, ISO 12207, ISO 15504, ISO 9001 e *Institute of Electrical and Electronics Engineers 1207 (IEEE1207)* (Calvo, 2018 & Claude, 2017).

Sin embargo, diversos estudios (como el de Calvo, 2018) indica que la aplicación de estos estándares en las pymes es difícil, porque se suponen una inversión en diversos rubros como: los recursos Humanos, tecnológicos y económicos que se verán reflejada a largo plazo y las considera como un gasto, además el poder utilizar este tipo de herramientas es mucho más complicado, por el tamaño reducido de los grupos de trabajo, los problemas de definición de roles dentro de estos mismos, las responsabilidades poco precisas, el tiempo empleado para ello y por los recursos restringidos con que cuentan este tipo de organización, además de no alcanzar a percibir a la calidad como un valor en sí mismo que hará prosperar a la organización, por el simple hecho que los modelos están orientados a grandes organizaciones (Calvo, 2018) (Pesado,2013).

Asimismo, en las pymes la aplicación de esos modelos se agrava aún más, porque existe un problema cultural relacionado con la adopción de normas y certificaciones (Calvo-Manzano José A, 2008). También las Muy Pequeñas Empresa (en inglés *Very Small Entity, VSE*), de acuerdo con la ISO 29110 están integradas de 1 a 25 empleados, percibieron los estándares de ingeniería como burocráticos y costosos de implementar, asimismo, no tienen formas para ser reconocidas como entidades que producen sistemas de calidad o productos de software en su dominio. Situación que genera su exclusión o aislamiento en las actividades económicas (Claude, 2017).

En años recientes las *VSE* han buscado garantizar la calidad de productos desarrollados, para ello se han diseñado y construido diferentes modelos y normas certificables; que se dividen en dos grandes grupos: los orientados a la calidad del producto, y al proceso, que han generado modelos orientados

a este tipo de organizaciones: El Modelo de Procesos para la industria del Software (MoProSoft) en México, Mejora de Procesos del Software Brasileño (MPS BR) en Brasil, y la Mejora de procesos software para pequeñas empresas (COMPETISOFT) como un proyecto Iberoamericano y la más reciente ISO/IEC 29110 que se establece como estándar en las Mejoras de Procesos para PYMES (Pesado, 2013).

Esta última de interés en este trabajo, por que presenta las características y requisitos de una VSE y aclara las razones de los perfiles, documentos, normas y guías. También introduce los procesos y los conceptos de normalización, definiendo la taxonomía de los perfiles ISO/IEC 29110 (Pino, 2018).

(Mas, 2005) realizó un estudio en Noruega sobre las grandes empresas (integradas con más de 200 desarrolladores) y las pymes (integradas con menos de 30 desarrolladores) y concluyó que hay diferencias fundamentales entre ellas:

- Con respecto al rendimiento general, las grandes empresas obtuvieron niveles más altos de éxito en la implementación de programas de Mejora de Procesos de Software (SPI) que las pequeñas, pero ambas alcanzaron valores similares en cuanto al éxito empresarial.
- La diferencia fundamental fue en los programas de SPI; En las grandes compañías despliegan sus buenas prácticas a través del establecimiento de modelos de procesos, procedimientos formales, guías, reglas, etc., en cambio las pequeñas obtienen sus mejores resultados derivados de la creatividad y de la dedicación de los recursos humanos implicados en el proyecto de mejora.

Esto significa que para las pequeñas empresas puedan implantar programas de mejora deben dedicar todo su esfuerzo en conseguir la participación de los empleados y en la exploración de nuevo conocimiento. En resumen, los procesos formales de las grandes organizaciones se suplen con la relación informal e interpersonal que se realiza en las pequeñas (Mas, 2005).

El Centro de desarrollo de software (CDS) del Instituto Tecnológico Superior del Sur de Guanajuato (ITSUR) tiene demanda para el desarrollo de estas herramientas tecnológicas en los sectores públicos y privados, sin embargo, presenta desventajas con respecto a las grandes empresas desarrolladoras, como lo son:

- La falta de tiempo asociada a la diversificación de actividades del reducido personal capacitado que tiene el CDS, pues los recursos humanos, además de cubrir con actividades de docencia también deben de realizar otras que comprenden su jornada laboral, situación

que dificulta el obtener y mantener certificaciones que proporcionan calidad a sus productos, situación que ha causado la pérdida de la certificación CMMI, que permitía al ITSUR realizar mejores prácticas en el desarrollo de software.

- El reducido presupuesto económico del CDS, para la realización de proyectos de desarrollo, que es reducido y no alcanza a cubrir los requisitos de pago y la implementación de certificaciones.
- La pérdida de la certificación CMMI ha ocasionado distintas formas de trabajar en el desarrollo de software, en las dos líneas de investigación, cada una de ellas opto por su mejor forma de trabajo, al igual que el personal ha adoptado distintas metodologías, debido a la falta de un estándar laboral y técnico, y
- La escasa información que se tienen en el CDS sobre otros estándares que otorgan el reconocimiento de calidad en los productos desarrollados.

Bajo esta problemática, se diseñará una propuesta metodológica para la implementación del ISO 29110 en el centro de desarrollo de Software del ITSUR, que servirá para tener productos de calidad y además para ser considerada como una de las empresas pioneras en el sur del estado de Guanajuato con la capacidad de responder a las demandas de formación de Recursos Humanos altamente calificado para el desarrollo de herramientas tecnológicas de alta calidad.

## Justificación

El desarrollo de software tiene ventajas competitivas, que generan mayores ganancias y dividendos para las empresas desarrolladoras de software. Una de estas ventajas son las certificaciones o estándares que permitan a la empresa tener un reconocimiento oficial que muestre que sus productos desarrollados tengan una calidad exigida por dichas normas o certificaciones. Para llegar a esto los procesos tecnológicos juegan un papel importante, porque con ellos se desarrolla paso a paso todas las actividades requeridas para llegar al objetivo final, pero si estos procesos no están bien definidos, se obtendrán productos deficientes y sin calidad. Los recursos materiales, humanos y tecnológicos con los que cuentan las empresas son importantes al conseguir una certificación, sobre todo en su aplicación que implica tiempo, recursos humanos y capital en la empresa. En las pequeñas y medianas empresas desarrolladoras de software (PYMES) se buscan implementar certificaciones o estándares que garanticen la calidad de sus productos para competir con las grandes empresas, sin embargo, se necesitan de todo tipo de recursos que la PYMES no poseen, es por eso, que se creó la norma ISO/IEC 29110, que adapta a las necesidades y a los recursos con las que cuentan las PYMES. Esta norma permite generar la calidad en el proceso y la implementación del desarrollo del software, que proporciona ventajas competitivas a estas empresas.

Por eso se busca diseñar una propuesta metodológica que permita al CDS del ITSUR tener las herramientas y el conocimiento tecnológico necesario para implementar el estándar ISO/IEC 29110.

Dicha propuesta traerá consigo diferentes beneficios:

- **Beneficios Administrativos:** se tendrá un mejor manejo y control de la documentación desde la fase de anteproyecto, la cual dictará si se consolida como un proyecto o no, todo lo acordado quedará de manifiesto en minutas y hojas de reuniones donde queda asentado que se realiza o no en el proyecto, la documentación se disminuye respecto a otras certificaciones lo que crea agilidad a los procesos de implementación de Software como a las tareas realizadas.
- **Beneficios operativos:** al ser una ISO que se enfoca a las muy pequeñas empresas, encaja de una manera precisa al centro de desarrollo de software, ya que se adapta a los diferentes modelos del desarrollo, con esto limitará las tareas y dará un camino a seguir para los desarrolladores, lo que disminuirá la carga de tareas, y era mas eficiente y eficaz las actividades que desarrollarán.
- **Beneficios académicos:** al ser implementado en una universidad, tiene beneficios no solo para los docentes del desarrollo, sino también para los estudiantes, ya que muchos de ellos

realizan sus practicas o residencias en el CDS , tendrán la oportunidad de trabajar con una certificación que es utilizada en el mundo laboral actual, además de obtener la experiencia de trabajar en un proyecto que cuente con esta certificación. Los docentes del CDS pueden enseñar en sus clases lo que esta certificación ofrece, como se desarrolla y como se implementa desde la etapa inicial hasta la terminación de esta. Este es uno de los principales beneficios que se obtendrá de esta metodología, ya que no solo quedará implementada en la institución, si no que muchas personas se verán beneficiadas de la misma.

- Beneficios económicos: Ya que esta certificación es especial para las muy pequeñas empresas (de 1-25 empleados) el costo de su implementación disminuye de forma considerable respecto a otras certificaciones, su implementación va de la mano al momento de realizar un proyecto, esto hará que los costos generados por el CDS disminuyen, y generen ingresos que les permitan invertirlos en proyectos a futuro, este beneficio también es muy provechoso para una desarrolladora perteneciente a una universidad, y por lo que se opto por cambiar el CMMI por el ISO/IEC 29110e.

# Objetivos

## ***Objetivo General.***

- Diseñar una propuesta metodológica para la implementación de la norma ISO/IEC 29110 en el centro de desarrollo de software del Instituto Tecnológico Superior del Sur de Guanajuato (ITSUR) para generar calidad en los productos desarrollados.

## ***Objetivos Específicos.***

- Realizar un diagnóstico técnico del centro de desarrollo de software para identificar debilidades y fortalezas para la mejora del software.
- Analizar la norma ISO/IEC 29110 para conocer las actividades y reglas a seguir para su implementación.
- Elaborar la propuesta metodológica de implementación de la ISO/IEC 29110 en el CDS.



# Capítulo II

## Marco teórico

### 2.1 Gestión de Procesos

La Gestión de Procesos de Software trata sobre gestionar exitosamente los procesos de trabajo asociados con el desarrollo, mantenimiento y soporte de productos de software. Por gestión exitosa se entiende que los productos y servicios generados por los procesos cumplen completamente con los requisitos del cliente interno y externo, y que ellos satisfacen los objetivos de negocio de la organización responsable de desarrollar los productos según Florac, 1997. Para la aplicación efectiva de la Gestión de Procesos de Software Florac, 1999 propone asumir cuatro responsabilidades claves que son: definir el proceso, medir el proceso, controlar el proceso y mejorar el proceso.

El objetivo de la gestión de procesos en el CDS del ITSUR es satisfacer los requisitos de los clientes, conocer lo que haremos y como lo haremos.

Con la gestión de procesos podemos obtener amplios beneficios, por ejemplo:

- Conocer lo que hacemos y cómo lo hacemos.
- Realizar mejoramiento por el solo hecho de describir un proceso.
- Aplicar métodos de mejora continua y aseguramiento de calidad que nos permitirán aumentar la eficiencia y la eficacia.
- Comparar nuestros procesos con las mejores prácticas del medio y así aprender y mejorar.
- Rediseñar un proceso para obtener rendimiento muchos mayores.
- Fortalecer la gestión del conocimiento, porque cada proceso levantado es conocimiento formal de la organización.
- Innovar a diferentes niveles de profundidad: proceso, actividad y tarea.
- Realizar verdaderamente control de gestión, porque parte del cambio en los procesos consiste en obtener información relevante, tal como incorporar indicadores en tiempo real y adecuadamente comparados en el tiempo (Carrasco, 2011).

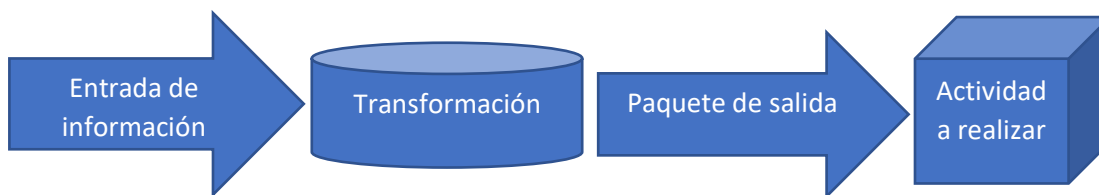
## 2.2 Proceso

El concepto de proceso lo introdujo por primera vez Adam Smith (Kim y Ramkaran, 2004), a partir del principio de dividir el trabajo en tareas y llevarlas a su mínima expresión, a fin de incrementar la productividad con el argumento de que esto conlleva el aumento de la destreza del personal, elimina retrabajos y amplía la capacidad de uso de las máquinas (Kim, 2004).

Los procesos, se pueden definir como secuencias ordenadas y lógicas de actividades de transformación, que parten de unas entradas (información, datos, especificaciones, requisitos, más medios materiales – máquinas, materias primas, consumibles, etc.), para alcanzar resultados programados, que se entregan a quienes lo han solicitado, los clientes de cada proceso (Zaratiegui, 1999).

En el CDS del ITSUR la carga de trabajo se divide en dos grandes grupos, cada uno de ellos tiene sus procesos a realizar de acuerdo con su línea de investigación. A cada miembro del grupo se le asigna procesos a realizar, los cuales son realizados en la forma que ellos deseen, pero con los resultados óptimos para el cliente. Esto nos da la pauta para implementar un estándar que ayude a la organización y gestión de los procesos realizados.

Para la realización correcta de un trabajo, lo ideal es fragmentarlo hasta el punto más pequeño, y de ahí crear pequeños procesos que permitan tener entradas de información para transformarla en actividades de salida que ayuden a ensamblar el trabajo programado.



*Ilustración 1 Proceso*

Ilustración elaborada por: el autor.

Los procesos deben de tener las siguientes características:

- Emplean Insumos, son el input de los procesos.
- Agregan un valor, transformación de los insumos.
- Utilizan los recursos de la organización, se dispone de los recursos humanos, materiales y financieros (medios, personas, equipos, tiempo, capital e inventario).

- Tienen un costo, el de los insumos y recursos de la organización involucrados en el proceso.
- Producen un resultado, el objetivo de un proceso es lograr un producto o servicio.
- Tienen un cliente externo o interno, que es el destinatario de lo generado por el proceso.
- En su ejecución participan generalmente varias áreas de la empresa, mientras la empresa está organizada verticalmente (áreas funcionales) el proceso tiene un flujo horizontal que puede involucrar a varias áreas funcionales (Flores, 2003).

## 2.3 Software

El software de computadora es el producto que construyen los programadores profesionales y al que después se le da mantenimiento durante un largo tiempo. Incluye programas que se ejecutan en una computadora de cualquier tamaño y arquitectura, el contenido se presenta a medida que se ejecutan los programas de cómputo e información descriptiva, tanto en una copia dura como en formatos virtuales que engloban virtualmente a cualesquiera medios electrónicos (Pressman, 2010).

En palabras más simples, el software es el conjunto de programas o aplicaciones, instrucciones y reglas informáticas que hacen posible el funcionamiento del equipo. Estos programas informáticos hacen posible la ejecución de tareas específicas dentro de un computador. Por ejemplo, los sistemas operativos, aplicaciones, navegadores web, juegos o programas. Estas características siempre trabajan de la mano. Mientras el Software aporta las operaciones, el hardware es el canal físico por el cual dichas funciones pueden realizarse (Goodwill Community Foundation, 2020).

El software ha sido un gran aliado para el ITSUR, ya que en el CDS se han desarrollado programas que han beneficiado de gran manera a la comunidad universitaria, tal es el caso del SICENET en el cual un alumno puede revisar su Kardex de calificaciones, dar de alta materias, y escoger los horarios y matriculas que el alumno desee. Otro Software importante para el ITSUR es la realización de los exámenes de admisión para los alumnos de nuevo ingreso, en los cuales, se olvidan los docentes del papel y se enfocan en las calificaciones que el software muestre, todo esto facilita la estadía en el ITSUR, tanto para los docentes, como a los alumnos en general.

## 2.4 Ingeniería de Software

La noción de Ingeniería del Software fue propuesta inicialmente en 1968 en una conferencia para discutir lo que en ese momento se llamó “La Crisis del Software”. Esta crisis del software fue el resultado de la introducción de las nuevas computadoras hardware basadas en circuitos integrados.

Su poder hizo que las aplicaciones hasta ese entonces irrealizables fueran una propuesta factible. El software resultante fue de órdenes de magnitud más grande y complejo que los sistemas de software previos (Sommerville, 2005).

En 1990 IEEE publica el estándar 610.12 que establece el glosario de términos para la Ingeniería del Software, en la cual se define como la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento del software (Pantaleo, 2012).

La ingeniería de software juega un papel importante en el desarrollo de las aplicaciones de software, con ellas los desarrolladores pueden guiarse con lineamientos establecidos para poder tener éxito o no en el proyecto desarrollado, el CDS del ITSUR siguen al pie de la letra los procesos y lineamientos de la ingeniería de software.

## 2.5 Proceso de software

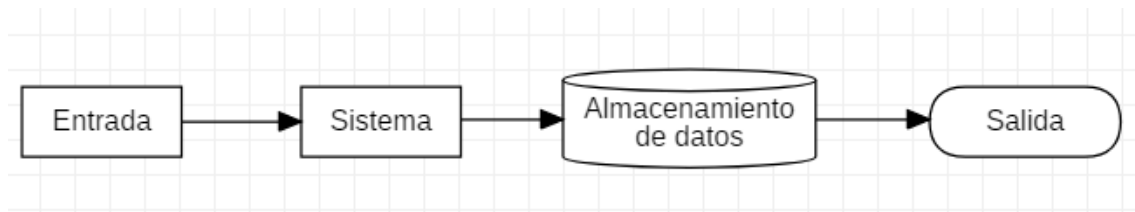
Los procesos de software son procesos también, como lo dice Osterweil (1987) es importante observar que los procesos de software son primero procesos, es decir, no es posible (y útil) para considerar a los procesos de software como entidades específicas y únicas que merecen un esfuerzo de investigación completamente autónomo e independiente. En pocas palabras como lo define Sommerville (2005) un proceso de software es un conjunto de actividades cuya meta es el desarrollo o evolución del software.

La tecnología de los procesos de software abarca tres áreas generales y superpuestas: modelado, modelos de ciclo de vida de procesos y mejora de procesos. Estas áreas abordan problemas muy críticos en la ingeniería de software. Un ejemplo típico de superposición es el modelado de procesos del ciclo de vida para lograr una mejora de procesos.

### **Modelado de procesos**

El modelado de procesos es una técnica para la organización y la documentación de los procesos de un sistema, sus entradas, sus salidas y sus formas de almacenamiento de datos. Una rama particular del modelado de procesos es el modelado de procesos de software, vale la pena aclarar que este no se centra únicamente en la descripción del proceso de software sino que va mucho más lejos (Fernández, 2006), esta técnica permite captar las características principales del proceso identificando su flujo y sus componentes para poder así comunicarlo con otros procesos,

comprender las relaciones entre sus partes y conocer sus actores, responsables y demás artefactos que intervienen en su ejecución.



*Ilustración 2 Componentes del modelado de procesos*

Elaborado por el autor.

Estos procesos pueden ser representados de diversas formas y cada una de ellas supe necesidades diferentes debido a características inherentes a las herramientas de modelado usadas y a las notaciones definidas para el entendimiento por parte de todos los participantes del proyecto (Bolaños, 2012).

Es así como aparecen herramientas que se usan para el modelado, como los diagramas de flujo de datos (Gane & sarson) una de las más utilizadas en el mundo (Fernández, 2006), propuestas como BPMN (Business Process Management Notation), EPC (Event-dri-ven Process Chain) (Dumas, 2005), que son estandarizadas por organizaciones internacionales como OMG (Object Management Group) y lenguajes como UML (Fernández, 2006) (Unified Modeling Language) y BPEL (Business Process Execution Language) (Web Service Business, 2007).

### **Ciclo de vida**

Un ciclo de vida de desarrollo de software (SDLC) es una construcción impuesta al desarrollo de un producto de software. Con frecuencia se considera un subconjunto del ciclo de vida del desarrollo de sistemas. Hay varios modelos para tales procesos, cada uno de los cuales describe enfoques para una variedad de tareas o actividades que tiene lugar durante el proceso de desarrollo de software (Bhuvanewari, 2013).

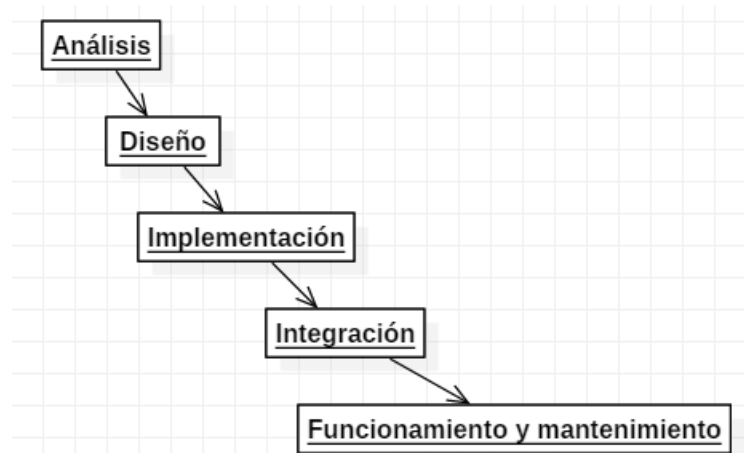
La mayoría de las organizaciones de desarrollo consideran necesario utilizar diferentes modelos de ciclo de vida dependiendo de la naturaleza del proyecto. Algunos de los enfoques que se utilizan en

la actualidad incluyen el modelo en cascada, el modelo en espiral, el modelo de desarrollo incremental y el modelo evolutivo, solo por nombrar algunos (Mead, 2000).

### **Modelo en Cascada**

Las actividades fundamentales del proceso de desarrollo de software se llevan a cabo como fases separadas y consecutivas. Estas actividades son: especificación (análisis y definición de requerimientos), implantación (diseño, codificación, validación) y mantenimiento. Los modelos en cascada consta básicamente de 5 fases que son (Cervantes, 2012):

- **Análisis y definición de requerimientos:** se trabaja a la par con el cliente para saber las necesidades y funciones que debe de tener el sistema, además de abordar las restricciones que debe de tener el mismo software, con esta información se produce el documento de “Especificación de Requerimientos del Sistema”.
- **Diseño del sistema y del software:** Durante esta etapa del ciclo en cascada, se identifican los requerimientos funcionales y no funcionales, es decir se separan los requerimientos que pertenecen al funcionamiento del software y cuales pertenecen a la interfaz del software, después de esto se establece la arquitectura a seguir en proyecto. Durante el diseño del software se identifican los subprocesos que tendrá el sistema y se describe su funcionamiento de forma individual y su relación del uno con el otro.
- **Implementación y validación del sistema:** En esta fase se codifica cada uno de los subsistemas obtenidos de la etapa de diseño de forma individual, para luego probar que cada uno de ellos cumpla con sus funciones tanto estéticas como de funcionamiento.
- **Integración y validación del sistema:** En esta etapa se integran cada uno de los subsistemas programados, para formar un sistema en sí, que cumpla con todos los requisitos tanto de funcionalidad como los requerimientos no funcionales. Cuando se prueba el sistema y pasa la revisión exitosamente, éste sistema se entrega al cliente.
- **Funcionamiento y mantenimiento:** En esta etapa se instala el sistema en el ordenador del cliente y se pone en marcha para su función, el mantenimiento implica ajustar errores que no se descubrieron en etapas anteriores en esta, así como dar mejoras y actualizaciones al mismo para crear mejor rebustes en el sistema.



*Ilustración 3 Modelo Cascada*

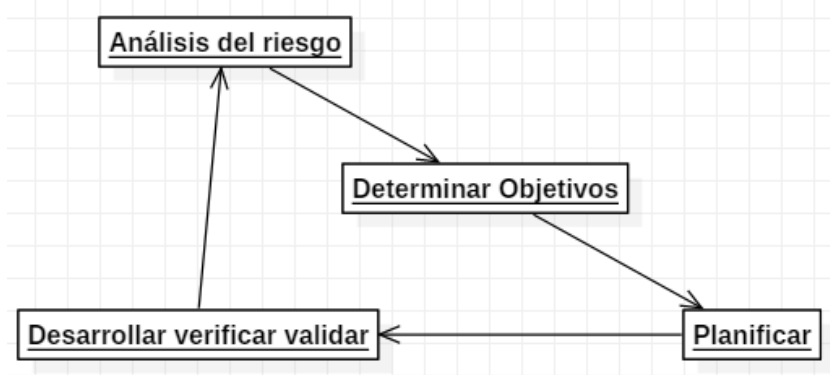
Elaborado por el autor.

### **Modelo en espiral**

El modelo espiral en el desarrollo del software es un modelo meta del ciclo de vida del software donde el esfuerzo del desarrollo es iterativo, tan pronto culmina un esfuerzo del desarrollo por ahí mismo comienza otro. Con cada iteración alrededor de la espiral, se crean sucesivas versiones del software cada vez más completas y, al final, el sistema de software ya queda totalmente funcional. La diferencia principal entre el modelo espiral y los diferentes modelos existentes, es la evaluación del riesgo. Además, en cada ejecución del desarrollo se sigue cuatro pasos principales (Galo, 2011):

1. **Determinar o fijar los objetivos:** En este primer paso se definen los objetivos específicos del sistema, así como las limitaciones del proceso y del desarrollo del software, además se diseña la planificación de la gestión y se identifican cada uno de los riesgos que puedan surgir en el proceso del desarrollo del software.
2. **Análisis del riesgo:** En este paso se realiza un diagnóstico y análisis de los riesgos resultantes de la actividad anterior, se define un plan de contingencia para reducir o acabar con los riesgos encontrados y posteriormente se realiza un análisis exhaustivo para encontrar estrategias para disminuir o erradicar estos riesgos.
3. **Desarrollar, verificar y validar:** En este tercer paso, después del análisis de riesgo, se elige la forma en la cual se decodificará el desarrollo del programa. Una vez elegida esta forma de programación, se empieza con el desarrollo del software.
4. **Planificar:** En este último paso del modelo en espiral, se revisa de forma completa el proyecto y se hace la observación si es necesario seguir con el modelo en espiral, o se

necesita de la ayuda de un ciclo posterior a este. Si se desea continuar con otro modelo se desarrollan los planes necesarios para la siguiente fase del proyecto actual.



*Ilustración 4 Modelo en Espiral*

Elaborado por el autor.

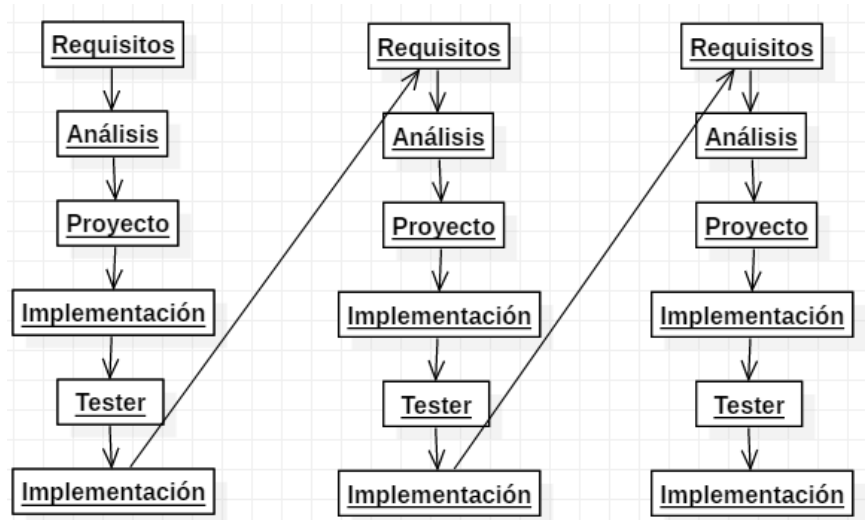
### **modelo Incremental**

El modelo incremental según Pressman (2006) combina elementos del modelo de cascada, aplicándose de forma interactiva. El modelo de proceso incremental es interactivo al igual que la creación de prototipos, más diferentes de la creación de prototipos, el objetivo incremental es presentar un producto operativo para cada incremento realizado. Este modelo es de gran utilidad cuando la empresa no cuenta con mano de obra disponible en el momento para una implementación completa, dentro del plazo estipulado. Este modelo cuenta con las siguientes fases:

- **Levantamiento de Requisitos:** En esta etapa del desarrollo, el cliente da a conocer los detalles acerca del proyecto, que funcionalidad requiere en él, los atajos las características y funciones del proyecto, también da a conocer sus alcances y limitaciones del proyecto a desarrollar.
- **Análisis de los requisitos:** En esta segunda etapa se analizan los requisitos levantados en la etapa anterior, en esta fase se separan los requisitos funcionales y los no funcionales, esta separación servirá para la programación de la fase siguiente, también se analiza el alcance del proyecto y las limitaciones que tendrá impuestas.



- **Proyecto:** En esta fase se programan los requisitos tanto funcionales como no funcionales, teniendo en cuenta los alcances y limitaciones del proyecto, así como el análisis de los requisitos de la sección anterior.
- **Implementación:** En esta fase se implementa una versión beta del programa en un ambiente similar al que tendrá el o los usuarios finales, esto se hace para hacer las pruebas finales del programa.
- **Tester:** En esta penúltima etapa del proceso incremental se realizan las pruebas del programa para ver que todo funciona en orden, en caso de que algún módulo presente algún fallo y no pase el testeado el software debe de remediar el error para poder ser entregado al cliente.
- **Implementación:** En esta etapa final, una vez pasadas las pruebas de testeado (fase anterior) se instala el programa en el ambiente del usuario, en dicho ambiente el programa trabajará tanto como el usuario final lo requiera.



*Ilustración 5 Modelo Incremental*

Elaborado por el autor.

Cabe resaltar que este modelo una vez terminado la etapa final puede regresar al análisis de los requerimientos, ya que sus fases pueden ser utilizadas para crear cada módulo o submódulo como se requiere y volver a comenzar con el modelo en cascada, (es por eso que se dice que es una combinación del modelo en cascada).

## **Modelo evolutivo**

Los modelos evolutivos son iterativos. Se caracterizan por la forma en que permiten a los ingenieros de software desarrollar versiones cada vez más completas del sistema. Los modelos evolutivos iteran sobre las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla a partir de los requerimientos prioritarios o los que están mejor definidos. Esta primera versión se refina en una nueva iteración con las peticiones del cliente para producir un sistema que satisfaga sus necesidades (Cervantes, 2012).

## **2.6 Metodología**

Una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información. Una gran variedad de estos marcos de trabajo han evolucionado durante los años, cada uno con sus propias fortalezas y debilidades. Cada una de las metodologías disponibles es más adecuada para tipos específicos de proyectos, basados en consideraciones técnicas, organizacionales de proyectos y de equipo. Una metodología de desarrollo de software o metodología de desarrollo de sistemas en ingeniería de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información (Maida, 2015).

## **2.7 Metodologías Ágiles**

En una reunión celebrada en febrero de 2001 en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de Software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que surgen a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Varias de las denominadas metodologías ágiles ya estaban siendo utilizadas con éxito en proyectos reales, pero les faltaba mayor difusión y reconocimiento (Letelier, 2006).

Según Cano (2012) las metodologías ágiles emergen como una posible respuesta para llenar ese vacío metodológico. Por estar especialmente orientadas para proyectos pequeños, las metodologías

ágiles constituyen una solución a medida para ese entorno, aportando una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto.

Las metodologías ágiles son flexibles, pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto. Los proyectos ágiles se subdividen en proyectos más pequeños mediante una lista ordenada de características. Cada proyecto es tratado de manera independiente y desarrolla un subconjunto de características durante un periodo de tiempo corto, de entre dos y seis semanas. La comunicación con el cliente es constante al punto de requerir un representante de él durante el desarrollo. Los proyectos son altamente colaborativos y se adaptan mejor a los cambios; de hecho, el cambio en los requerimientos es una característica esperada y deseada, al igual que las entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente (Ghosh, 2012).

Según Molina (2018) Entre las más notables metodologías de desarrollo ágil, se encuentran:

- Scrum (muy popular entre emprendimientos)
- Programación extrema (XP)
- Cristal Clear
- Mobile-D (ágil y extrema para móviles)
- Adaptive Software Development (ASD)
- Lean Development

## 2.8. Metodología SCRUM

En el año 1986 Takeuchi y Nonaka publicaron el artículo “The New Product Development Game” ellos se fijaron en empresas tecnológicas que, estando en el mismo entorno en que se encontraban otras empresas, realizaban productos en menos tiempo, de buena calidad y menos costes. Observaron empresas como Honda, HP, entre otras, se dieron cuenta de que el producto no seguía unas fases en las que había un equipo especializado en cada una de ellas, sino que se partía de unos requisitos muy generales y el producto lo realizaba un equipo multidisciplinar que trabajaba desde el comienzo del proyecto hasta el final (Gallego, 2012).

Según Gallego (2012) Se comparó esta forma de trabajo en equipo, con la colaboración que hacen los jugadores de Rugby y la utilización de una formación denominada SCRUM. SCRUM aparece como

una práctica destinada a los productos tecnológicos y será en 1993 cuando realmente Jeff Sutherland aplique un modelo de desarrollo de Software en Ease/Corporation.

La metodología de desarrollo de Software SCRUM, se define como un proceso de administración y control para el desarrollo de software, en el cual se deja de lado la complejidad de las metodologías de desarrollo de software tradicionales, para concentrarse en suplir las necesidades planteadas por el usuario, y mantener a los actores involucrados en el proceso de desarrollo, enfocados en alcanzar de una forma eficaz los objetivos trazados al inicio (Becerra, 2018).

La metodología SCRUM está representada esquemáticamente por la Figura 6.



Ilustración 6 Descripción conceptual de un ciclo SCRUM

La totalidad de los requerimientos a desarrollar, denominados historias de usuarios (*“user stories”*) son divididos en grupos en función de su prioridad relativa para luego ser implementados en ciclos de esfuerzos relativamente cortos (del orden de un mes de duración) llamados *“Sprints”*, las tareas son organizadas en el equipo de tal manera que las asignaciones y prioridades se revisan diariamente en una breve reunión llamada *“scrum”* que le da su nombre a la metodología. En este enfoque se siguen los principales criterios del Manifiesto obteniendo liberaciones parciales incrementales del producto bajo desarrollo. La evidencia es consistente que al abrazar la hoja de ruta y comprometer las inversiones necesarias para desplegar formalmente esta metodología también se abordan al mismo tiempo aspectos clave del despliegue de prácticas maduras del proceso (Colla, 2012).

El siguiente listado está compuesto por cinco ventajas que tiene el modelo Scrum sobre otro tipo de metodologías que se han logrado identificar, a través del uso de otras, en diferentes proyectos de Software:

- **Satisfacción del cliente:** el gran diferenciador de las metodologías ágiles es que hacen al cliente parte del equipo de trabajo y lo comprometen con el resultado final. El uso de metodologías tradicionales en contraposición a un modelo de tipo ágil y en especial a SCRUM, implica los siguientes riesgos:
  - Es posible que la funcionalidad que se había pactado en los documentos iniciales cambie debido a diversos motivos, como políticas de la compañía o cambio de personal.
  - Por razones administrativas, las personas que aprueban los documentos pueden ser diferentes a las que reciben el producto, por lo que se pueden tener discrepancias en cuanto a la entrega final.
  - Llegar a largas discusiones con el cliente, porque la interpretación de las frases en el documento tiene diferentes expectativas para los diferentes interesados.
  - Lograr que el cliente apruebe el sistema que se está entregando puede ser complicado al hacer al cliente o cambios gráficos.
- **Simplicidad:** Los eventos manejados por SCRUM están claramente identificados; indicando para cada uno: quienes participan, su objetivo, el tiempo que debe tomar y cuál es el resultado esperado. Lo cual en esencia facilita a los integrantes del equipo la adopción de la metodología.
- **Inspección:** uno de los componentes que resalta SCRUM, es la inspección y por ello, tres de sus eventos están orientados a estos objetivos: la reunión diaria, la revisión del sprint y la retrospectiva de este último. Este componente es uno de los preferidos en las organizaciones, ya que les permiten ver qué tan bien se va adaptando la metodología a su cultura y si los beneficios prometidos se están evidenciando.
- **Adaptación:** la mejor parte de la metodología es la disposición que tienen al cambio las características del producto. Esta adaptación beneficia a la organización en la medida que aporta a la satisfacción del cliente y los ingresos por ajuste.
- **Trabajo en equipo:** algo particularmente interesante de SCRUM es cómo logra la sinergia entre las personas que participan en el proceso, a tal punto que en cada iteración -ciclo de desarrollo-, el mismo equipo se adapta para mejorar. Esto también implica que cada

individuo sea reconocido como parte esencial del equipo, por lo cual el impacto del cambio de una persona puede llegar a ser alta. Comparativamente, SCRUM logra dar visibilidad al equipo de trabajo, ya que, en metodologías tradicionales, las personas que lo integraban no tenían relación directa con el cliente, lo cual, en términos de reconocimiento, es importante y vale la pena destacar (Rodríguez, 2015).

## 2.9. Metodología Extreme Programming (XP)

La programación extrema o *Extreme Programming*, es una disciplina de desarrollo de software basada en los métodos ágiles, que evidencia principios tales como el desarrollo incremental, la participación activa del cliente, el interés en las personas y no en los procesos como elemento principal, y aceptar el cambio y la simplicidad. El trabajo fundamental se publicó por Kent Beck en 1999, y tomó el nombre de Programación Extrema por las prácticas reconocidas en el desarrollo de software y por la participación del cliente en niveles extremos. Este método, al igual que RUP y MSF, también tiene principios los cuales son buenas prácticas para tener presentes en el desarrollo de Software (Pérez, 2011).

XP resalta una serie de valores y principios que deben tenerse en cuenta y practicarlos durante el tiempo de desarrollo que dure el proyecto. Al final se enuncian algunas características que deben tener los proyectos con XP. Estos valores son: La comunicación, la simplicidad, la retroalimentación y el coraje (Tobón, 2007).

Según Tobón y Carmona (2007) a partir de los valores se plantea una serie de prácticas que sirven de guía para los desarrolladores en esta metodología. Uno de los aspectos más importantes para XP son las doce reglas que se plantean, las cuales se caracterizan por su grado de simplicidad y por su enfoque de practicidad, además de que cada regla se complementa con las demás. Las 12 prácticas son las siguientes:

- El desarrollo está dirigido por pruebas.
- El juego de la planificación.
- Cliente in-situ.
- Programación en parejas.
- Entregas pequeñas.
- Refactorización sin piedad.
- Integración continua del código.

- Diseño simple.
- Utilización de metáforas del sistema.
- Propiedad colectiva del código.
- Convenciones de código.
- No trabajar horas extras.

## 2.10 Calidad

La calidad de software es de risa. Por supuesto no suscribimos esta sentencia, pero también está claro que hay mucho camino que recorrer hasta lograr la satisfacción del cliente o el usuario final. La calidad, sea de software o de cualquier otra cosa, requiere una visión integral. Algo bueno sólo en parte, no es muy satisfactorio. La percepción de calidad en la experiencia de uso del software no depende sólo del producto. El entorno (sistema) en que se ejecuta es igual de determinante. A su vez, la calidad del producto software es función de los procesos que lo han generado y la calidad del sistema función de su buen gobierno (Calero, 2010).

Una definición amplia de calidad, planteada en la norma UNE-EN ISO 402, expresa que “la calidad es el conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas” (AENOR).

Llevada esta definición al campo de la ingeniería de software, la IEEE Std 610 (1990), señala que “la calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”.

Por su parte Pressman (2005), se refiere a la calidad del software como “la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, estándares de desarrollo explícitamente documentados y características implícitas que se espera de todo software desarrollado profesionalmente”.

A la hora de definir la calidad de los SS (Sistemas de Software) se debe diferenciar entre la calidad del producto software y la calidad del proceso de desarrollo de éste (calidad de diseño y fabricación). No obstante, las metas que se establezcan para la calidad del producto van a determinar los objetivos del proceso de desarrollo, ya que la calidad del primero va a depender, entre otros aspectos, de éstos. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto (Mendoza, 2005).

## 2.11 CMMI

Un modelo es una representación simplificada del mundo. Los modelos de capacidad de madurez (CMM) contiene los elementos esenciales de procesos efectivos para uno o más cuerpos de conocimiento. Estos elementos se basan en los conceptos desarrollados por Crosby, Deming, Juran y Humphrey (Crosby 79, Juran 88, Deming 86, Humphrey 89). Al igual que otras CMM, los modelos de integración de modelos de madurez de capacidad (CMMI) proporcionan una guía para usar al desarrollar procesos.

Los modelos CMMI no son procesos ni descripciones de procesos. Los procesos reales que se utilizan en una organización dependen de muchos factores, incluidos el (los) dominio(s) de aplicación y la estructura y tamaño de la organización. En particular, las áreas de proceso de un modelo CMMI normalmente no se asignan una a una con los procesos utilizados en su organización (Carnegie Mellon Software Engineering Institute, 2002).

El nuevo CMMI brinda un marco con una estructura común para todas las disciplinas (Ingeniería de Software, Ingeniería de Sistemas, etc.) e incorpora una forma de representación llamada Continua (tomada de IPD-CMM y SE-CMM), orientada a medir la mejora en los procesos de manera individual en vez de hacerlo de manera conjunta como la representación por niveles del modelo original (escalonado) (Universidad Politécnica de Madrid, 2009).

El modelo CMMI define 22 áreas de proceso en la versión que integra el desarrollo de software e ingeniería de sistemas (CMM-SE/SW). Cada una de estas áreas de proceso se conforman como un conjunto de buenas prácticas en las actividades relacionadas con un determinado proceso del desarrollo de software (CMMI, 2009).

Entre los beneficios de implantar CMMI se pueden encontrar los siguientes (Qualitrain, 2009):

- Mejora alineación a objetivos de negocio.
- Mayor eficacia en la detección de errores a lo largo del ciclo de vida de los proyectos del Software, reduciendo drásticamente el número de errores que afecta directamente a los clientes y usuarios.
- Resultados más predecibles en los proyectos.
- Implementación de técnicas proactivas de gestión, mitigando los riesgos que afectan a los proyectos.



- Liberaciones de tensiones, malentendidos, y vacíos de responsabilidad en Proyectos Software.
- Disposición de información de gestión útil a la hora de tomar decisiones ya sean éstas relacionadas con la Gestión de Proyectos Software, bien con la mejora continua del Proceso de Software.
- Mejora en la calidad del producto.
- Lograr la satisfacción del cliente.
- Menores costos de desarrollo.
- Mayor rapidez de respuesta.
- Mejora la productividad.

## 2.12 Estándar ISO/IEC 29110

El estándar ISO/IEC 29110 Perfil básico es un conjunto de buenas prácticas en el desarrollo del software para asistir y evaluar a las PyMEs desarrolladoras de software en el proceso de mejora. Está compuesta del Proceso de Administración de Proyecto (AP) y del Proceso de Implementación Software (IS), cada uno de ellos posee un conjunto de roles, actividades y documentos externos, que se deben satisfacer al momento de evaluar el estado de los procesos (ISO/IEC 29110, 2011).

El estándar ISO/IEC 29110 ingeniería de Software: perfiles de ciclo de vida para entidades muy pequeñas tiene como objetivo acercar las buenas prácticas de ingeniería de software y gestión de proyectos a las VSE. Según Campos (2014), la norma ISO/IEC 29110 se divide en cinco partes de la siguiente manera: ISO/IEC TR 29110-1, ISO/IEC 29110-2, ISO/IEC 29110-3, ISO/IEC29110-4-1.

El proceso de Gestión del Proyecto establece y lleva a cabo de manera sistematizada, las tareas relacionadas con la gestión del proyecto, con la finalidad de lograr los objetivos del proyecto dentro de la calidad, tiempo y costos esperados. Este proceso está compuesto por 4 actividades: planificación del proyecto, ejecución del proyecto, evaluación y control del proyecto y, cierre del proyecto (Muñoz, 2018).

El proceso de Implementación de Software realiza de manera sistematizada, las actividades relacionadas con el desarrollo de software como: el análisis, el diseño, la construcción, la integración y, las pruebas; de acuerdo con los requerimientos especificados, ya sea para el desarrollo de un nuevo producto o la modificación de un producto. Este proceso está compuesto por 6 actividades:

inicio de implementación, análisis de requisitos, arquitectura y diseño detallado, construcción, integración y pruebas y, entrega del producto (Muñoz, 2018).

Actualmente uno de los estándares adecuados para su implementación dentro de las muy pequeñas empresas de desarrollo de productos y/o servicios de software que representan en promedio el 90% en México es el estándar ISO/IEC 2911. Con respecto a este estándar, en el 2006 fue presentado por los delegados mexicanos el modelo Moprosoft al grupo de trabajo 24 (WG24, por sus siglas en inglés) de la ISO. Posteriormente, este grupo decidió la adaptación de Moprosoft para cubrir los aspectos de las empresas muy pequeñas, tomando como base para la creación del ISO/IEC 29110 (Laporte, 2017).

# Capitulo III

## Metodología

La metodología de trabajo llevada a cabo en esta investigación cuenta con 5 fases como se muestra en la ilustración 7, cada una de las fases aquí mostradas será dividida en diferentes actividades que nos ayudarán a entender mejor el actuar de esta investigación. A continuación, veremos la primera fase de esta metodología, que lleva por nombre Diagnostico del CDS.



*Ilustración 7 metodología de trabajo*

### análisis de la certificación CMMI

Para analizar la norma CMMI se tomó como base su diagrama principal el cual lleva por nombre arquitectura de procesos el cual se muestra a continuación, este diagrama contiene todos los procesos y subprocesos que se deben de seguir para completar el CMMI. Se analizará cada uno de los procesos del diagrama.

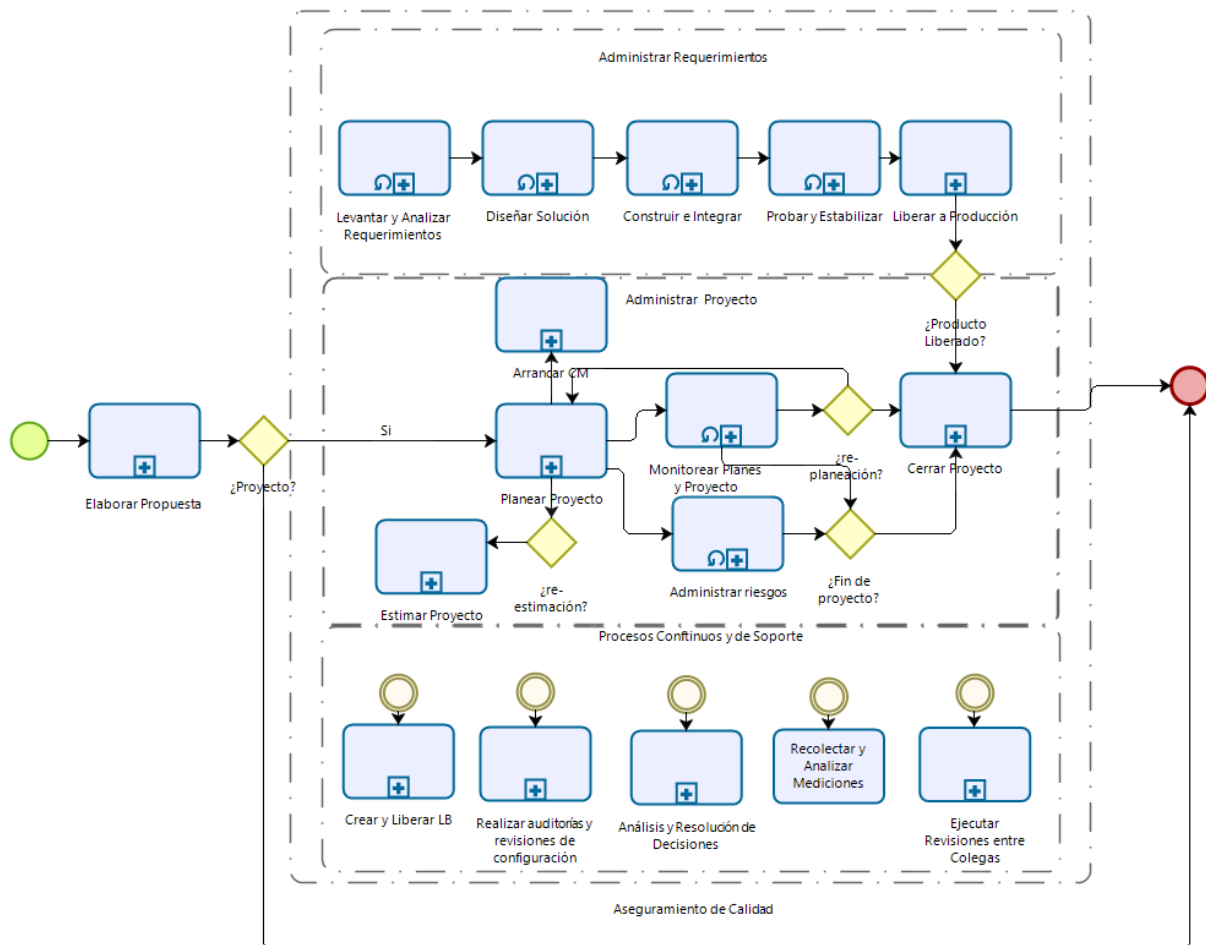


Ilustración 8 Arquitectura de Procesos

Para el análisis de cada proceso debemos de rescatar:

- los actores que participan en el proceso.
- el rol de cada participante en la certificación.
- Las actividades que contiene cada proceso.
- Los documentos de entrada de cada actividad.
- Los documentos de salida de cada actividad.
- Y las relaciones entre actividad y subprocesos.

### Análisis del CDS.

Para el análisis del CDS decidimos crear un diagrama de actividades que nos ayude a tener un mejor manejo de esta actividad. A continuación, se muestra este diagrama.



Ilustración 9 diagrama de actividades del Análisis del CDS.

### ***Análisis del CDS***

Para el análisis del CDS nos dirigimos hacia el ITSUR para ver la forma en la que se trabaja actualmente en el CDS y también como se trabaja en el pasado con la certificación CMMI, para ello tuvimos una serie de pláticas con uno de los principales encargados del CDS que nos explico ampliamente y a detalle la forma en la que se trabajaba y se trabaja en el desarrollo de software, también se tuvo pláticas con las personas que laboraban ahí y se nos mostró la estructura del CMMI, en la sección de resultados, se plasmaran de manera especifica todo lo recaudado en esta investigación.

### ***Detectar problema.***

Para llevar a cabo la formulación del problema central que aqueja el CDS, se tomó en cuenta el esquema principal que se utilizaba en la realización de los proyectos y las pláticas que se mantuvieron con los encargados del CDS, con dicha información se realizó un análisis a fondo para descubrir cual es la razón principal que está afectando el poder continuar con una certificación que permita tener a la institución una mejor calidad y prestigio a sus productos desarrollados.

Una vez obtenido el problema central, que está afectando las actividades del CDS, se deben de descubrir que efectos colaterales están surgiendo gracias a este problema, ya que el problema, tiene sus vertientes y sus efectos colaterales que puedan afectar a diferentes áreas, para esto se analizó

las prácticas que se obtuvieron con los actores principales del centro de desarrollo de software además de cruzarlo con el problema central, para obtener las consecuencias que se puedan desarrollar de estos mismos.

### *Solución del problema.*

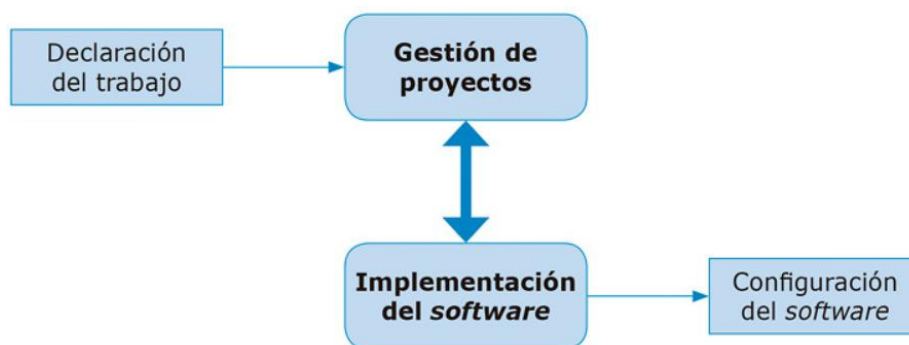
Para llegar a la solución del problema, se analizó tanto la certificación CMMI como la ISO/IEC 29110 para ver si el CDS del ITSUR era candidato para esta norma, a su vez se cruzó con el problema que aqueja el CDS para comprobar si con esta certificación se resolvía el problema o se tendría que aplicar otras medidas para solucionarlo. Mas adelante en este documento se muestra la solución de este problema.

### *Mejora continua.*

La mejora continua es muy importante en este tipo de actividades y asignaturas ya que día a día van surgiendo nuevas tecnologías que permiten tener mejor organización y mejores productos que ayuden a dar un salto de calidad a los CDS, por eso es importante llevar a cabo estas mejoras que ayuden a dar un salto de calidad a los productos finales realizados en el CDS.

### **Análisis de la norma ISO**

Para realizar el análisis de la norma ISO/IEC 29110 se debe la forma en la que opera, por lo tanto, se tuvo que leer la documentación principal de la norma, además se tuvo una serie de videoconferencias con una experta del tema, y la lectura de diferentes papers de como la ISO/IEC29110 estaba llevando a cabo su realización e implementación en diferentes empresas y universidades del mundo.



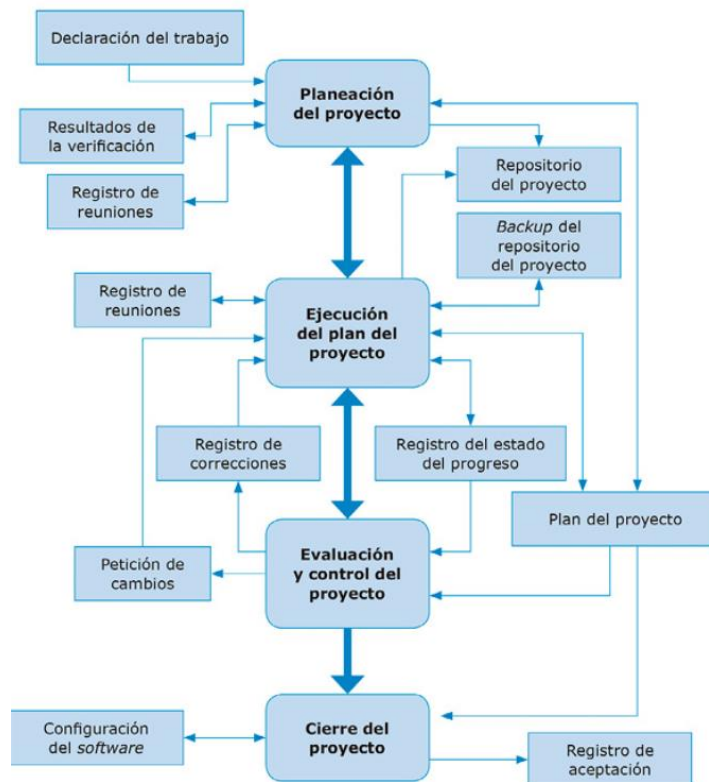
*Ilustración 10 Diagrama de trabajo de la norma ISO/IEC29110*

Para continuar con el análisis se tomó como partida el diagrama principal de procesos del perfil básico de la norma ISO/IEC 29110 que se muestra en la ilustración 10. Este diagrama tiene 2

procesos principales uno enfocado a la gestión de proyectos y el otro a la implementación del software, cada uno de estos proyectos tiene su propio diagrama de actividades que ayudan a cumplir con todos los requerimientos de la norma. A continuación, se muestran estos diagramas.

### *Gestión de proyectos*

La ilustración x muestra el diagrama de procesos de la gestión de proyectos, el cual es analizado siguiendo paso a paso cada actividad de este. En la sección de resultados se muestra El análisis de este diagrama.



*Ilustración 11 Diagrama de Gestión de Proyectos*

## Implementación del Software.

La ilustración x muestra el diagrama de procesos de la implementación del software, el cual es analizado siguiendo paso a paso cada actividad de este. En la sección de resultados se muestra el análisis de este diagrama.

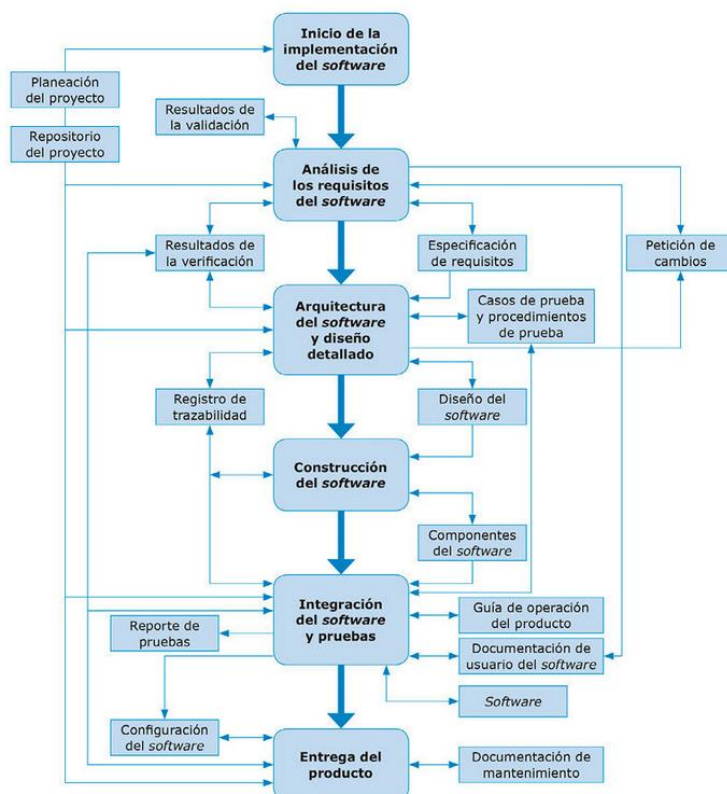
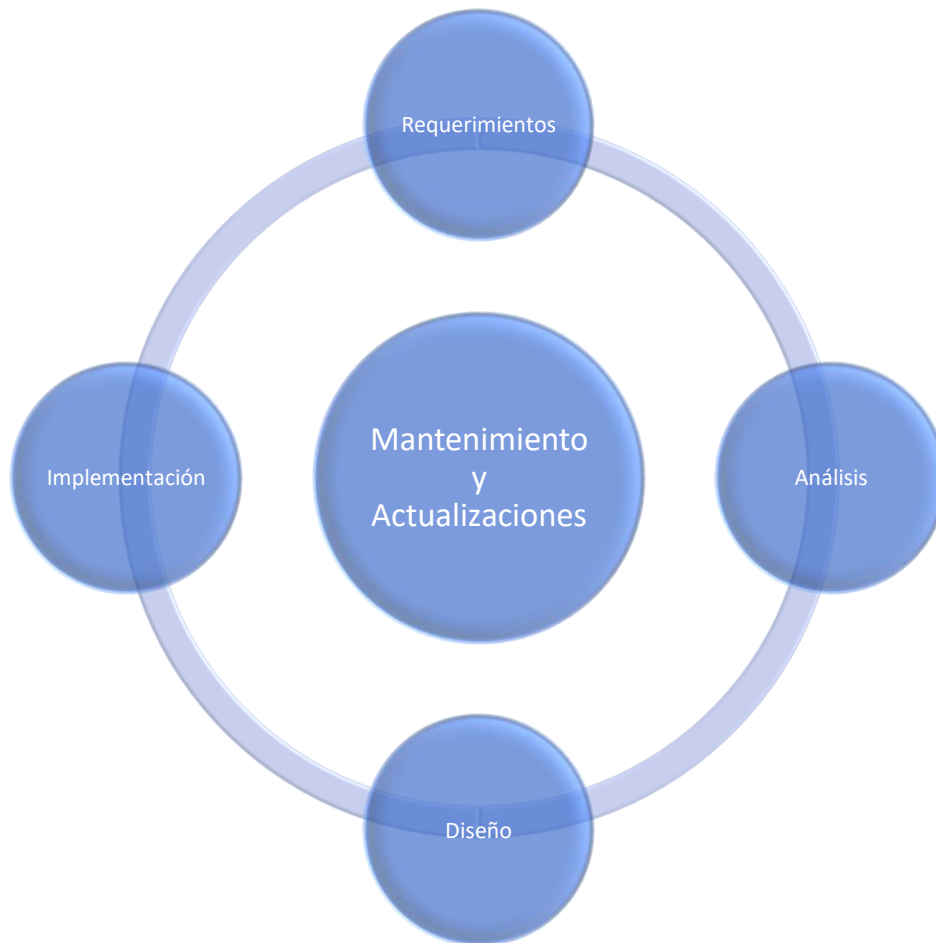


Ilustración 12 Diagrama de la Implementación del Software

## Mapeo entre la Norma ISO/IEC 29100 y la certificación CMMI.

Para lograr el mapeo entre ambas Normas, nos dimos a la tarea de crear una nueva metodología que ayudará a conseguir los resultados esperados del mapeo, y que en un futuro se pueda replicar en la realización de proyectos que garanticen cumplir con todos los requerimientos de la norma ISO/IEC 29110 y pueda encaminar al CDS a obtener dicha certificación.





*Ilustración 13 Metodología propuesta*

***Requerimientos.***

Para la realización de los requerimientos se separaron las actividades de la norma ISO/IEC 29110 y de la certificación CMMI, además de los actores que participan en ambas normas y las actividades y documentos que se deben de realizar en cada fase según su metodología, para luego analizarlas y ver el cruzamiento entre ambas.

***Análisis.***

Para el análisis se identificó las funciones de cumplimiento y no cumplimiento de la metodología ISO/IEC 29110, algunas estas funciones o actividades pueden cumplirse o no, dependiendo de las restricciones y las personas con las que cuente la organización que desea llevar a cabo su implementación, para lograr esto se debe de analizar de manera eficaz y eficiente el CDS para

descubrir si algunas funciones pueden pasar a un segundo plano o todas las funciones deben ser primordiales para llevar a cabo.

### *Diseño.*

Para llegar a diseñar la solución se tuvo que pasar tanto por la fase de requerimientos como el análisis de ambas certificaciones, una vez obtenido ambos análisis se realizó una tabla central que muestra las relaciones entre las actividades y documentos de ambas normas para así saber que documentos del CMMI se deben de modificar (ya sea ampliación, disminución, creación o eliminación) para cumplir con los estándares de la norma. Esta tabla la veremos mas adelante en la sección de resultados.

### *Implementación.*

La implementación no se llevo a cabo debido a la situación sanitaria que se esta viviendo en estos momentos, ya que hizo estragos en el proceso de implementar el diseño en uno de los proyectos que se tenía en el CDS. Esta fase y la siguiente todavía tienen un alto rango de investigación y en un futuro cuando mejoren las cosas, se plantea retomar la investigación en donde se quedo para completar la metodología propuesta.

### *Mantenimiento y actualizaciones.*

Esta fase tampoco se llevo a cabo por la situación sanitaria. En resumen, el mantenimiento y actualización de esta metodología es similar a una mejora continua, cada vez que se haga una actualización de una o ambas normas se debe de seguir este ciclo, para encontrar las modificaciones necesarias según las actualizaciones que surgieron o pueden surgir en ambas normas.

Para lograr este mapeo se debe de analizar la norma ISO/IEC 29110 como la certificación CMMI, una vez hecho este par de análisis se deben:

- de cruzar los resultados para saber qué actividades de CMMI pueden servirnos para cumplir con la norma ISO y cuáles no.
- Se deben de analizar cuales actividades no necesitan de modificaciones.
- Cuales actividades se pueden modificar para cumplir con la norma.
- Y que actividades se deben de crear desde cero.

En la sección de resultados se mostrará el mapeo que surgió entre las dos certificaciones el cual nos dará el camino a seguir para que el CDS valla en camino a conseguir la certificación de la norma ISO/IEC 29110.

# CAPITULO IV

## Resultados

### Análisis de la certificación CMMI.

#### Revisar esquema completo y verificar su lógica e integridad

Para poder diseñar una metodología que nos sirva de ayuda para lograr la certificación ISO/IEC 29110, se debe de analizar la forma en la que se trabaja en el CDS, los autores que participan en el, y que aciertos y errores tienen en su forma de trabajar. Todo esto servirá de base al momento de cruzar el análisis del CDS con el análisis de la norma ISO 29110, con esto se obtendrán conclusiones específicas que nos permitirá tomar algunas de las tareas ya implementadas en el CDS para validar las tareas que pide la norma ISO/IEC 29110 .

El siguiente esquema nos muestra la forma en la que trabaja la certificación CMMI, a este proceso se le llama “arquitectura de procesos”, y es el proceso principal para la toma de decisiones y la elaboración del proyecto.

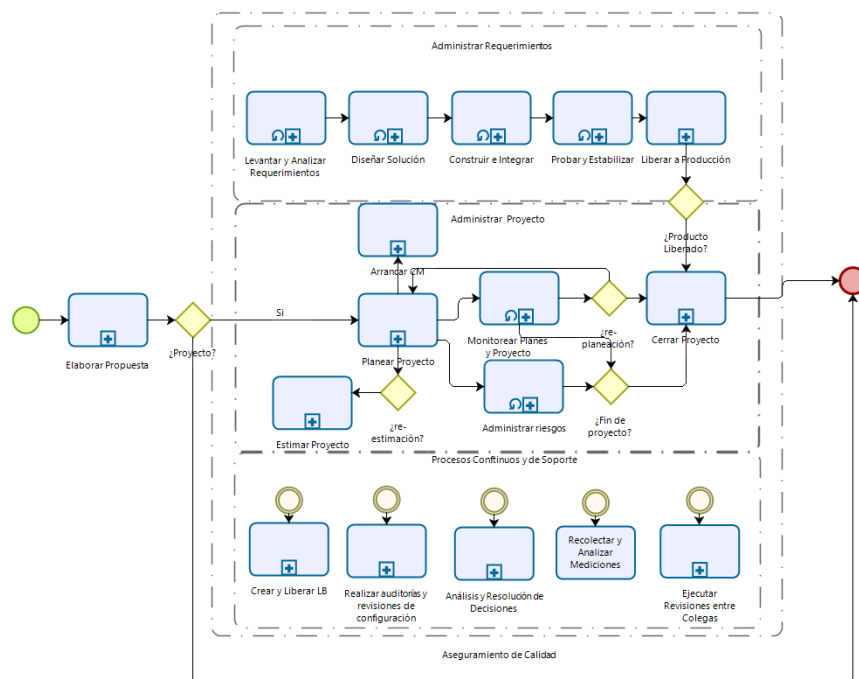


Ilustración 14 Arquitectura de procesos

En la ilustración 7 se puede observar que la arquitectura de procesos está dividida 3 etapas claves:

- **Administrar proyecto:** es la parte central del proyecto donde se lleva a cabo toda la administración y gestión de este.
- **Administrar requerimientos:** es la etapa donde se desarrolla y da solución a los requerimientos que el cliente pidió en el proyecto.
- **Procesos continuos y de soporte:** se ejercen las auditorías y revisiones de la configuración, además de analizar y solucionar las decisiones tomadas en el proyecto.

## Etapa 1. Administrar proyecto

En la etapa de “administrar proyecto” se realizan todas las actividades y subprocesos de la gestión y administración del proyecto. Contiene las actividades y subprocesos necesarios para lograr el éxito de ellas. A continuación, se muestra el diagrama de actividades que contiene esta primera etapa.

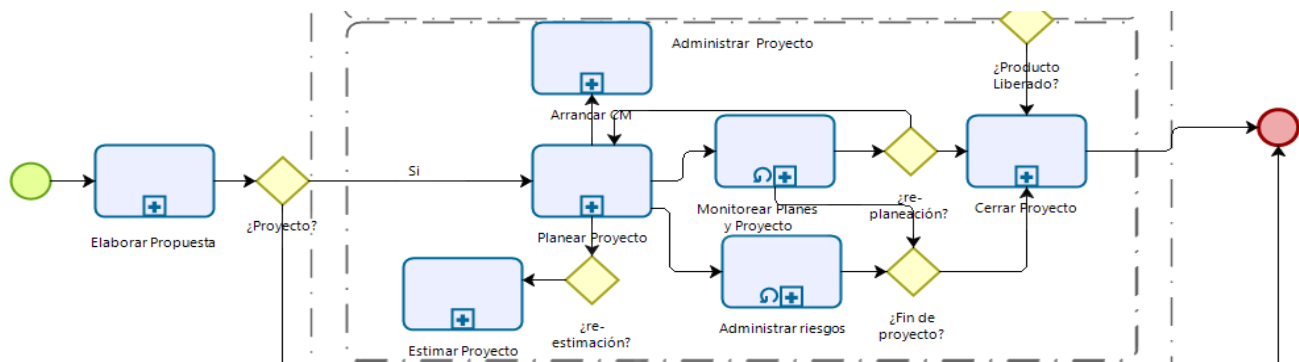


Ilustración 15 Fase1: Administrar Proyecto

## Elaborar Propuesta

Este proceso se ejecuta para identificar la visión y alcance del proyecto, de la misma forma se presenta al prospecto una propuesta técnica y económica. A continuación, se muestra el diagrama de actividades para Elaborar Propuesta.

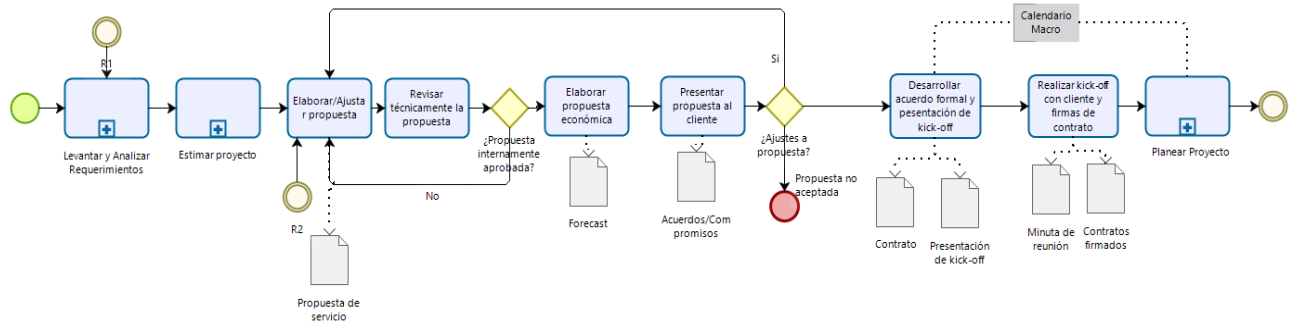


Ilustración 16 Elaborar propuesta

### Criterios de entrada/Entradas:

#### Criterios de entrada:

- El procedimiento inicia cuando se ha identificado una oportunidad de desarrollo de un proyecto.

#### Entradas:

- Documento de Visión.
- estimación.

### Criterios de salida/Salidas:

#### Criterios de salida:

- El procedimiento termina cuando se tiene un acuerdo formal que avale la relación de negocio entre el cliente y el CDS.

#### Salidas:

- Propuesta de Servicio.
- Contrato.
- Presentación de Kick-off.
- Forecast.

### Roles necesarios para el proceso (Recursos humanos):

- **Gerente del CDS:**
  - Desarrollar propuesta económica.

- Presentar propuesta al Cliente y negociar términos y condiciones del proyecto.
- Elaborar acuerdo formal de trabajo y entregarlo al Cliente para su firma.
- **Analista:**
  - Desarrollar estimación del proyecto.
  - Elaborar propuesta técnica.
- **Auditor:**
  - Asegurar la calidad de los productos y al apego al proceso

### Levantar y Analizar requerimientos.

Este subproceso se explicará en un diagrama más adelante.

### Estimar Proyecto.

Este subproceso se explicará en un diagrama más adelante.

### Elaborar/Ajustar propuesta

El Gerente del CDS elabora el documento de propuesta de servicio e integra como anexo el documento de visión.

### La propuesta describe:

- La solución técnica.
- Los compromisos establecidos, incluyendo los criterios definidos para evaluar y aceptar los requerimientos.
- Los riesgos relacionados a los requerimientos
- El alcance, descrito con claridad.
- La estimación del proyecto.

**NOTA:** El gerente general del CDS puede solicitar a un Líder de Proyecto el apoyo para la atención de la propuesta (lo que implica la elaboración, la coordinación de la revisión y la presentación y gestión con el cliente), al cual llamaremos de aquí en adelante Responsable de Propuesta. Generalmente el LP/Analista que se integre en esta fase del proyecto será el LP que administrará el proyecto. La selección dependerá de la disponibilidad y experiencia del sector del prospecto.

### Revisar técnicamente la propuesta

La propuesta de servicio se verifica por alguna de las personas que han participado en las reuniones de identificación de necesidades del cliente. Los puntos para verificar son:

- Necesidades del proyecto reflejadas en la propuesta del servicio.
- Validez de la solución técnica especificada.
- Identificación de requerimientos de alto nivel claros, precisos, verificables y sin ambigüedades.
- Identificación de exclusiones o puntos de consideración para la etapa de análisis.
- Estimación de esfuerzo.

El revisor comenta la propuesta y envía observaciones y defectos al Gerente del CDS (o responsable de propuesta) para su atención.

Una vez terminada esta actividad se hace la siguiente pregunta ¿Propuesta internamente aprobada? Si la respuesta es no tenemos que regresar a la actividad de Elaborar/Ajustar propuesta, para hacer las modificaciones necesarias que se tengan que cambiar. Una vez hecha estas modificaciones se continua con la actividad siguiente hasta llegar a una propuesta aprobada internamente, para continuar con la siguiente actividad la cual es Elaborar propuesta económica.

### **Elaborar propuesta económica**

El Gerente del CDS de acuerdo con la estimación de esfuerzo, elabora el cuadro de costos en el documento de forescat para realizar la propuesta económica.

### **Presentar propuesta al Cliente**

El Gerente del CDS/Responsable de Propuesta asignado entrega al Prospecto la propuesta. La entrega puede ser presencial para ser revisada en una reunión y/o por envío de correo electrónico y revisada por comunicación a distancia (llamadas telefónicas/ envío-recepción de correos).

Si se acuerda tener una reunión para revisión de la propuesta, deben asistir preferentemente:

- Por parte del CDS: el Gerente del CDS/Responsable de Propuesta, el personal que participó en el levantamiento de requerimientos y realización de la estimación.
- Por parte del prospecto: las personas que tengan un nivel de autoridad para aceptar la propuesta y la(s) personas(s) que fungirá(n) como responsable(s) del proyecto.

Sin importar el medio de comunicación manejado con el Prospecto el Gerente General del CDS/Responsable de Propuesta se asegura que se revisen los aspectos claves de la propuesta de servicio como:

- El alcance.



- Los requerimientos no funcionales o contractuales.
- Restricciones.
- La duración y *milestones* del proyecto.
- EL costo.
- Las responsabilidades y dependencias.

Al término de la revisión se debe lograr que exista un entendimiento común y una visión compartida de los requerimientos del proyecto con el Cliente.

**NOTA:** Por cada reunión que se haya tenido con el Prospecto se genera y envía una minuta con el registro de los acuerdos respecto al alcance, los compromisos y ajustes requeridos a la propuesta.

El Gerente del CDS/Responsable de Propuesta da seguimiento con el prospecto a fin de convertirlo en cliente.

Aquí se hace la pregunta ¿Ajustes a propuesta? Si la respuesta es sí, se debe de regresar hasta la actividad de Elaborar/Ajustar propuesta y seguir con los pasos antes mencionados. Si la respuesta es no se continua con la actividad siguiente. Pero si al cliente la propuesta no le resulta de su agrado o no es benéfica para el CDS, se da por terminado la elaboración de la propuesta y por ende el final del proyecto.

#### **Desarrollar acuerdo formal y presentación de Kick-off**

Cuando la propuesta se acuerda y acepta, el Gerente del CDS elabora el contrato indicado como ANEXOS tanto la propuesta de servicio como el documento de visión.

Si el cliente estipula el formato del acuerdo formal, el Gerente del CDS revisa el contenido, las cláusulas legales, que concuerde con la propuesta y una vez habido resuelto inconsistencias y negociando acuerdos, solicita la aprobación del Director del ITSUR.

El líder de Proyecto toma la plantilla presentación de Kick-off y desarrolla la presentación. Para su desarrollo toma la información de la propuesta de servicio, el documento de visión, el modelo de estimación utilizado y los temas de consideración comentados en la pre-venta y el contrato (p.e. restricciones, consideraciones, riesgos, etc.).

En caso de no haberse desarrollado como parte de la propuesta técnica un plan de trabajo, el Líder de Proyecto los desarrolla e integra en la presentación. En este momento, el plan de trabajo puede desarrollarse sin tanto desglose de las actividades, pero si es importante que se vea reflejadas:

- La fecha de inicio y fin.
- Las fases aplicables y el ciclo de vida.
- Principales milestone.
- Criterios globales para evaluar y aceptar los requerimientos.

### **Realizar Kick-off con clientes y firmas de contrato**

El Líder de Proyecto es el responsable de llevar la reunión con el Cliente. Durante la reunión se presentan y discuten temas como:

- Conformación del equipo por parte del CDS y el Cliente.
- Alcance del proyecto y objetivos a lograr.
- Fases del ciclo de vida.
- Principales milestones y fechas comprometidas.
- Restricciones, suposiciones y riesgos iniciales.
- Responsables de proveer aceptar y solicitar cambios a los requerimientos. Referirse al procedimiento de Levantar Requerimientos.

El Líder de Proyecto entrega el contrato y solicita al cliente su firma.

Los acuerdos y compromisos se registran en una minuta y al término de la reunión la envía a los involucrados.

### **Planear Proyecto**

Planear Proyecto es la última actividad de este subproceso, continuando con la etapa de administrar proyecto se hace la pregunta ¿Proyecto? Si la respuesta es no, se da por terminado el proyecto. En cambio, si se la respuesta es si, se continua con el subproceso de nombre Planear Proyecto el cual es desglosado a continuación.

### **Planear Proyecto**

Este proceso tiene como objetivo establecer un plan y estrategia de proyecto, que integre a todos los grupos o personas involucradas, y que se realice a partir de los procesos organizacionales ajustados a las características del proyecto.

Con la ejecución de este proceso se establecerá un presupuesto, un plan y un calendario del proyecto; mismo que serán revisados con los involucrados al proyecto para obtener un compromiso

de la planeación para el logro del objetivo final. A continuación, se muestra el diagrama de actividades para Planear Proyecto

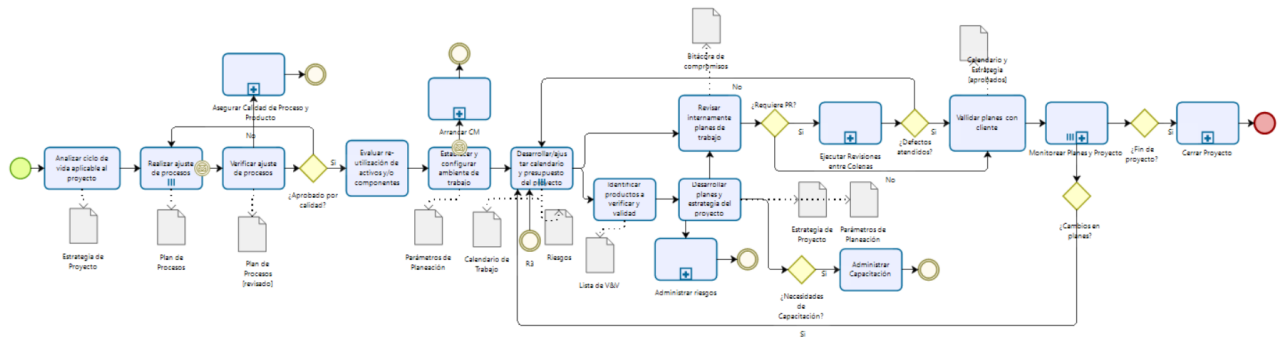


Ilustración 17 Planear Proyecto

### Criterios de entrada:

- El procedimiento inicia cuando se tiene registrado el proyecto y se ha asignado a un Líder de Proyecto.

### Entradas:

- Propuesta de servicio.
- Visión.
- Modelo de estimación utilizado.
- Notificación de asignación.
- Minutas/acuerdos/compromisos de kick-off internos y con cliente.
- LV de planeación.
- Plantilla de WBS Genérico.
- Plantilla de WBS Iterativo.
- Guía de ciclo de vida.
- DNC.

### Criterios de salida/Salidas:

### Criterios de salida:

- El procedimiento termina cuando el proyecto ha sido formalmente cerrado.

### Salidas:

- Estrategia de proyecto.
- Parámetros de Planeación.
- Calendario de Trabajo.
- Plan de procesos.
- Lista de verificación de planeación [aplicada]
- Riesgos registrados en la herramienta de administración de riesgos.
- Forecast.
- Lista de productos a verificar y validar.
- DNC [actualizado]

Roles necesarios para el proceso (Recursos humanos):

- Gerente del CDS:
  - Validar y aprobar las estrategias y planes de trabajo de los proyectos.
  - Configurar el Forecast inicial del proyecto y proporcionárselo al Líder del Proyecto.
  - Apoyar al Líder de Proyecto con las negociaciones requeridas con respecto a la planeación.
- Líder del Proyecto:
  - Desarrollar la estrategia y planes de proyecto.
  - Validar estrategia y plan de trabajo con los involucrados al proyecto.
  - Negociar y tomar acciones correctivas con respecto a la planeación del proyecto.
- Equipo de trabajo:
  - Retroalimentar al Líder de Proyecto sobre la estrategia y planes.
  - Identificar y comunicar al Líder de Proyecto riesgos con respecto al proyecto.
- Auditor:
  - Asesora al Líder de Proyecto en el ajuste de los procesos.
  - Validar y aprobar el plan de procesos.
  - Asegurar la calidad de los productos y el apego al proceso.

### Analizar ciclo de vida aplicable al proyecto

El Líder de Proyecto analiza la información del modelo de estimación, el resultado de la junta de Kick-off con el Cliente y los compromisos acordados y decide sobre la selección del ciclo de vida.

La decisión debe de tomarse considerando:

- Milestone acordados/comentados con el cliente.
- Certeza de los requerimientos.
- Recursos del proyecto.
- Duración del proyecto.

El Líder de Proyecto se apoya de la guía de ciclos de vida y el término de la selección se registra en la Estrategia de Proyecto.

### **Realizar ajustes de procesos**

El Líder de Proyecto desarrolla el plan de Procesos para seleccionar y/o adecuar los procesos organizacionales al proyecto. El objetivo de esta actividad es que el Líder de Proyecto dimensione el proyecto para comenzar a realizar la planeación.

Al finalizar el ajuste, envía el plan de procesos al Auditor para su validación y aprobación.

### **Planear Auditorías.**

Mas adelante se explicará este subproceso.

### **Verificar ajuste de procesos**

El auditor verifica el ajuste de procesos y asesora al Líder de Proyecto.

En esta fase del proceso, la verificación y aprobación del Auditor es sobre cuáles procesos aplican o no aplican al proyecto. El ajuste del cómo aplicaran los procesos, es un ajuste que se irá realizando a lo largo de ciclo de vida del proyecto. La validación se irá realizando de acuerdo con las auditorías de aseguramiento de calidad.

En este momento, el Auditor ejecuta su proceso para planificar y ejecutar las auditorías de calidad al proyecto.

A continuación, se hace la siguiente pregunta: ¿Aprobado por calidad? Si la respuesta nos debemos de regresar hasta la actividad de Realizar ajuste de procesos, y continuar con las actividades que explicamos anteriormente. En cambio, si la respuesta es si continuamos con la siguiente del diagrama de actividades de Planear Proyecto.

### **Evaluar reutilización de activos y/o componentes**

El Líder de Proyecto consulta el repositorio organizacional, para evaluar la re-utilización de activos (p.e. código, tomas de decisiones, riesgos, lecciones aprendidas, planes, indicadores, etc.) que apoyen a la ejecución del nuevo proyecto.

### **Establecer y configurar ambiente de trabajo**

El líder de Proyecto:

- Identifica la infraestructura necesaria para la ejecución de su proyecto. Si hubiera algo adicional a la configuración estándar de infraestructura (ver Estándares de Ambiente de Trabajo), la registra en la hoja “recursos” del documento de parámetros de planeación para su obtención. (p.e. licencias, memoria, dispositivos, etc.)
- Selecciona al responsable de configuración de su proyecto y se lo notifica por correo electrónico para que esté de inicio al procedimiento de Arrancar CM.

### **Desarrollar/Ajustar calendario y presupuesto del proyecto**

El Líder de Proyecto desarrolla calendario de trabajo a partir del WBS inicial del documento visión y tomando como guía la plantilla de WBS genérico o WBS iterativo.

El calendario incluye:

- Descomposición del WBS a nivel de actividades que generen un entregable. El listado de actividad son tanto técnicas, administrativas como de soporte.
- Fecha de inicio y fin del Proyecto.
- Dependencias entre tareas.
- Esfuerzo y duración de tareas planeados.
- Responsables asignados a cada tarea. Se verifica que los recursos asignados no estén sobre asignados en las cargas de trabajo.
- Milestones.

Una vez realizado el calendario de trabajo, el Líder de Proyecto analiza si la estimación concuerda con el calendario del proyecto y en caso de ser necesario comunica las diferencias al Gerente del CDS para que se tomen las acciones correctivas necesarias.

El Líder de Proyecto identifica restricciones o riesgos con respecto al calendario y los registra y administra en la herramienta de administración de riesgos.

Una vez elaborado el calendario de trabajo, el Líder de Proyecto analiza si la estimación concuerda con el calendario del proyecto y en caso de ser necesario comunica las diferencias al Gerente del CDS para que se tomen las acciones correctivas necesarias.

El Líder de proyecto identifica restricciones o riesgos con respecto al calendario y los registra y administra en la herramienta de administración de riesgos.

Una vez elaborado el calendario de trabajo lo envía al Gerente del CDS quien desarrolla el Forecast inicial y registra los costos del proyecto, el esquema de pagos y la proyección de horas a devengar por mes cada uno de los recursos asignados al proyecto.

### Identificar productos a verificar y validar

El Líder de Proyecto comienza a estructurar la estrategia de calidad aplicable al proyecto. Toma la Lista de Productos a V&V y comienza a seleccionar qué productos de trabajo requiere pasar por control de calidad, tomando en cuenta los siguientes criterios para su selección:

- Productos de trabajo que son importantes para otras fases del ciclo de vida del proyecto.
- La contribución del producto de trabajo al cumplimiento de los objetivos del proyecto.
- Productos de trabajo que ponen en riesgo el proyecto.
- Productos de trabajo que demostrarán al Cliente que los requerimientos han sido satisfechos.

Una vez seleccionados los productos, determina por cada producto a verificar:

- Los requerimientos a satisfacer por el producto de trabajo.
- El método a utilizar.
- El ambiente en que se ejecutará la verificación.

Y en cuanto a los productos a validar:

- El alcance de la validación (p.e. comportamiento operacional, mantenimiento, entrenamiento, interfaz de usuario).
- Identificación de principios claves, como características y fases de validación.
- El método a utilizar.
- El ambiente en que se ejecutará la validación.
- Restricciones relacionadas a la validación, las cuales se analizan en conjunto con los involucrados para tomar las acciones preventivas/correctivas necesarias.

Una vez definida la estrategia e identificados los productos de trabajo a verificar y validar, se integra la planeación dentro del calendario del proyecto.

### **Desarrollar Planes y Estrategia del Proyecto**

El Líder de Proyecto genera la estrategia de proyecto apoyándose del documento de Visión y la va validando con los involucrados para hacerlos partícipes y buscar su compromiso con el proyecto y los recursos necesarios para desarrollarlo.

Para el desarrollo de las estrategias de proyecto y de los parámetros de planeación el Líder:

- Establece el plan de comunicación que incluye:
  - Identificar a los principales involucrados en el proyecto y el rol que tomaran dentro del mismo. Si en la reunión de Kick-off con Cliente, no se identificó a los responsables en aceptar fin de fase y/o cambiar requerimientos, el Líder de Proyecto en esta actividad lo valida con el cliente.
  - Definir el tipo de reuniones a tener en el proyecto, así como la periodicidad y asistentes.
  - Establecer los mecanismos de comunicación.
- Establece el plan de recolección y administración de la información que abarca:
  - La información necesaria para el proyecto, en qué fase se requiere y quién es el responsable de proveerla. Ejemplos: información operativa del cliente, manuales, información técnica, de configuración, especificaciones, seguridad, etc., es decir, toda aquella información que sea insumo para ejecutar el proyecto.
  - Identificar el manejo que se le dará a la información (pública o confidencial)
  - Nivel de acceso a la información. Este dato se lo proporciona al CM del proyecto para su registro en el plan de configuración.
- Analiza los DNC's de su equipo de trabajo para evaluar si existen necesidades de capacitación que al cubrirlas apoyen a que su equipo de trabajo ejecute eficientemente sus actividades asignadas. Si se detecta la necesidad, actualiza el DNC, registra la necesidad en la hoja de capacitación de los parámetros de planeación y hace la solicitud de la capacitación al Gerente del CDS, quien ejecuta el PCD Administrar Capacitación.
- Registra los riesgos y restricciones identificados durante la planeación de acuerdo al procedimiento de Administrar Riesgos.



## Administrar Riesgos

Esta actividad funge también como un subproceso, el cual se explicara mas adelante.

## Revisar internamente planes de trabajo

El Líder de Proyecto coordina reuniones con los involucrados para revisar la estrategia del proyecto y el calendario de trabajo.

Entre las cosas que revisan son:

- Fechas y entregables a generar.
- Cargas de trabajo.
- Días festivos y vacaciones, para no interferir en lo comprometido en el proyecto.
- Necesidades de recursos y capacitación.
- Involucrados y estrategia de comunicación.
- Almacenamiento y distribución de información.
- Acuerdos de confidencialidad y seguridad de la información.
- Identificación de riesgos.
- Intervención y periodos la calidad del proyecto (de seguramiento de calidad y de pruebas)

En este caso de requerirse ajustes a la planeación, el Líder de Proyecto realiza las actualizaciones o negociaciones requeridas.

Al realizar esta revisión y llegar a los acuerdos correspondientes, se obtiene el compromiso interno para la ejecución del plan.

El Líder del Proyecto entrega al Cliente el calendario de trabajo para su revisión y le hace saber que ese plan es el que dirigirá las actividades del proyecto y el que se irá monitoreando en las reuniones de seguimiento.

En caso de requerirse ajustes a la planeación, el Líder de Proyecto realiza las actualizaciones o negociaciones requeridas con respecto a la planeación.

Después de esto se hace la pregunta si se requiere revisión entre pares, si es así se pasa a la actividad denominada Ejecutar Revisión entre colegas, y una vez acabada esta actividad se hace la pregunta si los defectos son atendidos o no, si la respuesta es no se regresa a la actividad de Desarrollar/ajustar calendario y presupuesto del proyecto, en cambio si los defectos son atendidos

nos dirigimos a la actividad de Validar Planes con Clientes. Pero en cambio si no se necesita revisión entre pares, pasamos directamente a Validar Planes con Clientes.

### **Ejecutar revisión entre Colegas**

Esta actividad también funge como un subproceso, es por eso por lo que se explicara más adelante cuando sea el momento necesario.

### **Validar calendario detallado con cliente**

El Líder de Proyecto entrega al Cliente el calendario de trabajo para su revisión y le hace saber que ese plan es el que dirigirá las actividades del proyecto y el que se irá monitoreando en las reuniones de seguimiento.

En caso de requerirse ajustes a la planeación, el Líder de Proyecto realiza las actualizaciones o negociaciones requeridas con respecto a la planeación.

### **Monitorear Planes y Proyecto**

Esta actividad también funge como un subproceso, el cual se explicará más adelante.

Después de esta actividad se hace la pregunta ¿Cambios en planes?, si la respuesta es sí, nos dirigimos a la actividad Desarrollar/Ajustar calendario y presupuesto del proyecto. Si la respuesta es no, se pregunta si es el fin del proyecto, si es así continuamos con la actividad cerrar proyecto, que se explica a continuación.

### **Cerrar Proyecto**

Esta actividad también funge como un subproceso, el cual se explicará más adelante.

Terminado el subproceso anterior, se hace la pregunta si es necesario una reestimación, si la respuesta es sí, nos dirigimos al subproceso Estimar Proyecto. si la respuesta es no, se debe dirigir al subproceso de nombre Arrancar CM.

### **Estimar proyecto**

Este proceso se ejecuta para, definir el alcance inicial mediante el desarrollo de un WBS, identificar los atributos de los productos de trabajo y tareas desglosadas en el WBS y establecer estimaciones de esfuerzo y tamaño de los componentes que conforman el WBS.

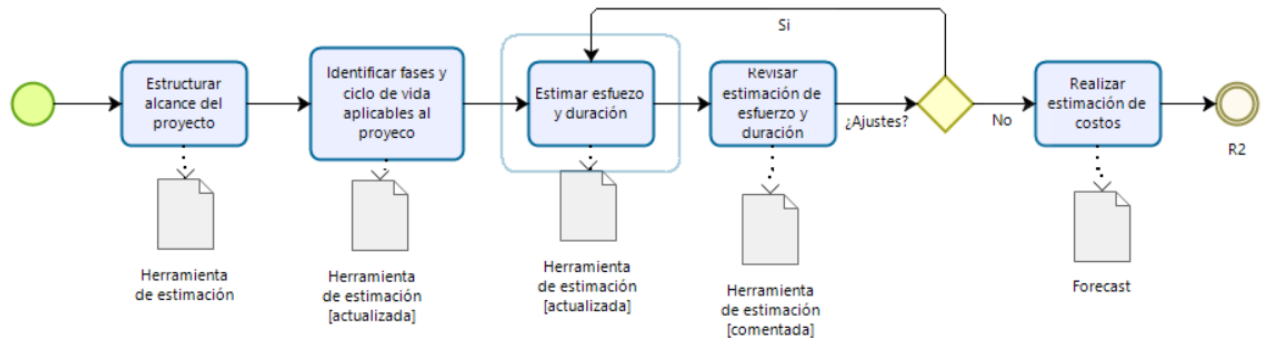


Ilustración 18 Estimar Proyecto

### Criterios de entrada/Entradas:

- El procedimiento inicia cuando se tiene la necesidad de estimar el esfuerzo y costo de requerimientos a desarrollar.

### Entradas:

- Requerimientos del producto, descritos en el documento de Visión y Alcance del Proyecto.

### Criterios de Salida:

- El procedimiento termina cuando se ha realizado la estimación de esfuerzo de los requerimientos.

### Salidas:

- Modelo de estimación puntos caso de uso aplicado al proyecto.

### Roles necesarios para el proceso (Recursos Humanos)

- **Responsable de la estimación:**
  - Desarrollar WBS y estimar el esfuerzo y duración del Proyecto.
  - Desarrollar calendario macro de trabajo.
- **Revisor:**
  - Revisar técnicamente el WBS, esfuerzo y duración.
- **Gerente del CDS:**
  - Desarrollar estimación de costo, con base a la estimación de esfuerzo.
- **Auditor:**
  - Asegurar la calidad de los productos y el apego al proceso.

### Estructurar Alcance del proyecto.

Si el líder del proyecto cree que es necesario estimar el proyecto, se recurre a las siguientes actividades.

El responsable de la estimación toma la información de los requerimientos funcionales y los divide en unidades lógicas (CU/Funciones) que le permitan visualizar la complejidad y esfuerzo requerido para cada uno de ellos. Lo anterior se conoce como WBS

Por cada Función/CU se identifican los componentes que pudieran ser reutilizables y se analiza el nivel de reutilización para decidir si es un porcentaje considerable que disminuirá el esfuerzo de construcción o se tomará como un porcentaje de contingencia sobre la estimación final.

La decisión final se documenta como raciocinio dentro de la hoja de estimación.

**Nota:** El WBS debe quedar desglosado a nivel sub-modulo y funcionalidad, ya que es la base para que el líder de Proyecto calendarice y asigne responsabilidad en su calendario.

### Identificar fases y ciclo de vida aplicables en el proyecto.

Una vez tenido en cuenta los criterios que se necesitan para evaluar el proyecto se deben de identificar las fases y el ciclo de vida que serán aplicables para el proyecto.

De acuerdo con las necesidades expresadas por el cliente, en cuanto a prioridades de uso del sistema, terminación de este, entregables comprometidos, etc., y el análisis de la capacidad interna de atención del proyecto, se seleccionan las fases aplicables al proyecto y el ciclo de vida a utilizar.

Si el ciclo de vida seleccionado fue cascada, al término del uso del método de estimación, dará el esfuerzo necesario para la construcción; con respecto a esta fase se asignan los porcentajes adicionales requeridos para cada una de las fases seleccionadas del proyecto.

Si el ciclo de vida seleccionado fue iterativo, por cada CU se indica la iteración a la que pertenece.

El raciocinio de asignación de porcentajes se documenta en la misma herramienta de estimación. Al término de esta actividad se obtiene una actualización del documento de herramientas de estimación actualizado al documento anterior.

### Estimar esfuerzo y duración.

Una vez concluida la actividad de identificar las fases y ciclo de vida del proyecto, ahora nos dirigimos a la actividad que lleva por nombre, Estimar Esfuerzo y Duración. En esta actividad el responsable de llevarla a cabo es el responsable de la propuesta, este se debe de referir a la hoja de instrucciones de la herramienta PP\_HER\_Modelo de estimación puntos de caso de uso para la utilización de este método. La hoja referida en esta actividad es el documento de estimación que se explicó en las fases

principales del diagrama de actividades de Estimar Proyecto. El documento saliente de esta actividad es la actualización de este documento.

### **Revisar estimación de esfuerzo y duración.**

Una vez terminada la actividad de estimar esfuerzo y duración, continuamos con la actividad de revisar estimación de esfuerzo y duración, donde el ejecutante es el revisor. Aquí el responsable de la estimación, solicita una verificación de la estimación a un revisor que tengan el contexto del proyecto y conozcan sobre la tecnología y/o Cliente al que se le entregará la propuesta (expertos). Los defectos encontrados se revisan en conjunto, se acuerdan y son corregidos.

En caso de tener dudas sobre el alcance de alguna unidad lógica o caso de uso que pudieran modificar el esfuerzo asignado, se solicita la intervención del Cliente para aclarar el alcance. En este caso, todo lo acordado con el Cliente se registra en minutas. Al término de esta actividad se debe entregar la herramienta de estimación comentada.

Una vez terminada la actividad de Revisar estimación de esfuerzo y duración, se hace la siguiente pregunta: **¿Ajustes?**

Si la respuesta es que si se necesitan ajustes al flujo del diagrama de actividades se regresa a la actividad de estimar esfuerzo y duración, y se siguen las actividades correspondientes, hasta llegar a este punto. Cuando la respuesta sea no se realiza la última actividad del proceso de estimación que lleva por nombre Realizar estimaciones de costo.

### **Realizar estimación de costos**

El gerente del CDS recaba la información de los esfuerzos estimados y realiza un cuadro de costos del proyecto en la plantilla de Forecast, considerando tanto costos directos como indirectos para analizar la rentabilidad del proyecto y darle un precio al cliente.

De acuerdo con el número de recursos estimados y su participación en el calendario global, se van identificando aquellos perfiles requeridos para la realización del proyecto.

### **Arrancar CM**

Mediante este proceso se da inicio al proceso de la administración de la configuración (CM).

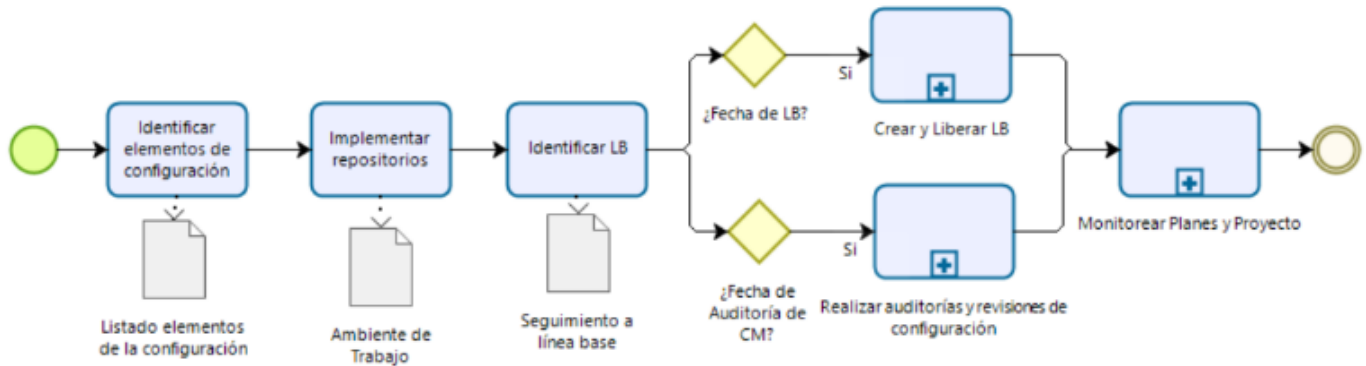


Ilustración 19 Arrancar CM

### Criterios de entrada/Entradas:

#### Criterios de Entrada:

- El procedimiento inicia cuando se realiza la asignación del CM al proyecto.

#### Entradas:

- Asignación del CM al proyecto.
- Calendario de Trabajo.

### Criterios de salida/Salidas:

- Se instalaron las herramientas que soportaran el proceso de CM.
- Se ha creado la estructura inicial del Proyecto.
- Elementos de la configuración identificados.
- Líneas base identificadas.

#### Salidas:

- Ambiente de trabajo.
- Identificación de elementos de configuración y sus niveles de acceso.
- Documento de seguimiento a líneas base.
- **Líder de Proyecto:**
  - Determina los tiempos de para la elaboración de las líneas base.
- **CM Manager:**
  - Monta los repositorios del proyecto.

- Configura el listado de elementos para el proyecto.
- Desarrolla la planificación de las LB.
- **Auditor:**
  - Asegurar la calidad de los productos y el apego al proceso.

### Identificar elementos de configuración

El CM del proyecto se refiere a la guía de CM en donde revisará:

- Las políticas para identificar qué productos de trabajo son puestos bajo configuración.
- La nomenclatura para llamar a los artefactos.
- El esquema de manejo de versiones.
- Nombramiento de las líneas base.

El CM, se refiere al documento Listado elementos de la configuración en la cejilla “documentación” y verifica que tanto los elementos de configuración como la estructura preestablecida son los adecuados para el proyecto, en caso de no aplicar adopta el documento y lo configura a las necesidades específicas del proyecto. De lo contrario se adopta como guía organizacional que define la estructura y elementos a administrar bajo configuración del proyecto.

El CM se asegura a lo largo del proyecto por medio de las auditorías de configuración que esos elementos se vayan subiendo al repositorio.

NOTA: cada autor responsable de generar el artefacto se encarga de subirlo al repositorio de acuerdo con las fechas establecidas en el calendario de trabajo. Siguiendo las políticas de nomenclatura de la guía CM\_GUI\_Guía de CM.

### Implementar repositorios

El CM Manager identifica recursos (servidor para la herramienta, lugar para el Server, red, etc.). En caso de requerirse alguna adquisición se lo comunica al Líder de proyecto para su planificación en los parámetros de planeación.

Una vez con todos los recursos disponibles comienza con la configuración de repositorios.

Para repositorio de documentación:

Se utiliza (Source Safe), tomando como base a la estructura inicial del proyecto de acuerdo con el Listado de elementos de la configuración cejilla de documentación.

Para repositorio de código:

El CM Manager implementa el repositorio de código conforme a las necesidades del lenguaje en el que se desarrollara la aplicación.

Instala las herramientas necesarias y crea la estructura inicial del proyecto de acuerdo con los elementos de la configuración correspondientes de acuerdo con el lenguaje de programación a utilizar. Utilice el documento “*Check List de Implementación de CM código*”.

El CM Manager le informa al Líder de proyecto que ha realizado la implementación de CM para su proyecto. En caso de que el Líder de proyecto sea quien esté a cargo del rol de cm mánager no se debe informar.

Para esta actividad apoyarse de las guías CM\_GUI\_Manual\_de\_visualSVNServer y CM\_GUI\_Manual\_de\_TortoiseSVN.

### **Identificar líneas Base (LB)**

El CM Manager toma el calendario de trabajo para identificar los periodos y/o fechas para la creación de las líneas base y comienza con el desarrollo del documento de Seguimiento Línea Base.

En esta etapa se establece:

- Identificador o nombre de la línea base.
- El contenido (el nombre de los archivos que contendrá la línea base).
- La fecha/periodo de creación.

Luego de esta última actividad se tiene dos brechas una de ellas es, si se tiene una fecha de auditoria de CM se procede a la actividad de Realizar auditorías y revisiones de configuración. La segunda brecha es si se tiene fecha de LB, y la actividad que se debe de realizar es Crear y Liberar LB.

### **Realizar Auditorías y revisiones de configuración**

Como esta actividad también funge como subproceso, se explicará más adelante.

### **Crear y Liberar LB**

Como esta actividad también funge como subproceso, se explicará más adelante.



## Monitorear Planes y Proyectos

Una vez finalizada estas dos brechas se juntan en una para continuar con la actividad final llamada Monitorear Planes y Proyectos. El cual será explicado más adelante.

## Administrar Riesgos

Este proceso se ejecuta para identificar problemas potenciales antes de que ocurran, de tal forma que las acciones para controlar los problemas puedan ser planeadas e involucrar oportunamente a los indicadores para mitigar un impacto adverso al logro de los objetivos.

El procedimiento puede ser utilizado tanto por los proyectos como por las diferentes áreas del CDS.

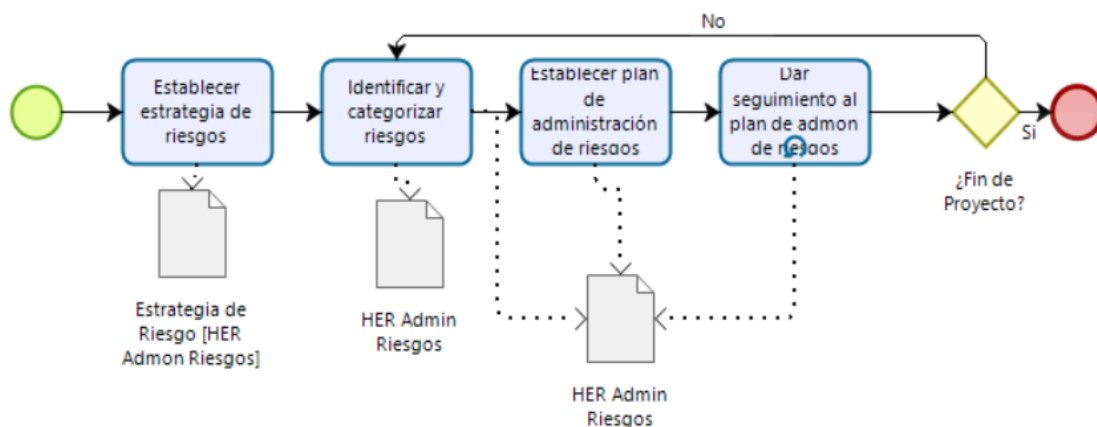


Ilustración 20 Administrar Riesgos

### Criterio de entrada/Entradas:

#### Criterios de entrada:

- El procedimiento tiene dos inicios, uno al arranque del proyecto y cada vez que se identifique una situación que pongan en riesgo el proyecto.

#### Entradas:

- Calendario de trabajo.
- Estrategia de proyecto.

#### Criterios de salida/Salidas:

#### Criterios de Salida:

- El procedimiento termina cuando el proyecto ha sido formalmente cerrado.

#### **Salidas:**

- Herramienta de administración de Riesgos.

#### **Roles necesarios para el proceso (Recursos Humanos):**

- **Gerente del CDS:**
  - Apoyar al Líder de Proyecto en la administración de riesgos del proyecto ( con la identificación, con planes de acción y/o con la ejecución de acciones para mitigar los riesgos).
- **Líder de Proyecto:**
  - Definir y comunicar estrategias de riesgos.
  - Registrar y mantener los riesgos en la herramienta de administración de Riesgos.
  - Coordinar las actividades de administración de riesgos.
- **Involucrados en el proyecto:**
  - Identificar y comunicar riesgos al Líder de Proyecto.
  - Ejecutar acciones para mitigar y controlar los riesgos.
  - Participar activamente en la administración de riesgos en el proyecto.
- **Auditor:**
  - Asegurar la calidad de los productos y el apego al proceso.
- **PMO:**
  - Dar seguimiento al plan de administración de riesgos.

#### **Establecer estrategia de riesgos**

El líder de Proyecto toma la herramienta de administración de riesgo y define los indicadores por tipo de riesgo que servirán como disparadores para escalar los riesgos y solicitar la participación de la Gerencia del CDS. Es importante cuidar que los parámetros de disparo por cada tipo de riesgo se encuentran directamente relacionados a los objetivos del proyecto y la vertiente de mayor peso en el triángulo de administración del proyecto (alcance, tiempo, costo).

#### **Identificar y categorizar los riesgos**

Cualquier involucrado en el proyecto (Equipo de trabajo, Gerencia del CDS, Auditor, Cliente) identifica riesgos que pudieran afectar al proyecto y los comunica al Líder del Proyecto.

Adicional a los riesgos identificados por los involucrados, el Líder de proyecto continuamente mediante las reuniones de seguimiento y elementos como el desglose del WBS en el plan de trabajo, identifica y documenta los riesgos en la herramienta de administración de riesgos.

El registro involucra:

- Categorizar el tipo de riesgo como:
  - Técnico – Calidad – Desempeño
  - Administrativo
  - Organizacional
  - Interno

EL objetivo de categorizar los riesgos es observar la relación entre los riesgos y facilitar la administración y acciones de mitigación.

- Describir el riesgo. Por ejemplo: requerimientos inciertos, afectación en operación del cliente, falta de recursos humanos, tecnología desconocida, etc.
- Detallar la consecuencia de que el riesgo se presente.
- Evaluar el riesgo. La evaluación consiste en asignar parámetros en una escala de muy baja, baja, media, alta y muy alta con respecto a la prioridad, impacto y severidad. La multiplicación de estos valores dará la magnitud de los riesgos.

### Establecer plan de administración de riesgos

Por cada riesgo identificado, el Líder de Proyecto establece una estrategia de atención del riesgo.

Las estrategias son:

- Evitar. Lo cual significa que el riesgo se va a evadir. Es decir, eliminar la causa que dispara el riesgo o cambiar la planeación del proyecto para cumplir con los objetivos.
- Transferir. Lo cual significa transferir el riesgo y las consecuencias a un tercero. En este caso el riesgo continúa, pero la responsabilidad de administración ahora pertenece al tercero.
- Mitigar. Lo cual significa reducir la probabilidad e impacto del riesgo a un nivel aceptable.
- Aceptar. Lo cual significa que no se van a tomar acciones para tratar de evitar o mitigar el riesgo. Esta acción se toma cuando el proyecto está consciente de las consecuencias del riesgo y que estas no impactan significativamente al proyecto. Se acepta el riesgo cuando la

relación entre actuar para mitigar vs aceptar es relativamente más costosa que la misma acción de aceptar.

Si la magnitud de atención del riesgo se muestra en color amarillo o rojo, el Líder del proyecto define un plan de mitigación y un plan de contingencia, así como un responsable y una fecha de atención. El objetivo de tener ambos planes es que, si las acciones tomadas para mitigar el riesgo no están dando resultados, entonces se procede a ejecutar el plan de contingencia.

Si la prioridad de atención del riesgo se muestra en color verde, basta con definir y ejecutar acciones para un plan de mitigación.

El Líder de Proyecto hace partícipes a los involucrados en el proyecto relacionados con los riesgos, en el establecimiento de los planes de administración de los riesgos.

#### **Dar seguimiento al plan de administración de riesgos**

El Líder de Proyecto da seguimiento a los planes de mitigación y contingencia de los riesgos y hace las actualizaciones requeridas.

Cada revisión de riesgos actualiza en la herramienta de administración de riesgos los valores de prioridad, impacto y severidad dependiendo de los resultados de las acciones para administrar los riesgos.

En caso de que in riesgo ocurra, indica la fecha real de ocurrencia y da seguimiento a que las acciones del plan de contingencia se realicen para reducir las consecuencias del riesgo.

El Líder de Proyecto comunica a los involucrados del estado y resultados de la administración de los riesgos, de acuerdo con lo establecido en el procedimiento Monitorear Planes y Proyecto. En caso de requerirse el escalamiento de algún riesgo, se escala a la PMO.

con esta actividad se da por terminado el subproceso de administrar riesgos y se puede proceder con la actividad siguiente de Monitorear planes y proyecto.

#### **Monitorear Planes y Proyecto**

Este proceso se ejecuta de manera continua mientras el proyecto este activo. El objetivo es proveer un entendimiento del progreso que permita tomar acciones oportunas cuando exista desviaciones contra lo planeado.

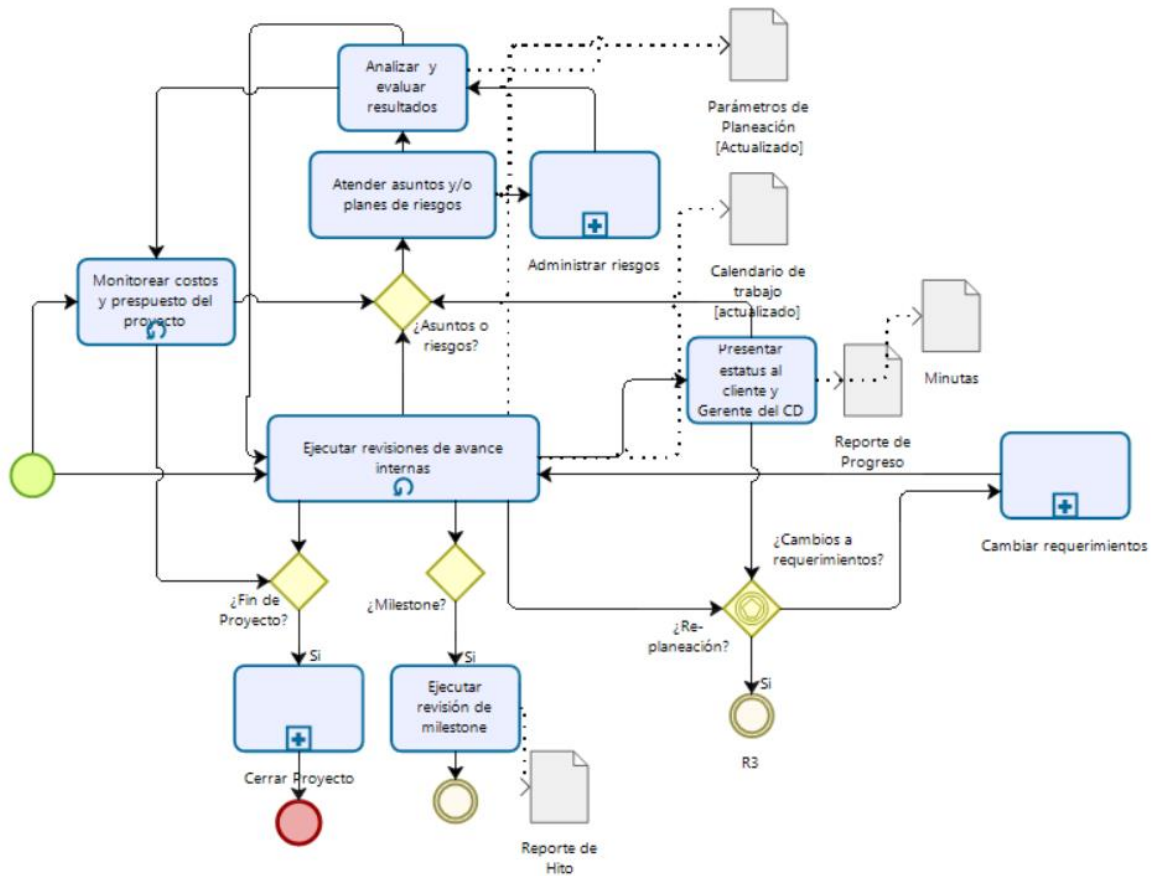


Ilustración 21 Monitorear Planes y Proyecto

**Criterios de entrada/Entradas:**

**Criterios de entrada:**

- El procedimiento inicia cuando se tiene el presupuesto y planes para un proyecto.

**Entradas:**

- Estrategia de proyecto.
- Calendario de trabajo.
- Parámetros de planeación.

**Criterios de salida/Salidas:**

**Criterios de salida:**

- El procedimiento termina cuando el proyecto ha sido formalmente cerrado.

**Salidas:**

- Calendario de trabajo (actualizado).
- Parámetros de planeación (actualizados).
- Reportes de Hito.
- Reportes de progreso.

**Roles necesarios para el proceso (Recursos humanos):****Gerente del CDS:**

- Monitorear el presupuesto del proyecto.
- Reportar desempeño del proyecto a la dirección.
- Apoyar al Líder de Proyecto en estrategias y acciones para controlar el proyecto.

**Líder de Proyecto:**

- Monitorear el desempeño del proyecto.
- Reportar estado del proyecto a la Gerencia del CDS y al Cliente.
- Tomar acciones preventivas y correctivas para controlar el proyecto.

**Equipo de Trabajo:**

- Reportar estado de sus actividades asignada al Líder de Proyecto.
- Identificar inconsistencias entre los requerimientos, productos de trabajo y planes del proyecto para comunicárselas al Líder de proyecto.
- Identificar inconsistencias entre los requerimientos, productos de trabajo y planes del proyecto para comunicárselas al Líder de Proyecto.

**Auditor:**

- Asegurar la calidad de los productos y el apego al proceso

**Ejecutar revisiones de avance internas**

En esta primera actividad el líder de proyecto realiza revisiones de progreso de acuerdo con el tipo y periodicidad establecida en la estrategia de proyecto y calendario de trabajo.

El monitoreo consiste en evaluar los valores reales contra los planeados. En cada revisión el líder actualiza:

- El avance y esfuerzo reales en el calendario de trabajo.
- En los parámetros de planeación actualiza valores de:
  - Los resultados de los datos y análisis de las mediciones.
  - Los recursos utilizados y provistos.
  - El avance en capacitación de los integrantes del equipo
  - El estado de compromisos
  - El estado de la configuración y control de la información
  - El control de riesgos (ver procedimiento de Administración de riesgos)
- El estatus de los asuntos en Mantis.

Adicional a los parámetros anteriores, el líder de proyecto identifica y evalúa:

- Como ha sido la participación de los actores y grupos involucrados en el proyecto.
- Qué tan bien está integrado y enfocado en el objetivo en el equipo de trabajo.
- Las dependencias y asuntos que deban ser controlados.
- Re-balanceo de cargas de trabajo.
- La existencia de lecciones aprendidas o mejores prácticas que puedan contribuir al repositorio organizacional.
- La identificación de sugerencias de mejora sobre el ajuste y la metodología.
- Que se estén generando los productos de trabajo para cubrir la totalidad de los requerimientos y que estas actividades estén alineadas con planeación.
- La existencia de solicitudes de cambio. Si es el caso, el Líder se refiere al proceso Administración de Requerimientos.
- Los resultados de las verificaciones (revisiones entre colegas y/o pruebas)

En caso de encontrarse asuntos, riesgos o desviaciones con respecto a los parámetros de planeación o en el proyecto en general, el Líder de proyecto los analiza y registra en mantis para tomar las acciones correctivas necesarias y mitigar los problemas. Para la administración de riesgos referirse al procedimiento Administrar Riesgos.

### **Presentar estatus al cliente y Gerente del CDS**

El líder de Proyecto desarrolla el reporte de progreso y lo presenta/envía al Cliente y al Gerente del CDS. Los valores del reporte de progreso se toman de la revisión que tuvo el Líder de Proyecto con su Equipo de Trabajo.

El Líder de Proyecto registra los acuerdos, compromisos, riesgos y asuntos derivados de la revisión, los cuales administra a lo largo del proyecto.

En caso de ser necesaria la intervención del Gerente del CDS en la toma de acciones para resolver asuntos que se presenten con el Cliente o ejecutar planes para la atención de riesgos, el Líder de Proyecto se asegura que el Gerente del CDS lo tenga presente.

### **Monitorear costos y presupuestos del proyecto**

De manera mensual, el Gerente del CDS concentra la información que los Líderes de Proyecto (reporte de progreso, parámetros de planeación y calendarios de trabajo actualizados) le hacen llegar y solicitan la comprobación de viáticos en haberse presentado. Con esta información actualiza los devengados y facturación de los proyectos en la herramienta del foescat (que en sesiones anteriores ya habíamos visto su funcionamiento).

En caso de ser necesario el Gerente del CDS valida la información con los involucrados y se toman las acciones correctivas necesarias.

### **Atender asuntos y/o planes de riesgos.**

Cuando se presenten asuntos en los proyectos, ya sea por ámbito interno o hacia con el Cliente, el Líder de Proyecto analiza la situación con los involucrados directos y se deciden las acciones correctivas necesarias.

Tanto el asunto, como los planes de acción se registran en mantis indicando qué se hará, quién lo hará y para cuando debe estar ejecutando el plan de acción.

Dependiendo de la naturaleza del asunto, el Líder de Proyecto da seguimiento con los involucrados y realiza las negociaciones necesarias.

En caso de referirse a un plan de riesgos, referirse al procedimiento de Administrar Riesgos. Nota: Esta actividad puede dispararse en cualquier momento del proyecto.



## **Administrar Riesgos**

Esta actividad ya se explicó anteriormente en este documento.

## **Analizar y evaluar resultados.**

Al cierre de las acciones correctivas, el Líder de Proyecto analiza la situación para determinar la efectividad de la acción y aprender de la situación.

La elección aprendida se comunica en la reunión de revisión de avance con el equipo y se registra en la siguiente revisión post-mortem.

## **Ejecutar revisiones de hito**

Al término de cada hito, el Líder de Proyecto desarrolla el reporte de hito registrando:

- Desempeño del proyecto (con respecto a la fase).
- Compromisos y riesgo (pendientes de la fase).
- Dependencias.
- Riesgos.
- Issues significativos de la fase y su impacto en el proyecto.
- Análisis de las decisiones tomadas durante la fase.
- Lecciones aprendidas, mejores prácticas y/o sugerencias de mejora.

El reporte se entrega y se revisa con el Cliente.

## **Cambiar requerimientos**

Esta actividad será explicada más adelante en este mismo documento, ya que también es un subproceso de las siguientes etapas de la metodología CMMI.

## **Cerrar Proyecto.**

Esta actividad será explicada a continuación ya que también funge como subproceso, y con esto damos por terminado los subprocesos y actividades de la etapa 1 de administrar proyecto.

## **Cerrar Proyecto**

Este proceso se ejecuta para formalizar el cierre de un proyecto. A continuación, mostramos el proceso de actividades de dicha actividad.

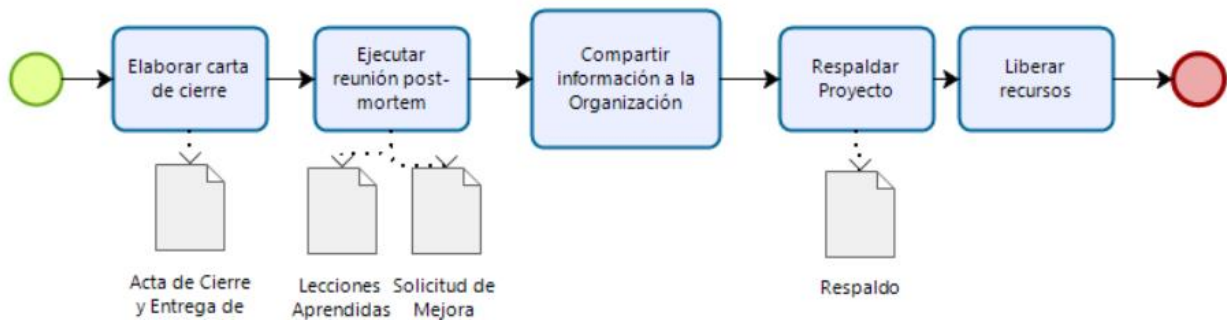


Ilustración 22 Cerrar proyecto

**Criterio de entrada/Entradas:**

**Criterios de entrada:**

- El procedimiento inicia cuando se han cumplido los compromisos de requerimientos con el Cliente.

**Entradas:**

- Calendario de trabajo.

**Criterio de salida/Salidas:**

**Criterios de salida:**

- EL procedimiento termina cuando se tiene la aprobación formal del cliente y se ha entregado el proyecto internamente.

**Salidas:**

- Actas de Cierre y Entrega de Software.
- Solicitud de Mejora.
- Reporte de Hito [Final].
- Respaldo del proyecto.
- Información del proyecto que se comparte en el repositorio organizacional.
- CV y matriz de habilidades, perfiles y experiencia (actualizada).

**Roles necesarios para el proceso (Recursos Humanos):**

**Gerente del CDS:**

- Elaborar y entregar al cliente el acta de entrega para su firma y aprobación.
- Actualizar la matriz de habilidades, perfiles y experiencia de los miembros liberados del proyecto.

**Líder de Proyecto:**

- Notificar al Gerente del CDS la liberación de los recursos tecnológicos y humanos asignados al proyecto.
- Entregar al Responsable de Calidad información compartible del proyecto.

**Equipo de Trabajo:**

- Identificar y registrar sugerencias de mejora.
- Actualizar CV al término de cada proyecto y entregarlo al Gerente del CDS para actualización de la matriz de habilidades, perfiles y experiencia.

**Administrador de la Configuración:**

- Realizar respaldo total del proyecto y entregar copias a Responsable de Calidad.

**Auditor:**

- Asegurar la calidad de los productos y el apego al proceso.

**Elaborar carta de cierre**

El gerente del CDS realiza el acta de cierre y entrega de software, y se le proporciona al Cliente para su firma y aprobación.

En caso de presentarse asuntos, el Líder de Proyecto toma las acciones necesarias para resolver la situación. A continuación, se muestra la imagen del acta cierre y entrega de software.

**Acta de cierre y entrega del software denominado**

**[Nombre del Software]**

TERMINACION DE LA PRESTACION DE SERVICIOS DE ACUERDO CON EL CONTRATO CELEBRADO EL [NUMERO DIA] [NUMERO EN LETRA] DE [AÑO] [AÑO EN LETRA] DAMOS POR TERMINADA LA RELACION DE TRABAJO CONFORME AL CONTRATO ANEXO.

CLIENTE: [NOMBRE DEL CLIENTE]

PRESTADOR DE SERVICIOS: ITSUR, CENTRO DE DESARROLLO DE SOFTWARE

De acuerdo a cada una de las cláusulas incluidas en el contrato de referencia se está de acuerdo de haber cumplido el PRESTADOR DE SERVICIOS con todas y cada una de ellas, entregando satisfactoriamente los siguientes resultados siendo el día de hoy [DIA NUMERO] de [NUMERO EN LETRA] de [AÑO NUMERO].

1. [Entregable 1]
2. [Entregable 2]

Firman al calce de común acuerdo las partes:

<hr/> [Nombre Persona] [Descripción del rol de la persona]	<hr/> [Nombre Persona] [Descripción del rol de la persona]
---	---

<hr/> [Nombre Persona] [Descripción del rol de la persona]	<hr/> [Nombre Persona] [Descripción del rol de la persona]
---	---



Av. Educación Superior No. 2000  
Col. Benito Juárez, Uriangato, Gto.  
C.P. 38800 Apaxtla Postal No. 61

Tels. 01 (445) 45 774 68 al 71 Ext. 110 Fax Ext. 127  
desarrollo@itsur.edu.mx  
www.itsur.edu.mx

*Ilustración 23 Acta de cierre y entrega de Software*

**Ejecutar reunión *post-mortem***

El Líder de Proyecto, coordina una sesión *post-mortem* con su equipo de trabajo. Esta revisión consiste en analizar el desempeño del proyecto de la fase terminada.

Antes de la reunión, cada participante se prepara con respuestas a:

- ¿Qué fue lo que hizo bien?
- ¿Qué debemos mejorar?

- ¿Qué debemos mantener?

En esta reunión se realiza particular énfasis en la retroalimentación del Cliente, en evaluar el desempeño y efectividad de los procesos establecidos en el plan de procesos y en concentrar los valores reales con respecto a la estimación.

El resultado de la reunión se registra en el reporte de lecciones aprendidas.

Los resultados de la revisión que puedan ser candidatos a sugerencias de mejora se registran como una solicitud de mejora para ser atendidos en un ciclo de mejora por el grupo de procesos.

### **Compartir información a la organización**

El líder de Proyecto entrega al responsable del repositorio de activos organizacionales (ver registro de repositorios) la información que como proyecto se generó y que puede ser reutilizada por los demás proyectos. Por ejemplo:

- Mediciones recolectadas.
- Control de riesgos.
- Tomas de decisiones.
- Estimaciones.
- Planes.
- Estrategias y resoluciones de asuntos.
- Componentes reutilizables.
- Reportes de Milestone con las lecciones aprendidas durante la ejecución del proyecto.
- Entregables y/o artefactos que pueden ser compartidos con los demás consultores o proyectos.

Cada miembro del equipo actualiza su CV con la nueva experiencia adquirida y se la envía al responsable de Capacitación para que actualice la matriz de habilidades perfiles y experiencia.

### **Respaldo Proyecto**

El responsable de la Configuración realiza el respaldo total del proyecto y lo almacena de acuerdo a las políticas de respaldos.

Liberar recursos.

El Gerente del CDS libera la asignación de recursos y realiza el cierre en el forescat.

El administrador de Configuración hace la liberación de recursos (hardware, software, espacios en discos, VPN's, Servidores, etc).

## Liberar Recursos

El Gerente del CDS libera la asignación de recursos y realiza el cierre en el forescat.

El Administrador de Configuración hace la liberación de recursos (hardware, software, espacios en disco, VPN's, servidores, etc.).

Con este último subproceso llamado Cierre de proyecto damos por terminado la etapa 1 y podemos comenzar con la etapa 2 de nombre Administrar Requerimientos.

## ETAPA 2. Administrar Requerimientos

En esta etapa se desarrollarán y desplegarán todos los requerimientos que el cliente tiene en mente para el proyecto. Después de levantar dichos requerimientos se procederá a diseñar la solución que plantea el cliente para finalmente construir el proyecto, y disponerse una vez terminado a probarlo y estabilizarlo para que sea liberado a producción.

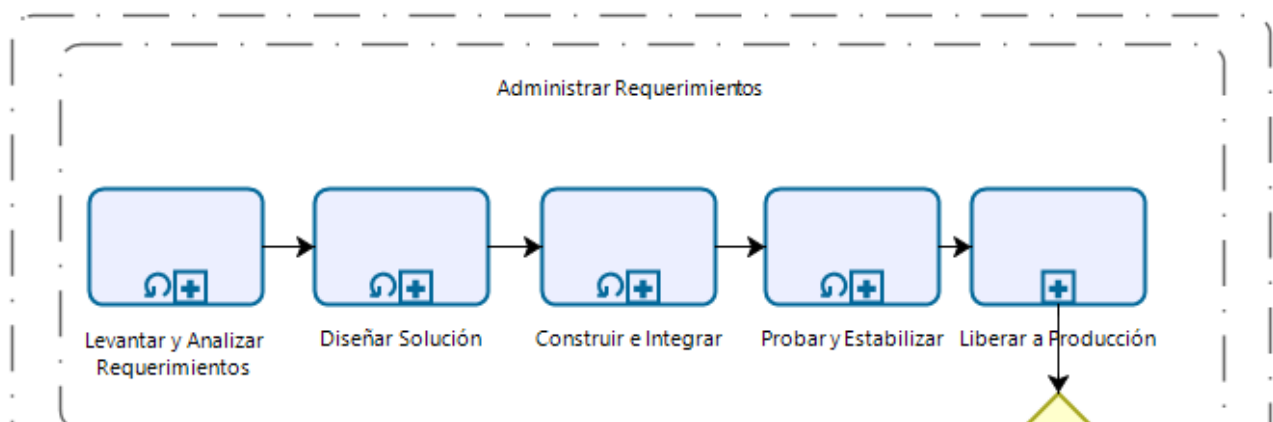


Ilustración 24 Etapa 2. Administrar Recursos

### Levantar y Analizar Requerimientos

Este proceso es el encargado de recolectar y acordar los requerimientos del cliente, con quiénes este último defina. Durante la recolección se evalúa la factibilidad técnica y se hace uso de técnicas o herramientas para lograr un entendimiento entre lo que el cliente espera y lo que se va a construir.

Derivado de la ejecución de este proceso se pueden disparar los procedimientos de "Cambiar requerimientos", "Diseñar Solución" y/o "Ejecutar revisiones entre colegas".

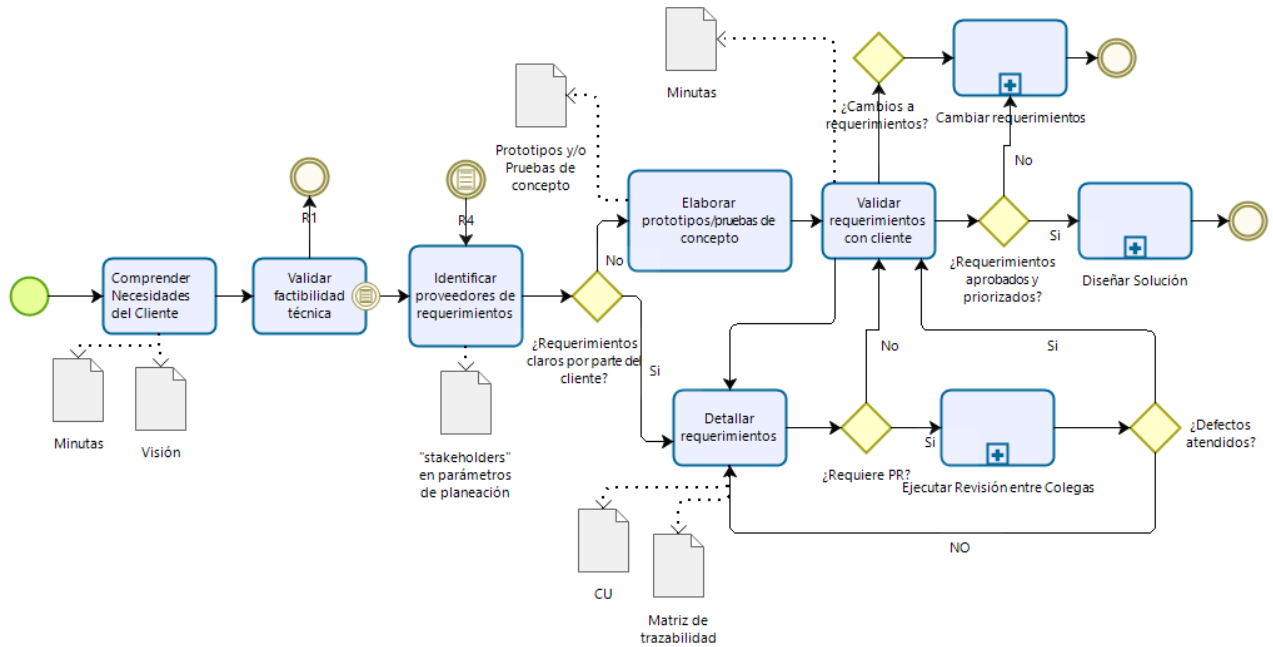


Ilustración 25 Levantar y analizar requerimientos

**Criterio de entrada/Entradas:**

**Criterios de entrada:**

- La identificación de necesidades y requerimientos de un Cliente.

**Entradas:**

- Requerimientos, necesidades y restricciones del Cliente.

**Criterios de salida/Salidas:**

**Criterios de salida:**

- Una vez aceptados y priorizados los requerimientos.

**Salidas:**

- Visión.
- Proveedores de requerimientos identificados en la hoja de "Stakeholders" del documento de parámetros de planeación.
- Prototipos / Pruebas de concepto.

- Especificación de requerimientos y casos de uso.
- Reporte de revisión entre colegas.
- Minutas de validación de requerimientos.
- Matriz de trazabilidad.

**Líder de proyecto:**

- Realizar las negociaciones y acuerdos con cliente con respecto a la priorización de los requerimientos.
- Exponer al Cliente situaciones de riesgo y restricciones hacia con los requerimientos.

**Analista:**

- Recolectar las necesidades y requerimientos del cliente.
- Elaborar prototipos y/o pruebas de concepto que apoyen el entendimiento de los requerimientos con el cliente (si fuera necesario)
- Evaluar factibilidad técnica de los requerimientos e identificar restricciones para el cumplimiento de los requerimientos.
- Transformar los requerimientos del cliente en caso de uso.
- Asegurar la compatibilidad entre requerimientos, la consistencia, que estén completos y que sea lo que el cliente espera de su funcionamiento.
- Actualizar matriz de trazabilidad.

**Cliente:**

- Determinar a los responsables en proveer/aceptar y cambiar los requerimientos.
- Priorizar los requerimientos.
- Validar la descripción funcional y no funcional de los requerimientos.
- Definir criterios de aceptación.

**Auditor:**

- Asegurar la calidad de los productos el apego al proceso.

**Comprender las necesidades del cliente (actividad a realizar en preventa/pre-análisis)**

El analista se reúne con el cliente (cuantas veces sea necesario) para identificar cuáles son las necesidades que derivaron la existencia de un sistema de información.



Durante esta interacción con el Cliente, el Analista:

- Identifica necesidades y requerimientos iniciales (técnicos y no técnicos) planteados por el Cliente.
- Identifica criterios de evaluación y aceptación de los requerimientos. Los criterios a este nivel son procesos globales de acuerdos y negociaciones con el cliente para la aceptación de los requerimientos (p.e. firmas y/o correos de aceptación, personas responsables, periodos de entrega, tipo de liberaciones (sw completo o por iteraciones), porcentajes de requerimientos, densidad de defectos, etc.), no así criterios específicos por cada requerimiento; estos últimos se detallarán con la elaboración de la especificación de requerimientos (formato de CU) en la fase de análisis de sistema.
- Entiende el uso y utilidad para el Cliente del resultado final del producto.
- Identifica el valor agregado que se puede transmitir al Cliente para satisfacer sus necesidades.
- Identifica los beneficios que se pueden lograr clasificando los requerimientos de acuerdo al tipo de necesidades que cubren; por ejemplo: solucionar problemas operativos, agregar nuevos procesos, realizar adecuaciones a procesos, incluir nuevas funcionalidades, dar mantenimiento de sistemas, entre otras.
- Determinar el nivel de compromiso del Cliente con el proyecto.
- Asegura que el alcance del proyecto está dentro de las capacidades del CDS.

El analista, cuando sea necesario, solicita que lo acompañe a las reuniones programadas, una persona con experiencia en el sector o tecnología de que se trate.

Por cada reunión se realiza una minuta y al término del ciclo se concentra la información en el documento de visión.

### Factibilidad técnica (actividad a realizar en preventa/preanálisis)

Una vez comprendida la actividad anterior, el Analista evalúa la factibilidad técnica para el desarrollo de los requerimientos, considerando:

- Restricciones y/o inconsistencias observadas
  - **de costos**, hace un paneo entre los gastos que van a ocasionar todas las actividades técnicas a realizar, tanto de desarrollo como de implementación de software con el presupuesto que tiene en mente el cliente.

- **de recursos materiales**, se revisan los recursos materiales con los que cuenta el CDS para la elaboración del proceso, de la misma forma se revisan los recursos materiales con los que cuenta el cliente.
- **Los recursos del tiempo** que dispone el CDS para lograr los acuerdos y compromisos que se hicieron en los documentos de minutas y de visión, y el tiempo con el que cuenta el cliente para la realización de reuniones y revisión de documentación que pudiera surgir a lo largo del proyecto.
- **Los recursos humanos**, si se cuenta con el personal adecuado y capacitado para poder realizar de buena forma el proyecto.
- **Limitaciones tecnológicas**, ver las limitaciones tecnológicas que tiene el CDS y que pudieran afectar la puesta en marcha del proyecto a desarrollar, una vez visto estas limitaciones, enseguida deberá de mirarse las limitaciones tecnológicas que tiene el cliente para una vez terminado el proyecto, descubrir si la instalación de este se pueda llevar a cabo con la tecnológica con la que cuenta, o si se tendrá que actualizar las mismas.
- **Riesgos**, existen ciertos tipos de riesgos a los que se deben de enfrentar al momento de realizar cualquier proyecto de base tecnológica en el que destacan los siguientes:
  - **Elaboración de la planificación**, una mala planificación puede llegar a retardar los lapsos y metas establecidos en el proyecto y en casos más extremos la cancelación del mismo proyecto.
  - **Requisitos**. Una mala toma de requisitos puede llegar a convertirse en el corto y mediano plazo en un dolor de cabeza, puesto que en los requisitos va incluido la idea principal del proyecto y todas las funciones y características que el cliente desea obtener de él, un mal entendimiento en la toma de requisitos entre el cliente y el CDS puede ocasionar un problema colosal en la manera en que se desarrollara y funcionara el proyecto en cuestión.
  - **Usuarios finales**, los usuarios finales insisten en nuevos requisitos una vez ya firmados y establecidos los requisitos planeados por ambas partes, y que fueron puestos ya en los documentos de minutas y visión.
  - **Clientes**. Los clientes insisten en nuevos requisitos, que no estaban planeados en los documentos de las minutas y visiones. Los ciclos de revisión/decisión del cliente para los planes, prototipos y especificaciones son mas lentos de lo que se esperaban. El cliente no participa en los ciclos de revisión de los planes, prototipos

y especificaciones, o es incapaz de hacerlo, resultando unos requisitos inestables y la necesidad de realizar cambios que consumen tiempo.

- **Producto.** Los módulos propensos a tener errores necesitan más trabajo de comprobación, diseño e implementación. El desarrollo de una interfaz de usuario inadecuada requiere volver a diseñarla y a implementarla. El requisito de trabajar con varios sistemas operativos necesita más tiempo del esperado. Depender de una tecnología que aún está en fase de desarrollo alarga la planificación.
- **Fuerzas Mayores.** El producto depende de las normativas del gobierno, que pueden cambiar de forma inesperada. El producto depende de estándares técnicos provisionales, que pueden cambiar de forma inesperada.
- **Personal.** La contratación tarda más de lo esperado. Las tareas preliminares (por ejemplo, formación, finalización de otros proyectos, adquisición de licencias) no se han completado a tiempo. La falta de relaciones entre la dirección y el equipo de desarrollo ralentiza la toma de decisiones. El personal necesita un tiempo extra para acostumbrarse a trabajar con hardware nuevo. No hay suficiente personal disponible para el proyecto.
- **Diseño e implementación.** Un diseño demasiado sencillo no cubre las cuestiones principales, con lo que hay que volver a diseñar e implementar. Un diseño demasiado complejo exige tener en cuenta complicaciones innecesarias e improductivas en la implementación. Se ha sobreestimado el ahorro en la planificación derivado del uso de herramientas para mejorar la productividad. Los componentes desarrollados por separado no se pueden integrar en forma sencilla, teniendo que volver a diseñar y repetir algunos trabajos.
- **Que puedan ser cubiertos los criterios de evaluación y aceptación del proyecto.** Se debe de realizar un estudio y análisis de todos los criterios y compromisos vistos en las reuniones con el cliente y planteados en los documentos de minutas y visión, así mismo se deben analizar todos los aspectos mencionados anteriormente para así lograr una evaluación concisa en la que se toman en cuenta todos los criterios, riesgos, restricciones y limitaciones tecnológicas de ambas partes para así lograr la aceptación o no del proyecto.

Una vez realizada esta evaluación se toma la primera decisión importante del proyecto, la aceptación o negación del proyecto. Si el resultado de la evaluación de los criterios es muy bajo y la evaluación muestra que no se logran cumplir con los criterios necesarios, la respuesta será un no,

lo cual dará por concluido la metodología a seguir y la elaboración del proyecto terminará en ese preciso momento.

En cambio, si los resultados de la evaluación se cumplen de buena manera y el cliente está decidido y convencido de la realización del proyecto, se continuará con la siguiente actividad de la metodología del CDS la cual es Identificar proveedores de requerimientos.

### **Identificar proveedores de requerimientos**

El líder de Proyecto solicita al Cliente defina quiénes son los involucrados responsables de proveer, aceptar y solicitar cambios a los requerimientos. El LP se puede apoyar de la guía REQM\_GUI\_Identificación de proveedores relevantes.

Los nombres los registra en la hoja de “Stakeholders” del documento de parámetros de planeación y analiza que se estén cubriendo todas las áreas involucradas en el desarrollo del proyecto.

El Líder del proyecto les expresa desde un inicio la importancia de su participación; entre las cuales se encuentra:

- Participar activamente en las fechas acordadas en el levantamiento y validación de requerimientos.
- Proveer retroalimentación sobre el entendimiento de los prototipos, pruebas de concepto y/o SRS que los analistas desarrollen durante la fase de análisis.
- Aceptar y solicitar cambios sobre los requerimientos.
- Definir en conjunto con el analista los criterios de aceptación de los requerimientos.

### **Elaborar prototipos/pruebas de concepto (actividad a realizar en fase de análisis)**

Si los proveedores de requerimientos no tienen claro la operación de los requerimientos, el Analista desarrolla prototipos y/o pruebas de concepto para apoyar al cliente a aterrizar los requerimientos y comprender por parte del CDS la construcción de estos.

El principal objetivo es lograr un entendimiento común de las necesidades del requerimiento y la construcción de este.

### **Detallar Requerimientos (actividad a realizar en fase de análisis)**

El Líder de proyecto programa reuniones con los responsables de proveer los requerimientos y los Analistas del Proyecto para llevar los requerimientos funcionales descritos en el documento de visión a un nivel de requerimientos de producto que serán visualizados como casos de uso.

Para esta actividad se hace uso de técnicas como entrevistas, lluvia de ideas, prototipos y/o casos de uso. Durante este periodo de análisis, el analista:

- Detalla con el cliente, necesidades, expectativas, restricciones de validación y/o verificación, e interfaces externas con las que tiene relación el producto.
- Transforma la información del documento de visión y la expresada por el o los proveedores de requerimientos en casos de uso. En caso de ser necesario obtiene información faltante y elimina conflictos que pudieran existir.
- Identifica y registra los requerimientos no funcionales y requerimientos operativos. Ver guías REQM\_GUI\_Identificación de requerimientos no funcionales para apoyo de esta actividad.
- Identifica requerimientos de interfaz externos al sistema e internas entre los componentes.
- Se identifican requerimientos derivados. Los requerimientos derivados son aquellos que no son explícitos por funcionalidad descrita por el cliente, pero que a nivel de componentes si son requeridos para cumplir con la funcionalidad requerida.
- Va analizando las dependencias y relaciones entre los requerimientos e identifica requerimientos clave que tengan fuerte influencia en costo, calendario, funcionalidad, ambiente o desempeño y a estos se les da seguimiento como riesgos durante el ciclo de vida para el proyecto.
- Identifica criterios de aceptación detallados por requerimientos, estos serán la base para el diseño de la arquitectura del sistema, y una entrada para la verificación y validación del producto. (Ejemplos de criterios de aceptación: tiempo de respuesta, disponibilidad, etc.).

El registro de este análisis y requerimientos se realiza tanto en el documento de casos de uso como en la matriz de trazabilidad.

**NOTA:** En la matriz de trazabilidad se registran tanto los requerimientos funcionales como los no funcionales, la diferencia radica en que los requerimientos funcionales se especifican en dónde se puede ver implantado ese requerimiento en cada fase del ciclo de vida del proyecto (análisis, diseño, código y dónde se avala su prueba) y para los requerimientos no funcionales se registra únicamente el requerimiento funcional derivado, en el que ya se podrá observar la implantación en las fases aplicables. La importancia de tener ambos, es para dar seguimiento a su cumplimiento.

En esta actividad obtendremos dos nuevos documentos que nos servirán a lo largo del proyecto los cuales son Casos de uso (CU) y el documento de matriz de trazabilidad.

NOTAS: Todos los compromisos realizados con el Cliente durante las revisiones y validaciones de los requerimientos se establecen en minutas. Dependerá de cada tipo de situación con el Cliente el método para establecer el compromiso; este podrá ser minutas con acuerdos de requerimientos aceptados y/o documentos de casos de uso firmados. El ajuste se registra en el plan de procesos.

### **Ejecutar Revisión entre pares**

Esta actividad se explicará más adelante en este documento, ya que funge también como un subproceso.

### **Diseñar Solución**

Esta actividad se explicará más adelante en este documento, ya que funge también como un subproceso.

### **Cambiar requerimientos**

Esta actividad se explicará más adelante en este documento, ya que funge también como un subproceso.

### **Diseñar Solución**

Este proceso se ejecuta para:

- Evaluar y seleccionar soluciones que satisfagan el conjunto de requerimientos.
- Realizar un diseño de alto y bajo nivel del producto a desarrollar.
- Diseñar los escenarios de prueba que satisfagan los criterios de aceptación de los requerimientos.
- Desarrollar documentación soporte del producto y componentes del producto.

Dependiendo del ciclo de vida seleccionado, este proceso se ejecuta de una sola vez después de haber terminado el análisis de los requerimientos, o puede irse ejecutando por iteración.

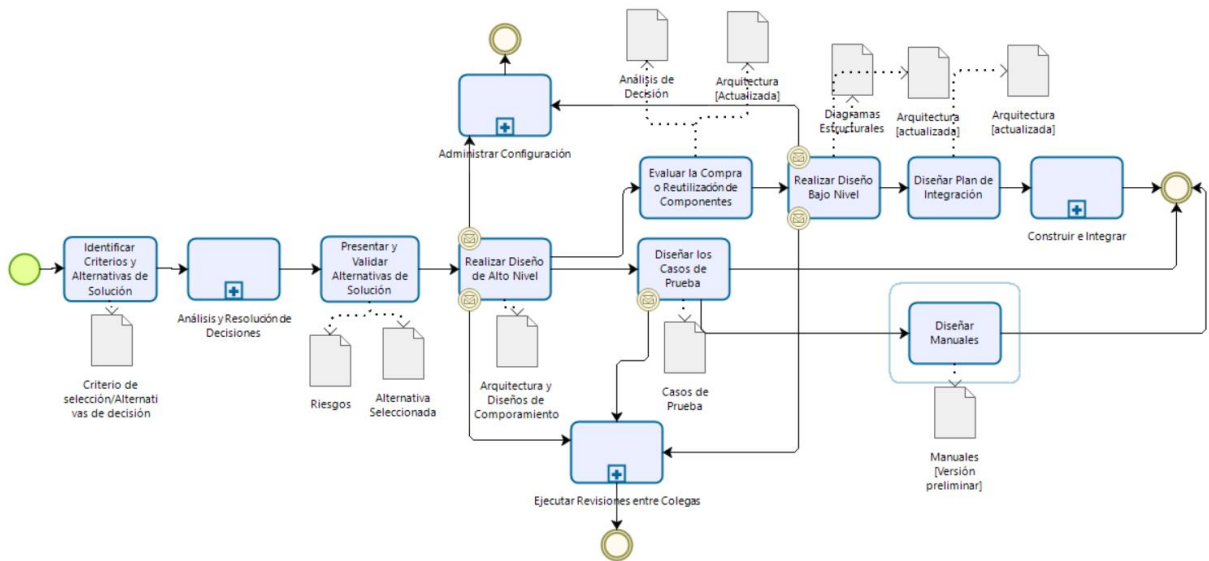


Ilustración 26 Diseñar Solución

**Criterios de entrada/Entradas:**

**Criterios de entrada:**

- El proceso inicia una vez que se tiene la línea base (completa o iteración) que integre los documentos de visión y los casos de uso.

**Entradas:**

- Prototipos / Pruebas de Concepto.
- Casos de Uso.
- Documento de Visión.
- Matriz de trazabilidad.
- Lista de V&V.

**Criterio de salida/Salida:**

**Criterios de salida:**

- El proceso termina cuando se tiene la línea base de la documentación soporte (diseño alto/bajo nivel, diseño de casos de prueba) para arrancar la construcción, integración y pruebas del producto.

**Salidas:**

- Evaluación y selección de alternativas de solución.
- Arquitectura del diseño (diseño alto nivel).
- Casos de prueba.
- Revisiones entre colegas de los productos de trabajo.
- Líneas base.
- Matriz de trazabilidad [actualizada]

Roles necesarios para el proceso (Recursos humanos):

**Analista:**

- Apoyar al diseñador en la evaluación de alternativas de solución.

**Diseñador:**

- Evaluar las alternativas de solución.
- Diseñar la arquitectura de la solución.
- Evaluar el uso o reutilización de componentes.
- Realizar el diseño de bajo nivel.
- Actualizar matriz de trazabilidad.

**Tester:**

- Diseñar los casos de prueba.

**Documentador:**

- Diseñar los manuales de solución.

**Líder de Proyecto:**

- Dar seguimiento a la fase conforme los planes de trabajo.
- Manejar asuntos o riesgos relacionados a la fase.

**CM:**

- Generar líneas base.

**Auditor:**

- Asegurar la calidad de los productos y el apego al proceso.



### **Líder técnico:**

- Coordinar y realizar la integración al ambiente beta.
- Realizar la integración de la base de datos.

### **Identificar criterios y alternativas de solución**

Para el desarrollo de la solución ya sea en un nivel global o nivel de componente, puede presentarse diferentes alternativas para abordar la solución.

En estos casos, el líder técnico en conjunto con el Diseñador y el Analista identifican y analizan esas posibles alternativas y definen un criterio de selección contra el cual se evaluarán cada una de las alternativas.

En la identificación de las alternativas se considera:

- Adquisición de componentes COTS.
- Reutilización de componentes.
- Aplicación de tecnologías de uso actual o nuevas tecnologías para ganar ventaja competitiva.
- Modularidad de los componentes.

Y siempre tomando en cuenta temas como costos, tiempos, desempeño y riesgos asociados.

El criterio de selección puede estar en función de:

- Costo (tiempo, recursos, dinero).
- Beneficios (desempeño, capacidad, efectividad).
- Riesgos (técnicos, costo, calendario).
- Calidad (tiempo de respuesta, seguridad).
- Requerimientos claves, problemas de diseño, restricciones.
- Requerimientos de componentes de producto.

La selección de las alternativas de solución, se realiza conforme el proceso de Análisis y Resolución de Decisiones.

La selección será la base para el diseño del producto. A continuación, se explicará el subproceso para analizar y tomar decisiones.

## Análisis y Resolución de Decisiones

Esta actividad se explicará más adelante en este documento, ya que también funge como un subproceso.

## Presentar y evaluar alternativas de solución

Una vez seleccionada la o las alternativas de solución se presentan al involucrado final de la aprobación de la evaluación (Cliente/Líder de Proyecto).

Durante este ejercicio se toma en cuenta los riesgos que esta decisión puede conllevar y las acciones que se puede tomar al respecto, de acuerdo con lo especificado en el procedimiento de Administrar Riesgos.

El líder técnico registra la selección tomada se registra en el documento de Arquitectura.

## Realizar diseño de alto nivel

El diseñador elabora un diseño de alto nivel en el documento de Arquitectura.

Para el desarrollo de la arquitectura se toma en cuenta:

- Los criterios de aceptación registrados en los casos de uso y el documento de visión.
- Reglas de negocio.
- Cuestiones de calidad y desempeño que sean críticos para el éxito de producto. (por ejemplo, modularidad, simplicidad portabilidad, escalabilidad, etc.)
- Los estándares de modelado de UML
- Las interfaces asociadas como lo son otros componentes de producto o elementos externos a la solución.
- La adherencia a los requerimientos.

Esto involucra el diseño de comportamiento, que son diagramas utilizados para visualizar, especificar, construir y documentar los aspectos dinámicos de un sistema:

- Diagrama de contexto: Muestra a través de flujos de datos las interacciones existentes entre los agentes externos y el sistema, sin describir en ningún momento la estructura del sistema de información. El sistema en su totalidad se representa como una sola entidad.
- Diagramas de caso de uso: Muestra el conjunto de casos de uso y actores (incluyendo sus relaciones). Estos diagramas se utilizan para ilustrar la vista de caso de uso del sistema.

- Diagrama de secuencia: Es un diagrama de iteración que enfatiza el orden en tiempo de los mensajes.
- Diagrama de colaboración: Es un diagrama de iteración que enfatiza la organización estructural de los objetos que envían y reciben mensajes. El diagrama de colaboración muestra un conjunto de objetos, ligas entre ellos y los mensajes enviados y recibidos por dichos objetos.
- Diagrama de estado: Muestra una máquina de estado, consistente en estados, transiciones, eventos y actividades. Estos diagramas enfatizan el comportamiento ordenado por eventos de un objeto.
- Diagrama de actividad: Muestra el flujo de una actividad a otra dentro del sistema. Se diseña para mostrar una visión simplificada de lo que ocurre durante una actividad o suceso.

La selección dependerá de cada proyecto; el ajuste queda en el plan de procesos.

Se debe emplear las herramientas StarUML y Enterprise Architect (escoger una o utilizar ambas, aunque se recomienda el uso de StarUML en cuanto a la arquitectura de alto nivel) para realizar los diagramas requeridos para el diseño de alto nivel.

Los diseños generados durante esta actividad forman parte del paquete técnico de datos, el cual tomarán como base los desarrolladores para codificar los componentes de producto, además de soportar estrategias de adquisición, implementación, liberación en producción y soporte.

Este paquete técnico de datos se actualiza conforme sea necesario y el ciclo de vida del proyecto va avanzando.

Una vez finalizados los diseños, los seleccionados para realizarles un PR son enviados a revisión de acuerdo con el procedimiento de Ejecutar Revisión entre Colegas (que anteriormente se explicó este procedimiento) y se resuelven los hallazgos encontrados.

Si se tiene planificada la ejecución de línea base (anteriormente vimos de que se trataba la LB), el Líder de Proyecto notifica al CM su elaboración.

### Realizar diseño de bajo nivel

El diseñador comienza a desarrollar los diagramas estructurales del sistema, que sirven para visualizar, especificar, construir y documentar los aspectos estáticos del sistema, entre ellos están:

- Diagrama de base de datos o modelo de datos.

- Diagrama de clase: Muestra un conjunto de clases, interfaces y sus relaciones.
- Diagrama de objetos: Muestra un conjunto de objetos y sus relaciones. A diferencia de los diagramas anteriores, estos diagramas se enfocan en la perspectiva de caso de uso, y prototipos.
- Diagramas de componentes: Muestra el conjunto de componentes y sus relaciones y se utilizan para ilustrar la vista de la implementación estática de un sistema.
- Diagrama de implantación: Muestra un conjunto de nodos y sus relaciones, se usan para ilustrar la vista de implantación estática de un sistema.

La selección dependerá de cada proyecto; el ajuste queda en el plan de procesos.

En cuanto al modelo de datos, se debe utilizar la herramienta de diseño propia del gestor de base de datos elegido (siempre y cuando este modelo pueda ser almacenado en un archivo para que pueda manejarse de forma individual como un elemento de la configuración), en caso de que el gestor no cuente con una herramienta de diseño, se debe usar la herramienta Enterprise Architect.

Al realizar el diseño de bajo nivel para cada caso de uso debe emplearse Enterprise Architecture o StarUML aunque se recomienda el uso de la primera herramienta.

Los diseños generados durante esta actividad forman parte del paquete técnico de datos, el cual tomarán como base los desarrolladores para codificar los componentes de producto, además de soportar estrategias de adquisición, implementación, liberación en producción y soporte.

Este paquete técnico de datos se actualiza conforme sea necesario y el ciclo de vida del proyecto va avanzando.

Una vez finalizados los diseños, los seleccionados para realizar un PR son enviados a revisión de acuerdo al procedimiento de Ejecutar Revisiones entre Colegas y se resuelven los hallazgos encontrados.

Si se tiene planificada la ejecución de línea base, el Líder de Proyecto notifica al CM su elaboración.

### **Diseñar los casos de prueba**

El Tester toma el paquete técnico de datos y la lista de V&V para comenzar el diseño de sus casos de prueba.

Si las pruebas serán automatizadas, diseña el escenario Testlink.

Si las pruebas son manuales diseña el escenario en el documento de caso de prueba.

Durante la elaboración de los casos de prueba el Tester debe conocer:

- Los datos de entrada de las pruebas, ya que el resultado de las pruebas depende de estos.
- Qué componentes interactúan en cada caso de prueba.
- Qué resultados debe devolver el componente según los datos de entrada utilizados en la prueba.
- Tolerancias permitidas.
- Rangos aceptables.
- Los criterios de aceptación.
- Reglas de negocio.

El Tester se apoya del analista para resolución de dudas, y en caso de encontrar problemas que pudieran afectar a la verificación y validación se le notifica el Líder de Proyecto para su resolución.

Una vez finalizados los diseños, los seleccionados para realizarles un PR son enviados a revisión de acuerdo con el procedimiento de Ejecutar Revisiones entre Colegas y se resuelven los hallazgos encontrados.

Si se tiene planificada la ejecución de línea base, el Líder de proyecto notifica al CM su elaboración. Ver procedimiento Administrar Configuración.

Nota: Los escenarios de pruebas son enviados preferentemente al Cliente para su validación; con el objetivo de tener otro punto de control en la validación de los requerimientos; ya que prácticamente los escenarios de prueba definen el comportamiento esperado del sistema. La decisión dependerá del contexto del proyecto.

### Diseñar Plan de Integración

El líder de proyecto designa a una persona responsable de realizar el plan de integración.

Esta persona toma el documento de arquitectura y desarrolla la vista de la implementación registrando:

- El procedimiento de integración, considerando:
  - Iteraciones y componentes que la conjuntan.
  - Políticas y/o convenciones de integración.
    - Cuando un componente de producto se encuentra listo para integrarse o es aceptable.

- Responsabilidades explícitas de integración entre los miembros del equipo desarrollo.
- Iteraciones de verificación.
- Las herramientas requeridas para la integración.
- Ambientes de integración.

### **Diseñar los manuales**

El documentador, comienza a desarrollar versiones preliminares de la documentación de entrega acordada para el proyecto, haciendo uso de los estándares definidos en el CDS. Esta documentación puede incluir:

- Manual de usuario.
- Manual de instalación u operación.
- Manual de mantenimiento.
- Ayuda en línea.
- Documentación técnica del producto o componentes de producto.

Esta actividad puede ser finalizada hasta el cierre de la construcción del producto o iteración.

### **Construir e integrar**

Esta es la última actividad para el subproceso de Diseñar Solución, a continuación, será explicado.

### **Construir e integrar**

Este proceso se ejecuta para construir e integrar el producto a partir de sus componentes, garantizando que la solución ya integrada funciona adecuadamente y puede ser entregada al cliente

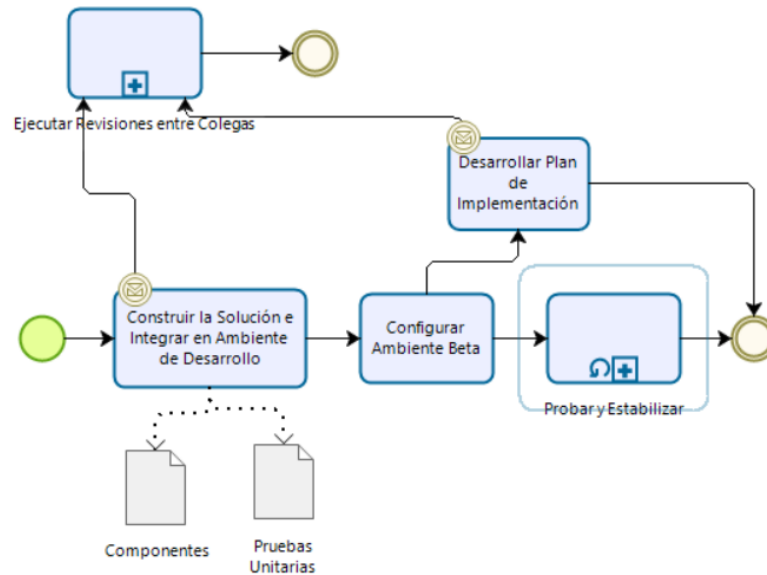


Ilustración 27 Construir e Integrar

#### Criterios de entrada/Entradas:

- El procedimiento inicia cuando se cuenta con un paquete técnico de datos el cuál un desarrollador puede basarse para iniciar la construcción.

#### Entradas:

- Línea base del paquete técnico de datos.
- Estándares de desarrollo.
- Plan de integración.
- Matriz de trazabilidad.

#### Criterios de salida/Salidas:

- El proceso termina cuando los componentes de producto han sido integrados, verificados y validados para ser liberados a producción.

#### Salidas:

- Solución construida.
- Verificaciones.
- Plan de integración/implementación.
- Matriz de trazabilidad [actualizada].

## **Roles necesarios para el proceso (Recursos humanos):**

### **Desarrollador:**

- Construir la solución.
- Ejecutar pruebas unitarias.
- Ejecutar revisiones entre colegas.
- Integrar la solución.

### **Tester:**

- Realizar la verificación planeada (pruebas integración/regresión/estrés/funcionales).

### **Líder técnico:**

- Coordinar y realizar la integración al ambiente beta.
- Realizar la integración de la base de datos.

### **Auditor:**

- Asegurar la calidad de los productos y apego al proceso.

## **Construir la solución e integrar en ambiente de desarrollo.**

Cada desarrollador configura su ambiente de desarrollo, haciendo la vinculación de la API de desarrollo con el manejador de versiones. Ver guía de CM para ayuda y seguir la estructura de carpetas definidas.

Los desarrolladores toman como base la arquitectura y los estándares de codificación y realizan el desarrollo del producto de acuerdo con el método y lenguaje especificado para el proyecto.

Al terminó de cada caso de uso implementado, el desarrollador responsable de su construcción ejecuta sus pruebas unitarias y actualiza el registro de trazabilidad en la matriz de trazabilidad.

- Si las pruebas unitarias se realizan automatizadas, el log de pruebas se relaciona en la documentación del código.
- Si las pruebas unitarias son sobre la misma prueba/compilación del componente, se registra el estatus de “prueba unitaria ejecutada” en la documentación del código.



La integración de los componentes en ambiente de desarrollo se va realizando de manera continua e incrementa hasta formar una integración parcial/final que sea migrada a una ambiente beta para realizar la verificación por calidad.

La integración se realiza de acuerdo con la secuencia, procedimiento y criterios descritos en la vista de implementación descrita en la arquitectura.

**NOTA:** Se realizan revisiones entre colegas del código de aquellos que hayan sido seleccionados en la lista de V&V, y siguiendo el procedimiento Ejecutar Revisiones entre Colegas.

Esta actividad se realiza durante toda la fase de construcción de ciclo de vida del proyecto; mediante el seguimiento que dé el líder de proyecto (reuniones técnicas y/o reuniones de progreso) con su equipo de trabajo se va revisando y asegurando:

- La compatibilidad entre interfaces (internas/externas)
- Que la secuencia, el procedimiento y los criterios de integración estén siendo exitosos.
- La cobertura de requerimientos mediante el seguimiento y actualización de la matriz de trazabilidad y la consistencia entre la especificación y la implementación.
- El apego a la configuración del producto y su compilación.

En caso de haberse encontrado con inconsistencias a la hora de seguir el plan de integración el líder técnico actualiza los elementos correspondientes.

Y en caso de presentarse asuntos o conflictos el líder de proyecto los registra y da seguimiento de acuerdo al procedimiento de Monitorear Planes y Proyectos.

### **Configurar ambiente beta**

El CM del proyecto, toma el documento de arquitectura y monta el ambiente descrito en la vista de implementación que es donde será probada la integración funcional de todos los componentes desarrollados.

Una vez implementada a solución, el Líder Técnico verifica que cumpla con las características descritas y realiza una prueba ad hoc.

Si la integración en la ambiente beta, se muestra estable, el Líder Técnico notifica al *Tester* que el ambiente se encuentra listo para que inicie el control de calidad.

El *tester* se basa en la arquitectura y lista de V&V para identificar rutas, usuarios y tipos de prueba a ejecutar.

### **Ejecutar Revisión entre Colegas**

Esta actividad será explicada más adelante en este documento, ya que tiene la función de ser un subproceso en sí.

### **Desarrollar plan de implementación**

El Líder Técnico toma el documento de arquitectura y desarrolla la vista de despliegue. Adicionalmente toma en cuenta los requerimientos operativos especificados en el documento de caso de uso.

La vista de despliegue describe el procedimiento para liberar la solución a producción; de manera que los asuntos presentados en la integración en ambiente beta se toman como lección aprendida para mejorar procedimientos y secuencia de integración en un ambiente de producción.

### **Probar y estabilizar**

Esta actividad, es explicada a continuación.

### **Probar y Estabilizar**

Este proceso se ejecuta para:

- Asegurar que los productos de trabajo seleccionados cumplan con sus requerimientos especificados.
- Demostrar que los productos de trabajo o componentes de producto cumplen con su propósito cuando son puestos en el ambiente determinado.

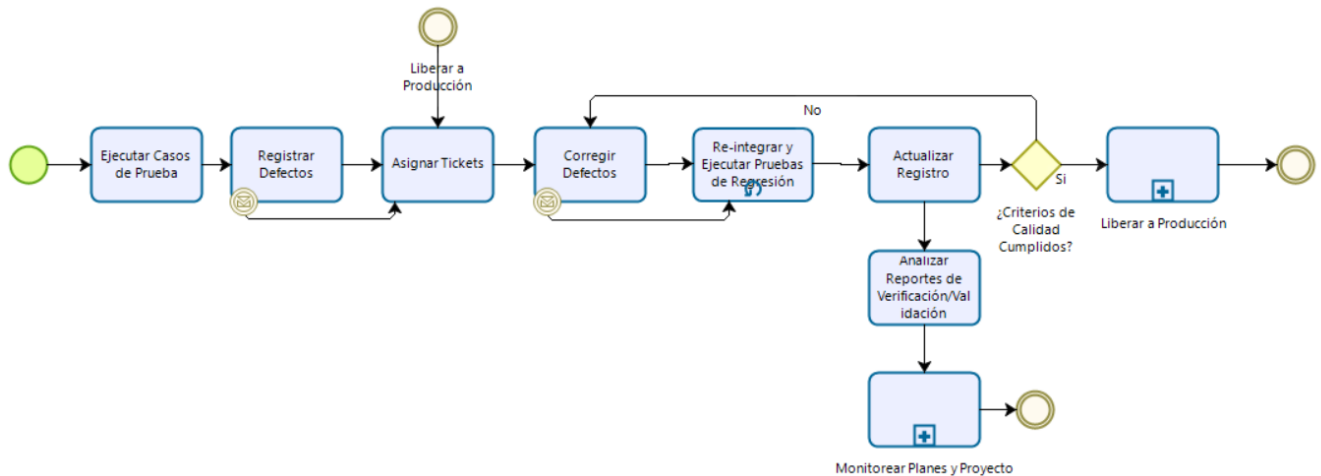


Ilustración 28 Probar y Estabilizar

**Criterio de entrada/Entradas:**

**Criterio de entrada:**

- El proceso arranca cuando existe un disparador en el plan de V&V y/o *ticket* en mantis sobre un componente a ser validado y/o verificado.

**Entradas:**

- Calendario de trabajo.
- Plan de V&V.
- Casos de prueba.
- *Ticket* en mantis [nuevos].

**Criterio de salida/Salidas:**

**Criterio de salida:**

- El proceso termina una vez que se hayan corregido las desviaciones encontradas en los productos de trabajo que fueron verificados y validados, así como el análisis de los resultados.

**Salidas:**

- Casos de prueba corridos.

- *Tickets* en mantis [nuevos/actualizados].
- Análisis de los resultados de la V&V y acciones correctivas.

Roles necesarios para el proceso (Recursos humanos):

**Tester:**

- Ejecutar los casos de prueba diseñados.
- Ejecutar los ciclos de prueba planeados de acuerdo con los métodos especificados.
- Dar de alta *tickets* en mantis de los defectos encontrados.
- Generar reportes de calidad y comunicarlos al Líder de Proyecto para su análisis.

**Líder de Proyecto:**

- Dar seguimiento a la calidad observada en el proyecto y tomar las acciones preventivas/correctivas necesarias.
- Administrar los *tickets* abiertos de su proyecto hacia su equipo de trabajo.

**Desarrolladores:**

- Atender los *tickets* asignados y actualizar los estatus correspondientes.

**Auditor:**

- Asegurar la calidad de los productos y el apego al proceso.

**Ejecutar Casos de Prueba**

El *Tester* toma sus casos de prueba y corre los escenarios diseñados.

**Registrar defectos**

En caso de encontrar defectos, los registra como un *ticket* en mantis, asignando como responsable al Líder de Proyecto. El registro incluye:

- El estatus de “nuevo”
- Descripción del defecto
- Tipificación
- Script de reproducción del defecto
- Integración de imagen del defecto arrojado (preferentemente para todos los casos)

### Asignar *Tickets*

El Líder de Proyecto, recibe la notificación de los defectos, y redistribuye los *tickets* entre su equipo de trabajo.

Esta actividad puede ser arrancada en dos momentos:

- Una vez registrados los defectos por parte del *Tester*
- Por defecto identificado por el Cliente y comunicado al Líder de Proyecto, en este caso quién registra el defecto es el Líder de Proyecto, lo registra en mantis como defecto encontrado en producción y lo asigna a un responsable de atenderlo.

### Corregir defectos

El desarrollador responsable de atender el defecto realiza la corrección en el componente correspondiente y realiza sus pruebas unitarias.

### Reintegrar y ejecutar pruebas de regresión

El Líder Técnico concentra los componentes relacionados a los *tickets* de estatus “resuelto” a red desplegar en beta y actualiza el conjunto de componentes para su siguiente iteración de pruebas.

Una vez reintegrado en ambiente beta, cambia el estatus del *ticket* a “resuelto”.

El *Tester*, verifica que los *tickets* de estatus “resuelto” ya no generen el defecto registrado y realiza las pruebas de regresión necesarias.

### Actualizar registro

Por cada iteración de pruebas, el *Tester* genera un reporte de estadísticas de los defectos encontrados y lo envía al Líder de Proyecto para su análisis y toma de acciones correctivas.

### Analizar Reportes de Verificación y Validación

El Líder de Proyecto toma el reporte de las verificaciones y validaciones ejecutadas y realiza un análisis de:

- Volumen/densidad de defectos que se han arrojado, en ambiente beta y producción.
- Tipo de defectos encontrados/fugados.
- La eficiencia de los métodos utilizados en la verificación y validación.
- Compara resultados actuales como los esperados.

El análisis de estos resultados le permite al Líder de Proyecto medir el desempeño técnico del proyecto y en caso de encontrarse desviaciones toma las acciones correctivas necesarias, y registra los asuntos y/o riesgos observados de acuerdo con el procedimiento Monitorear Planes y Proyecto.

### Monitorear Planes y Proyecto

Esta actividad ya fue explicada anteriormente en este documento.

### Liberar a producción

Esta actividad será explicada a continuación.

### Liberar a Producción

Este proceso se ejecuta para realizar liberaciones de software en un ambiente de producción.

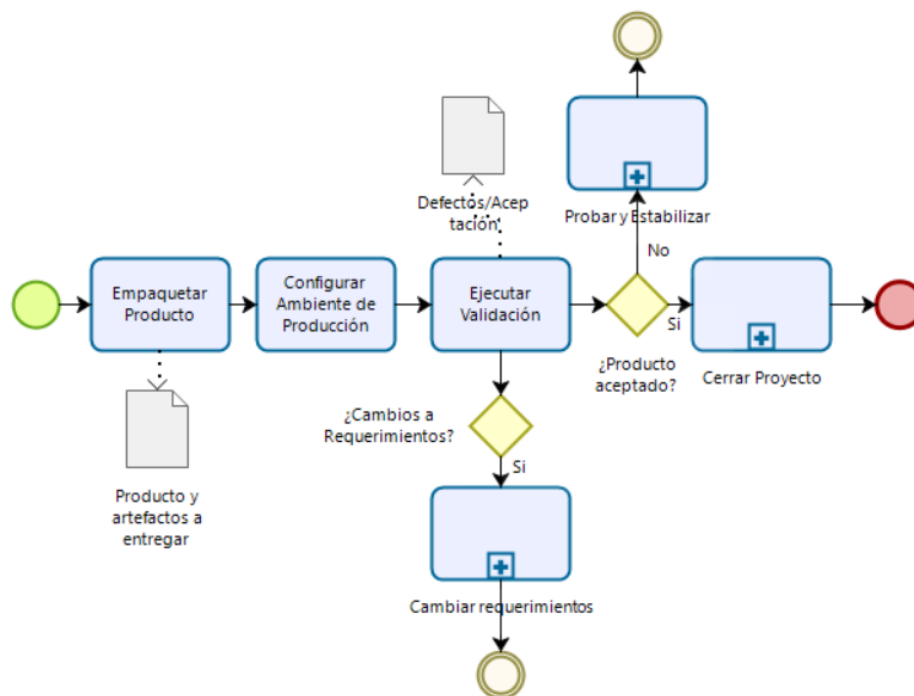


Ilustración 29 Liberar a producción

### Criterio de entrada/Entradas:

#### Criterios de entrada:

- El procedimiento arranca cuando se han pasado las pruebas de calidad internas.

#### Entradas:

- Paquete técnico de datos.

- Draft de los manuales.
- Línea base funcional de código.
- Plan de implementación.

**Criterio de salida/Salidas:**

**Criterios de salida:**

- El procedimiento termina una vez que el Cliente nos ha aceptado el producto liberado.

**Salidas:**

- Defectos en el software y/o aceptación por parte del Cliente.
- Solicitudes de cambio.

Roles necesarios para el proceso (Recursos Humanos):

**Líder técnico:**

- Coordinar la liberación del producto.
- Configurar el ambiente de producción.

**Líder de Proyecto:**

- Coordinar la validación del producto con el Cliente.

**Auditor:**

- Asegurar la calidad de los productos y el apego al proceso.

## **Empaquetar producto**

El líder Técnico coordina la ejecución de la vista de despliegue descrita en el documento de arquitectura, asegurándose que:

- Esté de acuerdo con los requerimientos operacionales de la especificación de requerimientos en el documento de casos de uso.
- Se cuenten con la línea base funcional del software. (es decir, que los defectos resultantes de la V&V hayan sido cerrados).
- Se cuenten con la línea base del paquete técnico de datos y los manuales a entregar al cliente (ibíd).

- Que los problemas identificados durante la vista de implantación (en ambiente beta) se encuentren resueltos y no se vean reflejados en la vista de despliegue (ambiente de producción).

### **Configurar ambiente de producción**

La ejecución de esta actividad dependerá de la responsabilidad especificada en el documento de casos de uso sobre quien se encargará de implementar el producto (Cliente o CDS) y el ambiente requerido (p.e. ambiente beta, o de producción del cliente).

El líder técnico se guía de lo establecido en la vista de despliegue y en el manual de instalación/operación aplicando los métodos de empaquetado definidos para preparar el sitio operacional en donde se implementará el producto.

### **Ejecutar validación**

El líder de Proyecto notifica/entrega al Cliente el producto acordado, junto con la documentación del paquete técnico de datos comprometida para el proyecto.

Se ejecutan el periodo de validación, según lo acordado en los requerimientos operativos y los métodos establecidos en la lista de V&V.

El líder de Proyecto revive los defectos reportados por el Cliente y analiza si se trata de un cambio a requerimiento o de un defecto en el producto. Si es defecto levanta un ticket en mantis y se corre el proceso de Probar y Estabilizar. Si se trata de un cambio a requerimiento, se corre el proceso Cambiar Requerimiento.

Esta actividad se corre por el periodo de tiempo acordado, y una vez que el producto ha sido aceptado por parte del Cliente, se inicia el procedimiento Cerrar Proyecto.

### **Cambiar requerimientos**

Esta actividad se explicará más adelante ya que también forma parte de un subproceso.

### **Probar y estabilizar**

Esta actividad se explicó anteriormente en este documento.

### **Cerrar Proyecto**

Esta actividad se explicó anteriormente en este documento.



Con el subproceso de Liberar a Producción se da por terminado la etapa dos la arquitectura de procesos, y ahora podemos empezar con la tercera y última etapa, la cual es el Aseguramiento de Calidad.

### Etapa 3. Procesos continuos y de soporte

En esta tercer y última etapa de la arquitectura de procesos, se crearán las líneas base para el desarrollo del proyecto, además de realizar las auditorías pertinentes para la revisión de la configuración. Se ejecutarán revisiones entre colegas para poder analizar y tomar soluciones para el bien del proyecto.

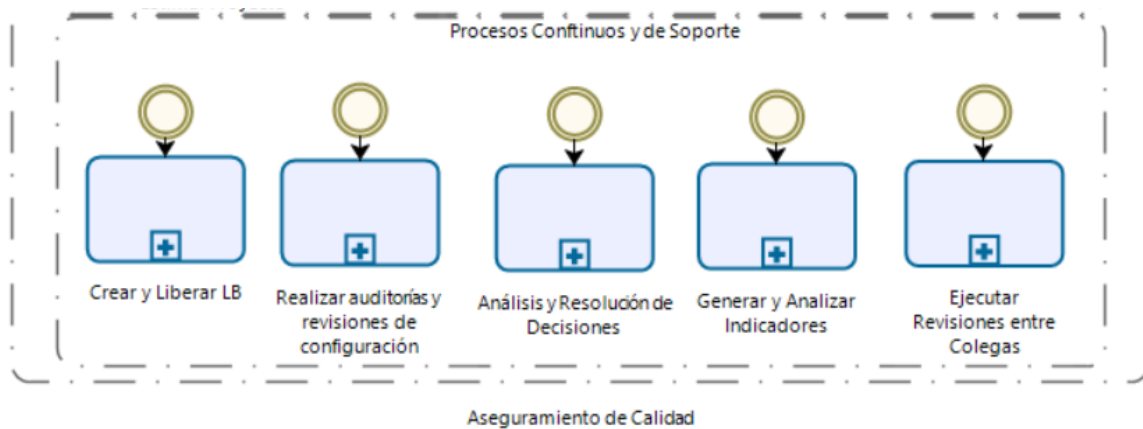


Ilustración 30 Procesos Continuos y de soporte

### Crear y liberar LB

Mediante este proceso se crean y liberan líneas base de código.

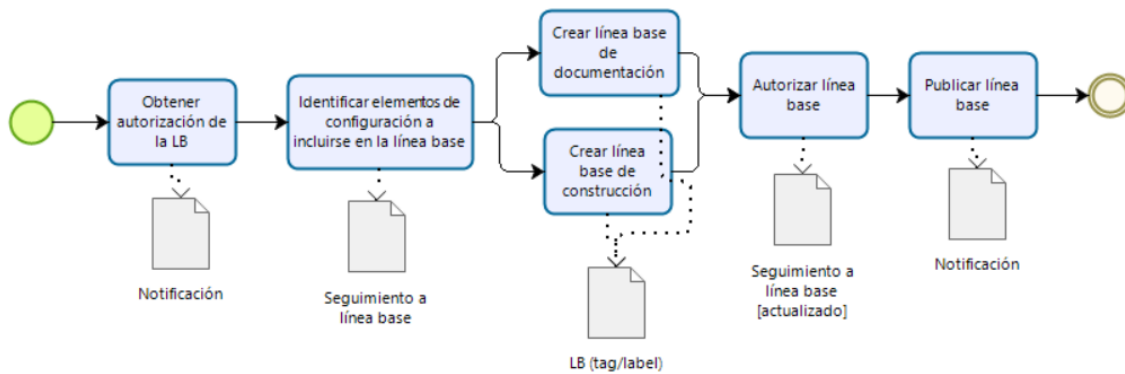


Ilustración 31 Crear y Liberar LB

### Criterios de entrada/Entradas:

- Se llega el tiempo de acuerdo con el plan del proyecto, para crear o liberar una línea base de código.

- Hubo un cambio a la línea base código.

**Criterios de salida/Salidas:**

- Aseguramiento a LB.
- Línea Base autorizada, publicada.

Roles necesarios para el proceso (Recursos Humanos):

**Líder de Proyecto:**

- Supervisa la ejecución de CM en el proyecto
- Audita la LB.

**CM Manager:**

- Participa en todas las actividades de este proceso.

***Configuration Control Board (CCB):***

- Evalúa y toma decisiones que afectan a las líneas base.

**Obtener autorización de la Línea Base**

Si la LB es nueva:

El CM determina según el calendario de trabajo que la actividad de crear LB se debe de ejecutar. El CM valida con el LP si los elementos se encuentran listos para la creación de la LB (si el LP es el CM del proyecto, la validación consiste en una revisión individual).

Si la LB es por control de cambios:

- Disparados por el cliente
  - El CM toma la solicitud de cambio e identifica los elementos que afectaran a la línea base.
- Disparados internamente por el CCB
  - El CM valida el motivo e impacto del cambio a la línea base en conjunto con el CCB y registra la evaluación en el documento de seguimiento a línea base como notas del registro de la LB.

## Identificar elementos de configuración a incluirse en línea base.

Si la LB es “nueva”:

- CM toma el documento de seguimiento a línea base y registra el nombre del documento físico a lo que corresponden el contenido de la LB.

Si la LB es por control de cambios:

- CM registra el nombre y contenido de la nueva línea base en el documento de seguimiento a línea base, registrando el nombre del documento físico a los que corresponden el contenido de la LB.

El CM se asegura que los elementos a incluir en la LB estén finalizados, que no estén bloqueados y que exista la evidencia de que el producto de trabajo es correcto.

La evidencia puede ser una aceptación por parte del cliente y/o control de calidad interno (por ejemplo: que los hallazgos de PR, pruebas y/o auditorías encontrados sobre el elemento se encuentren cerrados).

## Publicar en Línea base.

Una vez aprobada la LB, el CM notifica al equipo de trabajo que ha sido liberada una nueva línea informando el contenido de esta.

## Realizar Auditorías y Revisiones de Configuración

Proceso mediante el cual se realizan las auditorías a los elementos de la configuración en el sistema de la administración de la configuración.

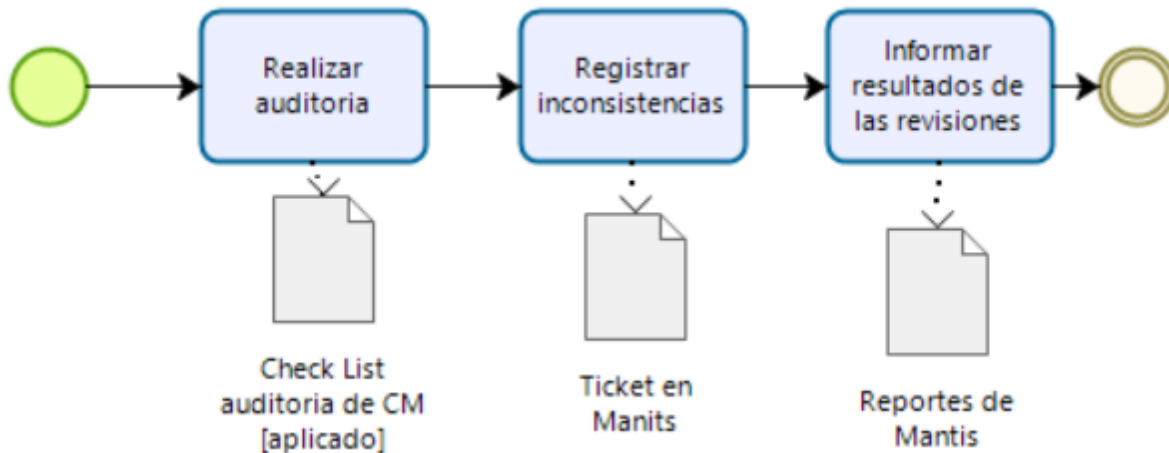


Ilustración 32 Realizar Auditorías y Revisiones de la Configuración

**Criterio de entrada/Entradas:**

- Se ha llegado el momento planeado para realizar la auditoría.

**Criterio de salida/Salidas:**

- Se registraron todas las inconsistencias

**Roles (Recursos Humanos):**

**CM Manager:**

- Es la persona encargada de llevar a cabo las auditorías, informar y dar seguimiento hasta la resolución de las no conformidades.

**Líder del Proyecto:**

- Se encarga de que se resuelvan las no conformidades encontradas.

**Realizar Auditorías**

El CM Manager realiza la auditoría de configuración aplicando el *Check List* auditoría de CM.

**Registra inconsistencias**

El CM Manager, registra en mantis los hallazgos encontrados en la auditoría, asignando como responsable al miembro del equipo de trabajo que corresponda.

La fecha de atención del hallazgo se establece para el siguiente día hábil.

El responsable atiende el hallazgo y cambia el estatus a “cerrado”.

Al día siguiente de la auditoría, el CM Manager verifica que los hallazgos han sido atendidos, y/o en su caso da seguimiento hasta su cierre.

En caso de que un hallazgo no haya sido atendido y/o dispare algún asunto y/o riesgo, el CM Manager se los comunica al LP para su intervención (en caso de que el CM Manager sea el mismo LP se omite esta comunicación y se pasa al punto 3 de este PCD).

### Informar resultados de las revisiones

El CM Manager extrae reportes del estatus de los hallazgos registrados en mantis y mediante las reuniones entre el LP y el equipo de trabajo, se da seguimiento a los hallazgos que aún estén abiertos y se atiende todo asuntos o riesgo disparado de esta actividad mediante el procedimiento de Monitorear Planes y Proyecto.

### Generar y Analizar indicadores

Es el proceso encargado de recolectar, analizar, resguardar y comunicar indicadores que apoyen a la toma de decisiones a nivel proyecto.

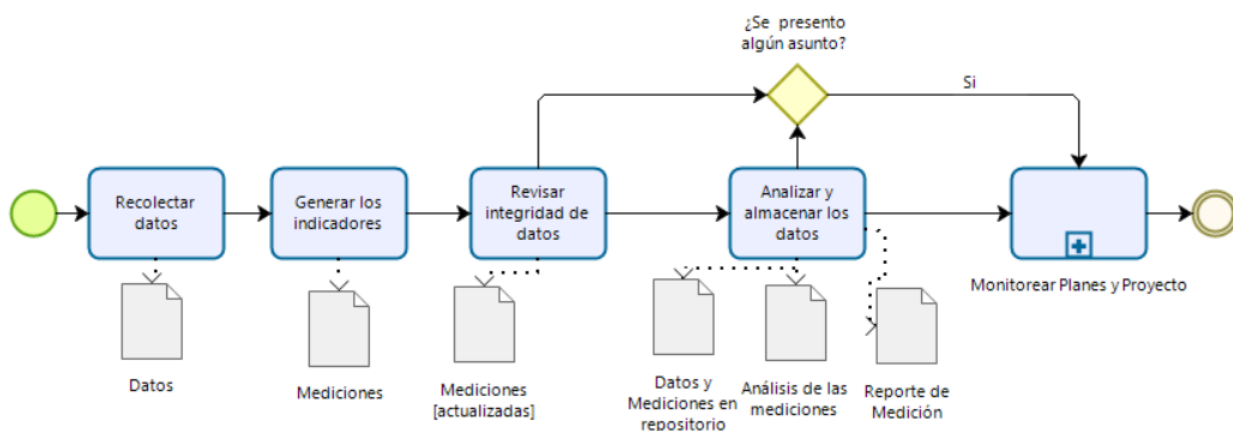


Ilustración 33 Generar y Analizar Indicadores

### Criterios de entrada/Entradas:

#### Criterios de entrada:

- El procedimiento inicia cuando se han identificado las mediciones a recolectar y analizar en el proyecto.

**Entradas:**

- Matriz de objetivos y necesidades de información.
- Definición del indicador.

**Criterio de salida/Salidas:****Criterios de salida:**

- El proceso termina una vez que el proyecto ha sido cerrado y durante toda la vida del proyecto se han tomado decisiones derivadas de los indicadores recolectados.

**Salidas:**

- Datos recolectados.
- Análisis y reportes de los indicadores.
- Repositorio de mediciones [actualizado].

Roles necesarios para el proceso (Recursos Humanos):

**Líder de Proyecto:**

- Asegurar la integridad de las mediciones.
- Recolectar las métricas que le corresponden y apegarse a los procedimientos establecidos para la medición, análisis y reporte de la información.

**Miembros del equipo:**

- Recolectar las métricas que le corresponden y apegarse a los procedimientos establecidos para la medición, análisis y reporte de la información.

**Responsables de métricas de la organización:**

- Asesorar a los miembros de los proyectos, en la recolección, análisis e interpretación de las mediciones.

**Auditor:**

- Asegurar la calidad de los productos y el apego al proceso.

## Recolectar datos

Cada miembro del CDS cumple con lo establecido en los archivos objetivos y necesidades de información y definición del indicador para recolectar y analizar las mediciones que estén bajo su responsabilidad.

El responsable de recolectar, analizar y recolectar alguna métrica se dirige al archivo objetivos y necesidades de información, para identificar que indicador se recolectará y dónde se encuentra ubicada su definición.

Por cada indicador se encuentra definido:

- Quién es el responsable de recolectar la información.
- Qué datos son lo que hay que recolectar.
- Qué herramientas se utilizarán y quién se las puede proveer.
- De qué forma se recolectará la información.
- Cuando y con qué frecuencia se recolectará.
- En qué lugar se almacenará la información y los medios para acceder a estos lugares.

El responsable de la recolección sigue los pasos descritos.

## Generar los indicadores

El responsable de la recolección ejecuta lo descrito en la sección de algoritmo de la definición del indicador.

## Revisar integridad de los datos

Una vez generados los indicadores, revisar la sección validación de datos del procedimiento del indicador, en donde se encontrará qué es lo que se debe revisar de tal indicador.

En caso de encontrar inconsistencias en la información, el recolector valida los datos con el responsable de haber generado la información. Se comenta, se resuelve y se actualiza el dato.

Si la magnitud de la inconsistencia se convierte en asunto, se le comunica al Líder de Proyecto para su intervención. (Si el recolector es el mismo Líder de Proyecto se omite este paso).

El líder de Proyecto registra el asunto en mantis y le da seguimiento de acuerdo al procedimiento de Monitorear Proyecto.



## Analizar los datos

Con los datos ya generados y validados, dirigirse a la sección de interpretación de la definición del indicador.

El análisis se toma como base para la toma de decisiones a nivel proyecto. Antes de comunicar los datos a la Gerencia, se realizan análisis y validaciones internas entre los miembros del equipo. En caso de observar asuntos o riesgos derivados del análisis, el Líder de Proyecto se apoya del procedimiento Monitorear Planes y Proyecto para su control.

La información se resguarda según lo descrito en la sección de almacenamiento de los datos de la definición del indicador.

La periodicidad de reporte del indicador se encuentra en la sección de reporte de datos de la definición del indicador.

## Monitorear Planes y Proyecto

Esta actividad ya fue explicada anteriormente en este mismo documento.

## Ejecutar Revisiones entre Colegas

Este proceso se ejecuta para encontrar y eliminar defectos de manera temprana.

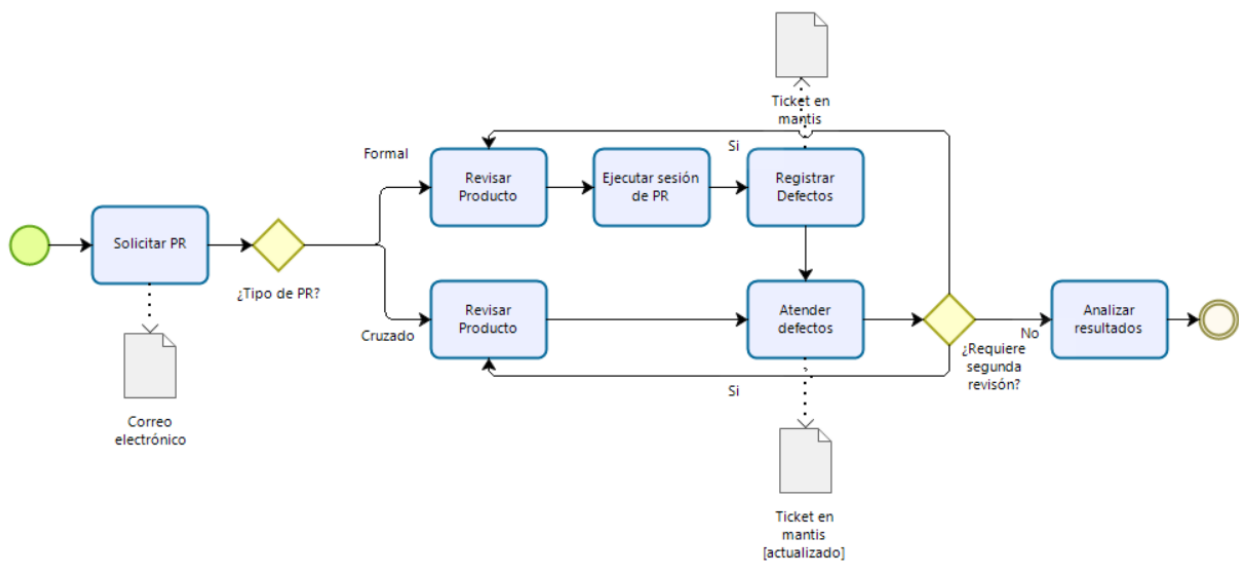


Ilustración 34 Ejecutar Revisión entre Colegas

**Criterio de entrada/Entradas:**

- El procedimiento inicia cuando existen productos de trabajo que requieren ser revisados dentro de un proyecto de acuerdo con lo estipulado en la lista de V&V.

**Entradas:**

- Calendario de trabajo.
- Lista de productos a V&V.
- Listas de verificación.

**Criterio de salida/Salidas:****Criterios de salida:**

- El procedimiento termina cuando se han cerrado los defectos encontrados en los productos de trabajo inspeccionados.

**Salidas:**

- *Tickets* en mantis [nuevos/actualizados].
- Análisis de los resultados las inspecciones y acciones correctivas.
- Productos revisados.

**Roles necesarios para el proceso (Recursos Humanos):****Autor:**

- Solicitar la revisión del producto de trabajo, enviando producto y lista de verificación.
- Atender los defectos encontrados a su producto de trabajo.
- Realizar una revisión personal antes de enviar el producto a revisión.

**Revisor:**

- Verificar el producto de trabajo contra la lista de verificación.
- Registrar los defectos encontrados en mantis (en caso de haber realizado revisión cruzada).

**Moderador:**

- Fungir como moderador en una sesión de revisión. Implica
  - Asegurarse que los revisores lleguen a la sesión con el producto revisado.

- Iniciar y terminar la sesión en el tiempo acordado.
- Dirigir la revisión hacia el producto y no hacia la persona.
- Evitar conflictos durante la sesión de revisión.
- Asignar el rol de escriba durante la sesión de revisión.

**Escriba:**

- Registrar los defectos encontrados en mantis durante la sesión de revisión.

**Auditor:**

- Asegurar la calidad de los productos y el apego al proceso.

## Solicitar PR

En la primera actividad del proceso de Ejecutar Revisión entre Colegas el autor de (los) producto(s) de trabajo solicita a los revisores la inspección de estos conforme a lo establecido en la lista de verificación y validación (V&V). En la notificación que es hecha por correo electrónico se especifica:

- El archivo o ruta donde se encuentra el material a revisar.
- La lista de verificación que se tomará en cuenta durante la revisión.
- Fecha requerida para la recepción de defectos encontrados (en caso de ser un PR cruzado).
- Fecha, hora y duración de la reunión (en caso de ser PR Formal).

Una vez cumplida con esta actividad, ahora tenemos dos caminos para realizar la revisión por pares: de manera formal o cruzada. A continuación, se describen cada una de estas formas de RP.

## Revisar el Producto

### *Forma cruzada*

Si se elige el camino de PR de manera cruzada, la primera actividad a desarrollar es revisar el producto, aquí el Revisor realiza la verificación contra la lista de verificación a aplicar. Los defectos encontrados, los registra en mantis (mantis es una herramienta de gestión de proyectos) y los asigna al autor del producto.

### Atender defectos

Una vez revisado el proyecto se continua con la actividad de atender defectos, en ella el autor corrige los defectos registrados en mantis y actualiza el estatus del *ticket*. Si se acordó que el

producto debería volver a ser revisado, el autor notifica a los revisores para que verifiquen que los defectos fueron atendidos. En cambio, si ya no es necesario otra revisión o la segunda revisión paso por el ok del revisor se continua con la fase final de analizar los resultados.

### **Analizar resultados**

En la actividad de Analizar resultados el Líder de proyecto extrae de mantis reportes de las verificaciones ejecutadas y realiza un análisis de:

- Volumen/densidad de defectos que se han arrojado, en ambiente beta y de producción.
- Tipo de defectos encontrados/fugados.
- La eficacia de los métodos utilizados en la verificación y validación.
- Compara resultados actuales con los esperados.
- El análisis de estos resultados le permite al líder de proyecto medir el desempeño técnico del proyecto y en caso de encontrarse desviaciones tomas las acciones correctivas necesarias, y registra los asuntos y/o riesgos observados de acuerdo con el procedimiento Monitorear Planes y Proyecto.

Con esto se da por terminado la actividad de Ejecutar Revisiones entre colegas por medio de la revisión cruzada.

### **Revisar Producto**

#### **revisión Formal.**

Si se elige la forma de revisión formal la primera actividad que se debe realizar es Revisar Producto. Aquí los revisores realizan la verificación contra la lista de verificación a aplicar. Anotan por medio de comentarios en el mismo producto de trabajo los defectos detectados.

### **Ejecutar Sesión de PR**

Una vez revisado el producto, se debe de ejecutar sesión de Pr, donde el moderador arranca la sesión siempre y cuando la mayoría de los revisores lleven consigo los defectos encontrados en su revisión (decisión a criterio del líder de proyecto, el objetivo es que se encuentren defectos que aporten valor al producto revisado). Para esta actividad los ejecutantes necesarios son: Revisor, Autor, Moderador y el Escriba.

## **Registrar Defectos**

Ahora se continua con la actividad de Registrar Defectos aquí el escriba registra los defectos en mantis, asignándolos al autor y realizando la clasificación del defecto.

## **Analizar Resultados**

Una vez registrados los defectos se preguntan si se requiere una segunda revisión si es así, se vuelven a iniciar la revisión entre pares de la manera formal, hasta que todos estén de acuerdo en continuar con la siguiente actividad. Una vez todos estén de acuerdo se analizan los resultados efectuados en la revisión entre colegas.

Con esta última actividad damos por concluido la etapa 3 y el análisis del esquema de la certificación CMMI en el CDS del ITSUR.

Ahora pasaremos a identificar los problemas funcionales que tienen el CDS, cuál es su problema central y que efectos colaterales puede tener dicho problema.

Con esto damos por terminado el esquema de trabajo del CDS del ITSUR. Ahora nos enfocaremos a identificar los problemas que se tiene en el CDS por no contar con una forma de trabajo estandarizada.

## Análisis del centro de desarrollo de software del ITSUR

Ya que se analizó el esquema de trabajo con el que operaba el CDS, podemos ver que problemas multifuncionales fueron los que surgieron al momento de trabajar con esta certificación.

- El primer problema que pudimos encontrar fue la falta de personal al momento de realizar las actividades de la certificación.

Esta certificación cuenta con una alta cantidad de actores, que desempeñan un papel importante en cada actividad y subproceso a realizar en la certificación. Cada actor tiene sus funciones bien especificadas y limitadas a realizar; al ser un gran número de actividades y de documentos de entrada y salida a realizar, cada personaje debe de realizar únicamente sus funciones, acordes al puesto que juegue en esta certificación. Si un actor quiere realizar las actividades y funciones de dos actores diferentes en el proceso de CMMI, se verá afectado en la carga de trabajo, verá reducido los tiempos de entrega, y la calidad disminuirá drásticamente, lo cual a la larga traerá en el futuro próximo del proyecto que se realizará en cuestión.

Esto pasó en el CDS, al tener poco personal, no se pudo tener un actor distinto para cada puesto, algunos actores debieron de jugar con dos o más roles necesarios para la certificación CMMI, lo cual vio afectada su carga de trabajo; ya que este personal no solo se encarga de llevar a buen puerto los proyectos y el CDS, sino que también, tienen funciones de profesores y de investigadores, lo que terminó por desgastarlos, además de tener muy poco tiempo para realizar sus funciones de profesores y/o investigadores.

Por este problema se optó por relajar la carga de trabajo, para esto se ocupó de alumnos que están cursando el Servicio Social o las Prácticas Profesionales para que ayuden con actividades puntuales a realizar en el CDS.

- El segundo problema que se encontró fue la reservación de citas para los clientes y toma de requerimientos.

La forma en que trabaja el CDS por medio de la certificación CMMI, hacía mucho hincapié en reuniones constantes con el cliente, esto con el fin de ver las pretensiones del cliente con el proyecto, revisar el plan de trabajo, el presupuesto con el que contaba, analizar los recursos con los que contaba el CDS, analizar los periféricos y softwares con los que cuenta el cliente para llegar a buen puerto.

Una vez aceptado el proyecto, se seguían las reuniones constantes con el cliente para dar seguimiento a los requerimientos tomados y al proyecto en sí, esto ocasionaba que el personal de RH (Recursos Humanos), se viera apretado de tiempo, ya que como se mencionó antes, tenían diferentes actividades en el plantel.

- El tercer problema que se encontró en el centro de desarrollo de software fue la falta de tiempo para continuar renovando la certificación CMMI y la falta de apoyo económico para lo mismo.

Es bien sabido que cuando tienes en mente tener una certificación debes tener mucho tiempo para lograr esto, el CDS puso todo el tiempo con el que disponían para lograr la certificación CMMI, y lo lograron, pero esta certificación tiene que irse renovando en periodos de tiempo corto, y una vez que se llegó una de tantas renovaciones, se optó por no renovarlas más, debido a lo apretado de la agenda que se tenía en RH, Las horas de clase, el tiempo para la investigación y los proyectos que se tenían en marcha, hicieron casi imposible lograrlo. El dinero jugo un papel importante en la renovación de esta certificación, ya que una certificación como esta es muy costosa y el CDS no es un ente que genere ganancias por sí solo, sino que depende de una institución, esta debe de proveer el dinero y los recursos a todos los departamentos, dificultando así la gestión para obtener los recursos necesarios para la renovación de esta.

- Perdida de la certificación CMMI y división en la forma de trabajar del CDS.

Con la perdida de la certificación CMMI, el centro de desarrollo de software busco una nueva forma de trabajo para sus proyectos. Las buenas prácticas que dejo esta certificación se siguieron usando por un lapso de tiempo, pero a la larga decidieron cambiar la forma de trabajo, dividiendo el CDS en dos grupos, los cuales son liderados por dos personas diferentes. Cada grupo tiene dos ramas diferentes dentro del CDS, pero a su vez comparten los mismos proyectos en sí.

Cuando se tiene un proyecto nuevo en puerta, se juntan todos los miembros del CDS, a escuchar la propuesta, ahí se toma la decisión a cual grupo de trabajo pertenece si al equipo de investigación, o al equipo de desarrollo, una vez elegido el grupo que tomara las riendas del proyecto, el líder de dicho grupo, sigue con la forma de trabajo que se implementó en el grupo y asigna las tareas y funciones a los miembros del equipo, una vez asignadas las tareas, cada miembro del equipo opta por la mejor forma de llevarlas a cabo, es decir dentro del equipo, se crea una nueva forma de trabajo en cada miembro.

Lo que ocasiona que no se tenga una metodología a seguir en la nueva forma de trabajo del CDS, lo que hace que cada uno trabaje a su manera. Una vez cumplidas sus funciones se entrega al líder del equipo para la revisión de estas y en conjunto se unen todas las piezas para dar por terminado el proyecto.

En pocas palabras la pérdida de la certificación CMMI ocasiono, que no se tenga una metodología única dentro del CDS, ya que cada uno de los miembros, trabaja con la metodología que le parezca adecuada a sus funciones y tareas, lo cual nos hace encontrar el problema principal que se explicara a continuación.



## Detectar Problema

No contar con una metodología única, que funja como base para el desarrollo de proyectos dentro del CDS, que se acople a las necesidades de sus miembros, tanto económicas, como de tiempo y de personal.

Conocer los efectos provocados por el problema central.

Los efectos colaterales, provocados por el problema central son:

- las diferentes formas de trabajo: el CDS tiene dos diferentes grupos de trabajo, cada uno de ellos tiene una forma única de cumplir con sus metas, sin importar la forma o la calidad del producto final, ellos siguen una metodología de software adecuada a sus fortalezas como a sus debilidades, lo cual hace que existan diferentes metodologías y modelos de programación en el CDS.
- diferentes tipos de recursos para el personal: cada actor que participa en los proyectos del CDS tiene sus actividades propias, esto ocasiona que cada uno de ellos tengan recursos limitados de manera individual, con los recursos que se les otorga ya sean económicos o de desarrollo deben de cumplir con las metas que se le exige, lo que crea problemas al momento de realizarlo, ya que cada uno tiene formas de trabajar y conseguir sus metas de diferentes maneras.
- no tener una única forma estandarizada de trabajo: el no tener una manera estandarizada de trabajo, cualquier actor puede seguir su trabajo de la manera que desee, lo que hace que existan diferentes formas de trabajo en un solo proyecto, lo que afecta el resultado final, ya que, al momento de juntar todas las piezas, se encuentra ramificaciones que no encajan del todo bien al momento de finalizar el proyecto.
- la división del CDS: hace que cada uno trabaje por su cuenta y al ritmo que ellos crean adecuados según sus habilidades, estos pueden seguir o no una metodología establecida en los libros, o crear sus propias metodologías.
- la falta de calidad en sus trabajos finales: la calidad del producto depende de las actividades realizadas por cada integrante del proyecto, cada ente al momento de trabajar de forma individual pone en cada una de ellas la calidad que para el parece la mejor; medir la calidad es algo trivial y para cada persona la calidad puede ser o no lo que otro persona piensa que lo es, por eso al momento de juntar las piezas del rompecabezas, se pueden notar que algunas de estas piezas tengas mas calidad que otras, y el problemas no es de los integrantes

en sí, sino de la falta de una metodología que asegure la calidad de forma uniforme y única para todos los miembros del equipo de desarrollo.

Bajo esta problemática, se diseñará una propuesta metodológica para la implementación del ISO 29110 en el centro de desarrollo de Software, que servirá para obtener productos de calidad y además para ser considerada como una de las empresas pioneras en el sur del estado de Guanajuato con la capacidad de responder a las demandas de formación de Recursos Humanos altamente calificado para el desarrollo de herramientas tecnológicas de alta calidad.

## solución del Problema

Para solucionar el problema que afecta el CDS del ITSUR es obtener la certificación del ISO/IEC 29110, ya que esta ayudará a generar una única metodología que rija en el CDS en la cual todos sus integrantes se enfoquen en la forma de realizar sus actividades y proceso para generar al final productos de alta calidad y confianza para sus clientes. Para esto se debe de tener un cruce entre la norma ISO/IEC 29110 y la certificación CMMI, los resultados de este cruce se encuentran más adelante en este documento.

### **Mejora continua.**

Para la realización de la mejora continua, se sugiere revisar cada 6 meses después de implementar la metodología formulada aquí para revisar los problemas que puedan surgir al momento de aplicar y de trabajar en esta nueva forma, descubrir el problema nuevo que aqueja el CDS y dar una solución factible para que sigan creando sistemas de productos finales de alta calidad,

## Análisis de la norma ISO

Al principio de este capítulo conocimos las actividades que conforman la certificación CMMI, ahora llego el momento de descubrir las actividades y funciones correspondientes al ISO 29110.

La Norma ISO/IEC 29110 propone en su perfil básico dos procesos de software que deberían estar implementados en todas las VSE: el proceso de gestión de proyectos (PM) y el proceso de implementación del software (SI). A continuación, veremos la ilustración correspondiente al proceso del perfil básico.



*Ilustración 35 Proceso del perfil básico*

En la imagen 28 se pueden observar dos procesos que están a la par.

- **la gestión de proyectos.** La cual tiene el propósito de establecer y llevar a cabo de manera sistemática las tareas del proyecto de implementación del software, lo que permite cumplir con los objetivos del proyecto en la calidad, el tiempo y los costos esperados.
- **Implementación del software.** El objetivo de este proceso es la realización sistemática del análisis, diseño, construcción, integración y pruebas de actividades para los productos de software nuevos o modificados de acuerdo con los requisitos especificados y el plan de ejecución.

Es importante resaltar que tanto el proceso de gestión de proyectos como el de implementación del software están relacionados entre sí de manera bidireccional. (véase la figura 28).

## Proceso de gestión de proyectos (PM)

### Objetivo del proceso de gestión de proyectos

El Objetivo de este proceso es establecer y llevar a cabo de manera sistemática las tareas del proceso de implementación del software, lo que permite cumplir con el alcance del proyecto en calidad, tiempo y costos esperados.

### Requisitos del proceso de gestión de proyectos

Este proceso cuenta con 15 requisitos de gestión de proyectos que se deben cumplir, los cuales se muestran en la tabla 1, estos requisitos permiten asegurar que la implementación del proceso de gestión de proyectos sea exitosa.

*Tabla 1 Requisitos del proceso de gestión de proyectos*

Ítem	Requisitos
A	Se deberá definir el alcance de trabajo para el proyecto, incluido sus entregables
B	Se deberán definir las tareas y los recursos asociados con el alcance del trabajo
C	Se deberán obtener la viabilidad de costo, la viabilidad técnica y el cronograma
D	Se deberán estimar el horario, el esfuerzo, el costo y la duración del trabajo, así como otras métricas, si fuera necesario
E	Se deberá planificar la asignación de recursos humanos
F	Se deberán preparar el plan de ejecución del proyecto según el alcance del trabajo, los recursos humanos planificados y las tareas definidas
G	Se deberá acordar el plan de ejecución con el cliente
H	Se deberán identificar y monitorizar los riesgos durante la ejecución del proyecto
I	Se deberá desarrollar e implementar una estrategia de control de versiones del software, incluyendo la copia de seguridad y la restauración
J	Se deberán identificar y controlar los elementos relevantes de la configuración de software, incluyendo su almacenamiento, línea base, manejo y modificaciones
K	Se deberá monitorizar y registrar el avance del proyecto según la planificación
L	Se deberán tomar medidas para ajustar y corregir desviaciones al plan de ejecución
M	Se deberán realizar las actividades de revisión entre el cliente y el equipo de trabajo para asegurar que el trabajo ha sido realizado y que cumple con los requisitos del software y plan de ejecución
N	Se deberán registrar y seguir los acuerdos resultantes de las actividades de revisión

O	Se deberá realizar el cierre del proyecto después de la aceptación del cliente
---	--

## Gestión de proyectos

La imagen 36 muestra el flujo de información entre las actividades del proceso de gestión de proyectos, incluyendo los productos de trabajo más relevantes y su relación.

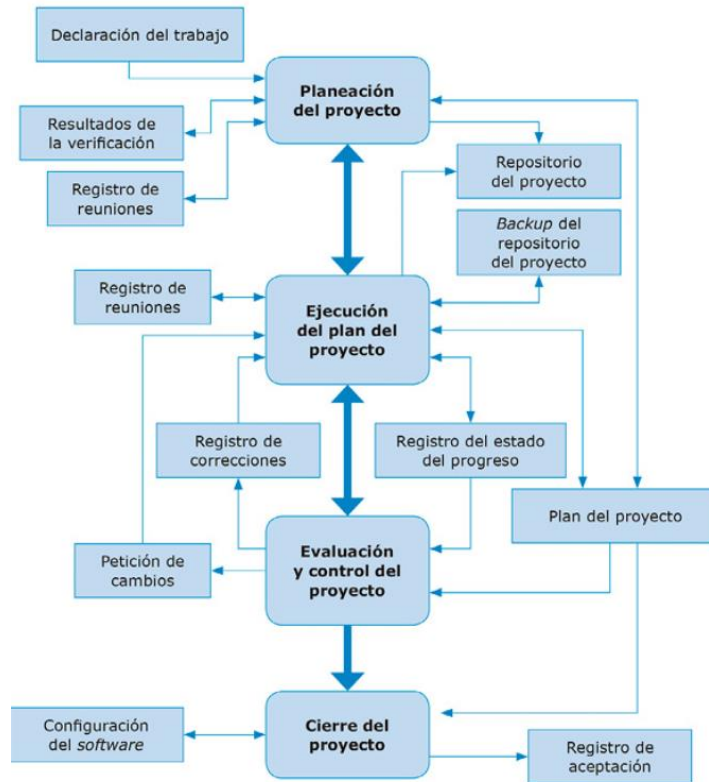


Ilustración 36 Diagrama del proceso de gestión de proyectos

### Tareas del proceso de gestión de proyectos

El proceso de gestión de proyectos (PM) utiliza la declaración de trabajo del cliente para elaborar el plan del proyecto. Las tareas de evaluación y control del proceso de PM comparan el progreso del proyecto con el plan del proyecto y se toman medidas para eliminar las desviaciones o incorporar cambios al plan del proyecto. La actividad de cierre de proyecto PM entrega la configuración de software, producida por SI, y recibe la aceptación del cliente para formalizar el proyecto. Se establece un repositorio proyectos para guardar los productos de trabajo y controlar sus proyectos (ISO/IEC TR 29110-5-1-2:2011)

El proceso de gestión de proyectos establece 4 actividades y 26 tareas asociadas a estas actividades que se deben cumplir, permitiendo garantizar que la implementación del citado proceso de gestión sea exitosa. A continuación, se muestran las actividades junto con su respectiva tabla de distribución de tareas que están asociadas a cada una de ellas, en las que los roles son:

Tabla 2 Roles PM

Roles	
PM	Project Manager (Jefe del Proyecto)
TL	Technical Leader (Responsable Técnico)
CUS	Customer (Cliente)
WT	Work Time (Equipo de Trabajo)
AN	Analyst (Analista)
DES	Designer (Diseñador)
PR	Programmer (Programador)

PM 1: planeación/planificación del proyecto (véase la tabla 3)

Tabla 3 Lista de tareas de PM1

Rol	Lista de tareas	Producto de entrada	Producto de salida
PM TL	PM1.1: Revisar la declaración del trabajo	Declaración del trabajo	declaración del trabajo [Revisada]
PM WT	PM 1.2: Definir con el cliente las instrucciones de entrega de cada uno de los entregables especificados en la declaración de trabajo	Declaración del trabajo [Revisada]	Plan del proyecto Instrucciones de entrega
PM TL	PM 1.3: Identificar las tareas específicas a realizar para producir los entregables y sus componentes de software identificados en la declaración de trabajo junto con la verificación, la validación, las revisiones	declaración del trabajo [Revisada]	Plan del proyecto -Tareas



	<p>del cliente y las tareas de equipo de trabajo para asegurar la calidad de los productos de trabajo.</p> <p>Identificar las tareas para llevar a cabo las instrucciones de entrega.</p> <p>Documentas las tareas</p>		
PM TL	<p>PM: 1.4 Establecer la duración estimada para realizar cada tarea</p>	<p>Plan del proyecto</p> <p>Tareas</p>	<p>Plan de proyecto</p> <p>Duración estimada</p>
PM TL	<p>PM 1.5: Identificar y documentar los recursos: humanos, materiales, equipos, herramientas, estándares, incluyendo la capacitación del equipo de trabajo para llevar a cabo el proyecto. Incluir en el calendario las fechas de cuando se necesitarán recursos y capacitaciones</p>	<p>declaración del Trabajo</p>	<p>Plan de proyecto</p> <p>Recursos</p>
PM TL	<p>PM 1.6: Establecer la composición del equipo de trabajo asignando roles y responsabilidades según los recursos</p>	<p>Plan del proyecto</p> <p>Recursos</p>	<p>Plan del proyecto</p> <p>composición del equipo de trabajo</p>
PM TL	<p>PM 1.7: Asignar fechas estimadas de inicio y de finalización para cada una de las tareas de proyecto</p>	<p>Plan del Proyecto</p> <p>Tareas</p> <p>Duración Estimada</p>	<p>Plan del Proyecto</p> <p>Programación de las tareas del proyecto</p>

	teniendo en cuenta los recursos asignados, la secuencia y la dependencia de las tareas	Composición del equipo de trabajo	
PM	PM 1.8: Calcular y documentar el proyecto en esfuerzo estimado y costo	Plan del proyecto Programación de las tareas del proyecto Recursos	Plan del proyecto Esfuerzo estimado y costo
PM TL	PM 1.9: Identificar y documentar los riesgos que pueden afectar el proyecto	Todos los elementos previamente definidos	Plan del proyecto Identificación de los riesgos del proyecto
PM TL	PM 1.10: Documentar la estrategia de control de versiones en el plan del proyecto		Plan del proyecto Estrategia de control de versiones
PM	PM 1.11: Generar el plan de proyecto de integración de los elementos previamente revisados y documentados.	Todos los elementos previamente definidos	Plan del proyecto *Tareas *Duración estimada *Recursos * composición del equipo de trabajo * programación de las tareas del proyecto *Esfuerzo estimado y costo * Identificación de los riesgos del proyecto *Estrategia de control de versiones

			* Instrucciones de entrega
PM TL	PM 1.12: Incluir descripción del producto, alcance, objetivos y resultados en el plan del proyecto	Declaración del trabajo *Descripción del producto *Alcance *Objetivos *Entregables	Plan del Proyecto *Descripción del producto *Alcance *Objetivos *Entregables
PM TL	PM 1.13: Verificar y obtener la aprobación del plan del proyecto. Verificar que todos los elementos del plan de proyecto sean viables y consistentes. Se documentan los resultados encontrados en una verificación y se lleva a cabo las correcciones hasta que el documento sea aprobado por el PM	Plan del Proyecto	Verificación de resultados *Plan del proyecto [Verificado]
PM CUS	PM 1.14: Revisar y aceptar el plan del proyecto. El cliente revisa y acepta el plan del proyecto, asegurándose de que los elementos del plan de proyecto coincidan con la declaración del trabajo	Plan del Proyecto [Verificado]	Registro de reunión Plan del proyecto [Aceptada]
PM TL	PM 1.15: Establecer el repositorio de proyectos utilizando la estrategia de control de versiones	Estrategia de control de versiones	Repositorio del proyecto

PM 2: Ejecución del plan de proyecto (véase la tabla 4)

Tabla 4 Lista de tareas de PM2

Rol	Lista de Tareas	Producto de entrada	Producto de salida
PM TL WT	PM 2.1: Monitorear la ejecución del plan del proyecto y registro de datos reales en el registro de estado de progreso	Plan de Proyecto	Registro de entrada de progreso
PM TL	PM 2.2: Analizar y evaluar la solicitud de cambio de costo, horario y efectos técnicos. La solicitud de cambio puede iniciarse externa o internamente por el equipo de trabajo. Actualizar el plan del proyecto si el cambio aceptado no afecta los acuerdos con el cliente. La solicitud de cambios que afecta a los acuerdos debe ser registrado por ambas partes (véase PM 2.4)	Solicitud de cambio [Iniciado] Plan del proyecto	Cambio de solicitud [evaluado] Plan del proyecto [actualizado]
PM TL WT	PM 2.3: Realizar reuniones de revisión con el equipo de trabajo, Identificar problemas, revisar estado de riesgo, registros de acuerdos y seguimiento hasta el cierre	Plan de Proyecto, registro de estado de progreso, registro de la corrección. Registro de reunión.	Registro de revisión [actualizado]

PM CUS TL WT	<p>PM 2.4: Conducir reuniones de revisión con el cliente, acuerdos de registro y seguimiento hasta el cierre.</p> <p>Solicitud de cambio iniciada por el cliente o iniciada por el equipo de trabajo, que afecta al cliente, debe ser negociado para llegar a la aceptación de ambas partes si es necesario, actualizar el plan del proyecto según el nuevo acuerdo con el cliente.</p>	<p>Plan del proyecto</p> <p>Registro de estado de progreso</p> <p>Solicitud de cambio [evaluado]</p> <p>Registro de Reunión</p>	<p>Registro de reunión [actualizado]</p> <p>Solicitud de cambio [aceptado]</p> <p>plan del proyecto [actualizado]</p>
PM	<p>PM 2.5: Realizar copia de seguridad según la estrategia de control de versión</p>	<p>Estrategia de control de versiones</p>	<p>Copia de seguridad del repositorio de proyectos</p>
PM	<p>PM 2.6: Realizar la recuperación del repositorio del proyecto utilizado, el repositorio del proyecto de respaldo, si es necesario</p>	<p>Copia de seguridad del repositorio de proyectos</p>	<p>Repositorio del proyecto [Recuperado]</p>

PM 3: Evaluación y control del proyecto (véase la tabla 5)

Tabla 5 Lista de tareas de PM3

Rol	Lista de Tareas	Producto de entrada	Producto de salida
	<p>PM 3.1: Evaluar el progreso del proyecto en relación</p>		

<p>PM TL WT</p>	<p>con el plan del proyecto, comparando: *Tareas reales contra tareas planificados *Resultados reales contra objetivos del proyecto establecidos *Asignación de recursos reales contra recursos previstos *Costo real contra presupuesto *Tiempo real contra tiempo previsto *Riesgo real contra los previamente identificados</p>	<p>Plan del proyecto Registro de estado de progreso</p>	<p>Registro de estado de progreso [evaluados]</p>
<p>PM TL WT</p>	<p>PM 3.2: Establecer acciones para conseguir desviaciones a problemas y riesgos identificados relativos a la realización del plan, según sea necesario, y documentados en el registro de correcciones y seguimiento hasta el cierre.</p>	<p>Registro de estado de progreso [Evaluados]</p>	<p>Corrección de registros</p>
<p>PM TL WT</p>	<p>PM 3.3: Identificar los cambios a requisitos o al plan del proyecto para abordar las principales desviaciones, riesgos potenciales o problemas relativos a la realización del</p>	<p>Registro de estado de progreso [Evaluados]</p>	<p>Solicitud de cambio de iniciada</p>

	plan, documento de solicitud de cambio y seguimiento hasta el cierre		
--	--	--	--

PM 4: Cierre del proyecto (véase la tabla 17)

Tabla 6 Lista de tareas PM 4

Rol	Lista de Tareas	Producto de entrada	Producto de salida
PM CUS	PM 4.1: Formalizar la finalización del proyecto según las instrucciones de entrega establecidos en el plan del proyecto, proporcionando apoyo de aceptación y obteniendo el registro de aceptación firmado	Plan del proyecto *Instrucciones de entrega Configuración del software [Entregado]	Registro de aceptación Configuración del software [Aceptado]
PM	PM 4.2: Actualizar el repositorio del proyecto	Configuración del software [aceptada] Repositorio del proyecto	Repositorio del proyecto [actualizado]

## Implementación del software (SI)

### Objetivo del proceso de implementación del software

El objetivo de este proceso es la realización sistemática del análisis, diseño, construcción, integración y pruebas de actividades para los productos de software nuevos o modificados de acuerdo con los requisitos especificados y el plan de ejecución.

### Requisitos del proceso de implementación del software

Para ello se cuentan con 19 requisitos de implementación del software que se deben cumplir, y que permiten asegurar que el proceso de implementación del software sea exitoso, tal como se muestra en la tabla 7.

Tabla 7 Requisitos del proceso implementación del Software

ítem	Requisitos
a	Se deberán comprender y revisar los requisitos del software y plan de ejecución por el equipo de trabajo
b	Se deberán definir los requisitos del software
c	Se deberán analizar los requisitos del software para su corrección y prueba
d	Se deberán acordar los requisitos del software con el cliente o el patrocinador del proyecto
e	Se deberán establecer y comunicar la línea base para los requisitos del software a los stakeholders (partes interesadas)
f	Se deberán evaluar los cambios en los requisitos software nivel costo, horario y efectos técnicos
g	Se deberá desarrollar la arquitectura de software y el diseño detallado, establecer una línea base y comunicarla a los stakeholders (partes interesadas)
h	Se deberán describir los componentes de software y sus interfaces internas y externas relevantes en la arquitectura software y diseño detallado
i	Se deberán establecer la consistencia y trazabilidad entre los registros software, la arquitectura del software y el diseño detallado software.
j	Se deberán producir los componentes de software definidos para el diseño detallado
k	Se deberá controlar y poner a disposición de los stakeholders la liberación de los artefactos
L	Se deberán realizar las pruebas unitarias para verificar la consistencia con los requisitos del software y el diseño detallado
m	Se deberá establecer la consistencia y trazabilidad entre los componentes de software, los requisitos y el diseño
n	Se deberá elaborar la documentación del usuario
o	Se deberá producir el software mediante la integración de los componentes de software



p	Se deberá probar y verificar el software, registrar los resultados y comunicarlos al equipo de trabajo
q	Se deberán corregir los defectos identificados en las revisiones, pruebas y verificaciones
r	Se deberá integrar y almacenar, en el repositorio del proyecto, la configuración del software; se deberá establecer con línea base final y comunicarla al equipo de trabajo y al cliente
s	Se deberá terminar y entregar el producto para su después de la validación por parte del patrocinador de proyecto o cliente

### Flujo de información del proceso de implementación del Software

La figura muestra el flujo de información entre las actividades del proceso de implementación del software, incluyendo los productos de trabajo más relevantes y su relación.

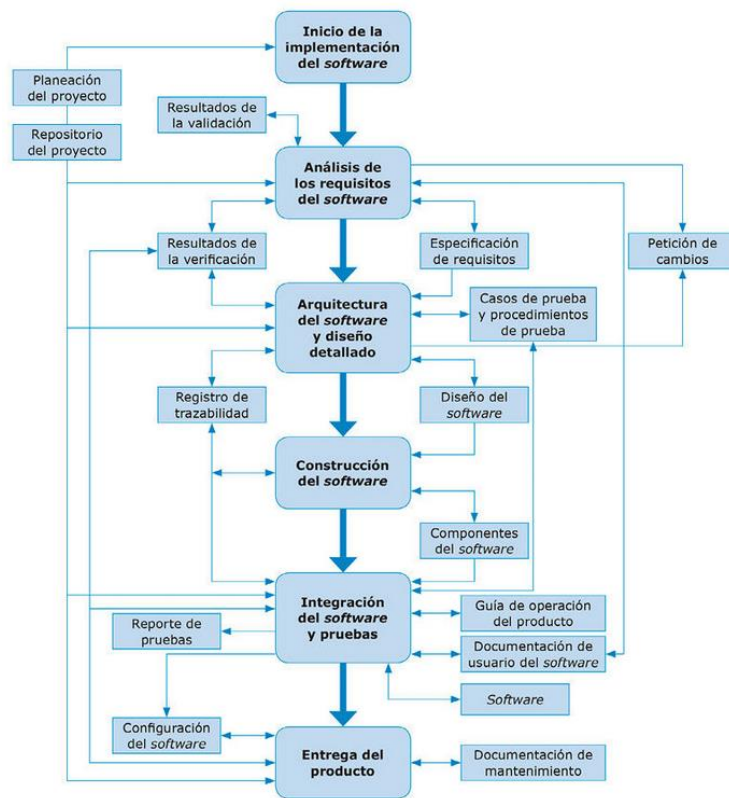


Ilustración 37 Diagrama del proceso de implementación del Software

## Tareas del proceso de implementación del software

El proceso de implementación del software establece 6 actividades y 32 tareas asociadas a estas actividades que se deben cumplir, permitiendo garantizar que la implementación del proceso de implementación del software sea exitosa.

Roles:

*Tabla 8 Roles Implementación del Software (SI)*

PM	Project Manager (Jefe del Proyecto)
TL	Technical Leader (Responsable Técnico)
CUS	Customer (Cliente)
WT	Work Time (Equipo de Trabajo)
AN	Analyst (Analista)
DES	Designer (Diseñador)
PR	Programmer (Programador)

A continuación, se muestran las actividades junto con su respectiva tabla de distribución de tareas, que están asociadas a cada una de ellas.

S1: inicio de la implementación del software (véase la tabla 9)

*Tabla 9 Lista de Tareas SI 1*

Rol	Lista de Tareas	Producto de entrada	Producto de salida
PM TL WT	SI 1.1: Revisar el plan actual de proyecto con los miembros del equipo de trabajo con el fin de alcanzar un entendimiento común y obtener su compromiso con el proyecto	Plan del proyecto	Plan del proyecto [revisado]
TL WT	SI 1.2: Establecer o actualizar el entorno de implementación.	Plan del proyecto [revisado]	

SI 2: Análisis de requisitos de Software (véase la tabla 10)

Tabla 10 Lista de Tareas SI2

Rol	Lista de Tareas	Producto de entrada	Producto de salida
TL WT	SI 2.1: Asignar tareas a los miembros del equipo de trabajo según su rol, basado en el plan del proyecto actual	Plan del proyecto [revisado] *Tareas	
AN CUS	SI 2.2: Documentar o actualizar la especificación de requisitos. Identificar y consultar fuentes de información (Cliente, usuarios, los sistemas anteriores, documentos, etc.) con el fin de obtener nuevos requisitos	Plan del proyecto *Descripción del producto	Especificación de requisitos
AN TL	SI 2.3: Verificar y obtener la aprobación de la especificación de requisitos Verificar la exactitud y la capacidad de prueba de la especificación de requisitos y su coherencia con la descripción del producto. Además, revisar que los requisitos son completos, sin ambigüedades y no contradictorios. Los resultados encontrados se documentan en los	Especificación de requisitos. Plan del proyecto *Descripción del producto	Resultados de la verificación Especificación de requisitos [verificada] Solicitud de cambio [iniciada]

	<p>resultados de la verificación y las correcciones se hacen hasta que el documento es aprobado por la AN. Si se necesitaban cambios importantes, iniciar una solicitud de cambio</p>		
<p>CUS AN</p>	<p>SI 2.4: Validar y obtener la aprobación de la especificación de requisitos. Validar que la especificación de requisitos satisface necesidades y expectativas acordadas incluyendo la usabilidad de la interfaz de usuario. Los resultados encontrados se documentan en los resultados de una validación y se hacen correcciones hasta que el documento sea aprobado por el CUS</p>	<p>Especificación de requisitos. [Verificado]</p>	<p>Resultados de la verificación Especificación de requisitos [validado]</p>
<p>AN</p>	<p>SI 2.5*: Realizar la versión preliminar de la documentación de usuario del software o actualizar el presente manual, si es apropiado *(opcional)</p>	<p>Especificación de requisitos [validada]</p>	<p>*Documentación del usuario del software [preliminar]</p>
<p>AN TL</p>	<p>SI 2.6*: Verificar y obtener la aprobación de la</p>	<p>*Documentación de usuario del software</p>	<p>Resultados de la verificación</p>

	<p>documentación de usuario del software, si es apropiado Verificar la consistencia de la documentación de usuario de software con la especificación de requisitos. Los resultados encontrados se documentan en los resultados de verificación y las correcciones se hacen hasta que el documento es aprobado por la AN. Si se necesitan cambios importantes, se inicia una solicitud de cambio</p> <p>*(opcional)</p>	<p>[preliminar] Especificación de requisitos</p>	<p>documentación del Software de usuario [preliminar, verificado]</p> <p>Solicitud de cambio iniciada</p>
TL	<p>SI 2.7: Incorporar la especificación de requisitos y la documentación* del software de usuario a la configuración de software en la línea base *(opcional)</p>	<p>Especificación de requisitos [validada]</p> <p>*Documentación del software de usuario [preliminar, verificado]</p>	<p>Configuración del software</p> <p>*Especificación de requisitos [validada, línea base]</p> <p>*Documentación de usuario software [preliminar, verificado, línea base]</p>

SI 3: Diseño detallado y arquitectura del Software (véase la tabla 11)

Tabla 11 Lista de Tareas de SI 3

Rol	Lista de Tareas	Producto de entrada	Producto de salida
-----	-----------------	---------------------	--------------------

TL AN DES	SI 3.1: Asignar tareas a los miembros del equipo de trabajo relacionadas con su función, según el plan del proyecto actual	Plan de proyecto *Tareas	
AN DES	SI 3.2: Entender la especificación de requisitos	Especificación de requisitos [validado, en línea base]	
AN DES	SI 3.3: Documentar o actualizar el diseño del software Analizar la especificación de requisitos para generar el diseño arquitectónico, su arreglo en subsistemas y componentes de software que definen las interfaces internas y externas Describir en detalle la apariencia y el comportamiento de la interfaz, basado en la especificación de requisitos, de tal manera que se puedan prever los recursos para su aplicación Proporcionar el detalle de los componentes de software y sus interfaces para permitir facilitar su construcción Generar o actualizar el registro de trazabilidad	Especificación de requisitos [validado, en línea base]	Diseño de Software Registro de trazabilidad

AN DES	<p>SI 3.4: Verificar y obtener la aprobación del diseño software.</p> <p>Verificar la precisión de la documentación del diseño de software, su viabilidad y consistencia con la especificación de requisitos. Verificar que el registro de trazabilidad contenga las relaciones adecuadas entre los requisitos y los elementos de diseño de software. Los resultados encontrados se documentan en los resultados de verificación y las correcciones se hacen hasta que el documento es aprobado por el DESS. Si se necesitan cambios importantes, iniciar una solicitud de cambio</p>	<p>Diseño de software. Registro de trazabilidad. Especificación de requisitos [validado, en línea base]</p>	<p>Resultados de la verificación Diseño de software [verificado] Registro de trazabilidad [verificado] Cambio de solicitud [iniciado]</p>
DES	<p>SI 3.5: Establecer o actualizar procedimientos de prueba y casos de prueba, para pruebas de integración, basados en la especificación de requisitos y diseño del software.</p> <p>El cliente proporciona datos de prueba, si es necesario</p>	<p>Especificación de requisitos [validado, en línea base] Diseño de software [verificado, en línea base]</p>	<p>Casos de prueba y procedimientos de prueba</p>

DES AN	<p>SI 3.6: Verificar y obtener la aprobación de los procedimientos de prueba y casos de prueba</p> <p>Verificar la consistencia entre la especificación de requisitos, el diseño de software y los casos de prueba. Los resultados encontrados se documentan en los resultados de verificación y las correcciones se hacen hasta que el documento es aprobado por el AN</p>	<p>casos de prueba y procedimientos de prueba.</p> <p>Especificación de requisitos [validada en línea base]</p> <p>Diseño de software [verificado, en línea base]</p>	<p>Casos de prueba de verificación de resultados y procedimientos de prueba [verificado]</p>
DES	<p>SI. 3.7: Actualizar el registro de trazabilidad incorporando los casos de prueba y procedimientos de prueba</p>	<p>Casos de prueba y procedimientos de prueba [verificado]</p> <p>Registro de trazabilidad [actualizado]</p>	<p>Registro de trazabilidad [actualizado]</p>
TL	<p>SI. 3.8: Incorporar el diseño software y el registro de trazabilidad para la configuración del software como parte de la línea base</p> <p>Incorporar los casos de prueba y procedimientos de prueba en el repositorio del proyecto</p>	<p>Diseño de software [verificado]</p> <p>Casos de prueba y procedimientos de prueba [verificado]</p> <p>Registro de trazabilidad [verificado]</p>	<p>Configuración del software</p> <p>*Software de diseño [verificado, línea base]</p> <p>*Casos de prueba y procedimientos de prueba [comprobado]</p> <p>*Registro de trazabilidad [Verificado, en línea base]</p>

SI 4: Construcción del Software (véase la tabla 12)



Tabla 12 Lista de tareas SI 4

Rol	Lista de Tareas	Producto de entrada	Producto de salida
TL PR	SI 4.1: Asignar tareas a los miembros del equipo de trabajo relacionadas con su función según el plan del proyecto actual	Plan del proyecto *Tareas	
PR	SI 4.2: entender el diseño de software	Diseño de software [verificado, en línea base]	
PR	SI 4.3: Construir o actualizar los componentes de software basados en la parte detallada del diseño de software	Diseño de software [verificado, en línea base] Registro de trazabilidad [verificado, en línea base]	Componentes de Software
PR	SI 4.4: Diseñar o actualizar casos de prueba de unidad y aplicarlos para verificar que los componentes de software implementan la parte detallada del diseño de software	Componentes de software	Componentes de Software [unidad probada]
PR	SI 4.5: Corregir los defectos encontrados hasta lograr una prueba exitosa de la unidad (alcanzando los criterios de salida)	Componentes de software [unidad aprobada]	Componentes de software [corregidos]
PR	SI 4.6: Actualizar el registro de trazabilidad incorporando los componentes de software construidos o modificados	Componentes de software [corregidos] Registro de trazabilidad [verificado, en línea base]	Registro de trazabilidad [actualizado]

PR	SI 4.7: Incorporar componentes de software y el registro de trazabilidad a la configuración del software construidos o modificados	Componentes de software [corregidos] Registro de trazabilidad [actualizado]	Configuración del software *Componentes software [corregidos, en línea base] *Registro de trazabilidad [actualizado en línea base]
----	--	--	--

SI: Pruebas e integración del Software (véase la tabla 13)

Tabla 13 Lista de tareas SI 5

Rol	Lista de Tareas	Producto de entrada	Producto de salida
TL PR	SI 5.1: asignar tareas a los miembros del equipo de trabajo relacionadas con su función según el plan del proyecto actual	Plan del proyecto *Tareas	
PR	SI 5.2: Integrar el software usando los componentes del software y actualizar los casos de prueba y procedimientos de prueba para las pruebas de integración, según sea necesario	Casos de prueba y procedimientos de prueba [verificado]	
PR	SI 5.3: Integrar el software usando los componentes del software y actualizar los casos de prueba y procedimientos de prueba para las pruebas de integración, según sea necesario	Componentes del software [corregidos, en línea base] Casos de prueba y procedimientos de prueba [verificado] Registro de trazabilidad [actualizado, en línea base]	Casos de prueba de software y métodos de prueba
PR CUS	SI 5.4: Realizar pruebas del software utilizando los casos de prueba y procedimientos de prueba para su integración y documentar los resultados en un informe de pruebas	Software Casos de prueba y procedimientos de prueba	Software [probado] Informe de prueba

PR	SI 5.5: Corregir los defectos encontrados y realizar pruebas de regresión hasta que se satisfagan los criterios de salida	Software [Probado] Casos de prueba, informe de prueba y procedimientos de prueba. Registro de trazabilidad [actualizado, en línea base]	Software [corregido] Informe de prueba [defectos eliminados]
PR	SI 5.6: Actualizar el registro de trazabilidad, si es necesario	Software [corregido] Registro de trazabilidad [actualizado, en línea Base]	Registro de Trazabilidad [actualizado]
PR	SI 5.7*: Documentar la guía de operación del producto o actualizar la guía actual, si es apropiado *(Opcional)	Software [probado]	*Guía de operación del producto
PR DES	SI 5.8: Verificar y obtener la aprobación de la guía de operación del producto*, si es apropiado (véase SI 5.7) Verificar la consistencia de la guía de operación del producto con el software. Los resultados encontrados se documentan en los resultados de verificación y las correcciones se hacen hasta que el documento es aprobado por el DES *(Opcional)	*Guía de operación del producto Software [aprobado]	Resultados de la verificación *Guía de operación del producto [verificado]
AN	SI 5.9*: Realizar la documentación de usuario del software o actualizar la	Software [probado] Documentación de usuario	Documentación de usuario del software

	<p>existente, si es apropiado *(opcional)</p>	<p>del software [preliminar]</p>	
<p>AN CUS</p>	<p>SI 5.10: Verificar y obtener la aprobación de la *documentación de usuario del software, si es apropiado (véase SI 5.9) Verificar la consistencia de la documentación del usuario del software con el software. Los resultados encontrados se documentan en los resultados de verificación y las correcciones se hacen hasta que el documento es aprobado por el CUS *(Opcional)</p>	<p>*Documentación del software de usuario Software [aprobado]</p>	<p>Resultados de la verificación *Documentación de usuario del software [verificado]</p>
<p>TL</p>	<p>SI 5.11: Incorporar los casos de prueba y procedimientos de prueba, software, registro de trazabilidad, informe de pruebas, *guía de operación del producto y *documentación de usuario del software a la configuración del software como parte de la línea de base *(Opcional)</p>	<p>Casos de prueba y procedimientos de prueba [probado] Prueba de informe de registro de trazabilidad [actualizado] Guía de operación del producto [verificado] Documentación de usuario del software [verificado]</p>	<p>Configuración del software *casos de prueba y procedimientos de prueba [en línea base] *Software [probado, en línea base] *Registro de trazabilidad [actualizado en línea base] *Informe de prueba [en línea base] *Guía de funcionamiento de producto [verificado, en línea base] *Documentación de</p>

			usuario software [verificado, en línea base]
--	--	--	---

SI 6: entrega del Producto (véase la tabla 14)

Tabla 14 Lista de tareas de SI 6

Rol	Lista de Tareas	Producto de entrada	Producto de salida
TL WT	SI 6.1: asignar tareas a los miembros del equipo de trabajo relacionadas con su función según el plan del proyecto actual	Plan del proyecto *Tareas	
DES	SI 6.2: entender la configuración del software	Configuración del software	
DES	SI 6.3: realizar la documentación de mantenimiento o actualizar la existente	Configuración del software	Documentación de mantenimiento
DES TL	SI 6.4: Verificar y obtener la aprobación de la documentación de mantenimiento Verificar la consistencia de la documentación de mantenimiento con la configuración de software. Los resultados encontrados se documentan en los resultados de verificación y las correcciones se hacen	Documentación de mantenimiento Configuración del software	Mantenimiento de resultados de verificación Documentación [verificado]

	hasta que el documento es aprobado por el TL		
TL	SI 6.5: incorporar la documentación de mantenimiento como parte de la línea base a la configuración de software	Configuración del software Documentación de mantenimiento [verificado]	Configuración del software *Mantenimiento documentación [verificado, en línea base]
TL	SI 6.6: realizar la entrega según las instrucciones de entrega	Plan de proyecto *Instrucciones de entrega Configuración del software	Configuración del software [entregado]

## Mapeo entre la norma ISO/IEC 29110 y CMMI

### Análisis

Para apoyar el cumplimiento de las VSE con los requisitos mencionados en la Norma ISO/IEC 29110-4, relacionamos dichos requisitos con las actividades propuestas por la parte ISO/IEC TR 29110-5-1-2. De esta manera, proponemos a las pequeñas empresas desarrolladoras de software un conjunto de actividades y productos de trabajo que les permitan cumplir los requisitos establecidos por la Norma ISO/IEC 29110-4-1, implantando un enfoque orientado a procesos para llevar a cabo el proceso de certificación.

El proceso de mapeo entre las partes 4-1 y 5-1-2 de la norma se hará realizando las siguientes actividades (Pino, 2010):

- Análisis de los modelos.
- Diseño del mapeo.
- Refinamiento y validación del mapeo.

### Análisis de los modelos

Como vimos, la Norma ISO/IEC 29110-4-1 contiene los requisitos necesarios para cumplir con los procesos de gestión de proyectos (PM) y con la implementación del software (SI); es decir describe en términos de requisitos cómo llevar a cabo gestión de proyectos e implementación del software.

Por su parte, la Norma ISO/IEC TR 29110-5-2 describe un conjunto de actividades, tareas, roles y productos de trabajo tanto del proceso de gestión de proyectos como del proceso de implementación del software.

Analizando la descripción tanto de requisitos como de actividades descritas por estas partes del modelo, cada uno de los requisitos de la ISO/IEC 29110-4-1 debe ser contrastado con el conjunto de tareas establecidas en el estándar ISO/IEC 29110-5-1-2 referentes a los procesos mencionados.



## Requerimientos

Esta actividad se desarrolla mediante la definición de una plantilla que muestra en el eje horizontal la definición de los requisitos de procesos especificados en la Norma ISO/IEC 29110-4-1 y, en el eje vertical, el conjunto relacionado de tareas con sus respectivos roles y productos de trabajo especificados en la Norma ISO/IEC TR 29110-5-1-2. Siguiendo el proceso descrito en la Norma ISO/IEC TR 29110-3-1, se lleva a cabo la relación entre estos dos modelos teniendo en cuenta el análisis de los estándares, que dan como resultado el esquema que se muestra en la tabla 15.

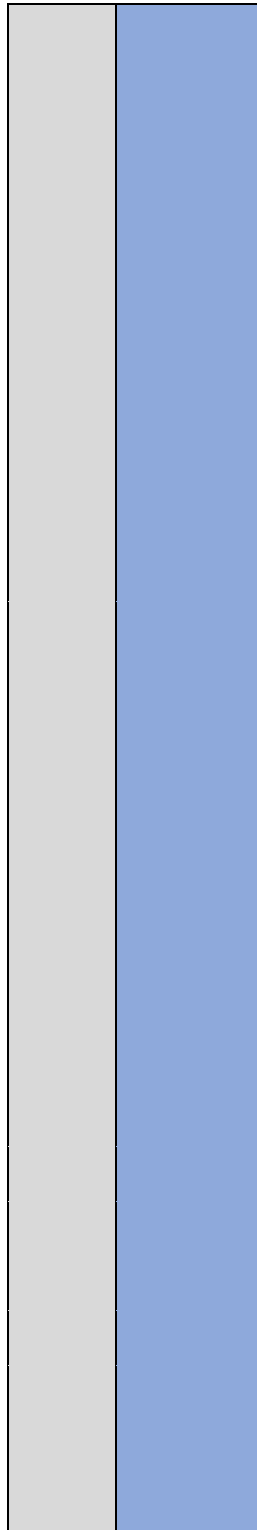
Tabla 15 Mapeo entre ISO/IEC 29110-4-1 e ISO TR 29110-5-1-2

			ISO/IEC TR 29110-5-1-2			
			Tarea	Producto de trabajo	Rol	
ISO/IEC 29110- 4-1	Requisitos de proceso	Proceso de gestión de proyectos (PM)	a	PM 1.1	Declaración del trabajo	PM
				PM 1.2	Declaración del trabajo [revisado]	TL
				PM 1.12	Declaración del trabajo (alcance, objetivos, entregables)	Cus
			b	PM 1.3	Plan del proyecto	PM, TL
			c	PM 1.4	Plan del proyecto (duración estimada)	PM, TL
			d	PM 1.5	Plan del proyecto (recursos)	PM, TL
				PM 1.8	Plan del proyecto (costos y esfuerzo estimados)	
			e	PM 1.6	Plan del proyecto (composición del equipo de trabajo)	PM, TL
			f	PM 1.7	Plan del proyecto (cronograma de tareas)	PM, TL
			g	PM 1.14	Registro de reuniones, plan del proyecto [aceptado]	PM, CUS
	h	PM 1.9	Plan del proyecto (identificación de riesgos del proyecto)	PM		

			i	PM 1.10	Plan del proyecto (estrategia de control de versiones)	PM, TL
				PM 1.15	Repositorio del proyecto	
				PM 2.5	Copia de seguridad del repositorio del proyecto	
				PM 2.6	Repositorio del proyecto [recuperado]	
				PM 4.2	Repositorio del proyecto [actualizado]	
			j	PM 1.11	Plan del proyecto (artefactos)	PM
			k	PM 2.1	Registro del estado del progreso	PM, TL, WT
				PM 3.1	Registro del estado del progreso [actualizado]	
				PM 3.2	Registro de correcciones	
			ISO/IEC 29110-4-1	Requisitos de proceso	Proceso de gestión de proyectos (PM)	l
	PM 2.2	Registro de cambios [evaluado], plan del proyecto [actualizado]				WT
	PM 2.3	Registro de reuniones				
	PM 3.2	Registro de correcciones				
	PM 3.3	Petición de cambios [iniciada]				
m	PM 1.14	Plan del proyecto [aceptado]				PM, CUS
	PM 2.4	Registro de reuniones [actualizado], petición de cambios [aceptado], plan del proyecto [actualizado]				TL, WT
n	PM 1.14	Plan del proyecto [aceptado]				PM, CUS
	PM 2.4	Registro de reuniones [actualizado], petición de cambios [aceptado], plan del proyecto [actualizado]				TL, WT

		o	PM 1.14	Plan del proyecto [aceptado]	PM, CUS
			PM 2.4	Registro de reuniones [actualizado], petición de cambios [aceptado], plan del proyecto [actualizado]	TL, WT
			PM 4.1	Registro de aceptación, configuración del software [aceptado]	
	Proceso de implementación del Software (SI)	a	SI 1.1	Plan del proyecto [revisado]	PM, TL, WT
			SI 3.2	Especificación de requisitos [validado, en línea base]	
		b	SI 2.1	Plan del proyecto [revisado]	TL, WT
			SI 3.1	Plan del proyecto [revisado] -tarea	AN, DES
			SI 4.1	Plan del proyecto [revisado] -tarea	PR
			SI 5.1	Plan del proyecto [revisado] -tarea	
			SI 6.1	Plan del proyecto [revisado] -tarea	
		c	SI 2.2	Especificación de requisitos	AN, CUS
			SI 2.3	Resultados de verificación, especificación de requisitos [verificado], petición de cambios [iniciado]	
		d	SI 2.4	Resultados de validación, especificación de requisitos [validado]	AN, CUS
		e	SI 2.7	La línea base para los requisitos del software deberá ser establecida y comunicada a las partes afectadas	TL

			f	PM 3.3	Los cambios en los requisitos del software deberán ser evaluados a nivel de costo, horario y efectos técnicos	PM, TL, WT
			g	SI 3.3	La arquitectura de software y el diseño detallado deberá ser desarrollado, una línea base será establecida y comunicada a las partes afectadas	AN
				SI 3.8		DES
				SI 4.2		TL PR
			h	SI 3.3	Diseño del software, registro de trazabilidad	AN, DES
			i	SI 3.4	Resultados de verificación diseño del software [verificado], registro de trazabilidad [verificado], petición de cambios [iniciado]	AN, DES
				SI 3.7	Registro de trazabilidad [actualizado]	PR
				SI 5.6	Registro de trazabilidad [actualizado]	PR
				SI 4.2	Componentes de software	PR
				SI 4.3	Componentes de software	
ISO/IEC 29110- 4-2	Requisitos de proceso	Proceso de implementación del Software (SI)	k	SI 4.6	La liberación de los artefactos deberá ser controlada y puesta a disposición de los interesados	PR
			l	SI 3.5	Casos de prueba y procedimientos de prueba	DES
				SI 3.6	Resultados de verificación, casos de prueba y procedimientos de prueba [verificado]	AN
				SI 4.4	Componentes de software [unidades probadas]	
				SI 5.2		



m	SI 3.4	Resultados de verificación, diseño del software [verificado], registro de trazabilidad [verificado], petición de cambios [iniciada]	AN, DES
	SI 3.7	Registro de trazabilidad [actualizado]	
n	SI 2.5	Documentación de usuario software [preliminar]	AN
	SI 2.6	Documentación de usuario software [preliminar, verificado], petición de cambios [iniciado]	
	SI 5.7	Guía de operación del producto	PR
	SI 5.8	Resultados de verificación, guía de operaciones del producto [verificado]	DES
	SI 5.9	Documentación de usuario Software	CUS
	SI 5.10	Documentación de usuario Software [verificado]	
o	SI 5.3	Software, casos de prueba y procedimientos de prueba	PR
p	SI 5.3	Software, casos de prueba y procedimientos de prueba	PR
	SI 5.4	Software probado, reporte de pruebas	CUS
q	SI 4.5	Componentes de software [corregidos]	PR
	SI 5.5	Software [corregido], reporte de pruebas [defectos eliminados]	
r	SI 3.8	Configuración del software	TL
	SI 4.7	Configuración del software	
	SI 5.11	Configuración del software	
s	SI 6.6	Configuración del software [entregado]	TL

## Identificar las funciones de cumplimiento y no cumplimiento

### Evaluación del ISO/IEC 29110-4-1

En este apartado se propone un esquema de evaluación para que las VSE puedan examinar el cumplimiento con los requisitos planteados en la Norma ISO/IEC 29110-4-1.

Para ello, se propone relacionar los requisitos definidos en la Norma ISO/IEC 29110-4-1 con las tareas y elementos definidos en la Norma ISO/IEC TR 29110-5-1-2 a manera de plantilla de apoyo a la evaluación para los procesos de gestión de proyectos (PM) e implementación del software (SI) (véase las tablas 16 y 17).

Esta plantilla de apoyo a la evaluación es un documento que contiene columnas que permiten al personal encargado de realizar la evaluación marcar los requisitos satisfechos por la VSE con base en los indicadores de cumplimiento de requisitos, representados por las tareas y evidencias definidas en los modelos. En la tabla 17 contiene las convenciones usadas para las evidencias del esquema de evaluación.

*Tabla 16 Convenciones para la plantilla de apoyo a la evaluación*

Acrónimo	Significado
PS	Producto de Salida
D	Artefacto Directo
I	Artefacto Indirecto

*Tabla 17 Plantilla de apoyo a la evaluación ISO/IEC 29110-4-1 para el proceso de PM*

ISO/IEC 29110-4-1	ISO/IEC TR 29110-5-1-2	Evidencia			
Gestión de proyectos PM	Actividad	Tareas	PS	D	I
a	PM 1	PM 1.1	Declaración del trabajo	Documento que describe el trabajo que se va a realizar relacionado con la implementación del software	Actas de reuniones para la elaboración de la oferta inicial del proyecto
		PM 1.2		Documento que especifica los elementos y componentes sw	Actas de reunión o correos que evidencian los componentes que se van a

b	PM 1		Plan de proyecto: definir instrucciones de entrega	que se van a entregar al cliente	entregar al cliente; Acta de reunión o correo con la aceptación de los acuerdos
		PM 1.3	Plan de proyecto: identificar tareas	Documento o procedimiento donde se identifican las tareas a llevar a cabo en el proyecto	No asignado
c	PM 1	PM 1.5	Plan de proyecto: identificar y documentar recursos	Documento o procedimiento donde se identifican los recursos para cumplir con el proyecto	No asignado
d	PM 1	PM 1.4	Plan de proyecto: establecer la duración estimada (tareas)	Documento o procedimiento donde se establece la duración estimada de las tareas	No asignado
		PM 1.5	Plan de proyecto: identificar y documentar recursos	Documento o procedimiento donde se identifican los recursos para cumplir con el proyecto	No asignado
		PM 1.7	Plan de proyecto: asignar cronograma para las tareas	Procedimiento de estimación de tamaño y esfuerzo de las tareas teniendo en cuenta los recursos con los que se cuenta Diagrama de Gantt en el que se describe la duración de las tareas	Horas imputadas a la estimación del tamaño y esfuerzo de las tareas, así como la duración
		PM 1.8	Plan de proyecto: calcular y documentar costos y esfuerzo estimado	Documento donde se establece el costo y esfuerzo estimado para la realización del proyecto	
e	PM 1	PM 1.6	Plan de proyecto: establecer la composición del equipo de trabajo	Listado de los participantes en el proyecto que poseen habilidades requeridas por el proyecto	Comunicación a las partes interesadas de los participantes en el proyecto (correos, wiki, portal, etc.)
f	PM 1	PM 1.11	Plan de proyecto: definir tareas, duración estimada, recursos, composición del equipo, cronograma de tareas, costo y esfuerzo, identificación de	Documento, carpeta o registro de almacenamiento donde se consolidan todos los elementos realizados para el plan de proyecto	No asignado

			riesgos, estrategia de control de versiones, instrucciones de entrega		
		PM 1.12	Plan de proyecto: incluir la descripción del producto, alcance, objetivos, entregables	Documento completo, carpeta o registro de almacenamiento del plan de proyecto que incluye todos los elementos mencionados en la etapa de planificación	No asignado
g	PM 1	PM 1.14	Registro de reuniones plan de proyecto [aceptado]	Registro de los acuerdos establecidos con el cliente y/o equipo de trabajo	Comunicación a las partes interesadas de los participantes en el proyecto (correos, wiki, portal, etc.) Acta de reunión o correo con la aceptación de los acuerdos
h	PM 1	PM 1.9	Plan de proyecto: identificación de riesgos	oferta o plan de proyecto donde se identifican los riesgos e inconvenientes posibles del proyecto	Horas imputadas en identificación y definición de riesgos
i	PM 1	PM 1.10	Plan de proyecto: documentar la estrategia de control de versiones	Documento que define la estrategia para el control de versiones, copias de seguridad y mecanismos de recuperación de información	Comunicación con los miembros del equipo de las estrategias de control de versiones (correos, wiki, portal, etc.)
		PM 1.15	Repositorio de proyecto	Establecimiento de un contenedor electrónico para almacenar los productos de trabajo y entregables del proyecto	No asignado
	PM 2	PM 2.5	Back up del repositorio de proyecto	repositorio usado para hacer copias de seguridad del repositorio del proyecto	No asignado
		PM 2.6	Repositorio de proyecto [recuperado]	Registro de recuperación del repositorio del proyecto	Comunicación a las partes interesadas sobre actividades de recuperación de información
	PM 4	PM 4.2	Repositorio de proyecto [actualizado]	Registro de actualización del repositorio del proyecto	comunicación a las partes interesadas sobre actividades de actualización de información
j	PM 3	PM 3.3	Petición de cambios [inicial]	Documento que identifica problemas de software y/o documentación, o deseos de mejora y peticiones de cambio	Comunicación con los miembros del equipo de las peticiones de cambio realizadas (correos, wiki, portal, etc.)



	PM 4	PM 4.2	Repositorio de proyecto [actualizado]	Registro de actualización del repositorio del proyecto	Comunicación con los miembros del equipo de las peticiones de cambio realizadas (correos, wiki, portal, etc.)
k	PM 2	PM 2.1	Registro de estado de progreso	Definición de controles para evaluar el avance del proyecto informes con el grado de avance del proyecto	Horas imputadas para la elaboración de controles y realización de informes con el grado de avance del proyecto
	PM 3	PM 3.1	Registro de estado de progreso [evaluado]	Registro de la evaluación del estado de progreso	Comunicación con los miembros del equipo de los resultados de evaluación del estado de progreso (correos, wiki, portal, etc.)
l	PM 1	PM 1.13	Verificación de resultados Plan de proyecto [verificado]	Documento que contiene los resultados de la verificación	Comunicación con los miembros del equipo de los resultados de verificación del estado de progreso (correos, wiki, portal, etc.)
	PM 2	PM 2.2	Petición de cambios [evaluada] Plan de proyecto [actualizado]	Resultados de evaluación de la petición de cambios Registro de actualización del plan de proyecto	Comunicación con los miembros del equipo y cliente de los resultados de evaluación de petición de cambios (correos, wiki, portal, etc.)
	PM 3	PM 3.2	Registro de correcciones	Identificar las actividades establecidas para corregir desviaciones o problemas que afecten el cumplimiento del plan Resultado de análisis y tratamiento de desviaciones	Horas imputadas para el análisis y tratamiento de desviaciones Comunicación a las partes interesadas del tratamiento de desviaciones y corrección (correo electrónico, actas, etc.)
		PM 3.3	Petición de cambios [inicial]	Documento que identifica problemas software y/o documentación, o deseos de mejora y peticiones de cambio	Comunicación con los miembros del equipo de las peticiones de cambio realizadas (correos, wiki, portal, etc.)
m	PM 2	PM 2.4	Registro de reuniones [actualizado] Petición de cambios [aceptada] Registro de reuniones	Registro de reuniones actualizado, Registro de aceptación de la petición de cambios	Comunicación con los miembros del equipo de las peticiones de cambio aceptadas y actividades de actualización de información (correos, wiki, portal, etc.)
n	PM 2	PM 2.2	Petición de cambios [evaluada] Plan de proyecto [actualizado]	Registro de evaluación de las peticiones de cambio Registro de actualización del plan de proyecto	Comunicación con los miembros del equipo y/o cliente de las peticiones de cambio evaluadas y actualización de información (correos, wiki, portal, etc.)

		PM 2.3	Registro de reuniones [actualizado]	Registro de actualización de reuniones	Comunicación con los miembros del equipo de las actividades de actualización de información (correos, wiki, portal, etc.)
		PM 2.4	Registro de reuniones [actualizado] Petición de cambios [aceptada] Registro de reuniones	Registro de reuniones actualizado, Registro de aceptación de la petición de cambios	Comunicación con los miembros del equipo de las peticiones de cambio aceptadas y actividades de actualización de información (correos, wiki, portal, etc.)
	PM 3	PM 3.2	Registro de correcciones	Identificar las actividades establecidas para corregir desviaciones o problemas que afecten el cumplimiento del plan Resultado de análisis y tratamiento de desviaciones	Horas imputadas para el análisis y tratamiento de desviaciones Comunicación a las partes interesadas del tratamiento de desviaciones y corrección (correo electrónico, actas, etc.)
o	PM 4	PM 4.1	Registro de aceptación, Configuración del software [aceptada]	Documento de aceptación del cliente para los productos entregados	Actas de reunión o correos que evidencian los componentes que se van a entregar al cliente Acta de reunión o correo con la aceptación de los acuerdos

Tabla 18 Plantilla de apoyo a la evaluación ISO/IEC 29110-4-1 para el proceso de SI

ISO/IEC 29110-4-1	ISO/IEC TR 29110-5-1-2		Evidencia		
Implementación del software SI	Actividad	Tareas	PS	D	I
a	SI 1	SI 1.1	Plan de proyecto [revisado]	No asignado	Horas imputadas para la revisión de proyecto Comunicación de la revisión a las partes (correo electrónico, actas, etc.)
		SI 1.2	No asignado	No asignado	Horas imputadas para el establecimiento o actualización del entorno de implementación Comunicación a las partes interesadas del establecimiento o actualización del entorno de implementación (correo electrónico, actas, etc.)

	SI 2	SI 2.1	No asignado	No asignado	Horas imputadas para la asignación de tareas de acuerdo con los roles del equipo de trabajo Comunicación a las partes interesadas de la distribución y asignación de tareas (correo electrónico, actas, etc.)
	SI 3	SI 3.1	No asignado	No asignado	Horas imputadas para la asignación de tareas de acuerdo con los roles del equipo de trabajo Comunicación a las partes interesadas de la distribución y asignación de tareas (correo electrónico, actas, etc.)
		SI 3.2	No asignado	No asignado	Horas imputadas para entender la especificación de requisitos de software
	SI 4	SI 4.1	No asignado	No asignado	Horas imputadas para la asignación de tareas de acuerdo con los roles del equipo de trabajo comunicación a las partes interesadas de la distribución y asignación de tareas (correo electrónico, actas, etc.)
	SI 5	SI 5.1	No asignado	No asignado	Horas imputadas para la asignación de tareas de acuerdo con los roles del equipo de trabajo comunicación a las partes interesadas de la distribución y asignación de tareas (correo electrónico, actas, etc.)
		SI 6	SI 6.1	No asignado	No asignado
b	SI 2	SI 2.2	Especificación de requisitos	Listado con los requisitos definidos para cada elemento del software en función de los requisitos del cliente y de las políticas y procedimientos aplicables a la organización	Actas de reunión o correos donde se evidencia el análisis y entendimiento en la definición o el cambio de los requisitos
c	SI 2	SI 2.3	Resultados de verificación Especificación de requisitos [verificado] Petición de cambios	Documento que contiene los resultados de la verificación y la solicitud de cambios	Comunicación con los miembros del equipo de los resultados de verificación y la petición de cambios (correos, wiki, portal, etc.)

d	SI 2	SI 2.4	Resultados de validación Especificación de requisitos [validada]	Documento que contiene los resultados de la validación de la especificación de requisitos	Comunicación con los miembros del equipo de los resultados de la validación de la petición de requisitos (correos, wiki, portal, etc.)
e	SI 2	SI 2.7	Configuración del software (especificación de requisitos, documentación de usuario)	Carpeta o tipo de almacenamiento que contenga la especificación de requisitos y la documentación de usuario	Comunicación con los miembros del equipo de los elementos que hacen parte de la configuración del software (correos, wiki, portal, etc.)
f	PM 3	PM 3.3	Petición de cambios	Documento que identifica problemas software y/o documentación, o deseos de mejora y peticiones de cambio	Comunicación con los miembros del equipo de los elementos que hacen parte de la configuración del software (correos, wiki, portal, etc.)
g	SI 3	SI 3.3	Diseño del software Registro de la trazabilidad	Documento con el diseño y la descripción de una arquitectura de alto nivel del sistema, los elementos hardware y software y las operaciones manuales, que garantiza que todos los requisitos del sistema se asignan entre los elementos Documento con la relación entre requisitos, elementos de diseño de software, componentes de software, casos de prueba y procedimientos de prueba	Actas de reuniones que evidencian el análisis la definición y la documentación del diseño de la arquitectura del sistema y la relación de los elementos del software. Comunicación a las partes interesadas del documento con el diseño de la arquitectura del sistema (correos, wiki, portal, etc.)
		SI 3.8	Configuración del software Diseño software [verificado, línea base], casos de prueba y procedimientos de prueba [verificado], registro de trazabilidad	Documento que contienen los resultados de la verificación de los elementos de la configuración del software y la línea base	Comunicación con los miembros del equipo de los resultados de verificación y los elementos asignados a la línea base (correos, wiki, portal, etc.)
		SI 4	SI 4.2	No asignado	No asignado
h	SI 3	SI 3.3			

			Diseño del software Registro de la trazabilidad	Documento con el diseño y la descripción de una arquitectura de alto nivel del sistema, los elementos hardware y software y las operaciones manuales, que garantiza que todos los requisitos del sistema se asignan entre los elementos Documento con la relación entre requisitos, elementos de diseño de software, componentes de software, casos de prueba y procedimientos de prueba	Actas de reuniones que evidencian el análisis la definición y la documentación del diseño de la arquitectura del sistema y la relación de los elementos del software. Comunicación a las partes interesadas del documento con el diseño de la arquitectura del sistema (correos, wiki, portal, etc.)
i	SI 3	SI 3.3	Diseño del software Registro de la trazabilidad	Documento con el diseño y la descripción de una arquitectura de alto nivel del sistema, los elementos hardware y software y las operaciones manuales, que garantiza que todos los requisitos del sistema se asignan entre los elementos Documento con la relación entre requisitos, elementos de diseño de software, componentes de software, casos de prueba y procedimientos de prueba	Actas de reuniones que evidencian el análisis la definición y la documentación del diseño de la arquitectura del sistema y la relación de los elementos del software. Comunicación a las partes interesadas del documento con el diseño de la arquitectura del sistema (correos, wiki, portal, etc.)
		SI 3.4	Resultado de verificación Diseño de software [verificado] Registro de trazabilidad [verificado] Petición de cambios	Documento que contiene los resultados de la verificación del diseño del software, registro de trazabilidad y la solicitud de cambios	Comunicación con los miembros del equipo de los resultados de verificación y la petición de cambios (correos, wiki, portal, etc.)
	SI 5	SI 5.6	Registro de trazabilidad [actualizado]	Documento con la relación entre requisitos, elementos de diseño de software, componentes de software, casos de prueba y procedimientos de prueba	Comunicación a las partes interesadas del documento que relacionan los elementos del sistema (correos, wiki, portales, etc.)

j	SI 4	SI 4.3	Componentes software	Conjunto relacionado de elementos software o unidades de código (código fuente, scripts, módulos, librerías, etc.)	Horas imputadas al establecimiento de los elementos de software o unidades de código
k	SI 4	SI 4.6	Registro de trazabilidad [actualizado]	Documento con la relación entre requisitos, elementos de diseño de software, componentes de software, casos de prueba y procedimientos de prueba	Comunicación a las partes interesadas del documento que relacionan los elementos del sistema (correos, wiki, portales, etc.)
l	SI 3	SI 3.5	Casos de prueba y procedimientos de prueba	Documento y/o procedimiento que contiene la planificación e información de las pruebas a realizar en las etapas de desarrollo del producto software	Horas imputadas al desarrollo de los casos de prueba y procedimientos de prueba Comunicación a las partes interesadas de los casos de prueba y procedimientos de pruebas (correos, wiki, portal, etc.)
		SI 3.6	Resultados de verificación Casos de prueba y procedimientos de prueba [verificados]	Documento que contiene los resultados de la verificación de los casos de prueba y procedimientos de prueba	Comunicación con los miembros del equipo de los resultados de verificación de los casos de prueba y procedimientos de prueba (correos, wiki, portal, etc.)
	SI 4	SI 4.4	Componentes software [unidades probadas]	Documento que contiene los resultados de las pruebas realizadas a los elementos software o a unidades de código (código fuente, scripts, módulos, librerías, etc.)	Horas imputadas a la ejecución de las pruebas de los elementos software o unidades de código
m	SI 3	SI 3.4	Resultado de verificación Diseño de software [verificado] Registro de trazabilidad[verificado] Petición de cambios	Documento que contiene los resultados de la verificación del diseño del software, registro de trazabilidad y la solicitud de cambios	Comunicación con los miembros del equipo de los resultados de verificación y la petición de cambios (correos, wiki, portal, etc.)
	SI 4	SI 4.6	Registro de trazabilidad [actualizado]	Documento con la relación entre requisitos, elementos de diseño de software, componentes de software, casos de prueba y procedimientos de prueba	Comunicación a las partes interesadas del documento que relacionan los elementos del sistema (correos, wiki, portales, etc.)
		SI 4.7			No asignado

			Configuración del software (componentes software [corregidos, línea base], registro de trazabilidad [actualizado, línea base])		Horas imputadas a la corrección y actualización de los elementos que forman parte de la configuración del software. Comunicación con los miembros del equipo de los resultados de corrección y actualización de los elementos asignados a la línea base (correos, wiki, portal, etc.)
n	SI 2	SI 2.5	Documentación de usuario software	Material impreso o electrónico que proporciona información a los usuarios de software	No asignado
		SI 2.6	Resultados de la verificación Documentación de usuario software [verificado] Petición de cambios	Documento que contiene los resultados de la verificación del documento de usuario del software y la solicitud de cambios	Comunicación con los miembros del equipo de los resultados de verificación y la petición de cambios (correos, wiki, portal, etc.)
	SI 5	SI 5.7	Guía de operación del producto	Material impreso o electrónico que describe el proceso de instalación y ejecución de un producto software, así como sus operaciones comunes	No asignado
		SI 5.8	Resultados de la verificación guía de operación del producto [verificada]	Documento que contiene los resultados de la verificación de la guía de operación del producto	Comunicación con los miembros del equipo de los resultados de verificación (correos, wiki, portal, etc.)
		SI 5.9	Documentación de usuario software	Material impreso o electrónico que proporciona información a los usuarios de software	No asignado
		SI 5.10	Resultado de verificación Documentación de usuarios software [verificado]	Documento que contienen los resultados de la verificación del documento de usuario del software	Comunicación con los miembros del equipo de los resultados de verificación (correos, wiki, portal, etc.)
	SI 6	SI 6.3	Documentación de mantenimiento	Documento que describe la configuración del software y el entorno utilizado para el desarrollo y pruebas (compiladores, herramientas	Horas imputadas al establecimiento y definición de los criterios de mantenimiento Comunicación con los miembros del equipo de los criterios de mantenimiento (correos, wiki, portal, etc.)

				de diseño, construcción y pruebas)	
		SI 6.4	Resultados de verificación Documentación de mantenimiento [verificado]	Documento que contiene los resultados de la verificación de la documentación de mantenimiento	Comunicación con los miembros del equipo de los resultados de verificación (correos, wiki, portal, etc.)
		SI 6.5	Configuración del software Documentación de mantenimiento [verificado, línea base]	Documento que contiene los resultados de la verificación de la documentación de mantenimiento y su establecimiento en la línea base	Comunicación con los miembros del equipo de los resultados de verificación (correos, wiki, portal, etc.)
o	SI 5	SI 5.3	Software Casos de prueba y procedimientos de prueba	Elemento de software (código fuente de software y código ejecutable) para un cliente, constituido por una colección de componentes de software integrados	No asignado
p	SI 5	SI 5.2	No asignado	No asignado	Horas imputadas a entender los casos de prueba y los procedimientos, establecer o actualizar el entorno de pruebas
		SI 5.3	Software Casos de prueba y procedimientos de prueba	Elemento de software (código fuente de software y código ejecutable) para un cliente, constituido por una colección de componentes de software integrados	No asignado
		SI 5.4	Software [probado] Informe de pruebas	Documento de ejecución de pruebas para los elementos de software (código fuente de software y código ejecutable)	Horas imputadas a la realización y ejecución de las pruebas para los elementos de software Comunicación a los miembros del equipo de los resultados de las pruebas realizadas a los elementos de software
q	SI 3	SI 3.7	Registro de trazabilidad [actualizado]	Documento con la relación entre requisitos, elementos de diseño de software, componentes de software, casos de prueba y procedimientos de prueba	Comunicación a las partes interesadas del documento que relacionan los elementos del sistema (correos, wiki, portales, etc.)
	SI 4	SI 4.5	Componentes software [corregidos]	Conjunto relacionado de elementos software o unidades de código (código	Horas imputadas a la corrección de los elementos de software o unidades de código



				fuentes, scripts, módulos, librerías, etc.)	
	SI 5	SI 5.5	Software [corregido] Informe de pruebas [defectos eliminados]	Elementos de software (código fuente de software y código ejecutable) [corregido] Documento de ejecución de pruebas para los elementos de software (código fuente de software y código ejecutable)	Horas imputadas a la corrección del software y eliminación de defectos en los informes de pruebas Comunicación a las partes interesadas de las actividades de corrección y eliminación de defectos
r	SI 4	SI 4.7	Configuración del software (componentes software [corregidos, línea base], registro de trazabilidad [actualizado, línea base])	No asignado	Horas imputadas a la corrección y actualización de los elementos que forman parte de la configuración del software. Comunicación con los miembros del equipo de los resultados de corrección y actualización de los elementos asignados a la línea base (correo, wiki, portal, etc.)
	SI 5	SI 5.11	Configuración del software Casos de prueba y procedimientos de prueba [línea base]	Carpeta o tipo de almacenamiento que contiene todos los elementos establecidos en la configuración del software Casos de prueba y procedimientos de prueba establecidos como parte de la línea base	Comunicación a los miembros del equipo del establecimiento de los elementos dentro de la configuración del software y establecimiento de los casos de prueba y procedimientos de prueba como parte de la línea base (correos, wiki, portal, etc.)
	SI 6	SI 6.2	No asignado	No asignado	Horas imputadas a entender la configuración de software
		SI 6.5	Configuración del software Documentación de mantenimiento [verificado, línea base]	Documento que contiene los resultados de la verificación de la documentación de mantenimiento y su establecimiento en la línea base	Comunicación con los miembros del equipo de los resultados de verificación (correos, wiki, portal, etc.)
s	SI 6	SI 6.6	Configuración del software [entregado]	Carpeta o tipo de almacenamiento que contiene todos los elementos establecidos en la configuración del software a ser entregado al cliente	Comunicación a las partes interesadas de la entrega de los elementos de la configuración del software (correos, wiki, portal, etc.)

## Definición de indicadores de cumplimiento de requisitos

Para que una organización cumpla con los requisitos establecidos en el estándar ISO/IEC 29110-4-1 se puede apoyar de una serie de indicadores de satisfacción de los requisitos logrando satisfacer un conjunto de tareas y productos de trabajo presentados en el estándar ISO/IEC TR 29110-5-1-2. La relación de las tareas descritas por el estándar ISO/IEC TR 29110-5-1-2 que apoyan el cumplimiento de los requisitos del estándar ISO/IEC 29110-4-1 se mostró en las columnas tareas y evidencias de las tablas 17 y 18.

La tabla 19 presenta una descripción de los componentes deseados y las características de los productos de trabajo que la organización debe mostrar para satisfacer una tarea definida en el estándar ISO/IEC TR 29110-5-2.

El cumplimiento de las tareas y los productos de trabajo permiten a la organización satisfacer los requisitos del estándar ISO/IEC 29110-4-1. Si la organización logra evidenciar a través de los productos de trabajo establecidos en la tabla 29 el cumplimiento de los requisitos presentados en la tabla 28 para el proceso de gestión de proyectos (PM, Project Management) y en la tabla 29 para el proceso de implementación del software (SI, Software Implementation), se podrían garantizar la conformidad de los procesos de la organización con respecto a los requisitos establecidos en el perfil básico del estándar ISO/IEC 29110-4-1.

*Tabla 19 Descripción y características de los productos de trabajo*

ID Producto de trabajo	Producto de trabajo	Características	Tarea
PT1	Declaración del trabajo	Descripción del trabajo realizado relacionado con la implementación del software:	PM 1.1
		*Descripción del producto	
		*Propósito	
		*Requisitos generales del cliente	
		*Descripción del alcance	
		*Objetivos del proyecto	
		*Listas de productos que se van a entregar al cliente	
PT2	Instrucciones de entrega	Incluye las siguientes características:	PM 1.2
		*Elementos requeridos para la liberación del producto identificados (hardware, software, documentación)	
		*Requisitos de entrega	
		*Orden secuencial de tareas a realizar	
		*Liberaciones aplicables identificadas	
		*Identificar todos los componentes software con la información de versión	

		*Identificar cualquier procedimiento necesario de copia de seguridad y recuperación	
		*Los estados aplicables son: verificado, aceptado, actualizado, revisado	
PT3	Plan de proyecto	Presenta cómo se ejecutarán los procesos y las actividades del proyecto para asegurar la finalización exitosa del proyecto y la calidad de los productos entregables. Incluye los siguientes elementos:	PM 1.4
			PM 1.5
		*Descripción del producto	PM 1.6
		*Propósito	PM 1.7
		*Requisitos generales del cliente	PM 1.8
		*Descripción del alcance	PM1.9
		*Objetivos del proyecto	PM1.2
		*Lista de productos que se van a entregar al cliente	PM1.3
		*Tareas, incluyendo la verificación, la validación y las revisiones por parte del cliente y el equipo de trabajo. Las tareas pueden ser representadas como una estructura de desglose de trabajo	
		*Duración estimada de las tareas	
		Recursos (humanos, materiales, estándares, equipos y herramientas), incluyendo capacitaciones:	
			PM 1.10
		*Composición del equipo de trabajo	PM 1.11
		*Cronograma de tarea, inicio y finalización estimada para cada tarea, relación y dependencia de las tareas	PM 1.12
		*Costos y esfuerzo estimado	PM 1.13
		*Identificación de riesgo de proyecto	SI 1.1
		*Estrategia de control de versiones (herramienta para repositorio de producto o mecanismo identificado; ubicación y mecanismos de acceso para el repositorio especificados: identificación y control de versiones definido; mecanismos para la copia de seguridad y recuperación definidos; mecanismos de almacenamiento, manejo y entrega especificados)	
		*Instrucciones de entrega	
		*Elementos requeridos para la liberación del producto identificados (hardware, software, documentación, etc.)	
PT4	Registro de reuniones	Registro de los acuerdos establecidos con el cliente y/o equipo de trabajo. Incluyendo:	
		*Propósito de la reunión	
		*Asistentes	PM 1.14
		*Lugar y fecha	PM 2.3
		*Referencia a minutos anteriores	PM 2.4
		*Los logros alcanzados	
		*Identificación de inconvenientes superados	
*Próxima reunión			
		El estado aplicable es: actualizado	
PT5	Repositorio del proyecto	Contenedor electrónico para almacenar los productos de trabajo y entregables del proyecto. Incluyendo:	
		*Almacenamiento de productos de trabajo del proyecto	
		*Almacenamiento de productos entregables liberados	
		*Capacidades de almacenamiento y recuperación	PM 1.15

		*Capacidad para buscar contenidos	PM 4.2
		*Listas de contenidos con su descripción	
		*Capacidad para compartir y transferir productos de trabajo entre miembros del equipo	
		*Control efectivo sobre el acceso	
		*Descripción del mantenimiento de productos	
		*Recuperación de versiones de productos de trabajo	
		*Reporte de estado de producto	
		Los estados aplicables son: recuperado y actualizado	
PT6	Registro de estado de progreso	Almacena el estado del proyecto contra el plan de proyecto. Incluyendo:	
		*Estado de las tareas actuales contra las planeadas	
		*Estado de los resultados actuales contra los objetivos establecidos	PM 2.1
		*Estado de la asignación actual de los recursos contra los recursos planeados	PM 3.1
		*Estado actual de riesgo contra los previamente identificados	
		*Almacenar cualquier anomalía para las tareas y su justificación	
		El estado aplicable es: evaluado	
PT7	Petición de cambios	Identifica problemas software y/o documentación, o deseos de mejora y peticiones de cambio. Incluyendo:	PM 2.2
		*Identificación de cambios al propósito	PM 2.4
		*Identificación de estado de peticiones (nuevas, aceptadas, rechazadas)	PM 3.3
		*Identificación de la información del contacto solicitante	PM 2.3
		*Sistemas afectados	SI 2.3
		*Impacto asociado con la documentación	SI 2.6
		*Estimar impacto y fecha	SI 3.4
		Los estados aplicables son: iniciado, evaluado y aceptado	
PT 8	Copia de seguridad del repositorio del proyecto	Repositorio usado para hacer copias de seguridad del repositorio del proyecto y, si es necesario, para la recuperación de información	PM 2.5
			PM 2.6
PT 9	Registro de correcciones	Identifica las actividades establecidas para corregir desviaciones o problemas que afecten el cumplimiento del plan.	
		Incluyendo:	
		*Identificar el problema inicial	
		*Definir una solución	PM 3.2
		*Identificar las acciones correctivas tomadas	SI 4.5
		*Identificar el propietario de esas acciones definidas	SI 4.7
		*Identificar fechas de apertura y cierre	SI 5.5
*Contener un indicador de estado			
		*Indicar las acciones a seguir	
PT 10	Registro de aceptación	Documento de aceptación del cliente para los entregables del proyecto. Incluyendo:	
		*Registro de recepción de la entrega	PM 4.1
		*Identificar la fecha de recibido	
		*Identificar los elementos entregados	
		*Almacenar las verificaciones de los clientes y los criterios de aceptación	
		*Firma de recibido por el cliente	

PT 11	Configuración del software	Un conjunto de productos software único y coherente. Incluyendo:	
		*Especificación de requisitos	
		*Diseño de software	PM 4.1
		*Registro de trazabilidad	SI 2.7
		*Componentes software	SI 3.8
		*Software	SI 4.7
		*Casos de prueba y procedimientos de prueba	SI 5.11
		*Reporte de pruebas	SI 6.5
		*Guía de operación del producto	SI 6.6
		*Documentación de usuario del software	
		*Documentación de mantenimiento	
		Los estados aplicables son: entregado y aceptado	
PT 12	Estrategia de control de versiones	Incluye las siguientes características:	
		*Herramientas o mecanismos de repositorio de productos identificados	
		*Ubicación y mecanismos de acceso al repositorio especificado	
		*Identificación y control de versiones definido	PM 1.10
		*Mecanismos de copia de seguridad y recuperación definidos	
		*Mecanismos para el almacenamiento, el manejo y la liberación (incluyendo archivo y su recuperación) especificados	
PT 13	Resultados de verificación	Incluye las siguientes características:	PM 1.3
		*Participantes	SI 2.3
		*Fecha	SI 2.6
		*Lugar	SI 3.4
		*Duración	SI 3.6
		*Lista de chequeo con la verificación	SI 3.8
		*Elementos que pasaron la verificación	SI 5.8
		*Elementos que fallaron la verificación	SI 5.10
		*Elementos pendientes de verificación	SI 6.4
		*Defectos identificados durante la verificación	SI 6.5
PT 14	Resultados de validación	Incluye las siguientes características:	
		*Participantes	
		*Fecha	
		*Lugar	SI 2.4
		*Duración	
		*Lista de chequeo con la validación	
PT 15	Documentación de mantenimiento	Documento que incluye las siguientes características:	
		*Incluye o hace referencia a todos los elementos de la configuración de software desarrollados durante la implementación	SI 6.3
		*Identifica el entorno utilizado para el desarrollo y las pruebas (compiladores, herramientas de diseño, construcción y herramientas de pruebas)	SI 6.4
		*Está escrito en términos que el personal de mantenimiento puede entender	SI 6.5
		Los estados aplicables son: verificados y alineado	
PT 16		Incluye las siguientes características:	

	Especificación de requisitos	*Introducción	
		*Descripción	
		*Descripción general del software y su uso dentro del alcance del negocio del cliente	
		Descripción de los requisitos:	
		*Funcionalidad: necesidad que debe ser satisfecha por el software cuando es utilizado en condiciones específicas. La funcionalidad debe ser adecuada, precisa y segura	
		*Interfaz de usuario: definición de las características de interfaz de usuario que permite entender y aprender el software fácilmente para que el usuario pueda realizar sus tareas	
		*Interfaces externas: definición de interfaces con otro software o hardware	SI 2.2
		*Fiabilidad: especificación del nivel de ejecución del software relativo a la madurez, la tolerancia a fallos y la recuperación	SI 2.3
		*Eficiencia: especificación del nivel de ejecución del software relativo al tiempo y el uso de los recursos	SI 2.4
		*Mantenimiento: descripción de los elementos que facilitan la comprensión y ejecución de las futuras modificaciones de software	SI 2.7
		*Portabilidad: descripción de las características del software que permiten su transferencia de un lugar a otro	
		*Limitaciones/limitaciones de diseño y construcción: necesidades impuestas por el cliente	
		*Interoperabilidad: capacidad que tienen dos o más sistemas o componentes de software para cambiar información mutuamente y usarla	
		*Reutilización: característica de cualquier producto/subproducto, o parte de él, para que pueda ser utilizado por varios usuarios como producto final, en el propio desarrollo de software o en la ejecución de otros productos de software	
		*Necesidades legales y regulativas: impuestos por leyes, reglamentos, etc.	
Cada requisito es identificado, único y es verificable o puede ser evaluado			
Los estados aplicables son: verificado, validado y puesto en la línea base			
PT 17	Software	Elemento Software (código fuente de software y código ejecutable) para un cliente, constituido por una colección de componentes de software integrados	SI 5.3
		Los estados aplicables son: probado y alineado	SI 5.4
			SI 5.5
PT 18	Componentes del software	Un conjunto de unidades de código relacionadas	SI 4.3
			SI 4.4
		Los estados aplicables son: unidad probada, corregida y puesta en la línea base	SI 4.5
			SI 4.7
PT 19	Configuración del software	Un conjunto único y coherente de productos de software que incluye:	
		*Especificación de requisitos	
		*Diseño de software	
		*Registro de trazabilidad	SI 2.7
		*Componentes software	SI 3.8
		*Software	SI 5.11
		*Casos de prueba y procedimientos de prueba	SI 6.5
		*Informe de prueba	SI 6.6
		*Guía de funcionamiento del producto	
		*Documentación del usuario del software	
*Documentación de mantenimiento			

		Los estados aplicables son: entregado y aceptado	
PT 20	Diseño del software	Información textual y gráfica sobre la estructura del software	
		Esta estructura puede incluir las siguientes partes:	
		*Diseño de la arquitectura de software de alto nivel: describe la estructura software:	
		*Identifica los componentes de software requeridos	
		*Identifica la relación entre los componentes de software	SI 3.3
		*Se considerará cualquier requisito: características del rendimiento del software; hardware, software e interfaces humanas; características de seguridad; requisitos de diseño de base de datos: atributos de manejo de errores y recuperación	SI 3.4 SI 3.8
		*Diseño detallado de software de bajo nivel: incluye detalles de los componentes de software para facilitar su construcción y prueba en el entorno de programación:	
		*Proporciona un diseño detallado (podría representarse como un prototipo, organigrama, entidad diagrama relación, pseudocódigo, etc.)	
		*Proporciona una especificación de las necesidades de almacenamiento de datos	
		*Establece las convenciones de nomenclatura de datos necesarios	
		*Define el formato de las estructuras de datos requeridos	
		*Define los campos de datos y el propósito de cada elemento de datos requeridos	
		*Proporciona las especificaciones de la estructura del programa	
		Los estados aplicables son: verificado y puesto en línea base	
PT 21	Casos de prueba y procedimientos de prueba	Elementos necesarios para probar el código. El caso de prueba puede incluir lo siguiente:	
		*Identifica el caso de prueba	
		*Elementos de prueba	
		*Especificaciones de entrada	
		*Especificaciones de salida	SI 3.5
		*Las necesidades del ambiente de prueba	SI 3.6
		*Dependencias de la interfaz	SI 3.8
		Los procedimientos de prueba pueden incluir lo siguiente:	SI 5.3
		*Identifica: nombre de la prueba, descripción de la prueba y fecha de finalización de la prueba	SI 5.11
		*Identifica posibles problemas de implementación	
		*Identifica a la persona que completó el procedimiento de prueba	
*Identifica los prerrequisitos			
*Identifica los pasos del procedimiento incluyendo el número de paso, la acción requerida por el probador y los resultados esperados			
		Los estados aplicables son: verificado y puesto en línea base	
PT 22	Informe de pruebas	Documenta la ejecución de las pruebas. Puede incluir lo siguiente:	
		*Un resumen de cada defecto	
		*Identifica el caso de prueba correspondiente	
		*Identifica al probador que encontró cada defecto	
		*Identifica la gravedad de cada defecto	SI 5.5
		*Identifica la(s) función(es) afectada(s) por cada defecto	
		*Identifica la fecha en que se originó cada defecto	
		*Identifica la fecha en que se resolvió cada defecto	
		*Identifica a la persona que resolvió cada defecto	

		El estatus aplicable es: puesto en línea base	
PT 23	Registro de trazabilidad	Documenta la relación entre los requisitos incluidos en la especificación de requisitos, elementos de diseño de software, componentes de software, casos de prueba y procedimientos de prueba. Puede incluir lo siguiente:	SI 3.4
			SI 3.7
		*Identifica los requisitos de la especificación de los requisitos a rastrear	SI 3.8
		*Proporciona mapeo hacia adelante y hacia atrás de los requerimientos hacia los elementos del diseño de software, componentes de software, casos de prueba y procedimientos de prueba	SI 4.6
			SI 4.7
		Los estados aplicables son: verificado, puestos en la línea base y actualizado	SI 5.6

Con esta última tabla se da por terminado el análisis de la norma ISO/IEC 29110, esto quiere decir que tenemos los recursos necesarios para crear una metodología única que ayude al CDS del ITSUR a conseguir la certificación ISO/IEC 29110.



## Diseño de la propuesta Metodológica

Ya vimos las actividades y documentaciones que necesitamos para elaborar la metodología que nos ayudará a conseguir la certificación ISO/IEC 29110, para esto, se creó una tabla en la cual mostrará las actividades y documentos necesarios para conseguir la certificación ISO/IEC 29110, dicha tabla tendrá un mapeo entre las actividades CMMI y el ISO/IEC 29110 para descubrir que actividades nos pueden ayudar a obtener esta certificación, cuales hacen falta y cuales necesitan modificaciones para completar las actividades necesarias. A continuación, en la tabla 20 se muestra el mapeo entre estas dos grandes certificaciones del mundo del Software.

*Tabla 20 Resultados de mapeo CMMI y ISO/IEC 29110*

producto del trabajo	ISO/IEC29110	CMMI	características	De importancia	CMMI
<b>Declaración del trabajo</b>	si	no	descripción del producto	si	si
			Propósito	si	no
			Requisitos generales del cliente	si	no
			descripción del alcance	si	si
			Objetivos del proyecto	si	si
			Listas de productos que se van a entregar al cliente	si	si
<b>Instrucciones de entrega</b>	si	no	Elementos requeridos para la liberación del producto liberado (Hardware, Software, documentación)	si	no
			Requisitos de entrega	Si	no
			Orden secuencial de tareas a realizar	si	no
			Liberaciones aplicables identificadas	si	si
			Identificar todos los componentes software con la	si	no

			información de versión		
			Identificar cualquier procedimiento necesario de copia de seguridad y recuperación	si	no
<b>Plan de proyecto</b>	si	si	Descripción del producto	si	si
			Propósito	si	si
			Requisitos generales del cliente	si	no
			descripción del alcance	si	si
			Objetivos del proyecto	si	si
			lista de productos que se van a entregar el cliente	si	si
			*Tareas, incluyendo la verificación, la validación y las revisiones por parte del cliente y el equipo de trabajo. Las tareas pueden ser representadas como una estructura de desglose de trabajo	si	si
			*Duración estimada de las tareas	no	no
			Recursos (humanos, materiales, estándares, equipos y herramientas), incluyendo capacitaciones:	si	si
			*Composición del equipo de trabajo	si	si
			*Cronograma de tarea, inicio y finalización estimada para cada tarea,	si	si

			relación y dependencia de las tareas		
			*Costos y esfuerzo estimado	si	si
			*Identificación de riesgo de proyecto	si	si
			*Estrategia de control de versiones (herramienta para repositorio de producto o mecanismo identificado; ubicación y mecanismos de acceso para el repositorio especificados: identificación y control de versiones definido; mecanismos para la copia de seguridad y recuperación definidos; mecanismos de almacenamiento, manejo y entrega especificados)	si	si
			*Instrucciones de entrega	no	no
			*Elementos requeridos para la liberación del producto identificados (hardware, software, documentación, etc.)	si	no
<b>Registro de reuniones</b>	si	si	*Propósito de la reunión	si	si
			*Asistentes	si	si
			*Lugar y fecha	si	si
			*Referencia a minutos anteriores	si	si
			*Los logros alcanzados	si	si
			*Identificación de inconvenientes superados	si	si
			*Próxima reunión	si	si
<b>Repositorio del Proyecto</b>	si	si	*Almacenamiento de productos de trabajo del proyecto	si	si
			*Almacenamiento de productos entregables liberados	si	si

			*Capacidades de almacenamiento y recuperación	si	si
			*Capacidad para buscar contenidos	si	no
			*Listas de contenidos con su descripción	si	si
			*Capacidad para compartir y transferir productos de trabajo entre miembros del equipo	si	si
			*Control efectivo sobre el acceso	si	si
			*Descripción del mantenimiento de productos	si	si
			*Recuperación de versiones de productos de trabajo	si	si
			*Reporte de estado de producto	si	no
<b>Registro de estado de Progreso</b>	si	si	*Estado de las tareas actuales contra las planeadas	si	no
			*Estado de los resultados actuales contra los objetivos establecidos	si	no
			*Estado de la asignación actual de los recursos contra los recursos planeados	si	no
			*Estado actual de riesgo contra los previamente identificados	si	no

			*Almacenar cualquier anomalía para las tareas y su justificación	si	no
<b>Petición de cambios</b>	si	si	*Identificación de cambios al propósito	si	no
			*Identificación de estado de peticiones (nuevas, aceptadas, rechazadas)	si	no
			*Identificación de la información del contacto solicitante	si	si
			*Sistemas afectados	si	no
			*Impacto asociado con la documentación	si	no
			*Estimar impacto y fecha	si	si
<b>Copia de seguridad del repositorio del proyecto</b>	si	si	Repositorio usado para hacer copias de seguridad del repositorio del proyecto y, si es necesario, para la recuperación de información	si	si
<b>Registro de correcciones</b>	si	si	*Identificar el problema inicial	si	no
			*Definir una solución	si	no
			*Identificar las acciones correctivas tomadas	si	si
			*Identificar el propietario de esas acciones definidas	si	no
			*Identificar fechas de apertura y cierre	si	si
			*Contener un indicador de estado	si	no
			*Indicar las acciones a seguir	si	si

<b>Registro de aceptación</b>	si	si	*Registro de recepción de la entrega	si	si
			*Identificar la fecha de recibido	si	no
			*Identificar los elementos entregados	si	si
			*Almacenar las verificaciones de los clientes y los criterios de aceptación	si	no
			*Firma de recibido por el cliente	si	si
<b>Configuración del Software</b>	si	si	*Especificación de requisitos	si	no
			*Diseño de software	si	si
			*Registro de trazabilidad	si	si
			*Componentes software	si	si
			*Software	si	si
			*Casos de prueba y procedimientos de prueba	si	si
			*Reporte de pruebas	si	si
			*Guía de operación del producto	si	no
			*Documentación de usuario del software	si	si
*Documentación de mantenimiento	si	no			
<b>Estrategia de control de versiones</b>	si	no	*Herramientas o mecanismos de repositorio de productos identificados	si	si

			*Ubicación y mecanismos de acceso al repositorio especificado	si	si
			*Identificación y control de versiones definido	si	si
			*Mecanismos de copia de seguridad y recuperación definidos	si	si
			*Mecanismos para el almacenamiento, el manejo y la liberación (incluyendo archivo y su recuperación)	si	si
			especificados	no	no
<b>Resultados de verificación</b>	si	no	*Participantes	si	si
			*Fecha	si	si
			*Lugar	si	si
			*Duración	si	no
			*Lista de chequeo con la verificación	si	si
			*Elementos que pasaron la verificación	si	si
			*Elementos que fallaron la verificación	si	no
			*Elementos pendientes de verificación	si	no
			*Defectos identificados durante la verificación	si	si
<b>Resultados de validación</b>	si	no	Participantes	si	si
			fecha	si	si
			Lugar	si	si
			Duración	si	no
			Lista de chequeo con la validación	si	si

<b>Documento de mantenimiento</b>	si	si	*Incluye o hace referencia a todos los elementos de la configuración de software desarrollados durante la implementación	si	si
			*Identifica el entorno utilizado para el desarrollo y las pruebas (compiladores, herramientas de diseño, construcción y herramientas de pruebas)	si	si
			*Está escrito en términos que el personal de mantenimiento puede entender	si	si
<b>Especificación de requisitos</b>	si	si	*Introducción	si	si
			*Descripción	si	si
			*Descripción general del software y su uso dentro del alcance del negocio del cliente	si	si
			Descripción de los requisitos:	si	si
			*Funcionalidad: necesidad que debe ser satisfecha por el software cuando es utilizado en condiciones específicas. La funcionalidad debe ser adecuada, precisa y segura	si	si
			*Interfaz de usuario: definición de las características de interfaz de usuario que permite entender y aprender el software fácilmente para que el usuario pueda realizar sus tareas	si	si
			*Interfaces externas: definición de interfaces con otro software o hardware	si	si
			*Fiabilidad: especificación del nivel de ejecución del software relativo a la madurez, la tolerancia a fallos y la recuperación	si	si



			*Eficiencia: especificación del nivel de ejecución del software relativo al tiempo y el uso de los recursos	si	si
			*Mantenimiento: descripción de los elementos que facilitan la comprensión y ejecución de las futuras modificaciones de software	si	si
			*Portabilidad: descripción de las características del software que permiten su transferencia de un lugar a otro	si	si
			*Limitaciones/limitaciones de diseño y construcción: necesidades impuestas por el cliente	si	si
			*Interoperabilidad: capacidad que tienen dos o más sistemas o componentes de software para cambiar información mutuamente y usarla	si	si
			*Reutilización: característica de cualquier producto/subproducto, o parte de él, para que pueda ser utilizado por varios usuarios como producto final, en el propio desarrollo de software o en la ejecución de otros productos de software	si	si
			*Necesidades legales y regulativas: impuestos por leyes, reglamentos, etc.	si	si
			Cada requisito es identificado, único y es verificable o puede ser evaluado	si	si

<b>Software</b>	si	si	Elemento Software (código fuente de software y código ejecutable) para un cliente, constituido por una colección de componentes de software integrados	si	si
<b>Componentes de Software</b>	si	si	Un conjunto de unidades de código relacionadas	si	si
<b>Configuración del Software</b>	si	si	*Especificación de requisitos	si	si
			*Diseño de software	si	si
			*Registro de trazabilidad	si	si
			*Componentes software	si	si
			*Software	si	si
			*Casos de prueba y procedimientos de prueba	si	si
			*Informe de prueba	si	si
			*Guía de funcionamiento del producto	si	si
			*Documentación del usuario del software	si	si
			*Documentación de mantenimiento	si	si
<b>Diseño del Software</b>	si	si	Información textual y gráfica sobre la estructura del software	si	si
			Esta estructura puede incluir las siguientes partes:		
			*Diseño de la arquitectura de software de alto nivel: describe la estructura software:	si	si
			*Identifica los componentes de software requeridos	si	si

			*Identifica la relación entre los componentes de software	si	si
			*Se considerará cualquier requisito: características del rendimiento del software; hardware, software e interfaces humanas; características de seguridad; requisitos de diseño de base de datos: atributos de manejo de errores y recuperación	si	si
			*Diseño detallado de software de bajo nivel: incluye detalles de los componentes de software para facilitar su construcción y prueba en el entorno de programación:	si	si
			*Proporciona un diseño detallado (podría representarse como un prototipo, organigrama, entidad diagrama relación, pseudocódigo, etc.)	si	si
			*Proporciona una especificación de las necesidades de almacenamiento de datos	si	si
			*Establece las convenciones de nomenclatura de datos necesarios	si	si
			*Define el formato de las estructuras de datos requeridos	si	si
			*Define los campos de datos y el propósito de cada elemento de datos requeridos	si	si
			*Proporciona las especificaciones de la estructura del programa	si	si

<b>Casos de prueba y procedimientos de prueba</b>	si	si	*Identifica el caso de prueba	si	si
			*Elementos de prueba	si	si
			*Especificaciones de entrada	si	si
			*Especificaciones de salida	si	si
			*Las necesidades del ambiente de prueba	si	si
			*Dependencias de la interfaz	si	si
			Los procedimientos de prueba pueden incluir lo siguiente:		
			*Identifica: nombre de la prueba, descripción de la prueba y fecha de finalización de la prueba	si	si
			*Identifica posibles problemas de implementación	si	si
			*Identifica a la persona que completó el procedimiento de prueba	si	si
			*Identifica los prerrequisitos	si	si
			*Identifica los pasos del procedimiento incluyendo el número de paso, la acción requerida por el probador y los resultados esperados	si	si
<b>Informe de pruebas</b>	si	si	*Un resumen de cada defecto	si	si
			*Identifica el caso de prueba correspondiente	si	si

			*Identifica al probador que encontró cada defecto	si	si
			*Identifica la gravedad de cada defecto	si	si
			*identifica la(s) función(es) afectada(s) por cada defecto	si	si
			*Identifica la fecha en que se originó cada defecto	si	no
			*Identifica la fecha en que se resolvió cada defecto	si	no
			*Identifica a la persona que resolvió cada defecto	si	si
<b>Registro de trazabilidad</b>	si	si	Documenta la relación entre los requisitos incluidos en la especificación de requisitos, elementos de diseño de software, componentes de software, casos de prueba y procedimientos de prueba. Puede incluir lo siguiente:	si	si
			*Identifica los requisitos de la especificación de los requisitos a rastrear	si	si
			*Proporciona mapeo hacia adelante y hacia atrás de los requerimientos hacia los elementos del diseño de software, componentes de software, casos de prueba y procedimientos de prueba	si	no

Como se puede observar en la tabla número 20, la mayoría de los productos de trabajo son semejantes entre la forma que se trabajaba en CDS del ITSUR con certificación CMMII con los

productos de trabajo del estándar ISO/IEC 29110, lo que permite avanzar a pasos agigantados a conseguir la certificación de este estándar.

Para lograr esta metodología de la certificación se recomienda editar los documentos con los que se trabajaba antiguamente en el CDS; si bien los documentos cumplen con lo que se pide en la norma ISO/IEC 29110, estos contienen información más extendida de la que se pide en la norma ya que es una certificación más robusta en todos los sentidos.

A su vez se debe de crear los documentos que no tiene el CMMI de acuerdo a la norma ISO/IEC 29110, y se debe de completar las características que tampoco aparecen como acreditadas, aunado a esto se debe englobar los roles de CMII a los roles del estándar ISO/IEC 29110, a continuación, se muestra cómo deben de quedar los roles de CMMI convertidos a la norma ISO/IEC 29110.

Tabla 21 Mapeo de roles entre CMMI e ISO/IEC 29110

CMMI	ISO/IEC29110E
líder del proyecto	Project Manager
CM Manager	Technical Leader
Administrador QA	Work Time
Auditor	Work Time
Tester	Work Time
Analista	Analyst
Arquitecto	Designer
No definido	Programmer
No definido	Customer

Como se muestra en la tabla 21 se tienen roles similares entre las dos certificaciones, como es el caso del líder del proyecto y el analista, encontramos también diferencias en la certificación ISO/IEC 29110e, en ella se define un programador y un cliente mientras que en la metodología CMMI no, a su vez en los roles CMMI se cuenta con un administrador QA, un Auditor y un Tester, en el estándar ISO/IEC 29110E estos tres roles perfectamente caben en lo que se denomina el rol de Work time.

Con el mapeo de los productos de trabajo y los roles entre certificaciones, el Centro de Desarrollo de Software del ITSUR tiene una gran ventaja para conseguir la certificación ISO/IEC 29110, ya que cuenta con los elementos necesarios para llevar a cabo esta certificación, sería mas conveniente contar con esta certificación ya que los pasos y entregables que se necesitan son menores que a la certificación CMMI, también los roles de trabajo se pueden adaptar de mejor manera a la forma de trabajo del CDS, las reuniones con los clientes se tratarían de manera mas eficaz y eficiente para ambas partes, y lo mas importante de todo es que gracias a la forma de trabajo con CMMI ya se dispone con los documentos , actividades y procesos que se necesitan para conseguir la certificación ISO/IEC 29110.

## Conclusiones

En la presente investigación se desglosaron dos certificaciones importantes en el mundo del desarrollo de Software, la certificación CMMI y el estándar ISO/IEC 29110e, de la misma forma, se describe la forma de trabajar del centro de desarrollo de software del Instituto Superior del Sur de Guanajuato (ITSUR).

En la primera parte del documento se describen los procedimientos pertenecientes a la certificación CMMI, tales procedimientos contienen actividades y procesos que se necesitan cumplir al pie de la letra para conseguir los resultados y la calidad que se necesita para esta certificación. Cada proceso y actividad tienen documentos de entrada y salida que se deben de cumplir para lograr el éxito del proceso, de estos documentos se encargan los actores según sus funciones y tareas a los que están ligados con dicha certificación. El cliente también es parte importante de esta certificación, ya que va ligado a varias actividades y subprocesos de la certificación CMMI, en los cuales sus principales funciones son dar los requerimientos que necesitan para lograr el proyecto, asistir a las reuniones de avance del proyecto, solicitar cambios en el programa y asistir a reuniones en sesiones de cambios y reuniones de emergencias. Todos estos factores hacen que la certificación CMMI sea de las mejores certificaciones en el mundo del Software.

En la segunda parte de este documento se habló de la forma de trabajar en el CDS del ITSUR, en esta parte se hizo hincapié en los problemas que tenía este, los principales problemas fueron la falta de recursos económicos y humanos para dar seguimiento a la certificación CMMI, dichos problemas causaron la pérdida de esta certificación, esto desmonto en un problema agudo con los que cuenta en estos momentos el CDS. La pérdida de la certificación desencadenó la falta de una metodología para trabajar en el CDS, ahora se trabaja con dos grupos diferentes para la realización de los proyectos, estos grupos a su vez se dividen el trabajo en cada participante el cual tiene su propia forma de trabajo, esto ocasiona tener diferentes metodologías para la realización de un proyecto, lo que ocasiona problemas en el equipo y la calidad del producto final.

La tercera parte nos habla acerca de la ISO/IEC 29110E en la cual vimos que se divide en dos grandes procedimientos, uno encargado a la administración del proyecto y el otro a la implementación del software, dichos procesos se dividen en diagramas de actividades, las cuales tienen sus propias tareas y actividades a realizar, cada una de estas actividades cuentan con documentos de entrada y



salidas que ayudan a tener una mejor gestión y calidad de entrega del producto final, de igual manera que en la certificación CMMI los clientes forman parte importante de este ISO, ya que de él depende los requerimientos del proyecto y enterarse de los avances del mismo, de la misma forma puede solicitar cambios de este y dar opiniones de que les parecieron los cambios, todo esto llevado en minutas y documentos firmados de conformidad y de enterado de la misma forma que en CMMI.

En la cuarta y última parte del documento se hizo un mapeo entre estas dos certificaciones de software lo que nos dio la pauta para crear una metodología que permita al Centro de Desarrollo de Software del ITSUR lograr la certificación ISO/IEC 29100e que les permita crear de nueva forma Software de alta calidad y dar un nuevo plus a esta importante institución del sur de Guanajuato.

## Bibliografía

- A., O. A. (2011). Cuatro enfoques metodológicos para el desarrollo de Software RUP - MSF - XP - SCRUM. *inventum*, 64-78.
- Abarca Romero Melissa, A. M. (2015). Implementación del Perfil Básico de la ISO/IEC 29110 de una Pequeña Empresa Desarrolladora de Software: Lecciones aprendidas. *Conference: Industrial (INDUSTRIAL 2015) en XVIII Congreso Ibero Americano de Ingeniería de Software (CIBSE 2015), At Lima, Peru*, 1-12.
- AENOR. (s.f.). *Gestión de la calidad y aseguramiento de la calidad*. AENOR.
- Alejandro, F. K. (2003). Auditoría a los procesos en las empresas. *Quipukamayoc*, 43-57.
- Alfonso Fuggetta, E. D. (2000). Proceso de Software. *FOSE Track* , 1-12.
- Anaya, R. (2006). Una visión de la enseñanza de la ingeniería de software como apoyo al mejoramiento de las empresas de software. *Universidad EAFIT*, 60-76.
- Andrés Navarro Cadavid, J. D. (2013). Revisión de metodologías ágiles para el desarrollo de Software. *Prospectiva Una nueva visión para la ingeniería*, 30-39.
- Barney, J. (2001). Resource-based theories of competitive advantage: A ten year retrospective on the resource-based view. *Journal of Management*, 643.
- Barrios-García Jorge Alberto, S.-M. L.-R.-G. (2018). Implementación de la norma ISO/IEC 29110 de ingeniería de Software en Instituciones Académicas. *Revista de Tecnología y Educación*, 1-6.
- Becerra, J. A., & Vanegas, C. D. (2018). Propuesta de un método para desarrollar Sistemas de Información Geográfica a partir de la metodología de desarrollo ágil-SCRUM. *Cuaderno Activa*, 29-41.
- Beynon, D. (2007). Interpreting Capability Maturity model Integration (CMMI) for business Development organizations in the Government and Ind. *Tech. Rep. Technical Note CMU/SEI-2007-NT-004*.
- Brian Molina Montero, H. V. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales*, 114-121.
- Brown, W. M. (1998). *AntiPatterns. Refactoring Software, Architectures, and Projects in Crisis*. NY, John Wiley & Sons. ISBN\_0-471-19713-0.
- Calero, C., Moraga, M. Á., & Piattini, M. G. (2010). *Calidad del producto y proceso Software*. Madrid: RA-MA.
- Calvo-Manzano José A, G. J. (2008). Perfiles del ciclo de vida del software para pequeñas empresas: los informes técnicos ISO/IEC 29110. *REICIS. Revista Española de Innovación, Calidad e ingeniería del software*, 96-108.

- Campos, E. M., Gordón, M. L., Palacios, R. C., & Seco, A. d. (2014). Towards Measuring the Impact of the ISO/IEC 29110 Standard: A systematic Review. *Communications in Computer and Information Science*, 1-12.
- Cano, J. H., Letelier, P., & Penadés, M. C. (2012). Metodologías Ágiles en el Desarrollo de Software. *DSIC*, 1-8.
- Carnegie Mellon Software Engineering Institute. (2002). *Capability Maturity Model Integration (CMMI), Version 1.1*. Pittsburgh: U.S. Department of Defense.
- Carrasco, J. B. (2011). *Gestión de Procesos (alineados con la estrategia)*. Santiago de Chile: Editorial Evolución S.A.
- Castillo-Salinas Luis, S.-G. S.-R.-G. (2020). Evaluation of the implementation of a subset of ISO/IEC 29110 Software Implementation process in four teams of undergraduate students of Ecuador. An empirical software engineering experiment. *ElSevier*, 1-19.
- Cátedra de Mejora de Procesos de Software en el Espacio Iberoamericano de la Universidad Politécnica de Madrid. (2009). *CMMI Guía para la integración de procesos y la mejora de productos*. Madrid: Universidad Politecnica de Madrid.
- César Camisón, S. C. (2006). *Gesrtión de la calidad: Conceptos, enfoques, modelos y sistemas*. Madrid: Pearson Educación S.A.
- Claude Y Laporte, M. M. (2017). The Education of Students About ISO/IEC 29110 Software Engineering Standards and Their Implementations in Very Small Entities. *Canada International Humanitarian Technology Conference (IHTC)*, 94-98.
- CMMI. (30 de Septiembre de 2009). *CMMI*. Obtenido de <http://3ecubo.es/articulos/41>
- Colla, P. E. (2012). Marco para evaluar el valor en metodología SCRUM. *Argentine Symposium on Software Engineering, ASSE*, 32-46.
- de León Sigg Maria, V. C. (2020). Uso del Estándar ISO/IEC29110 para Entrenar Estudiantes en Procesos de Ingeniería de Software. *risti*, 60-72.
- Deming, W. (1986). *Calidad, Productividad y Competitividad. La salida de la crisis*. Madrid, España: Díaz de Santos, S.A.
- E. Turban, J. E.-P. (2005). Knowledge Management. In *Decision Support Systems and Intelligent Systems*, Pearson, Uppers Saddle River, NJ (U.S.A.): Prentice Hall,, 487.
- Egusquiza Herrada Hugo Jhonny, N. M. (2016). *Modelo de mejora del ciclo de vida del desarrollo de Softwae con referencia a la ISO/IEC 29110 Caso: Mype Holinsys*. Lima: Universidad de Lima.
- Evans M.W., S. C. (2005). The Myth of best practices silver bullet. *Crosstalk*.
- Fahey, L. (1989). *Discovering your firm's strongest competitive advantage*. editorial: In L. Fahey.
- Fernández, V. (2006). Desarrollo de Sistemas de Información. *UPC*, 175-176.

- Florac, W. a. (1999). *Measuring the software Process. Statistical Process Control for Software Process Improvement* . Addison Wesley.
- Florac, W. R. (1997). Practical Software Measurement: Measuring for Process Management and Improvement. *Pittsburgh, Software Engineering Institut, Camegie Mellon University*, 1-12.
- Gallego, M. T. (2012). Metodología SCRUM. En M. T. Gallego, *Manuel Trigas Gallego* (pág. 55). TFC.
- García-Ochoa Mónica, B. D. (2015). Ventajas competitivas de las empresas de Telefonía Móvil en América Latina, Análisis desde la perspectiva de los grupos estratégicos. *El trimestre Económico*, 88-116.
- Ghosh, S. (26 de 11 de 2012). *Internet*. Obtenido de <http://pm.umd.edu/files/public/documents/student-papers/2011/EVM%20in%20Water-fall%20and%20Agile%20Software%20Project%20by%20Sam%20Ghosh.pdf> [Acceso Junio 1, 2013].
- Gildardo, C. H. (2013). Ventajas competitivas de las empresas cárnicas en sinaloa. *Revista EAN*, 40-43.
- Goodwill Community Foundation, Inc. (12 de octubre de 2020). *GCF Aprende Libre*. Obtenido de GCF Aprende Libre: <https://edu.gcfglobal.org/es/informatica-basica/que-es-hardware-y-software/1/>
- Grant, R. (1996). Prospering In Dynamically-Competitive Enviroments: Organizational Capability As Knowledge Integration. *Organization Science*, 375-387.
- HPC, H. P. (4 de Agosto de 2006). *Software arquitectura, Technology Area*. Obtenido de Software arquitectura, Technology Area: [https://www.spider.hpc.navy.mil/index.cfm?RID=TTE\\_OT\\_1000025](https://www.spider.hpc.navy.mil/index.cfm?RID=TTE_OT_1000025)
- Humphrey, W. (1997). *Introduction to the Personal Software Process*. Massachusetts: Addison Wesley Logman, Inc.
- ISO/IEC 29119:2011. (2011). *Software engineering*. ISO.
- J, C. O., & Fuentes, M. d. (2012). Taxonomía de los modelos y metodologías de desarrollo de softare más utilizados . *Universidades*, 37-47.
- J. Cervantes Gómez, M. D. (2012). Taxonomía de los modelos y metodologías de desarrollo de Software más Utilizados. *Universidades UDUAL*, 37-47.
- J. Gupta, S. S. (2004). An overview of knowledge management. *Idea Group Inc.*, 1.
- J.R., Z. (1999). La gestión por procesos: Su papel e importancia en la empresa. *Economía Industrial*, 81-88.
- José H. Canos, P. L. (2012). Metodologías Ágiles en el Desarrollo de Software. *Repositorio institucional de la Universidad de las Tunas*, 1-8.
- Letelier, P., & Penadés, M. C. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *CYTA*, 1-5.

- Luis E. Mendoza, M. A. (2004). Prototipo de Modelo Sistemático de Calidad (MOSCA) del software. *Computación y Sistemas*, 196-217.
- Luis García Paucar, C. Y. (2015). Implementation and Certification of ISO/IEC 29110 in an IT Startup in Peru. *Software Engineering Processes*, 16-29.
- M. Dumas, W. V. (2005). *Process-Aware Information*. New Jersey: John Wiley & Sons, Inc.
- Madruñero Edwin, C. J. (2018). Implementación del estándar ISO/IEC 29110 en el proceso de desarrollo de software de la dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte. *UTN Repositorio Digital Universidad Técnica del Norte*, 1-6.
- Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de Software*. Argentina: UCA.
- Marsal, J. M. (2002). La gestión del conocimiento en las organizaciones. *ch. VII Herramientas de la gestión del conocimiento, CH IX gestión del cambio*, 60-68; 87-94.
- Mas Antónia, A. E. (2005). La mejora de los procesos de software en las pequeñas y medianas empresas (pyme). Un nuevo modelo y su aplicación a un caso real. *REICIS. Revista Española de Innovación, calidad e ingeniería del software*, 7-29.
- Mathison Luis, G. J. (2007). Innovación: Factor Clave para Lograr Ventajas competitivas. *NEGOTIUM*, 46-83.
- Mauricio, M. R. (2018). SAPPO. Sistema de autoevaluación en la nube del estándar ISO/IEC 29110 para pequeñas empresas de desarrollo de Software. *Actas del I congreso Internacional de Ingeniería de Sistemas*, 107-116.
- Mendoza, L. E., Pérez, M. A., & Grimán, A. C. (2005). Prototipo de Modelo Sistemático de Calidad (MOSCA) del software. *Computación y Sistemas*, 196-217.
- Miguel E. Morales-Trujillo, H. O. (2013). From MoProSoft Level 2 to ISO/IEC 29110 Basic Profile: Bridging the gap. *CLEI Electronic Journal*, 1-20.
- Muñoz Mirna, M. J. (2018). Implementación del Estándar ISO/IEC 29110 en Centros de Desarrollo de Software de Universidades Mexicanas: Experiencia del Estado de Zacatecas. *XI seminario Internacional de Ciencias de la Computación*, 2-12.
- Muñoz Mirna, M. J. (2018). Implementación del estándar ISO/IEC 29110 en centros de Desarrollos de software de Universidades Mexicanas: Experiencia del Estado de Guanajuato. *risti Revista Ibérica de Sistemas y Tecnologías de Información*, 43-54.
- Nancy R. Mead, R. E. (2000). Life-Cycle Models for survivable Systems. *Software Engineering Institute Carnegie Mellon University*, 1-5.
- Norberto Osorio Beltrán, G. C. (2011). Gestión de calidad en desarrollo de software. *Management of quality in software development*, 65-69.
- Ordoñez Hugo, C. C. (2019). ISO 29100 en Colombia: de la teoría a la práctica. *PRESS*, 1-15.
- Organización Internacional de Normalización ISO, Norma Internacional ISO 9000. (2005). Sistemas de gestión de calidad. *Sistemas de gestión de calidad - Fundamentos y Vocabulario*. Suiza.

- Organización Internacional de Normalización ISO, Norma Internacional ISO 9000. (2005). *Sistemas de gestión de la calidad - Fundamentos y Vocabularios*. Suiza. *Sistemas de gestión de la calidad - Fundamentos y Vocabularios*. Suiza.
- OSD Comptroller iCenter. (2004). *Best Practices & Benchmarking. Making Worthwhile Comparisons*, 9.
- Osterweil, L. (1987). *Procesos de software son software también*. *IEEE Computer Society Prensa*, 2-13.
- Penadés, L. P. (2012). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. *Laboratorio de Sistemas de Información. Departamento de Sistemas Informáticos y Computación. Facultad de Informática. Universidad Politécnica de Valencia*, 1-17.
- Pesado Patricia, B. R. (2013). *Mejora de Procesos en el desarrollo de Sistemas de Software y en Procesos de Gestión. Experiencias en PyMEs. XV Workshop de investigadores en ciencias de la computacion* , 581-585.
- Pino Correa Francisco J., P. A. (2018). *ISO/IEC 29110 Para procesos de software en las pequeñas empresas*. Madrid: AENOR Internacional, S.A.U.
- Pino Correa Francisco J., P. A. (2018). *ISO/IEC 29110 Para procesos software en las pequeñas empresas*. España: AENOR.
- PMI Project Management Institute. (2013). *PMBok- A guide to the Project Management Body of Knowledge*.
- Porter, M. E. (1990). *Ventaja Competitiva: creación y sostenimiento de un desempeño superior*. . San Juan Tlahuac, México D.F.: Patria.
- Porter, M. E. (1991). *La ventaja competitiva de las naciones*. Buenos Aires: Vergara.
- Pressman, R. (2002). *Ingeniería del Software. Un enfoque práctico*. España: McGraw-Hill Interamericana.
- Qualitrain. (30 de Septiembre de 2009). *Beneficios del proceso de Software bajo el modelo CMMI*. Obtenido de <http://www.qualitrain.com.mx/index.php/Procesos>
- R., G. F. (2011). *Modelo en espiral de un proyecto de desarrollo de Software*. *UNEMI Universidad Estatal De Milagro*, 1-9.
- Rinaldo, G. P. (201). *Ingeniería de Software* . México: Alfaomega.
- Rodríguez, C., & Dorado, R. (2015). *¿Porqué implementar SCRUM?* *ONTARE*, 125-144.
- Roger, P. S. (2010). *Ingeniería del Software Un Enfoque Práctico*. México, D.F.: McGraw-Hill Interamericana Editores S.A DE C.V.
- Ross, P. (1993). *Total Quality Management*:. Delray Beach: ST. Lucie Press.
- Saderra, I., & Jorba, L. (1993). *El secreto de la calidad Japonesa*. Barcelona: Marcombo.

- Sandro Javier Bolaños Castro, S. A. (2012). LMPS como propuesta alterna a BPMN para el modelado de procesos de software. *Tecnura*, 157-170.
- Sila, I. &. (2002). An investigation of the total quality management survey based research published between 1989 and 2000: a literature review. *International Journal of Quality and Reliability Management*, 902-970.
- Sommerville, I. (2005). *Ingeniería del software*. Madrid: Pearson Educacion S.A.
- Sommerville, I. (2005). *Ingeniería del software*. Madrid: Pearson Education, S.A.
- Sommerville, I. (2005). *Ingeniería del Software,, Séptima edición*. Madrid: Pearson Educación.
- T. Bhuvaneswari, S. P. (2013). A Survey on Software Development Life Cycle Models. *International Journal of Computer Science and Mobile Computing*, 262-267.
- Tobón, L. M., & Carmona, L. E. (2007). *Caso Práctico de la metodología ágil XP al desarrollo de Software*. Pereira: Universidad Tecnológica de Pereira.
- United Kingdom Government. (2011). *The Green Book: appraisal and evaluation in central government. Treasury Guidance*. London: HM Treasury, Original edition.
- Web Service Business. (2007). *Process Execution Language Version 2.0*. OASIS Standard.
- Williams, T. (2008). How do organizations Learn Lesson From Projects And Do They? *IEEE Transactions on Engineering Management*, 55, 248-266.