

FORMULACIÓN DEL MODELO DE GESTIÓN DE PROCESOS, BAJO EL ENFOQUE DE  
ASEGURAMIENTO DE LA CALIDAD, BASADO EN EL CICLO DE MEJORA CONTINUA  
PHVA DE EDWARDS DEMING, PARA EL LABORATORIO DE LA INDUSTRIA  
ACADÉMICA EN DESARROLLO DE SOFTWARE, PARA LA FACULTAD DE  
INGENIERIA DE LA UCO.

LINA MARÍA GIRALDO CASTAÑO  
MARTÍN DAVID MOSQUERA PÉREZ.

UNIVERSIDAD CATÓLICA DE ORIENTE  
FACULTAD DE INGENIERÍAS  
INGENIERÍA INDUSTRIAL  
RIONEGRO ANTIOQUIA

2019

FORMULACIÓN DEL MODELO DE GESTIÓN DE PROCESOS, BAJO EL ENFOQUE DE  
ASEGURAMIENTO DE LA CALIDAD, BASADO EN EL CICLO DE MEJORA CONTINUA  
PHVA DE EDWAR DEMING PARA EL LABORATORIO DE LA INDUSTRIA  
ACADÉMICA EN DESARROLLO DE SOFTWARE, PARA LA FACULTAD DE  
INGENIERIA DE LA UCO.

LINA MARÍA GIRALDO CASTAÑO  
MARTÍN DAVID MOSQUERA PÉREZ.

Trabajo de grado para optar por el título de:

INGENIERO INDUSTRIAL

Asesores:

Luz Mery Ríos Álzate

Diego Andrés Aguirre Cardona

UNIVERSIDAD CATÓLICA DE ORIENTE

FACULTAD DE INGENIERÍAS

INGENIERÍA INDUSTRIAL

RIONEGRO ANTIOQUIA

2019

## DEDICATORIA

Esta tesis la dedico a:

Primeramente, dedico este logro a Dios por haberme dado la vida, a mis padres Walter Mosquera López y María Ninelly Pérez Hernández por su inigualable esfuerzo, amor, y comprensión que me han permitido cumplir uno de mis más grandes sueños; además por siempre confiar en mí y creer en todas mis expectativas, por sus sabios consejos y sobre todo por día a día inculcar en mí que siempre a pesar de las adversidades las cosas se pueden lograr teniendo presente que Dios está conmigo hasta en medio de los momentos más difíciles.

A mi hermana Sara Manuela Mosquera Pérez por su cariño incondicional y por brindarme todo su apoyo durante mi proceso de formación. A cada uno de los miembros de mi familia, y amigos que siempre creyeron en mí, y siempre con sus consejos, sus oraciones hicieron de mí una mejor persona y me acompañan en el cumplimiento de mis metas.

Además, en esta dedicatoria no quiero dejar pasar por alto el nombre de mi abuela María Cenelia López López, porque en su vida siempre creyó en mí creando en su interior un orgullo inmenso por cada uno de mis logros, y aun desde el cielo guía mis pasos por el buen camino, y me llena de bendiciones que hacen que cumpla cada uno de mis objetivos.

Finalmente, no quiero pasar por alto el dedicar también este logro a la Fundación Club Campestre que desde mis 15 años me abrió sus puertas, brindándome experiencias únicas e irrepetibles con las que hoy puedo decir que han hecho algo muy grande por mí, formándome en valores y en estilos de vida que hacen de mí un hombre de bien para la sociedad.

*MARTIN DAVID MOSQUERA PEREZ.*

Esta tesis la dedico a:

Primeramente, a Dios por darme la posibilidad de estudiar y vencer todos los obstáculos, a mi Padre José Leonel Giraldo Zuluaga que desde el cielo sé que se siente orgulloso de mis logros alcanzados, a mi madre Martha Castaño Ocampo por su inigualable apoyo, paciencia, y amor, que me ha permitido cumplir siempre con mis sueños y metas, y a mi hermano Duván Giraldo Castaño, por siempre estar a mi lado, por su sacrificio y comprensión porque gracias a él soy profesional en Ingeniería Industrial. Finalmente quiero dedicar a todos mis amigos y familiares que creyeron en mí y siempre me apoyaron en las más duras adversidades.

*LINA MARIA GIRALDO CASTAÑO*

## AGRADECIMIENTOS

Agradecemos a Dios primeramente por habernos dado la vida, por siempre guiarnos por el buen camino a lo largo de nuestras vidas, y por siempre estar con nosotros en los momentos más tensos dándonos fortaleza.

Gracias a nuestros padres porque son el eje y pilar fundamental de nuestros sueños, por siempre confiar en la grandeza de nuestras metas, y porque a través de sus valores, concejos y principios que nos han inculcado han hecho de nosotros personas prosperas en nuestro día a día.

Queremos agradecer a cada uno de los docentes de la facultad de ingenierías que estuvieron con nosotros durante todo el proceso de formación, quienes a través de la transmisión de sus conocimientos no solo inculcaron temas para nuestras carreras, sino algo muy importante que es el “ser primero humano, luego profesional”; de manera especial queremos agradecer a los docentes Luz Mery Ríos Álzate y Diego Andrés Aguirre Cardona quienes fueron los tutores de nuestro proyecto de investigación y con su constante acompañamiento, paciencia y dedicación hicieron que esto fuera posible.

## CONTENIDO

	Pág.
ANTECEDENTES	12
1. PLANTEAMIENTO DEL PROBLEMA	15
2. PREGUNTA DE INVESTIGACIÓN	17
3. JUSTIFICACIÓN	18
4. OBJETIVOS	21
4.1. General	21
4.2. Específicos	21
5. MARCO TEORICO	22
5.1. Procesos	22
5.2. Enfoque de procesos	22
5.3. Gestión	22
5.4. Modelo de gestión	22
5.5. Modelos de gestión de procesos	22
5.6. Calidad	22
5.7. Ciclo de mejora continua PHVA	23
5.8. Estándares de calidad	24
5.9. Software	24
5.10. Desarrollo de software	25
5.11. Ciclo de vida del software	25
5.12. Actividades generales del proceso de desarrollo de software	26
5.13. Especificaciones del software	27
5.14. Diseño e implementación del software	28
5.15. Validación del software	29
5.16. Metodologías ágiles para el desarrollo de Software	30
5.16.1 <i>Principales valores de una metodología ágil</i>	31
5.17. ISO 12207: 2017 Ingeniería de sistemas y software. Procesos del ciclo de vida del software.	31
5.18. CMMI Buenas prácticas para el desarrollo de software	31
6. METODOLOGIA	39

6.1. Alcance	39
6.2. Descripción detallada de la metodología	39
7. RESULTADOS	41
7.1. Objetivo	41
7.1.1 <i>Relación de la norma ISO/IEC 12207 y CMMI</i>	41
8. CONCLUSIONES	82
REFERENCIAS BIBLIOGRÁFICAS	84
ANEXOS	89

## LISTA DE ILUSTRACIONES

	Pág.
Ilustración 1. Componentes del proceso de software	24
Ilustración 2. Actividades básicas del proceso de desarrollo de Software.	26
Ilustración 3. Especificación del Software	27
Ilustración 4. Etapas del diseño e implementación del Software	28
Ilustración 5. Validación del Software	30
Ilustración 6. Áreas de proceso para alcanzar las buenas prácticas	32
Ilustración 7. Procesos de la norma ISO/ IEC 12207	41
Ilustración 8. Procesos de acuerdo (ISO/IEC 12207)	42
Ilustración 9. Proceso de acuerdo (CMMI)	43
Ilustración 10. Procesos de soporte (ISO/IEC 12207)	44
Ilustración 11. Procesos de soporte (CMMI)	46
Ilustración 12. Procesos técnicos (ISO/IEC12207).	48
Ilustración 13. Procesos técnicos (CMMI)	49
Ilustración 14. Gestión de requisitos	51
Ilustración 15. Integración del producto	53
Ilustración 16. Proceso de verificación	55
Ilustración 17. Proceso de validación.	56
Ilustración 18. Proceso de gestión técnica	58
Ilustración 19. Proceso de planificación del proyecto	60
Ilustración 20. Proceso de evaluación y control del proyecto	62
Ilustración 21. Análisis de decisiones y resolución.	64
Ilustración 22. Gestión de riesgos	65
Ilustración 23. Proceso de gestión de la configuración	66
Ilustración 24: Enfoque en procesos de la organización	67
Ilustración 25: Roles de los desarrolladores de Software en la actualidad.	90
Ilustración 26: Enfoque de los desarrollos de Software en la actualidad	91
Ilustración 27: Metodologías implementadas en la actualidad	92
Ilustración 28: Uso de validaciones sobre la metodología usada dentro del desarrollo de proyectos	92



Ilustración 29: Programas para desarrollo de Software	100
Ilustración 30: Lenguajes de programación	101
Ilustración 31: Nivel inicial de modelos de calidad de Software.	104
Ilustración 32: Nivel administrativo de modelos de calidad de Software	105
Ilustración 33: Nivel definido de modelos de calidad de Software.	106
Ilustración 34: Nivel administrados cuantitativamente de modelos de calidad de Software.	107
Ilustración 35. Proceso metodología SCRUM	114
Ilustración 36: Proceso metodología Crystal	115
Ilustración 37: Dynamic Systems Development method (DSDM)	116
Ilustración 38: Adaptive Software Development (ASD)	117
Ilustración 39: Feature- Driven Development (FDD)	118
Ilustración 40: Características norma ISO/ IEC 12207	120
Ilustración 41: Procesos de la norma ISO/ IEC 12207	121
Ilustración 42: Factores de calidad que se aseguran bajo la ISO/ IEC 12207	122

## LISTA DE TABLAS

	Pág.
Tabla 1. Relación norma ISO 12207 VS guía CMMI.	34
Tabla 2. Descripción actividades metodológicas.	39

## LISTA DE ANEXOS

	Pág.
Anexo 1. Procesos de desarrollo de software hoy	89
Anexo 2. Definiciones de calidad de varios autores.	113
Anexo 3. Metodologías ágiles	114
Anexo 4. Norma ISO/ IEC 12207.	120

## ANTECEDENTES

En la actualidad el desarrollo de software ha sido una de las tantas líneas de procesos existentes que ha tenido un gran apogeo, esto debido a que las compañías están en la exploración constante de estandarizar y mejorar sus procesos, volviéndolo más sistemáticos y eficientes. En Colombia existen varias universidades dedicadas al desarrollo de software, donde se encuentran espacios prácticos de enseñanza relacionados al programa de Ingeniería de Sistemas que le sirve de guía al estudiante para crear, implementar y administrar diferentes soluciones de software que contribuyan al desarrollo de empresas tanto públicas como privadas, buscando aumentar los indicadores de productividad y optimizar diversos procesos, además de crear valor agregado y promover buenas prácticas con el objetivo de presentar soluciones que se ajusten a las principales problemáticas que se presenten en las diferentes operaciones diarias en la industria, y lo más importante, que esas soluciones que se planteen además de suplir las necesidades, sean soluciones o alternativas que aseguren la calidad en los diferentes procesos. En Colombia, los principales laboratorios de desarrollo de Software se encuentran localizados en ciudades como Villavicencio, Bucaramanga, y Medellín.

Es importante destacar que más que desarrollar un software que mitigue alguna problemática de la industria, ha sido y sigue siendo pieza clave tener en el día a día como eje fundamental la gestión por procesos y dicha gestión debe estar soportada u orientada por un modelo que dé pista al enfoque y alcance de los diferentes procesos; es por esto que la gestión por procesos se ha considerado por tener un rol administrativo dentro de las organizaciones, y la evolución de esta a través de los tiempos se encuentra encaminada a la administración por su constante búsqueda tratando de encontrar la manera más eficiente y eficaz para alcanzar los objetivos y metas que tiene cada organización.

Por otro lado, la gestión por procesos se dice que dio origen en la primera y segunda etapa de la revolución industrial cuando los capitalistas identificaron la necesidad de incrementar sus riquezas llevando a realizar una división del trabajo y la especialización con lo que se dio origen a las llamadas estructuras organizacionales y dando origen además al mecanismo como teoría organizacional.

La gestión por procesos de manera paralela se ha desarrollado con la calidad y ambas se trabajan en función de las cambiantes necesidades que presentan las organizaciones buscando la satisfacción del cliente quien con el pasar de los años cada vez cambia y exige más en los productos o servicios de los cuales obtiene beneficios propios.

Entrando más en la línea del tiempo, en la década de 1950 el departamento de defensa de los Estados Unidos de América (USA) en conjunto con la NASA desarrollaron el concepto de aseguramiento de la calidad pensando en la responsabilidad que tienen las compañías por la conformidad de los productos, y dicha responsabilidad se extiende en todas las áreas de la empresa que pueden afectar el resultado para el cliente. El aseguramiento de la calidad se basa del uso de diferentes métodos o herramientas para apoyar todo lo relacionado con el alcance de resultados de calidad con alto índice de eficiencia y eficacia buscando generar una gran confianza en el cliente y consumidor final.

A media de que el departamento de defensa de la USA y la NASA enfatizaban en desarrollar el concepto de calidad el señor Edwar Deming introdujo en Japón el ciclo Planear, Hacer, Verificar y Actuar más conocido como “Ciclo PHVA” con el fin de tener una herramienta para mejorar y gestionar las actividades de las organizaciones; en la actualidad este ciclo es utilizado por gran cantidad de organizaciones como una herramienta de gran ayuda para actuar e intervenir sobre los procesos, no sobre personas, generando estrategias de mejora continua y

viendo en este ciclo una oportunidad de obtener una fácil adaptación a un entorno cambiante y de constante evolución como el que hoy enfrentan la diferentes empresas.

Finalmente, tomando como referente toda la información presentada en la sesión de antecedentes es importante resaltar el complemento fundamental de la formulación de un modelo de gestión de procesos con un enfoque de aseguramiento de la calidad basado en el ciclo PHVA dentro del proyecto de un laboratorio de la industria del software para el programa de ingeniería de sistemas de la facultad de ingenierías de la Universidad Católica de Oriente (UCO).

El complemento entre los componentes anteriormente mencionados es bastante amplio puesto que comprende como principal actor al cliente y la importancia del mismo dentro de las actividades de la organización, en este caso dentro del laboratorio de desarrollo de Software, además la formulación del modelo de gestión de procesos en este caso permite definir un flujo de trabajo eficiente y adaptado a necesidades en lo que comprende desarrollo de software, este modelo paralelo al ciclo PHVA aporta a la organización, para este caso al laboratorio a constantemente realizar un feedback y constante evaluación de posibles mejoras en las operaciones que se lleven a cabo; además ayuda a centralizar el manejo de todo el proceso dentro de una sola herramienta en la que se controlan y se definen las diferentes responsabilidades, roles, actividades que se desarrollan y la medición de los resultados obtenidos.

Además, apoyado en el ciclo PHVA el modelo de gestión de proceso se plantea para que este sea bastante efectivo dentro de la organización en lo referente al manejo y disposición de documentación y, sobre todo trabajar de manera estandarizada para reducir tiempos y costos para que la organización pueda centrarse en su principal grupo de interés, que para este caso son los clientes.

Mediante una encuesta realizada a las diferentes empresas desarrolladoras de software de Medellín se encontró lo siguiente, *ver el Anexo 1*.

## 1. PLANTEAMIENTO DEL PROBLEMA

La Universidad Católica de Oriente requiere espacios físicos y de gestión para el desarrollo de diversas aplicaciones de software que se ajusten a sus necesidades de carácter interno, dentro de la academia donde se encuentran desarrollos de programas con diferentes aplicabilidades por parte de los estudiantes de semestres avanzados en ingeniería de sistemas; y que adicionalmente atiendan expectativas de carácter externo tales como el desarrollo de prácticas, consultorio empresarial, procesos de innovación en desarrollo de software, articulación e integración del impulso del proceso académico con el proceso de investigación en la formación de las disciplinas de la ingeniería, entre otros.

Adicionalmente los estudiantes de Ingeniería de Sistemas y otros programas de pregrado y postgrado tienen dificultades al momento de realizar la entrega de su trabajo de grado, esto debido en gran parte a la carencia de diferentes programas en los cuales los estudiantes puedan dar cierto grado de aplicabilidad al conocimiento adquirido durante todo el proceso de formación; siendo este motivo de postergación de las graduaciones de cierto número de estudiantes.

Se centra entonces la problemática, en la carencia de una estructura que permita unir las necesidades ya mencionadas para posibilitar tanto a los estudiantes como a los usuarios externos un buen manejo de los productos desarrollados en materia de Software.

La principal idea para de cierto modo dar solución a la problemática ya mencionada es formular un modelo de gestión para el laboratorio de desarrollo de software para la universidad, teniendo como base el ciclo PHVA, integrar un modelo de desarrollo de acuerdo con las tendencias de empresas de desarrollo incluyendo buenas prácticas de CMMI.

Con lo anterior, se piensa dar un aporte significativo al proyecto matriz que consiste en la implementación de un laboratorio de desarrollo dentro de las instalaciones de la Universidad Católica de Oriente, proponiendo un modelo que se pueda ver como una herramienta de gestión

que tiene como fundamento un enfoque basado en los diferentes procesos lo que lleva a que durante la transformación de actividades en resultados se tengan productos que contengan un resultado final eficiente desde el punto de vista de la implementación y la calidad.

Además, con el modelo propuesto se lanza una visión a una constante normalización o estandarización de los diferentes procedimientos buscando tener un proceso que genere un valor agregado sobre las entradas para finalmente luego de realizar todo un paso a paso en un resultado viendo a la organización como un conjunto de procesos interconectados.



## 2. PREGUNTA DE INVESTIGACIÓN

¿Cómo formular un modelo de gestión de proceso aplicado a un laboratorio de la industria académica en desarrollo de software, para la facultad de ingeniería de la UCO, bajo el enfoque de aseguramiento de la calidad, apoyado por el ciclo de mejora continua PHVA de Edward Deming?

### 3. JUSTIFICACIÓN

El proyecto traerá beneficios para la Universidad puesto que las dinámicas de desarrollo de software están a la orden del día para numerosas necesidades en la población en general.

Específicamente beneficiaría a:

- Los estudiantes en general:

Podrán aspirar a la realización de proyectos del banco de ideas, para completar su plan de estudios en lo referente a trabajo de grado. Esto se logrará bajo la autorización del comité de currículo y su certificación por el concejo de facultad. Todos los proyectos de grado deben ser guiados bajo el grupo GIMU y debe estar inscritos ante I+D UCO.

- Grupo de profesores de tiempo completo:

Son los profesores que permanecen un total de 40 horas semanales en funciones académicas y administrativas del programa. Todos los docentes del programa están llamados a apoyar la investigación desde diferentes frentes.

Anualmente se deben generar al menos diez proyectos que se deben colocar en el banco de ideas. Estas ideas no tienen lineamientos específicos, son justamente pequeñas propuestas para ser tenidas en cuenta al momento de formular los proyectos. En este punto toda la comunidad puede aportar ideas, las cuales serán moderadas por los profesores de tiempo completo del programa.

- Empresas del sector:

La relación Universidad – Empresa – Estado se puede materializar al involucrar al sector productivo público o privado a los proyectos de investigación que adelante el programa. De esta forma, se materializan las necesidades del sector y se buscan alternativas de solución conjunta a las problemáticas trazadas. También se potencia el intercambio de buenas prácticas y de mejora

continúa en el cual sea posible buscar ayudas con las empresas del sector, proceso en el que se puede aprender y aportar.

Por otro lado, es importante mencionar de manera más clara la razón de ser del proyecto, lo cual de cierto modo da más credibilidad o justifica la importancia y el porqué es importante la implementación de un laboratorio de desarrollo de software visto desde la perspectiva interna y externa.

De lo anterior se desprenden un número de factores que deben ser considerados como el corazón o razón de ser del proyecto; primeramente se piensa en la importancia de que los estudiantes puedan simular un ambiente empresarial desde lo académico mediante la realización o ejecución de diferentes proyectos de desarrollo de software de tal manera que cuando los estudiantes se vean enfrentados a un ambiente real tengan nociones y por medio de esto los mismos se pueden destacar en un ambiente empresarial una vez terminen su ciclo académico, de este primer factor cabe resaltar que de esto pueden surgir diferentes proyectos de grado por lo que es fundamental que los docentes brinden un acompañamiento constante a los estudiantes mientras se ejecutan los proyectos que se lleven a cabo.

Como segundo ítem se piensa que otra de las razones que se pueden planear para justificar el proyecto es por la importancia de generar espacios interdisciplinarios donde la ingeniería de sistemas como principal actor se pueda unir con otras carreras para complementarse en la ejecución de proyectos enfocados en el desarrollo de software y calidad de este.

Además, el realizar la formulación del modelo de gestión de procesos bajo el enfoque de aseguramiento de la calidad basado en el ciclo PHVA, permite orientar al proyecto matriz en lo relacionado con la sistematización de los procesos con el objetivo de lograr los resultados esperados.

Finalmente, al realizar la formulación del modelo tomando como base el ciclo PHVA se busca tener calidad en los procesos mediante la mejora continua teniendo como resultado una cantidad mínima de errores, de defectos y reprocesos lo que lleva a minimizar tiempos de ciclo de producción reduciendo el consumo de recursos que es directamente proporcional con la disminución de costos en las operaciones.

## 4. OBJETIVOS

### 4.1. General

Formular un modelo de gestión de proceso para el laboratorio de la industria académica en desarrollo de software, para la facultad de ingeniería de la UCO, bajo el enfoque de aseguramiento de la calidad, basado en el ciclo de mejora continua PHVA de Edward Deming.

### 4.2. Específicos

- Investigar sobre modelos de desarrollo software, modelos de calidad, y herramientas de uso gratuito, que sean aplicables a laboratorios de desarrollo de software en entornos de educación terciaria.
- Diseñar un modelo de gestión de proceso para el laboratorio de la industria académica en desarrollo de software para la facultad de ingenierías de la Universidad Católica de Oriente.
- Plantear el modelo de gestión de proceso para el laboratorio de la industria académica en desarrollo de software para la facultad de ingenierías de la Universidad Católica de Oriente.

## 5. MARCO TEORICO

### 5.1. Procesos

Son un conjunto de acciones, o actividades que determinan una trayectoria de un estado inicial y un estado final de algún cuerpo, sistema o ambiente, donde se genera una transformación, desplazamiento o cambio de este.

### 5.2. Enfoque de procesos

Está basado en mejorar la eficiencia y eficacia de la organización para lograr los objetivos definidos, donde implica aumentar la satisfacción del cliente mediante el cumplimiento de los requisitos. Permitiendo controlar y gestionar el valor de las actividades enfocadas en el cliente.

### 5.3. Gestión

Es entrelazar las funciones para la creación de políticas corporativas, donde le permita organizar, planear, controlar y dirigir los recursos de la organización enfocados al cumplimiento de los objetivos de la política establecida.

### 5.4. Modelo de gestión

Es un esquema utilizado para la administración de entidades públicas como privadas, para alcanzar los objetivos planteados, mediante el desarrollo de políticas y acciones.

### 5.5. Modelos de gestión de procesos

Este modelo permite que se abandonen los modelos tradicionales de gestión, y se adopten al modelo basado en la mejora continua y la consecución de nuevos objetivos.

### 5.6. Calidad

Permite a las empresas diferenciarse de las demás, volviéndose más competitivas y obtener mejores resultados, favoreciendo al crecimiento y desarrollo empresarial, buscando satisfacer las necesidades y requerimientos de un grupo objetivo.

El aseguramiento de la calidad es la actividad que proporciona evidencia de que se puede confiar en que la función calidad se ha llevado a cabo con efectividad. Existen 7 enfoques de calidad, los cuales son:

1. **Enfoque al cliente:** Es entender a nuestros clientes, sus necesidades y expectativas.
2. **Enfoque estratégico:** Deben tratar esto como un objetivo estratégico clave, para ello se debe dar la creación de una visión estratégica y la implementación de ésta en toda la empresa.
3. **Liderazgo Enfocado:** El compromiso de los líderes, su conducción activa de la estrategia y un compromiso positivo constante con su aplicación.
4. **Enfoque en procesos:** Los resultados son impulsados por la aplicación efectiva de los procesos correspondientes.
5. **Orientación a las personas:** Los procesos sólo son eficaces en la entrega de valor para el cliente si se asocian con comportamientos apropiados de los individuos involucrados.
6. **Enfoque científico:** La gestión de la calidad se basa fundamentalmente en el método científico: Planificar, Hacer, Verificar y Actuar.
7. **Mejora Continua, Innovación y Aprendizaje:** La mejora de procesos en una organización, no se trata simplemente de responder a problemas se trata de buscar soluciones en forma proactiva.

Ver anexo 2, donde se definen la calidad mediante otros autores.

#### 5.7. Ciclo de mejora continua PHVA

El ciclo PHVA fue presentada por Deming a partir del año 1950, el cual consta de 4 pasos; planear, hacer, verificar y actuar. Los resultados de la implementación de este ciclo en las empresas se ven reflejados en la mejora integral de la competitividad, de productos y servicios, mejorando continuamente la calidad.

### 5.8. Estándares de calidad

Es un referente que facilita las tareas y define las condiciones mínimas para que los aspectos y características de un producto y/o servicio cumplan con los requisitos de los clientes y los satisfagan de manera eficiente. Además, ofrece las modificaciones de mejora para el producto, proceso o servicio.

Existen tres tipos de estándares:

- Estándares oficiales: son aprobados por un organismo de estandarización.
- Estándares facto: son elaborados por un comité, orientados por una entidad u organismo.
- Estándares de acuerdo: son definidos por un pacto entre proveedores, productos y usuarios.

### 5.9. Software

Se refiere a un programa o conjunto de programas, como procedimientos, datos y pautas que permiten realizar distintas tareas en un sistema informático.

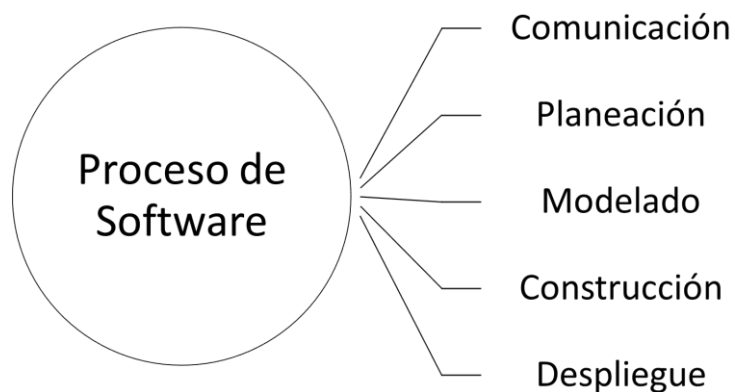


Ilustración 1. Componentes del proceso de software. Fuente: Elaboración propia.



- En la comunicación se tiene como propósito definir los requerimientos, la idea es trabajar con los clientes de manera conjunta. De acuerdo con esto es una actividad que se realiza durante todo el proyecto y su frecuencia varía de acuerdo con el proyecto y los clientes.

- La planeación no es exclusiva del proceso de software cualquier proceso requiere la determinación de las actividades, las acciones y las tareas, entendiendo por actividad algo que busca un objetivo amplio, por acción diferentes tareas que generan un producto importante del trabajo y la tarea se enfocan en un objetivo pequeño, pero bien definido que produce un resultado tangible.

- En el modelado se diseña el software, se utiliza los patrones, en esta actividad se realizan los planos del software.

- En la construcción se genera el código, de acuerdo con el modelado.

- En el despliegue se entrega al consumidor para que este lo evalúe y le dé retroalimentación, que se basa en dichas evaluaciones.

Las cinco actividades estructurales genéricas mencionadas anteriormente, se usan durante el desarrollo de programas pequeños y sencillos, en la creación de aplicaciones web grandes y en la ingeniería de sistemas enormes y complejos basados en computadoras.

#### 5.10. Desarrollo de software

- Para el desarrollo del software se interviene con varias personas las cuales cumplen primero con el análisis del sistema donde llegan todas las necesidades y requerimientos del cliente, ya con las necesidades establecidas el programador hace el diseño y codificación del software donde posteriormente se prueba, aprueba y se instala.

#### 5.11. Ciclo de vida del software

- Describe desde el inicio del software hasta el final, también los procesos intermedios de verificación y validación del procedimiento del software para el cumplimiento de

los requisitos establecidos por el cliente. También permite que los errores se detecten lo antes posibles, y por lo tanto garantizar la calidad del software.

### 5.12. Actividades generales del proceso de desarrollo de software

Las cuatro actividades básicas del proceso de desarrollo de software son: especificación, diseño e implementación, validación y evolución, ya cada una de las diferentes metodologías organizan las actividades de forma distinta, en la siguiente ilustración **¡Error! No se encuentra el origen de la referencia.** se muestra las actividades básicas y el flujo, teniendo en cuenta que la evolución es la adaptación del software al medio, que la entrega no es proceso final si el inicio de la evolución.

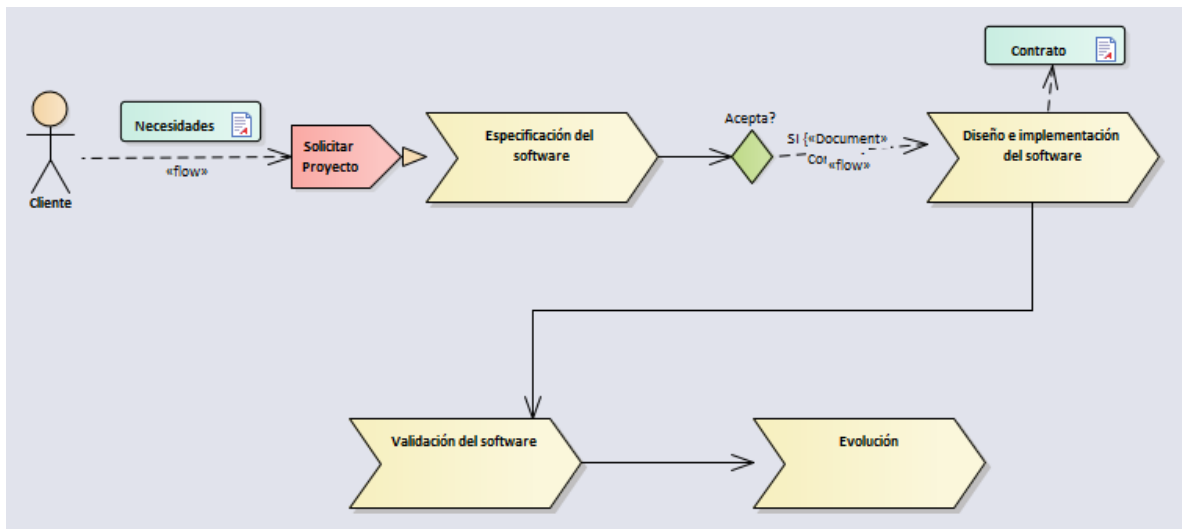


Ilustración 2. Actividades básicas del proceso de desarrollo de Software. Fuente: Elaboración propia.

El proceso descrito en la ilustración representa cada fase necesaria para lo que requiere la estructuración y llegar a la funcionalidad de un Software según unos requerimientos; este proceso inicia con la determinación de una serie de necesidades las cuales son identificadas entre las partes interesadas, posteriormente, en base a los requerimientos de Software se realiza una solicitud del proyecto lo cual lleva a crear unas especificaciones enfocadas en las necesidades que se tienen, si se aceptan las especificaciones y requerimientos que tiene el cliente se avanza a la

etapa que contempla todo el diseño e implementación de un Software para una final validación y evolución del Software; dicho proceso constantemente es retroalimentado con un enfoque en el ciclo PHVA (Planificar, hacer, verificar y actuar) buscando siempre una mejora continua y con una visión de que el cliente final obtenga un Software optimo el cual se ajuste a sus necesidades y requerimientos.

### 5.13. Especificaciones del software

Consiste en el proceso de comprensión y definición de los servicios que se requieren dentro del sistema y de identificación de las restricciones de funcionamiento y desarrollo de este. Esta es la etapa más crítica del proceso de desarrollo de software.

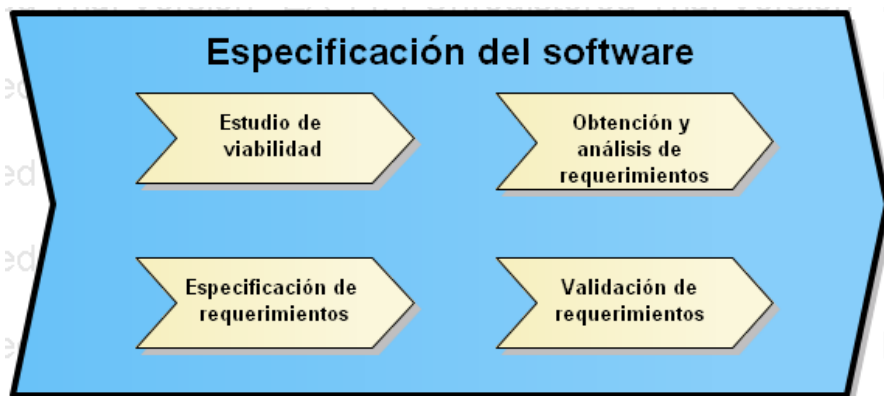


Ilustración 3. Especificación del Software. Fuente: Elaboración propia.

La especificación del Software está compuesta por otras 4 fases: Estudio de viabilidad. Se estudia si el proyecto es viable técnica, financiera y operativamente; la Obtención y análisis de requerimientos. Es la aplicación de técnicas de recolección con el fin de entender mejor el proceso y la Especificación de requerimientos es donde se traduce la información recopilada durante la actividad de análisis en un documento que define un conjunto de requerimientos y la Validación de requerimientos se comprueba la veracidad, consistencia y completitud de los requerimientos. Durante este proceso, inevitablemente se descubren errores en el documento de requerimientos. Se debe modificar entonces para corregir estos problemas.

#### 5.14. Diseño e implementación del software

La fase de implementación del desarrollo de software consiste en convertir la especificación del sistema en un sistema ejecutable. Esto implica los procesos de diseño y programación de software, pero, si se emplea un enfoque evolutivo de desarrollo, también puede implicar un refinamiento de la especificación del software.

Un diseño de software consiste en realizar una descripción de la estructura del software que se va a desarrollar, los datos que son parte del sistema, las interfaces entre los componentes del sistema y, algunas veces, los algoritmos utilizados.

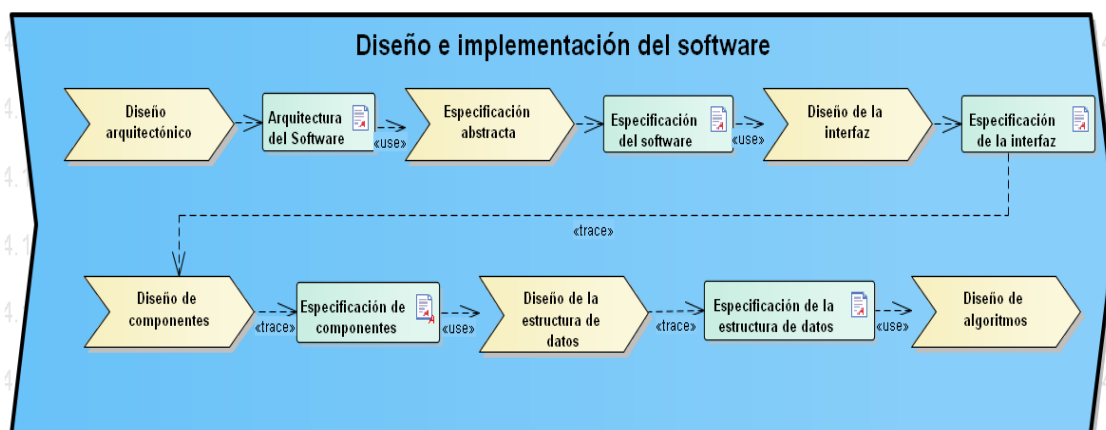


Ilustración 4. Etapas del diseño e implementación del Software. Fuente: Elaboración propia.

Como se presenta en la ilustración, esta etapa está compuesta por una serie de actividades bastante extensas que definen lo siguiente:

1. Diseño arquitectónico. Cada uno de los subsistemas que constituyen el sistema y sus relaciones se identifican y documentan.
2. Especificación abstracta. Para cada uno de los subsistemas se realiza una descripción abstracta de sus servicios y las restricciones bajo las cuales debe operar.

3. Diseño de la interfaz. Para cada subsistema se diseña y documenta su interfaz con diferentes subsistemas. Esta descripción de la interfaz debe ser inequívoca, puesto que esta permite que el subsistema se emplee sin conocimiento de su funcionamiento.

4. Diseño de componentes. Se asignan diferentes servicios a cada uno de los componentes y se diseñan sus respectivas interfaces.

5. Diseño de la estructura de datos. Se diseña con un profundo detalle y se realiza la respectiva especificación de la estructura de datos empleada en la ejecución del sistema.

6. Diseño de algoritmos. Se diseñan en detalle, para posteriormente, especificar los algoritmos utilizados para suministrar los servicios.

La programación es una actividad personal, y no existe un paso a paso que se siga comúnmente.

Algunos programadores, inician sus proyectos con los componentes que comprenden, los desarrollan y luego continúan con los que interpretan con menor grado de apropiación. Otros toman una orientación opuesta, trabajando con los componentes que son más familiares hasta el final, puesto que saben cómo desarrollarlos. Algunos desarrolladores optan por determinar los datos al inicio del proceso y estos los usan para conducir el desarrollo del programa; otros dejan los datos sin especificar tanto como sea posible.

#### 5.15. Validación del software

La validación del software o, empleando una forma más general, la verificación y validación (V & V) es empleada para plasmar que el sistema se ajusta a su respectiva especificación y que este cumple las expectativas del interesado.

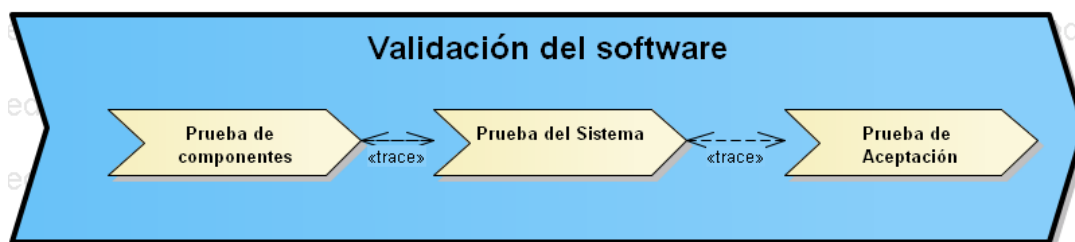


Ilustración 5. Validación del Software. Fuente: Elaboración propia.

El proceso representado en la ilustración, o también conocido como proceso de pruebas consta de tres fases, en las cuales se prueban cada uno de los componentes del sistema, la integración de este, y finalmente, el sistema con los datos del cliente. Las etapas del proceso de pruebas se definen de la siguiente manera:

1. Prueba de componentes (o unidades). En esta fase, se prueban los componentes individuales para asegurar que estos funcionan correctamente.
2. Prueba del sistema. Cada uno de los componentes se integran para dar forma al sistema. Este proceso se enfoca en encontrar errores que son el resultado de diferentes interacciones no previstas entre los componentes y su respectiva interfaz.
3. Prueba de aceptación. Esta es la etapa final en el proceso de pruebas, antes de que el sistema entre en funcionamiento. Se realizan pruebas con los datos suministrados por el cliente y con datos de prueba simulados. Debido a la gran diferencia que existe entre los datos reales y los de prueba, la prueba de aceptación la gran mayoría de veces puede revelar errores y omisiones en la definición de requerimientos del sistema. También puede presentar problemas en los requerimientos donde los recursos del sistema no se ajustan a las necesidades del usuario o donde el desempeño del sistema es inaceptable.

#### 5.16. Metodologías ágiles para el desarrollo de Software

Las metodologías ágiles (como por ejemplo XP, SCRUM, DSDM, Crystal, etc....), hacen parte del movimiento de desarrollo ágil y eficiente de software, que se enfocan en la

adaptabilidad ante posibles cambios como medio para incrementar las posibilidades de éxito de un proyecto.

Principales valores de una metodología ágil:

Una metodología ágil es la que tiene como principales valores (Martin, Robert C. et al., 2019).

- Personas e interacciones sobre diferentes procesos y herramientas.
- Software en funcionamiento basado en documentación extensiva.
- Colaboración con el cliente sobre la negociación contractual.
- Respuesta ante cambios para seguir un plan de acción.

Para más y mejor contextualización del tema ver anexo 3: tipos de mitologías ágiles.

#### 5.17. ISO 12207: 2017 Ingeniería de sistemas y software. Procesos del ciclo de vida del software.

- La norma ISO 12207 está encargada de todo lo relacionado con el ciclo de vida del desarrollo de software, desde la conceptualización de la idea, hasta el producto final que es la implementación del software. Consta de procesos de adquisición y de suministros del proyecto y servicios, donde se establecen pautas para su control y mantenimiento.

El principal objetivo de la norma es proporcionar una estructura común, para los proveedores como los desarrolladores y compradores, para que sea utilizado el mismo lenguaje, y también proporciona la mejora continua de los procesos.

Ver Anexo 4: ISO 12207

#### 5.18. CMMI Buenas prácticas para el desarrollo de software

Proporciona una orientación para la implementación de buenas prácticas de una organización de desarrollo, estos modelos se centran en las actividades de desarrollo de productos y servicios de calidad que logren satisfacer las necesidades de las partes interesadas.

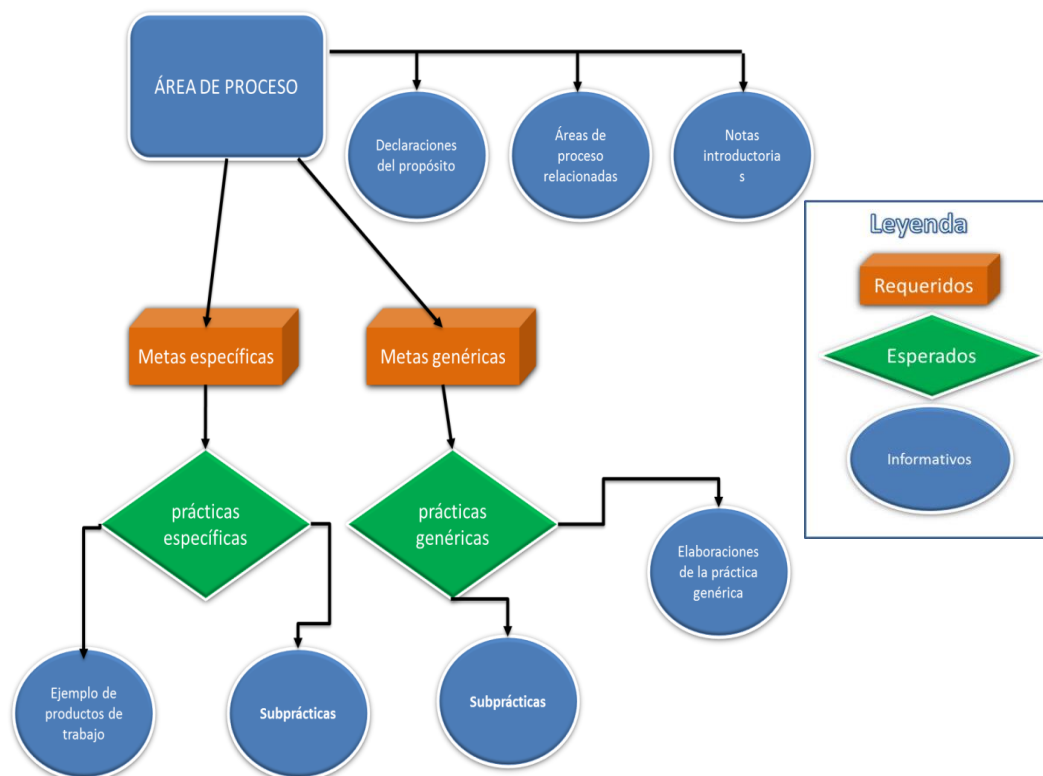


Ilustración 6. Áreas de proceso para alcanzar las buenas prácticas. Fuente: Elaboración propia.

A continuación, una explicación de cada componente de los que contiene la ilustración:

- Declaraciones del propósito: Describe la finalidad del área de proceso.
- Notas introductorias: Describe cada uno de los conceptos principales cubiertos por el área de proceso.
- Áreas de proceso relacionadas: Estas enumeran las referencias a áreas de proceso relacionadas y refleja las relaciones de alto nivel entre las áreas de proceso.
- Metas específicas: describe cada una de las características únicas que deben estar presentes para satisfacer el área de proceso. Están compuestas por prácticas y unas prácticas específicas es el desarrollo de una actividad que se piensa que es importante para llegar a la meta específica asociada. Las prácticas específicas, describen todas las actividades que se espera que produzcan un impacto positivo en el logro de las metas específicas de un área de proceso. A su



vez, las prácticas específicas tienen ejemplo de productos de trabajo, estas enumeran resultados de muestra de una práctica específica y las sub-prácticas, realizando una descripción detallada que brinda una orientación para la interpretación e implementación de una práctica específica o genérica.

➤ Metas genéricas: Describen todas las características que deben de ser tenidas en cuenta para institucionalizar los procesos que implementan un área de proceso. Compuesta de prácticas genéricas asociadas con una meta genérica, describen cada una de las actividades que se consideran importantes para lograr la meta genérica y contribuir a la institucionalización de los procesos asociados con un área de proceso, y las elaboraciones de la práctica genérica aparecen luego, para orientar en la forma en que pueden ser aplicadas de forma única las prácticas genéricas a las áreas de proceso.

Las 22 áreas de proceso mencionadas anteriormente se presentan a continuación por orden alfabético de sus respectivos acrónimos en inglés:

- Análisis Causal y Resolución (CAR).
- Gestión de Configuración (CM).
- Análisis de Decisiones y Resolución (DAR).
- Gestión Integrada del Proyecto (IPM).
- Medición y Análisis (MA).
- Definición de Procesos de la Organización (OPD).
- Enfoque en Procesos de la Organización (OPF).
- Gestión del Rendimiento de la Organización (OPM).
- Rendimiento de Procesos de la Organización (OPP).
- Formación en la Organización (OT).

- Integración del Producto (PI).
- Monitorización y Control del Proyecto (PMC).
- Planificación del Proyecto (PP).
- Aseguramiento de la Calidad del Proceso y del Producto (PPQA).
- Gestión Cuantitativa del Proyecto (QPM).
- Desarrollo de Requisitos (RD).
- Gestión de Requisitos (REQM).
- Gestión de Riesgos (RSKM).
- Gestión de Acuerdos con Proveedores (SAM).
- Solución Técnica (TS).
- Validación (VAL).

Verificación (VER).

Ver anexo 5: CMMI

A continuación, se hace las relaciones entre la norma ISO 12207 Y CMMI:

Tabla 1. Relación norma ISO 12207 VS guía CMMI.

Norma ISO 12207	CMMI 1.3. SAM
Proceso de Acuerdo	SG1 Se establecen y mantienen los acuerdos con los proveedores. SP1.1 Determinar el tipo de adquisición para cada producto o componente del producto a adquirir. SP1.2 Seleccionar los proveedores en base a una evaluación de su habilidad, para dar cumplimiento a los requerimientos especificados y criterios establecidos. SP1.3 Establecer y mantener acuerdos con proveedores.
Proceso de adquisición	SG2 Los acuerdos con los proveedores deben satisfacerse tanto por el proyecto como por el proveedor. SP2.1 Desarrollar las actividades con el proveedor como se especifican en el acuerdo con proveedor.
Proceso de suministro	SG2 Los acuerdos con los proveedores deben satisfacerse tanto por el proyecto como por el proveedor. SP2.2 Asegurar que el acuerdo con proveedor se cumple antes de

Norma ISO 12207	CMMI 1.3. SAM
	aceptar el producto adquirido. SP2.3 Asegurar la transición de productos adquiridos del proveedor.

Norma ISO 12207	CMMI OPD
Proceso de soporte	<p>SG 1 Establecer los activos de proceso de la organización.</p> <p>SP 1.1 Establecer los procesos estándar.</p> <p>SP 1.2 Establecer las descripciones de los modelos de ciclo de vida.</p> <p>SP 1.3 Establecer los criterios y las guías de adaptación.</p> <p>SP 1.4 Establecer el repositorio de mediciones de la organización.</p> <p>SP 1.5 Establecer la biblioteca de activos de proceso de la organización.</p> <p>SP 1.6 Establecer los estándares del entorno de trabajo.</p> <p>SP 1.7 Establecer las reglas y guías para los equipos</p>

Norma ISO 12207	CMMI 1.3
<p>6.4.2 Proceso de definición de necesidades y requisitos de las partes.</p> <p>6.4.3 Proceso de definición de requisitos del sistema/software</p>	<p>Desarrollo de requisitos.</p> <p>SG 1 Desarrollar los requisitos de cliente.</p> <p>SP 1.1 Educar las necesidades.</p> <p>SP 1.2 Transformar las necesidades de las partes interesadas en requisitos de cliente.</p> <p>SG 2 Desarrollar los requisitos de producto.</p> <p>SP 2.1 Establecer los requisitos de producto y de componente de producto.</p> <p>SP 2.2 Asignar los requisitos de componente de producto.</p> <p>SP 2.3 Identificar los requisitos de interfaz.</p> <p>SG 3 Analizar y validar los requisitos.</p> <p>SP 3.1 Establecer los conceptos y los escenarios de operación.</p> <p>SP 3.2 Establecer una definición de la funcionalidad y de los atributos de calidad requeridos.</p> <p>SP 3.3 Analizar los requisitos.</p> <p>SP 3.4 Analizar los requisitos para conseguir un equilibrio.</p> <p>SP 3.5 Validar los requisitos.</p>
6.4.3 proceso de definición de requisitos del sistema/software.	<p>REQM Gestión de requisitos</p> <p>SG 1 Gestionar los requisitos.</p> <p>SP 1.1 Comprender los requisitos.</p> <p>SP 1.2 Obtener el compromiso sobre los requisitos.</p>

Norma ISO 12207	CMMI 1.3
	<p>SP 1.3 Gestionar los cambios a los requisitos.</p> <p>SP 1.4 Mantener la trazabilidad bidireccional de los requisitos.</p> <p>SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos.</p>
6.4.8 Proceso de integración	<p>PI Proceso de integración</p> <p>SG 1 Prepararse para la integración del producto.</p> <p>SP 1.1 Establecer una estrategia de integración.</p> <p>SP 1.2 Establecer el entorno de integración del producto.</p> <p>SP 1.3 Establecer los procedimientos y los criterios de integración del producto.</p> <p>SG 2 Asegurar la compatibilidad de las interfaces.</p> <p>SP 2.1 Revisar la completitud de las descripciones de las interfaces.</p> <p>SP 2.2 Gestionar las interfaces.</p> <p>SG 3 Ensamblar los componentes de producto y entregar el producto.</p> <p>SP 3.1 Confirmar la disponibilidad de los componentes de producto para la integración.</p> <p>SP 3.2 Ensamblar los componentes de producto.</p> <p>SP 3.3 Evaluar los componentes de producto ensamblados.</p> <p>SP 3.4 Empaquetar y entregar el producto o componente de producto.</p>
6.4.9 Proceso de verificación	<p>VER. Verificación</p> <p>SG 1 Preparar la verificación.</p> <p>SP 1.1 Seleccionar los productos de trabajo para la verificación.</p> <p>SP 1.2 Establecer el entorno de verificación.</p> <p>SP 1.3 Establecer los procedimientos y los criterios de verificación.</p> <p>SG 2 Realizar las revisiones entre pares.</p> <p>SP 2.1 Preparar las revisiones entre pares.</p> <p>SP 2.2 Realizar las revisiones entre pares.</p> <p>SP 2.3 Analizar los datos de las revisiones entre pares.</p> <p>SG 3 Verificar los productos de trabajo seleccionados.</p> <p>SP 3.1 Realizar la verificación.</p> <p>SP 3.2 Analizar los resultados de la verificación.</p>
6.4.11 Proceso de validación	<p>SG1 Preparar la validación.</p> <p>SP 1.1 Seleccionar los productos para la validación.</p> <p>SP 1.2 Establecer el entorno de validación.</p> <p>SP 1.3 Establecer los procedimientos y los criterios de validación.</p>

Norma ISO 12207	CMMI 1.3
	SG 2 Validar el producto o los componentes de producto. SP 2.1 Realizar la validación. SP 2.2 Analizar los resultados de la validación.

Norma ISO 12207	CMMI
6.3.1 Proceso de planificación del proceso	<p>IPM Gestión integrada del proyecto</p> <p>SG 1 Utilizar el proceso definido del proyecto.</p> <p>SP 1.1 Establecer el proceso definido del proyecto.</p> <p>SP 1.2 Utilizar los activos de proceso de la organización para planificar las actividades del proyecto.</p> <p>SP 1.3 Establecer el entorno de trabajo del proyecto.</p> <p>SP 1.4 Integrar los planes.</p> <p>SP 1.5 Gestionar el proyecto utilizando planes integrados. SP 1.6 Establecer los equipos.</p> <p>SP 1.7 Contribuir a los activos de proceso de la organización.</p> <p>SG 2 Coordinar y colaborar con las partes interesadas relevantes.</p> <p>SP 2.1 Gestionar la involucración de las partes interesadas. SP 2.2 Gestionar las dependencias.</p> <p>SP 2.3 Resolver las cuestiones de coordinación.</p>
6.3.2 Proceso de evaluación y control del proyecto	<p>PMC. Monitorización y control del proyecto.</p> <p>SG 1 Monitorizar el proyecto frente al plan.</p> <p>SP 1.1 Monitorizar los parámetros de planificación del proyecto.</p> <p>SP 1.2 Monitorizar los compromisos.</p> <p>SP 1.3 Monitorizar los riesgos del proyecto.</p> <p>SP 1.4 Monitorizar la gestión de los datos.</p> <p>SP 1.5 Monitorizar la involucración de las partes interesadas.</p> <p>SP 1.6 Llevar a cabo las revisiones del progreso.</p> <p>SP 1.7 Llevar a cabo las revisiones de hitos.</p> <p>SG 2 Gestionar las acciones correctivas hasta su cierre.</p> <p>SP 2.1 Analizar las cuestiones.</p> <p>SP 2.2 Llevar a cabo las acciones correctivas.</p> <p>SP 2.3 Gestionar las acciones correctivas.</p>
6.3.3 Proceso de gestión de decisiones	<p>DAR. Análisis de decisiones y resolución.</p> <p>SG 1 Evaluar las alternativas.</p> <p>SP 1.1 Establecer las guías para el análisis de decisiones.</p> <p>SP 1.2 Establecer los criterios de evaluación. SP 1.3 Identificar las soluciones alternativas.</p> <p>SP 1.4 Seleccionar los métodos de evaluación.</p> <p>SP 1.5 Evaluar las soluciones alternativas.</p> <p>SP 1.6 Seleccionar las soluciones.</p>
	<p>SG 1. Preparar la gestión de riesgos.</p> <p>SP 1.1 Determinar las fuentes y las categorías de riesgos.</p>

Norma ISO 12207	CMMI
6.3.4 Proceso de gestión de riesgos	SP 1.2 Definir los parámetros de riesgos. SP 1.3 Establecer una estrategia de gestión de riesgos. SG 2 Identificar y analizar los riesgos. SP 2.1 Identificar los riesgos. SP 2.2 Evaluar, clasificar y priorizar los riesgos. SG 3 Mitigar los riesgos. SP 3.1 Desarrollar los planes de mitigación de riesgos. SP 3.2 Implementar los planes de mitigación de riesgos.
6.3.5 Proceso de gestión de la configuración	CM. Gestión de configuración. SG 1 Establecer las líneas base. SP 1.1 Identificar los elementos de configuración. SP 1.2 Establecer un sistema de gestión de configuración. SP 1.3 Crear o liberar las líneas base. SG 2 Seguir y controlar los cambios. SP 2.1 Seguir las peticiones de cambio. SP 2.2 Controlar los elementos de configuración. SG 3 Establecer la integridad. SP 3.1 Establecer los registros de gestión de configuración. SP 3.2 Realizar auditorías de configuración.

Fuente: Elaboración propia

## 6. METODOLOGIA

### 6.1. Alcance

El modelo de gestión de procesos bajo el enfoque de aseguramiento de la calidad, basado en el ciclo PHVA para el laboratorio de la industria académica en desarrollo de software para la Universidad Católica de Oriente el cual se formula mediante un desarrollo presentado en este informe busca aportar ciertos lineamientos o ideas al proyecto matriz para interrelacionar de manera secuencial cada uno de los componentes que hacen parte del proceso de desarrollo de software, esto teniendo en cuenta la recolección y análisis de información de índole cuantitativo y cualitativo como por ejemplo tendencias en desarrollo de Software en la actualidad, lo cual es clave para realizar la estructuración de un laboratorio de desarrollo de software desde los frentes productivos y administrativos.

### 6.2. Descripción detallada de la metodología

Buscando llegar al cumplimiento de los objetivos establecidos, se plantea llevar a cabo una serie de actividades que comprenden desde la investigación hasta el llegar a formular el modelo de gestión, tales actividades son generales y deben ser abordadas de manera secuencial y estructurada de la siguiente manera. (Ver tabla)

Tabla 2. Descripción actividades metodológicas.

Actividad N°	Descripción de la actividad.
1	Investigación acerca del alcance y lineamientos para aplicación y sostenibilidad del ciclo PHVA dentro de procesos, además es importante obtener un conocimiento básico sobre diferentes terminologías empleadas en desarrollo de software y en la estructuración de modelo de gestión tomando como eje fundamental los procesos.

2	Conocimiento básico del proceso de desarrollo de software e investigación acerca de modelos de gestión empleados en los últimos años en diferentes instituciones enfocadas a desarrollar productos de software
3	Obtención de información específica del paso a paso que tiene como resultado final un software y diferentes indicadores para medir los procesos con la finalidad de tener idea del proceso para así realizar un análisis y procesamiento de la información que permita poder formular del modelo de gestión de procesos.
4	Dar dimensión al modelo formulado con base en la información recolectada durante el proceso teniendo en cuenta diferentes factores como necesidades, metas y estrategias.

Fuente: Elaboración propia.

De la información presentada en la Tabla anterior se tiene un amplio enfoque en el aseguramiento de la calidad puesto que con la misma y el ciclo PHVA lo que se hace es mejorar constantemente la eficacia y eficiencia de los procesos mediante el modelo formulado, lo anterior es importante puesto que la calidad que percibe el cliente puede ser tan positiva o negativa por el método con que la organización lleva a cabo cada una de las actividades que se ven reflejadas en un producto.



## 7. RESULTADOS

### 7.1. Objetivo 1

Investigar sobre modelos de desarrollo de software, modelos de calidad, y herramientas de uso gratuito, que sean aplicables a laboratorios para la industria (o fábrica) de desarrollo de software en entornos de educación terciaria.

Para el alcance de este objetivo inicialmente se realiza una relación entre las normas ISO/IEC 12207 y la guía CMMI.

Relación de la norma ISO/IEC 12207 y CMMI:

En esta sesión se presenta de manera clara y precisa el cómo se relación la norma ISO/IEC 12207 y la guía CMMI en materia de desarrollo de Software, y cómo los procesos de cada una de estas interactúan y se muestra interrelacionadas buscando un mismo fin orientado desde diferentes enfoques. Por tanto, a continuación, se muestran los principales procesos de la norma ya mencionada y posteriormente se hará un desglose en el cual se presentan los procesos de esta y la relación de esta con algunos procesos de CMMI.

La norma ISO/ IEC 12207 está compuesta de los siguientes procesos:

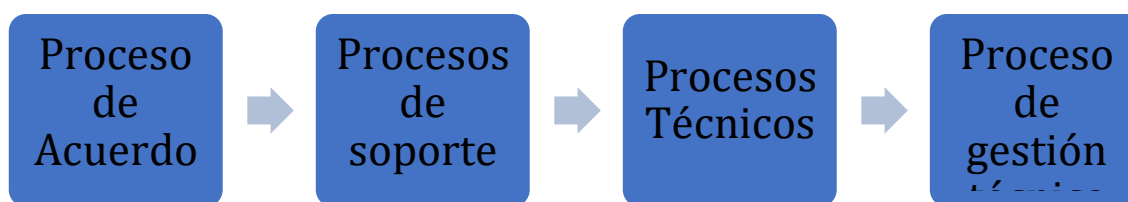


Ilustración 7. Procesos de la norma ISO/ IEC 12207. Fuente: Elaboración propia.

Cada uno de los procesos contenidos en la ilustración 7 se define, se representan y se relacionan de la siguiente manera:

- **Procesos de acuerdo:** Tienen como objetivo toda la relación con compra y con subcontratación de los insumos necesarios para el desarrollo.

El proceso de acuerdo está dividido en tres procesos adicionales, en la siguiente ilustración se muestra cada subproceso con su objetivo:

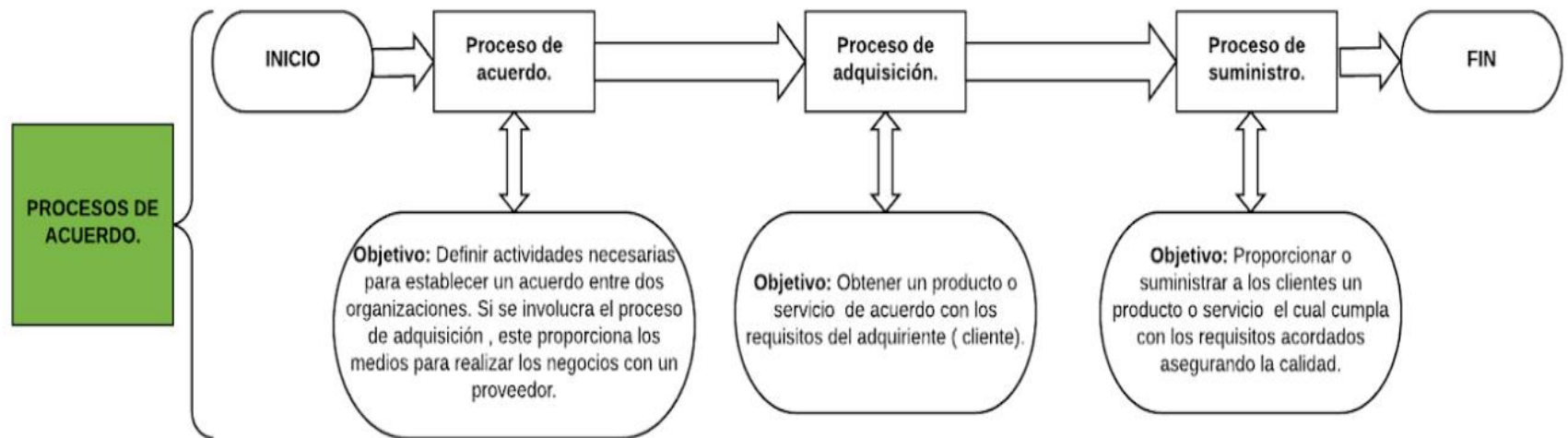


Ilustración 8. Procesos de acuerdo (ISO/IEC 12207). Fuente: Elaboración propia.

En la guía CMMI Versión 1.3, tiene el área de proceso Gestión de acuerdo con los proveedores (SAM - Supplier Agreement Management), la cual tiene como objetivo gestionar lo relacionado con la adquisición de productos y servicios del proveedor. El área de proceso se describe a continuación:

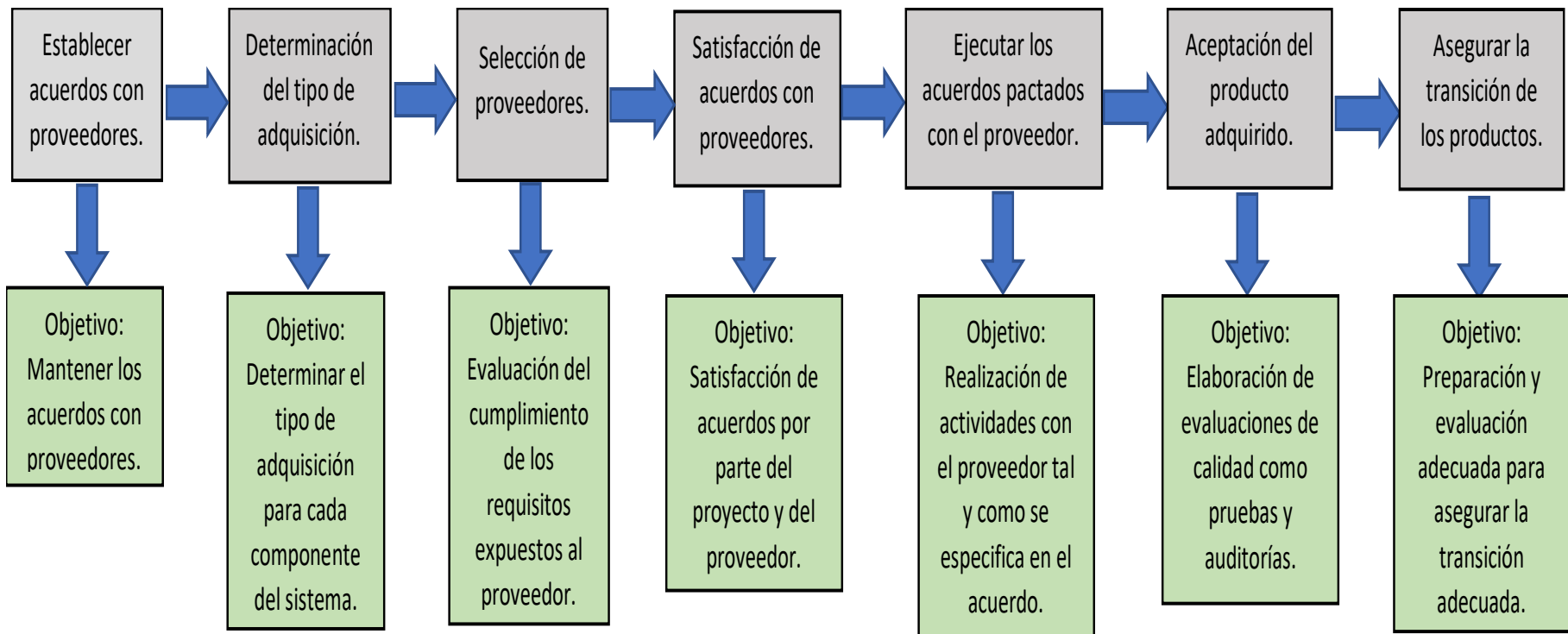


Ilustración 9. Proceso de acuerdo (CMMI). Fuente: Elaboración propia.

- Proceso de soporte: A continuación, se presenta el diagrama de proceso correspondiente a los procesos de soporte el cual está basado en la norma ISO 12207, además, se demuestran los subprocesos y su respectivo objetivo en lo que se refiere a desarrollo de Software.

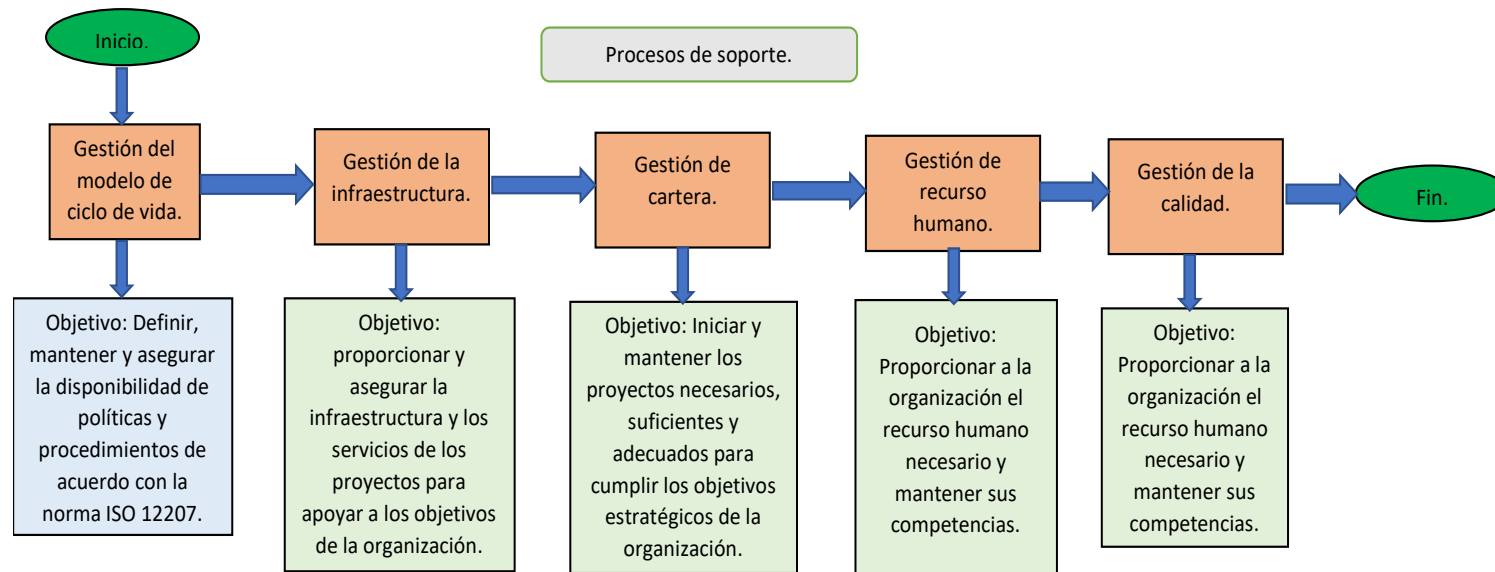


Ilustración 10. Procesos de soporte (ISO/IEC 12207). Fuente: Elaboración propia.

Algunos de los procesos plasmados en la **¡Error! No se encuentra el origen de la referencia.**, se relacionan o tiene un común denominador con ciertos procesos de la guía CMMI, a continuación, se plasma el desarrollo en el cual se evidencia la relación de estas dos normas:

Para el caso del proceso de gestión de la infraestructura de la norma ISO 12207 se tiene una relación con los procesos de soporte (OPD) de la guía CMMI el cual aporta o brinda como idea ciertas metas y prácticas específicas apuntando a tener un nivel de madurez 3 en cuanto al desarrollo de Software.

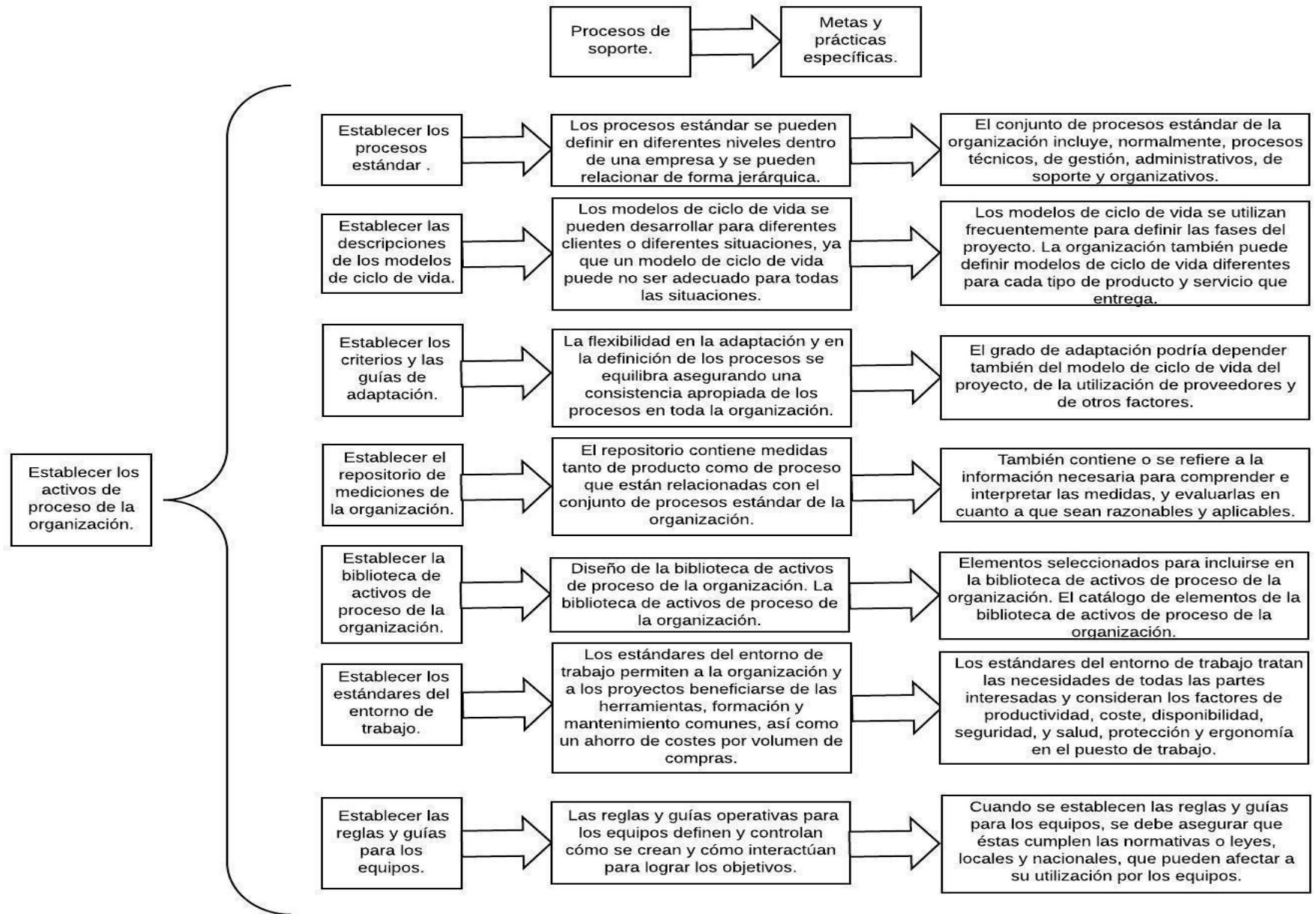


Ilustración 11. Procesos de soporte (CMMI). Fuente: Elaboración propia.

Al analizar el proceso de gestión de infraestructura y al relacionarlo representado en la ilustración 11 con el proceso de soporte que presenta CMMI mostrado en la ilustración 11 se puede afirmar con certeza de que ambos están ligados en cuanto al aseguramiento de una infraestructura adecuada de acuerdo a las necesidades que se presenten, además las dos normas en este sentido van muy encaminadas al desarrollo y ciclo de vida de lo que se va a desarrollar y a sus procesos principales de manera que se contribuya al buen desarrollo de las actividades. Dentro de las actividades que se encuentra en la relación de estos dos procesos se tiene que ambos apuntan a lograr objetivos que apoyan a otros procesos y además llevan a cabo una función especializada.

Para los procesos de gestión de conocimiento de la norma ISO 12207 se observa que al igual que los procesos de gestión de infraestructura están relacionados con los procesos de soporte, pero para este caso tiene un enfoque no tanto a lo material sino más ligado a talento humano.

La relación de estos dos procesos es la encargada del estudio de cómo se aprovechan y se obtienen más oportunidades en el medio con el conocimiento que se tiene dentro del sistema, además establecer todo lo relacionado a reglas y guías para los equipos de trabajo, también van encaminadas a todo lo referente a las mediciones de diferentes indicadores que se presentan dentro de los procesos desarrollados.

- Procesos técnicos: Continuando con lo anterior se presenta el diagrama de Procesos Técnicos de la norma ISO 12207, donde se busca integral con algunos subprocesos de la guía CMMI, para el desarrollo de software.

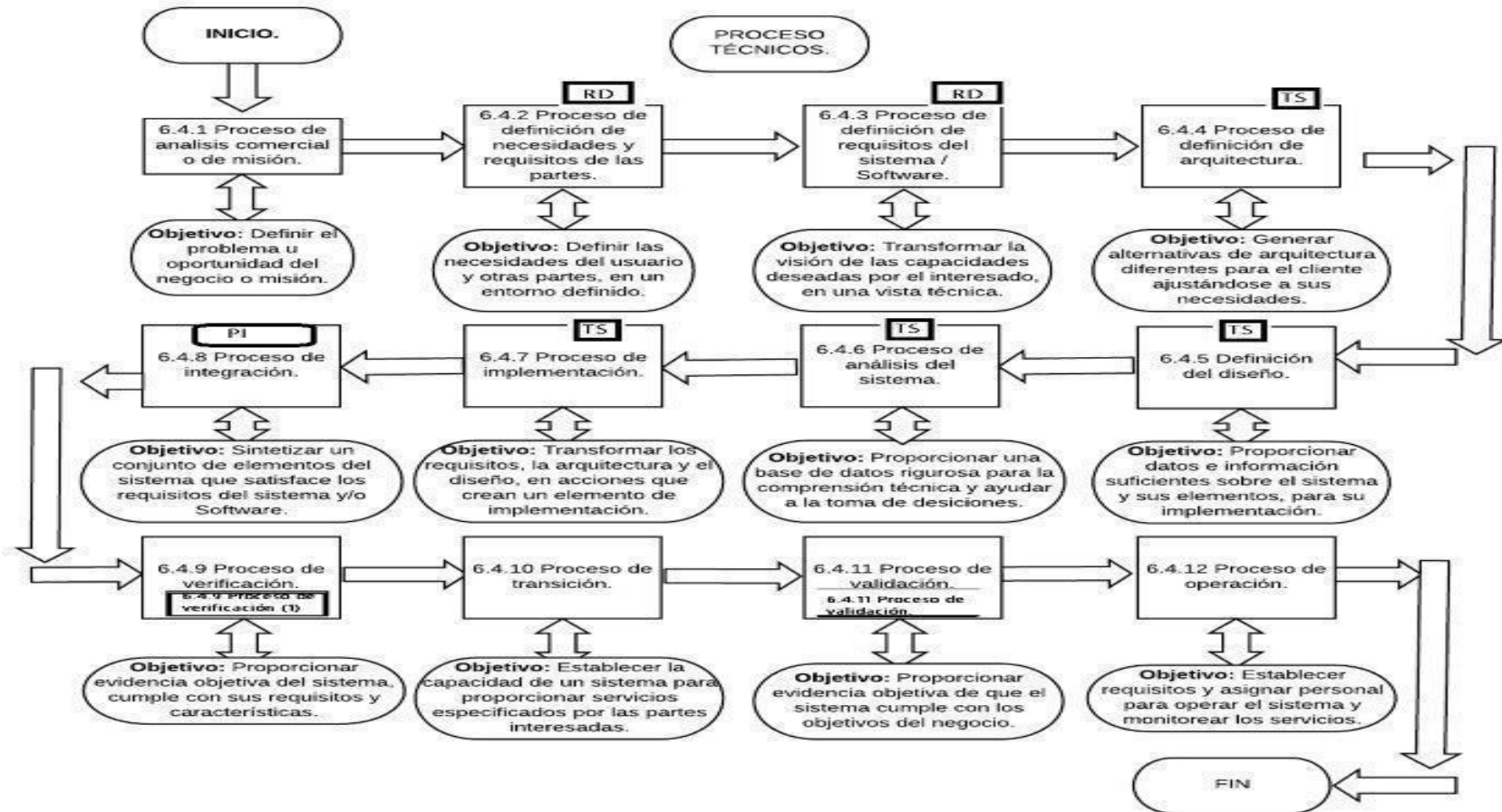


Ilustración 12. Procesos técnicos (ISO/IEC12207). Fuente: Elaboración propia.



Basados en algunos procesos que están relacionados con la norma de CMMI, se diseña el siguiente diagrama:

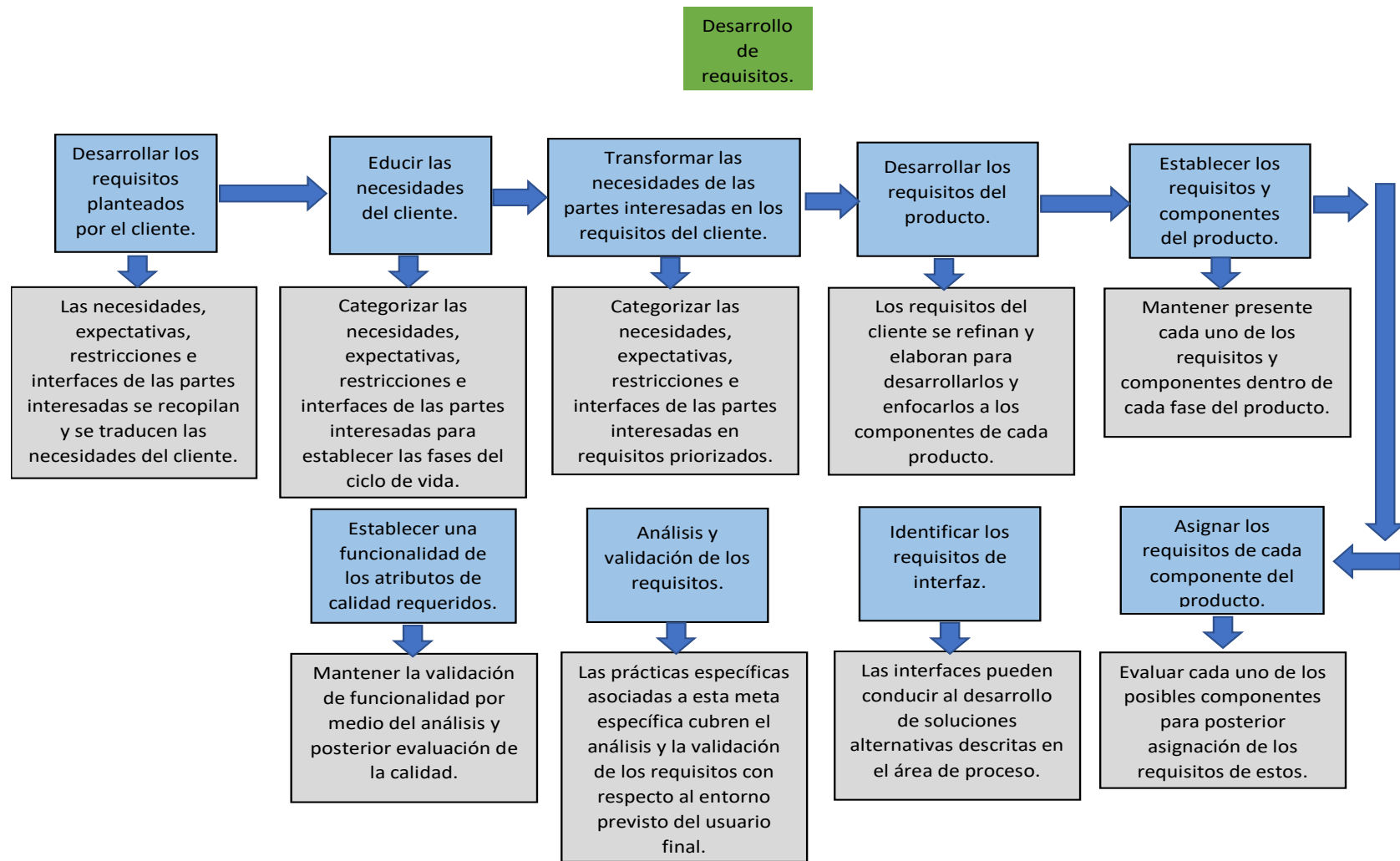


Ilustración 13. Procesos técnicos (CMMI). Fuente: Elaboración propia.

El proceso de definición de necesidades y requisitos de las partes y el proceso de definición de requisitos del sistema /software de la norma ISO/ IEC 12207, está relacionado con dos procesos de la guía CMMI el desarrollo de requisitos, el propósito de este proceso es educir, analizar y establecer los requisitos de cliente, de producto y de componente de producto. Las fases para lograr el objetivo de este proceso, es primero desarrollar los requisitos del cliente donde se especifican las necesidades y expectativas de las partes interesadas, luego se desarrolla los requisitos del producto donde identifica los componentes , incluyendo objetos, personas y proceso, por último se debe analizar y validar los requisitos para determinar qué impacto tendrá el entorno de operación previsto sobre la capacidad para satisfacer las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas.

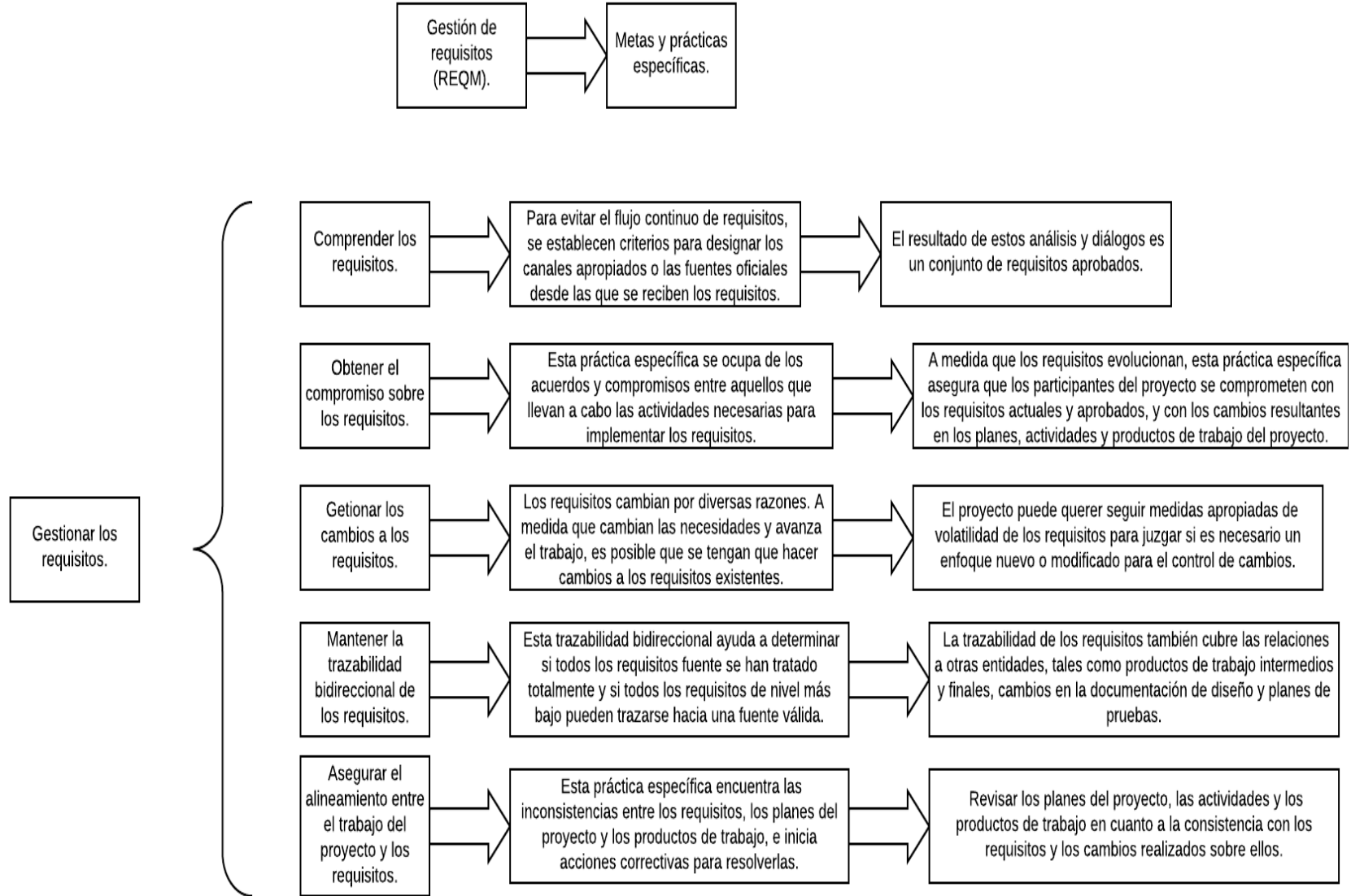


Ilustración 14. Gestión de requisitos. Fuente: Elaboración propia.

Para el proceso de gestionar los requisitos, el propósito para lograr cumplir estos en los productos y los componentes del proyecto, se debe asegurar la alineación entre esos requisitos, los planes y los productos de trabajo del proyecto. Para ello primero se comprende las restricciones por medio de los canales apropiados o las fuentes oficiales para ser aprobados, luego se obtiene el compromiso sobre las exigencias. Por otro lado, se ocupa de los acuerdos y compromisos para implementar los requisitos, y asegurar la responsabilidad de los participantes del proyecto para ir alineados a los requisitos aprobados, también se gestiona los cambios a las necesidades que se da a medida que cambian las mismas y avanza el trabajo, esto se da si es necesario el cambio a los requisitos existentes. Se mantiene la trazabilidad bidireccional de los requisitos y se asegura los alineamientos entre el trabajo del proyecto.

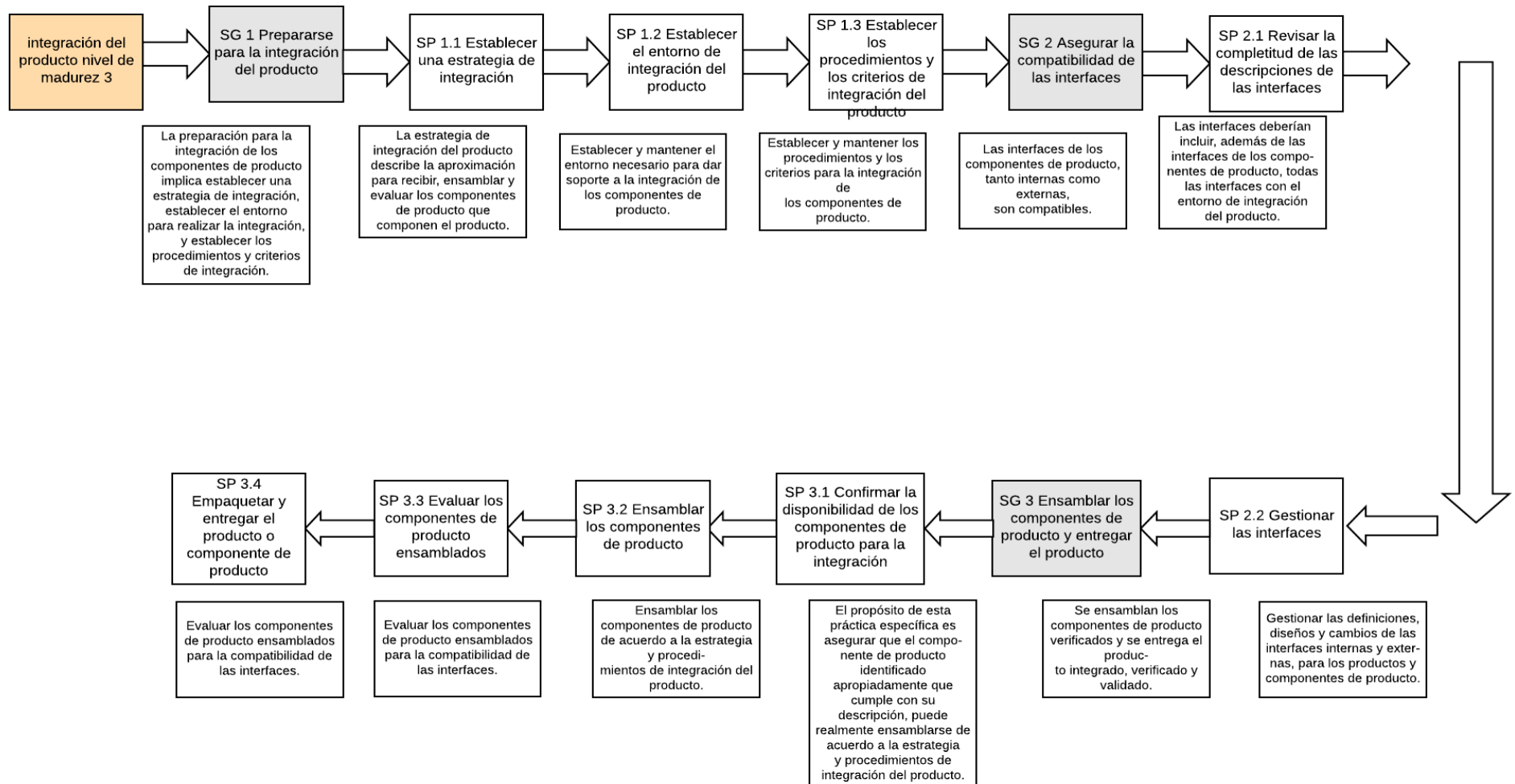


Ilustración 15. Integración del producto. Fuente: Elaboración propia.

Para el subproceso de integración, se encuentra relacionada con la integración del producto donde se halla tres metas y prácticas específicas, una vez que se identifican las necesidades, se desarrolla un programa de formación que las trate. Primero se establece una estrategia de integración, el entorno del producto, procedimientos y los criterios de la unificación. Luego se debe asegurar la compatibilidad de las interfaces externas e internas, y también se debe gestionar las definiciones, diseños y los cambios de las interfaces tanto para el producto y sus componentes. Y por último se ensamblan los componentes de producto verificados y se entrega el producto integrado, validado e integrado.

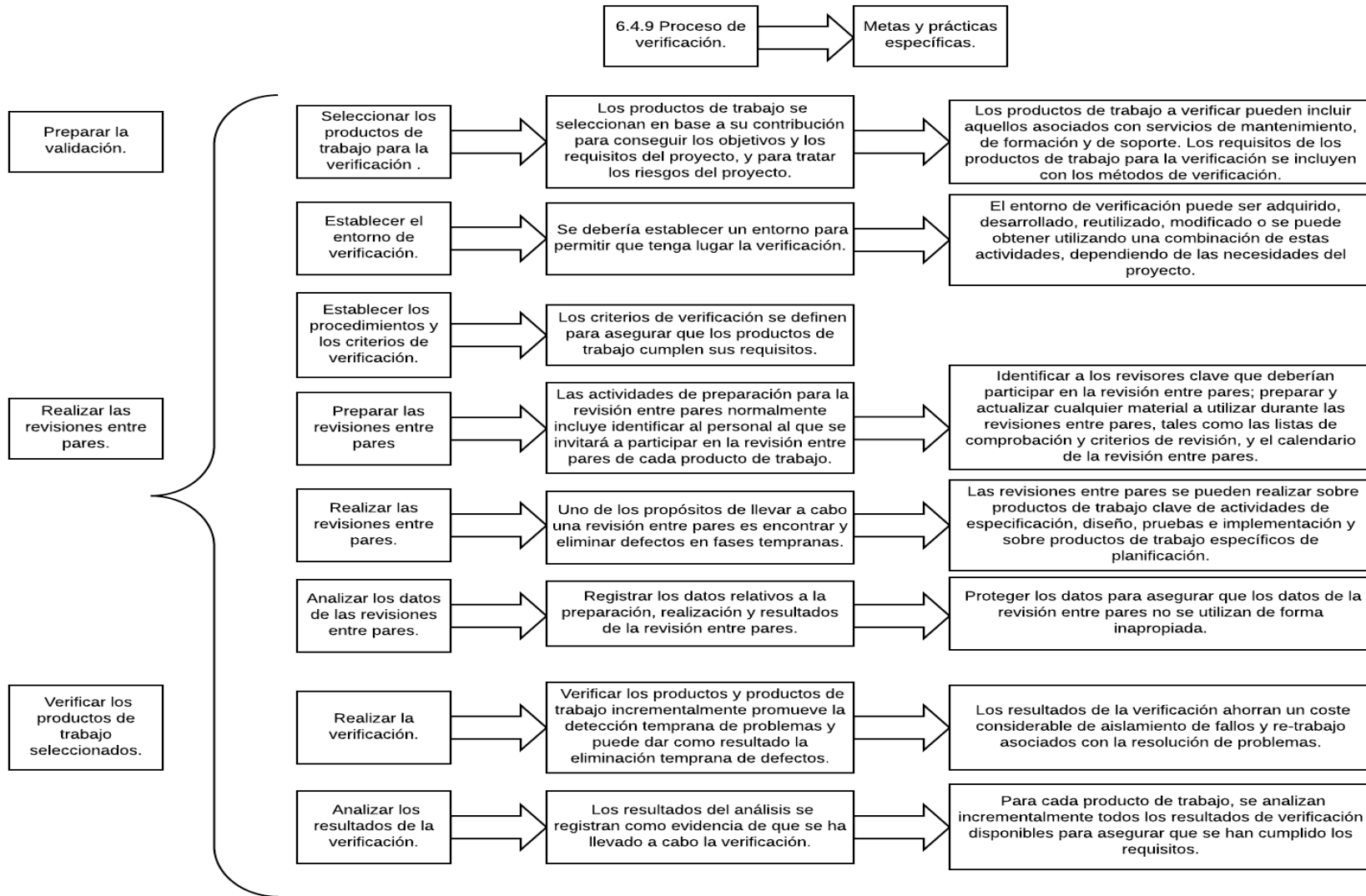


Ilustración 16. Proceso de verificación. Fuente: Elaboración propia.

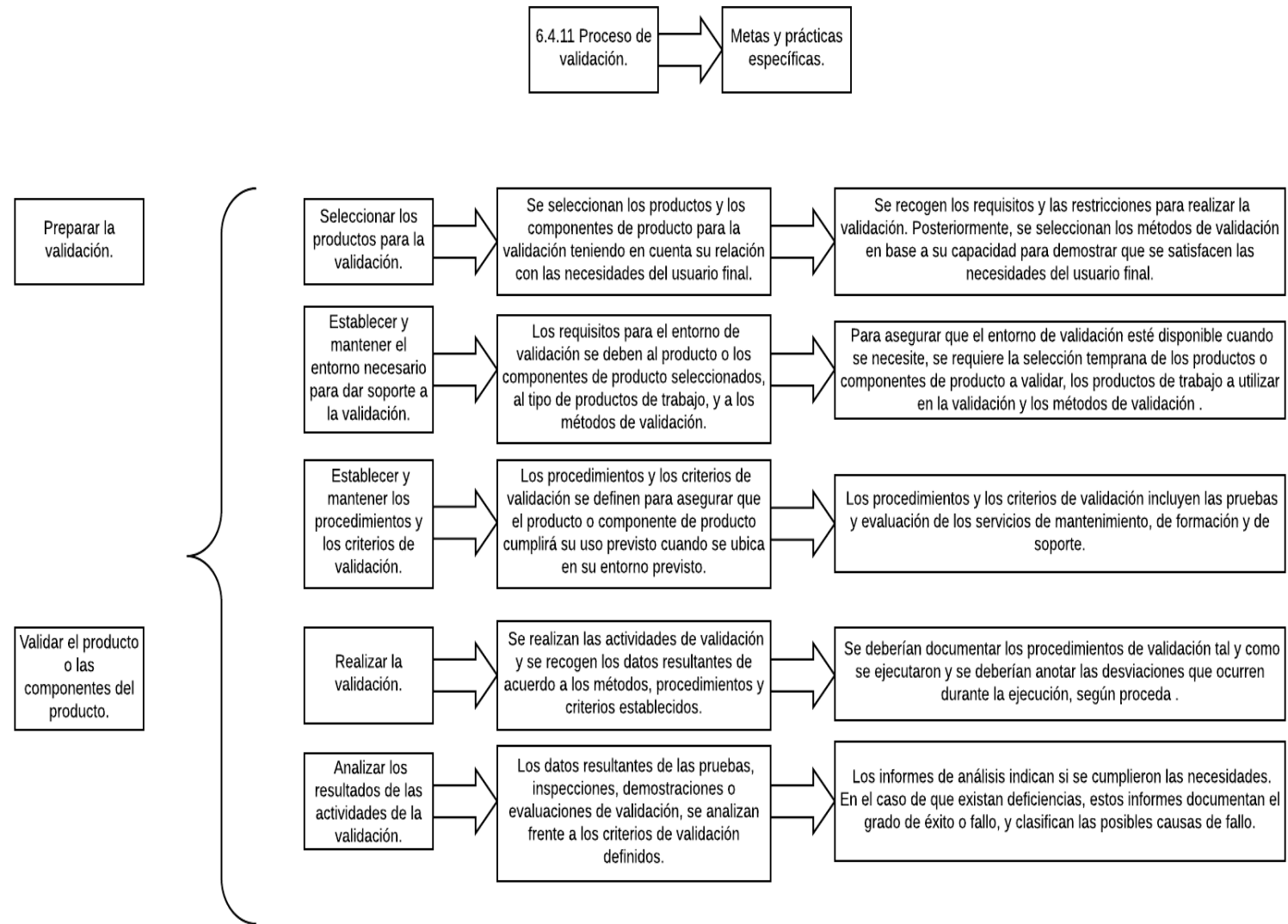


Ilustración 17. Proceso de validación. Fuente: Elaboración propia.



En cuanto al proceso de verificación de la ISO 12207 donde se proporciona la evidencia objetiva del sistema, el cumplimiento de sus requisitos y características. Simultáneamente para la norma CMMI el proceso de verificación está dada por tres importantes pasos, el primero es preparar la validación donde se debe seleccionar los productos de trabajo para su verificación, se establece el entorno y procedimientos y los criterios de verificación. Después se realiza las revisiones entre pares en el cual primero se prepara, luego se realiza y por último se analiza los datos de la revisión. Y finalmente la verificación de los productos de trabajo seleccionados, en el cual se analiza los datos de las revisiones entre pares, se hace la verificación y se analiza los resultados del control.

Los procesos de validación de la ISO 12207 el cual proporciona evidencia objetiva de que el sistema cumple con los objetivos del negocio, en comparación con el proceso de validación de la norma CMMI se encuentra que la norma CMMI cuenta con dos pasos, el primero el preparar la validación por medio de la selección de los productos para la validación, donde se debe establecer y mantener el entorno necesario para dar soporte a la admisión y el último paso es validar el producto o las componentes del producto, donde se analiza los resultados de las actividades.

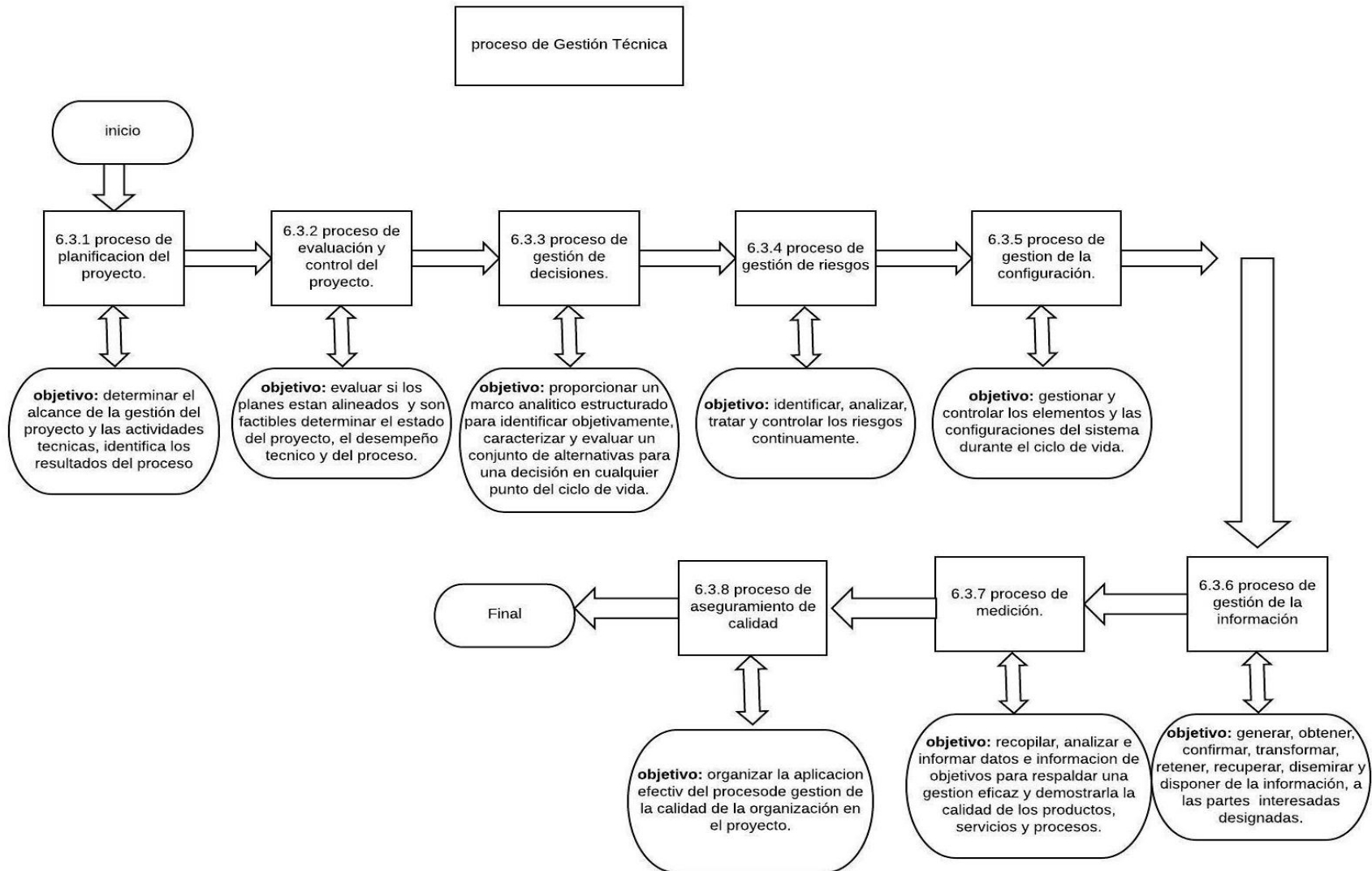


Ilustración 18. Proceso de gestión técnica. Fuente: Elaboración propia.

Algunos de los procesos plasmados en el diagrama anterior se relacionan o tienen un común denominador con ciertos procesos de la norma CMMI, a continuación, se realizará el desarrollo en el cual se presenta la relación de las dos normas:

Para el caso del proceso de planificación del proyecto de la norma ISO 12207 se tiene que este tiene gran relación con el SG de CMMI (utilizar el proceso definido del proyecto) el cual define unas metas y prácticas específicas en cierto nivel de madurez, para este caso.

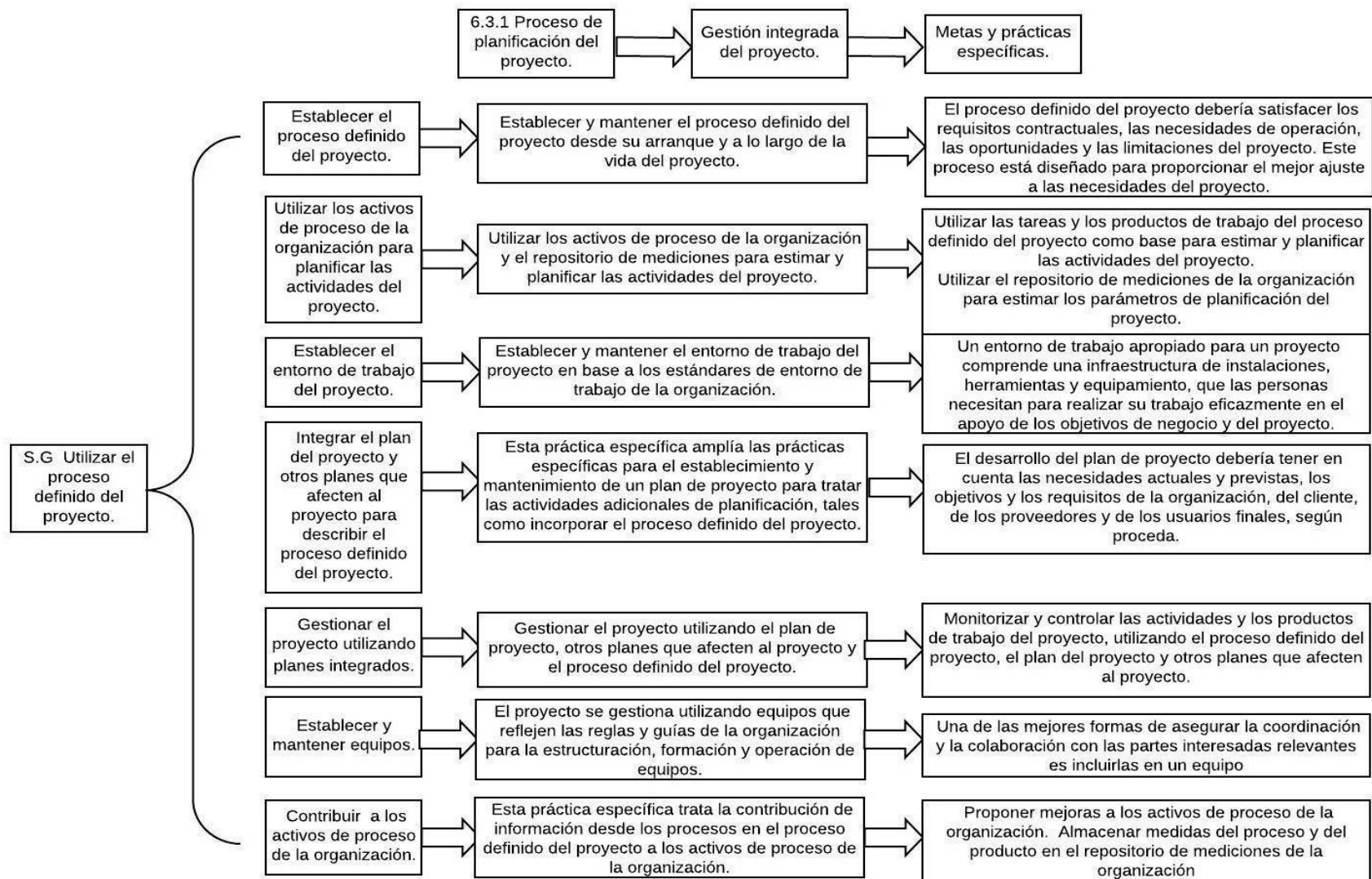


Ilustración 19. Proceso de planificación del proyecto. Fuente: Elaboración propia.

Si se analiza este proceso en los diagramas de las dos normas, se concluye que para este tipo de proceso ambas tienen un mismo enfoque, el cual es la definición y planificación del proyecto apoyados en la identificación de un ciclo de vida del producto (PHVA) de tal forma que cuando se dé la ejecución del proyecto, se pueda cumplir con el alcance en el tiempo establecido y con los costos presupuestados.

Además de una u otra forma ambas normas se enfocan en la iniciación de proyecto puesto que después de la identificación de es este lo que se hace es una completa definición de los objetivos, el alcance, la calidad, se estima como se llevara a cabo y se hace una evaluación de los riesgos, además se crean estimaciones de tiempos, costes teniendo en cuenta los recursos humanos materiales y financieros disponibles.

Por otro lado, para el proceso de evaluación y control del proyecto de la norma ISO 12207 se tiene que esta se relaciona con el monitoreo del proyecto frente al plan y a la gestión de acciones correctivas hasta el cierre del proyecto, al igual que el proceso definido anteriormente.

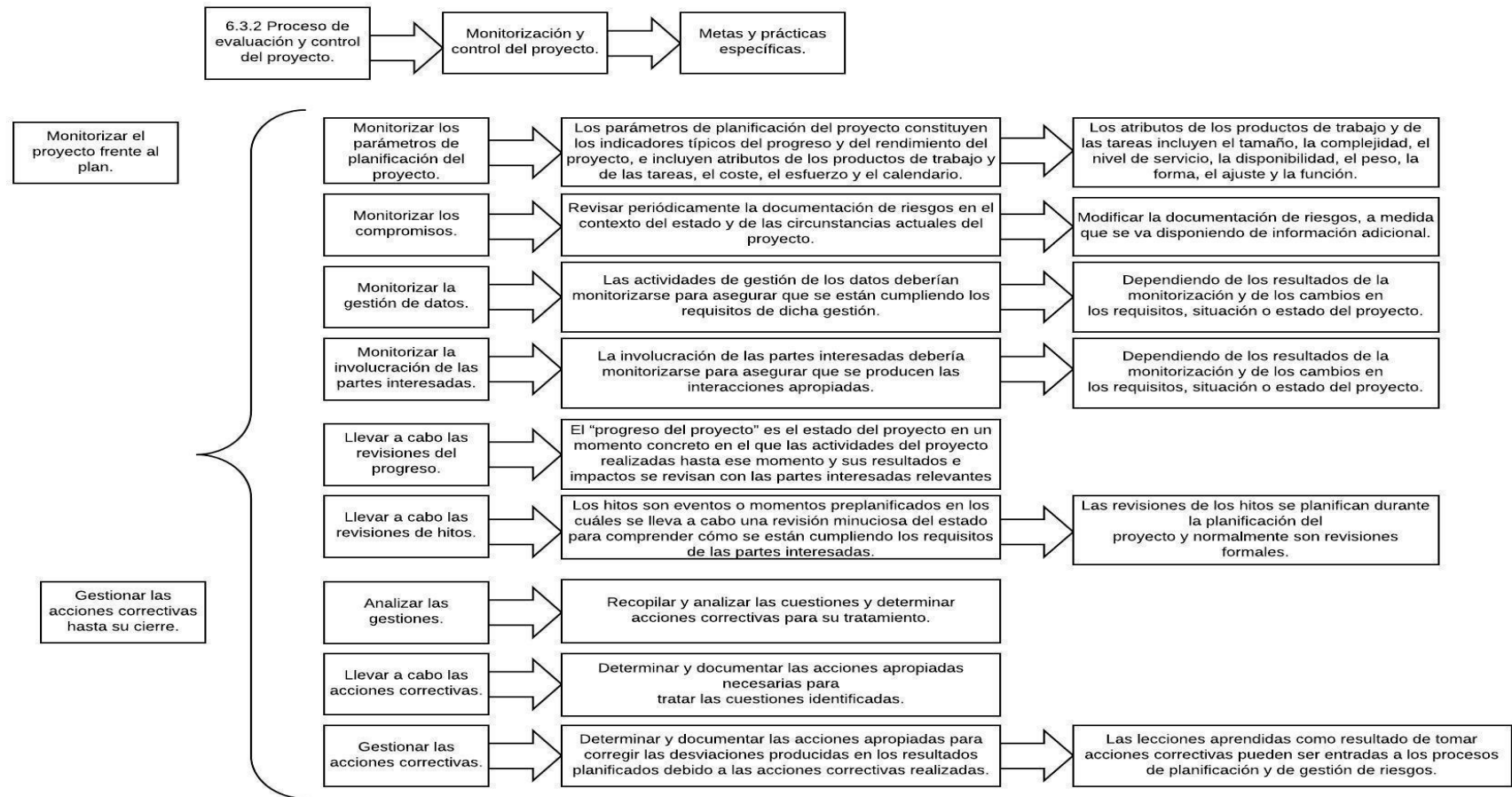


Ilustración 20. Proceso de evaluación y control del proyecto. Fuente: Elaboración propia.

Al analizar el proceso en los dos diagramas se evidencia que tienen como objetivo principal todo lo relacionado al monitoreo y gestión de lo planificado, todo esto con el fin de administrar el equipo del proyecto, es decir discutir regularmente el progreso de cada fase del proyecto y determinar las prioridades siguientes. Se realiza constante monitoreo sobre el avance real del proyecto para que este se pueda adaptar el a los cambios contextuales.

Para el Proceso de gestión de decisiones de la norma ISO 12207 se tiene una relación con el análisis de decisiones y resolución de la norma CMMI, teniendo como objetivo común una pre evaluación de diferentes factores determinantes al momento de la toma de decisiones dentro de los procesos, esto teniendo en cuenta la importancia de una buena decisión desde que se tienen las primeras conversaciones de ejecución entre las partes interesadas, hasta que se tiene un proyecto estructurado y que cumple con los requisitos y expectativas de un cliente como consumidor final.

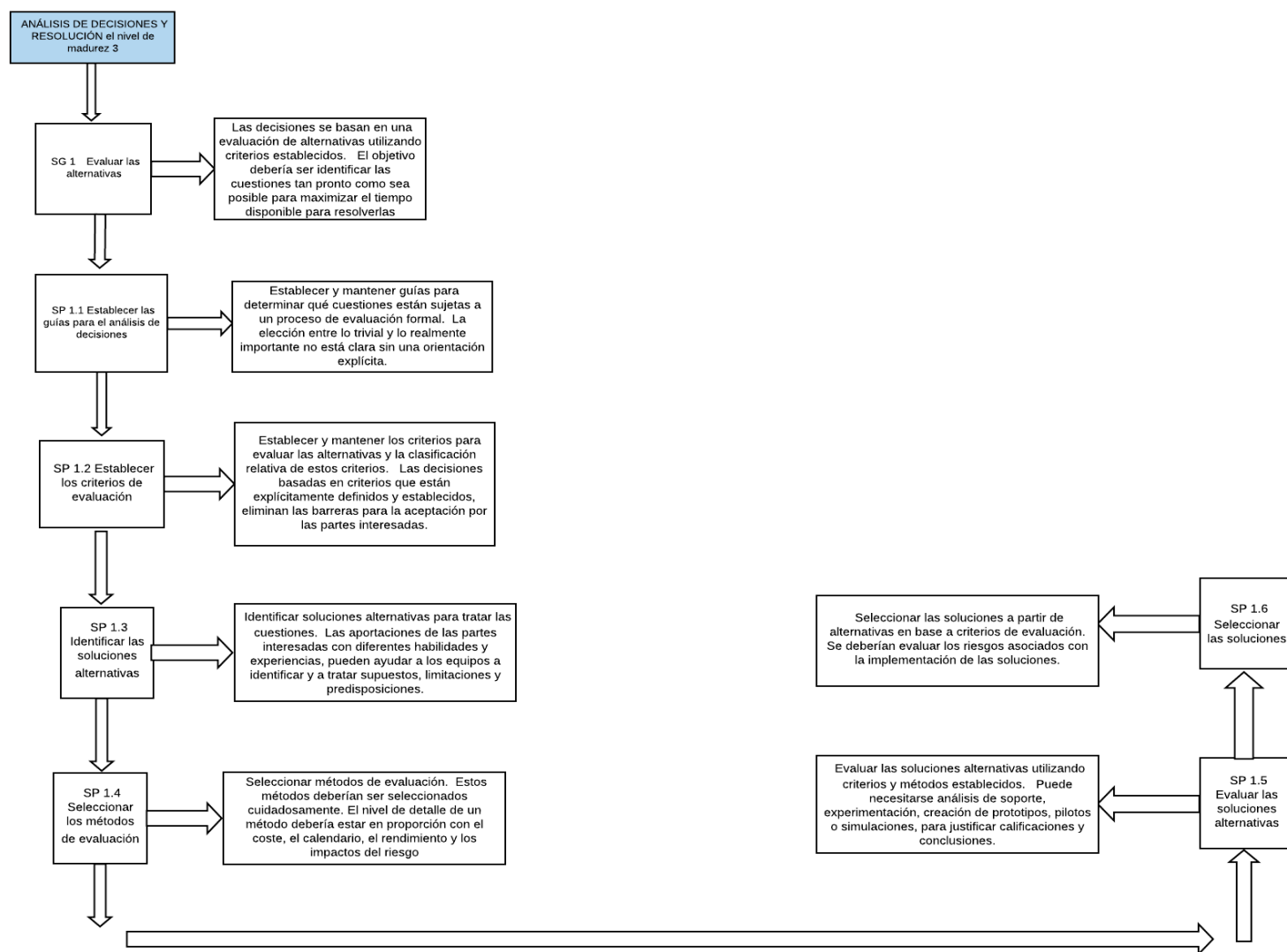


Ilustración 21. Análisis de decisiones y resolución. Fuente: Elaboración propia.

Para este proceso, ambas normas lo que buscan es llevar a cabo las valoraciones oportunas sobre lo planeado y lo ejecutado, esto para posteriormente realizar una retroalimentación y evaluar puntos débiles para aplicar las medidas correctivas bajo un ciclo PHVA a fin de constantemente estar evaluando el cumplimiento de metas y objetivos establecidos dentro del proyecto.



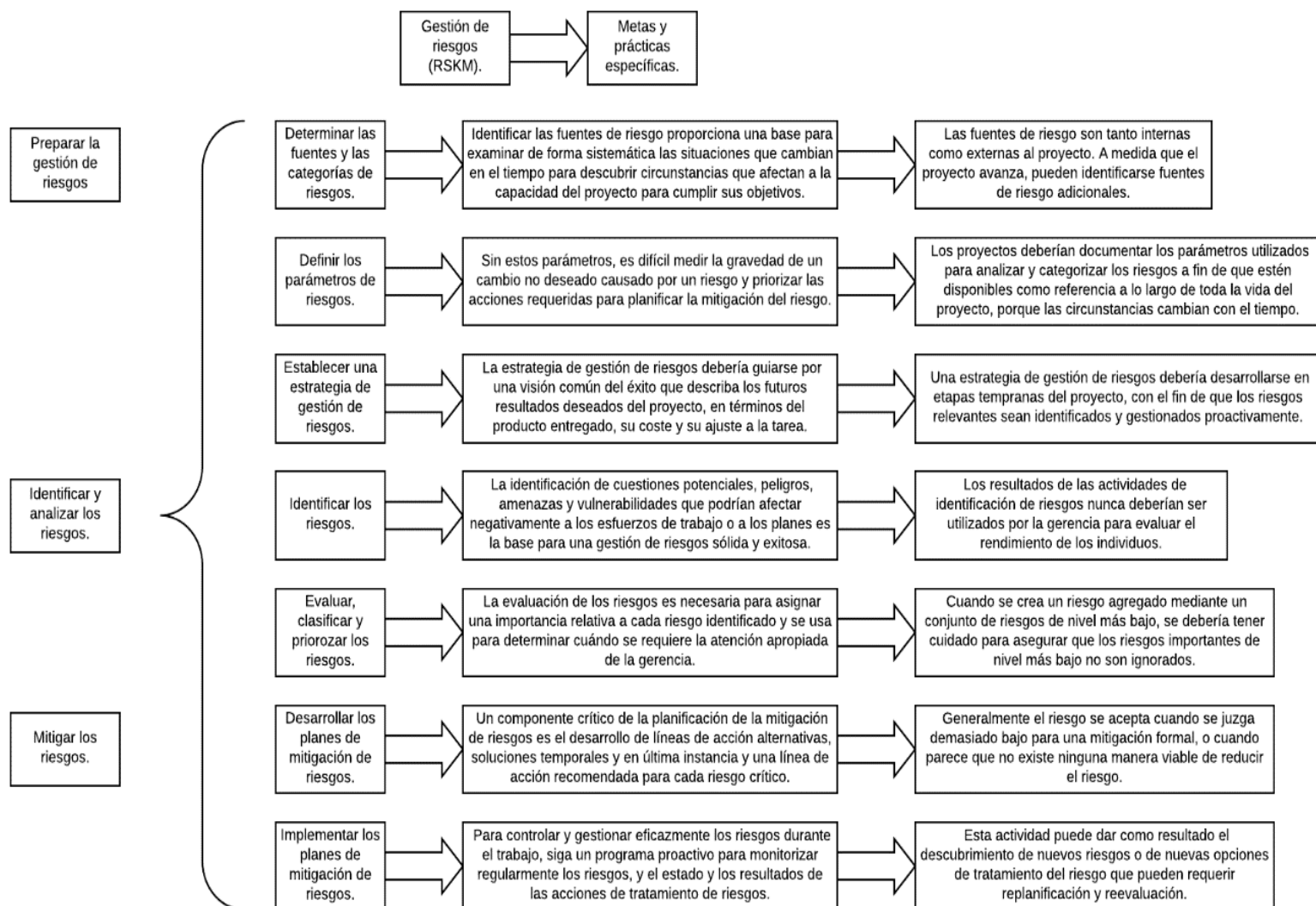


Ilustración 22. Gestión de riesgos. Fuente: Elaboración propia.

Para el Proceso de gestión de riesgos de la norma ISO 12207, se tiene una relación con la gestión de riesgos definida por CMMI como RSKM. El común denominador de ambas normas es el identificar, analizar y tratar todo lo relacionado a riesgos desde la planificación del proyecto hasta el cierre de este.

Como se mencionó anteriormente el común de ambas normas para este proceso es el identificar, analizar y tratar todo lo relacionado a riesgos intentando el incremento de la probabilidad y el impacto de eventos positivos, y disminuir la probabilidad y el impacto de eventos negativos para el proyecto. Según ambas normas, se dice que una vez se determinan los riesgos que pueden afectar el proyecto estos se documentan con sus respectivas características y posibles efectos dentro del desarrollo.

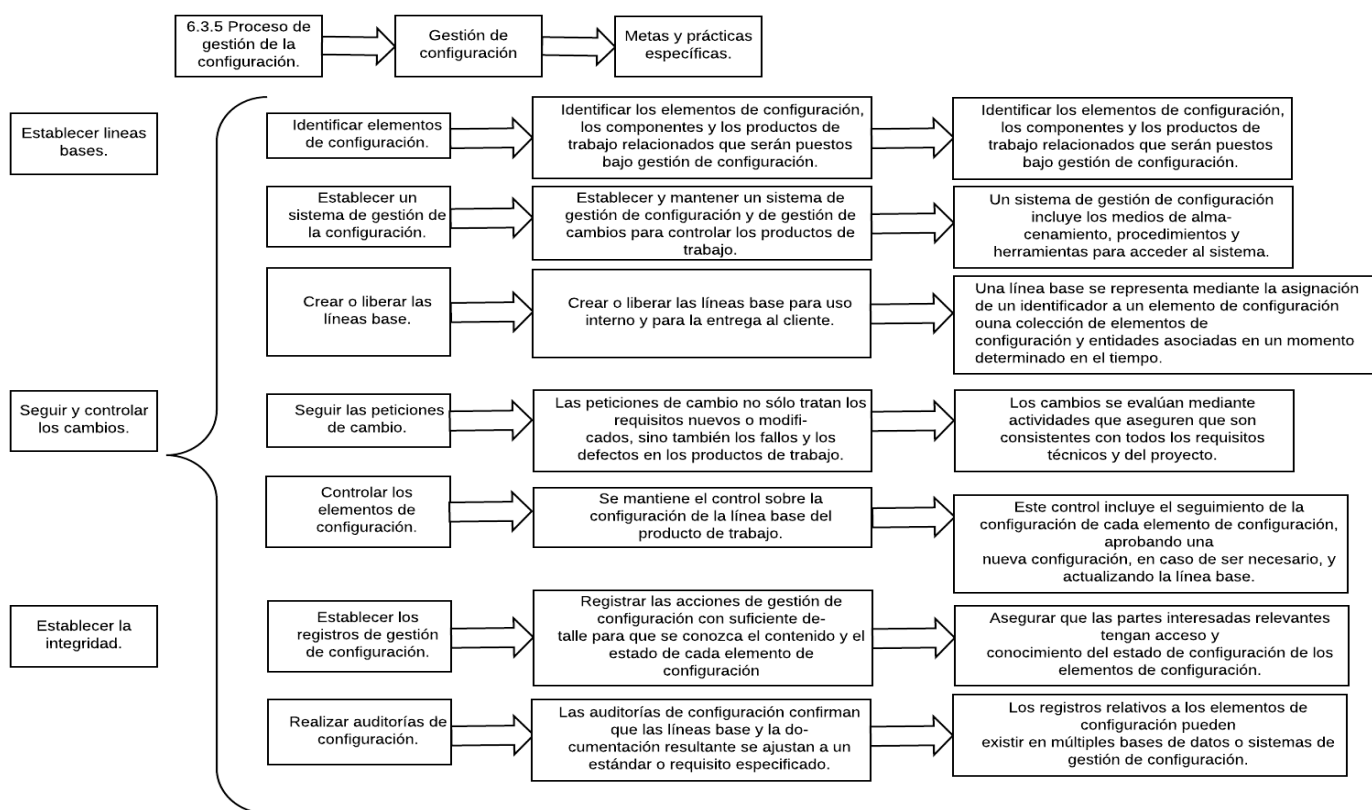


Ilustración 23. Proceso de gestión de la configuración. Fuente: Elaboración propia.

Por otro lado, para el Proceso de gestión de la configuración de la norma ISO 12207 se tiene una relación directa con tres factores de la norma CMMI que son: Establecer líneas bases, seguir y controlar los cambios y establecer la integridad del proyecto como se observa a continuación:

Dentro del proyecto ambas normas tienen un enfoque de ayudar en cuanto al control de cambios dentro del sistema sin impedir seriamente los cambios de manera justificada, si analizamos ambas normas para este proceso nos damos cuenta que tiene una dirección a identificar y establecer los elementos en el sistema, controlando así, el cambio de estos a lo largo de su ciclo de vida, registrando y reportando constantemente el estado de los elementos y las solicitudes de cambio, para verificar que los elementos estén completos y que sean los correctos.

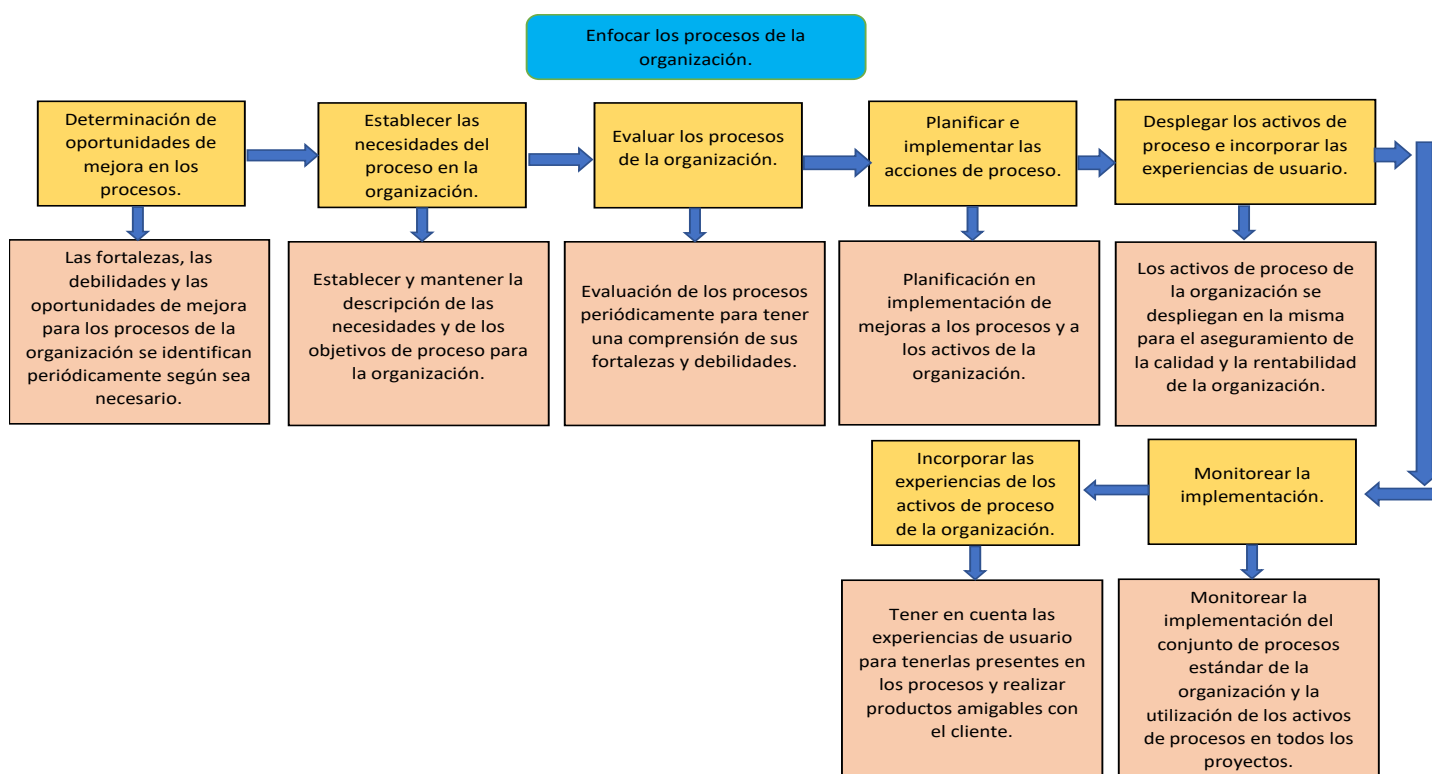


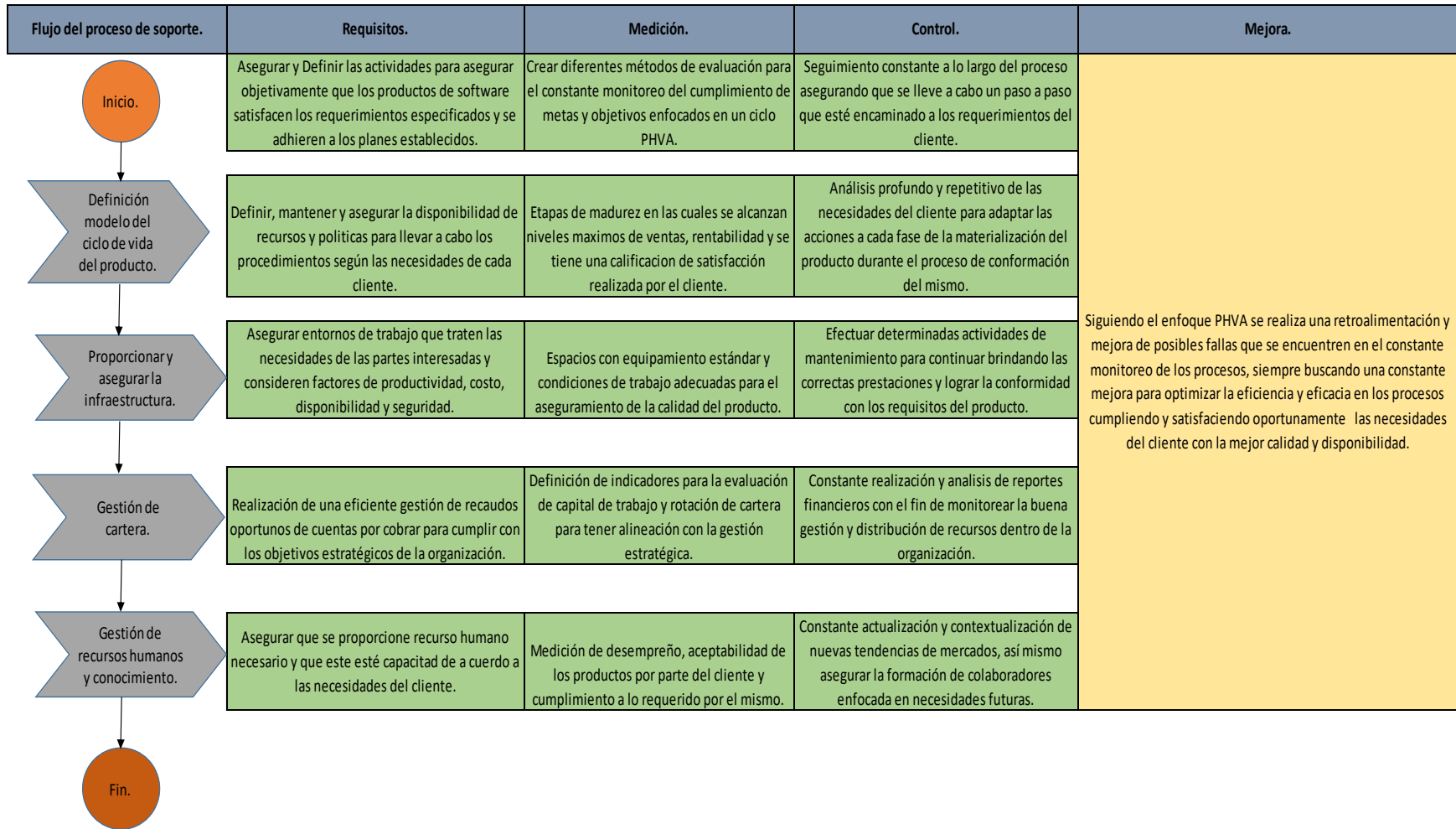
Ilustración 24: Enfoque en procesos de la organización. Fuente: Elaboración propia.

## 7.2. Objetivo 2

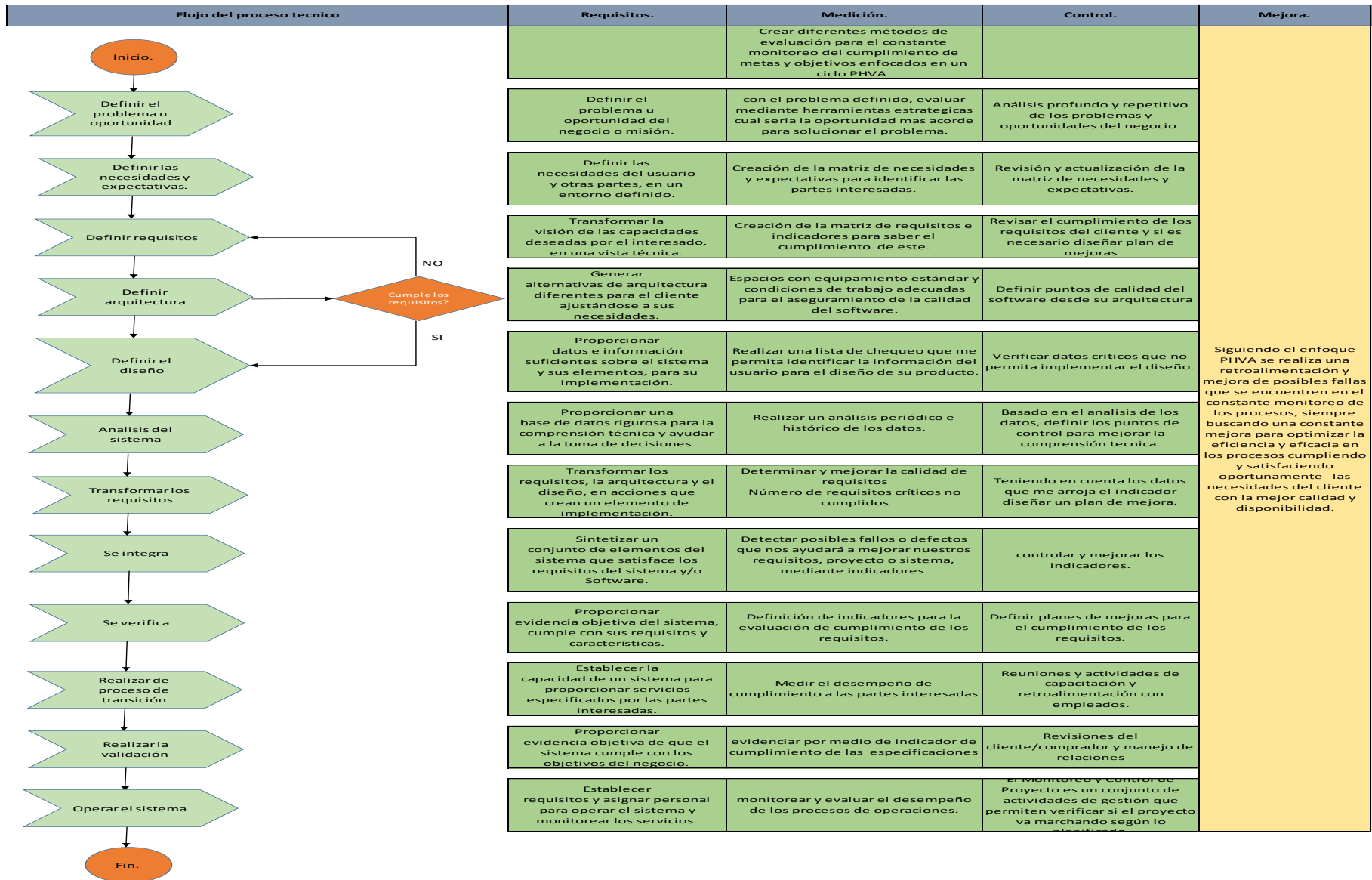
Diseñar un modelo de gestión de procesos para el laboratorio de la industria académica en desarrollo de software para la facultad de ingenierías de la Universidad Católica de Oriente.

A continuación, se realiza un desarrollo de cada uno de los procesos en el cual se presenta los sub procesos de cada uno y donde se define los requisitos, medición y control de esta para asegurar la calidad en cada fase del proceso de desarrollo de Software.


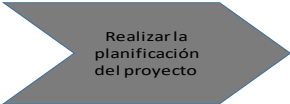

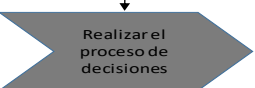
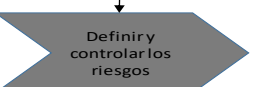
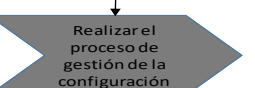
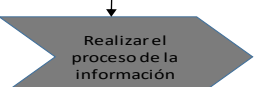
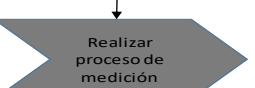
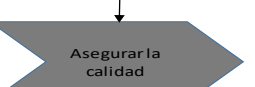

• Proceso de soporte.



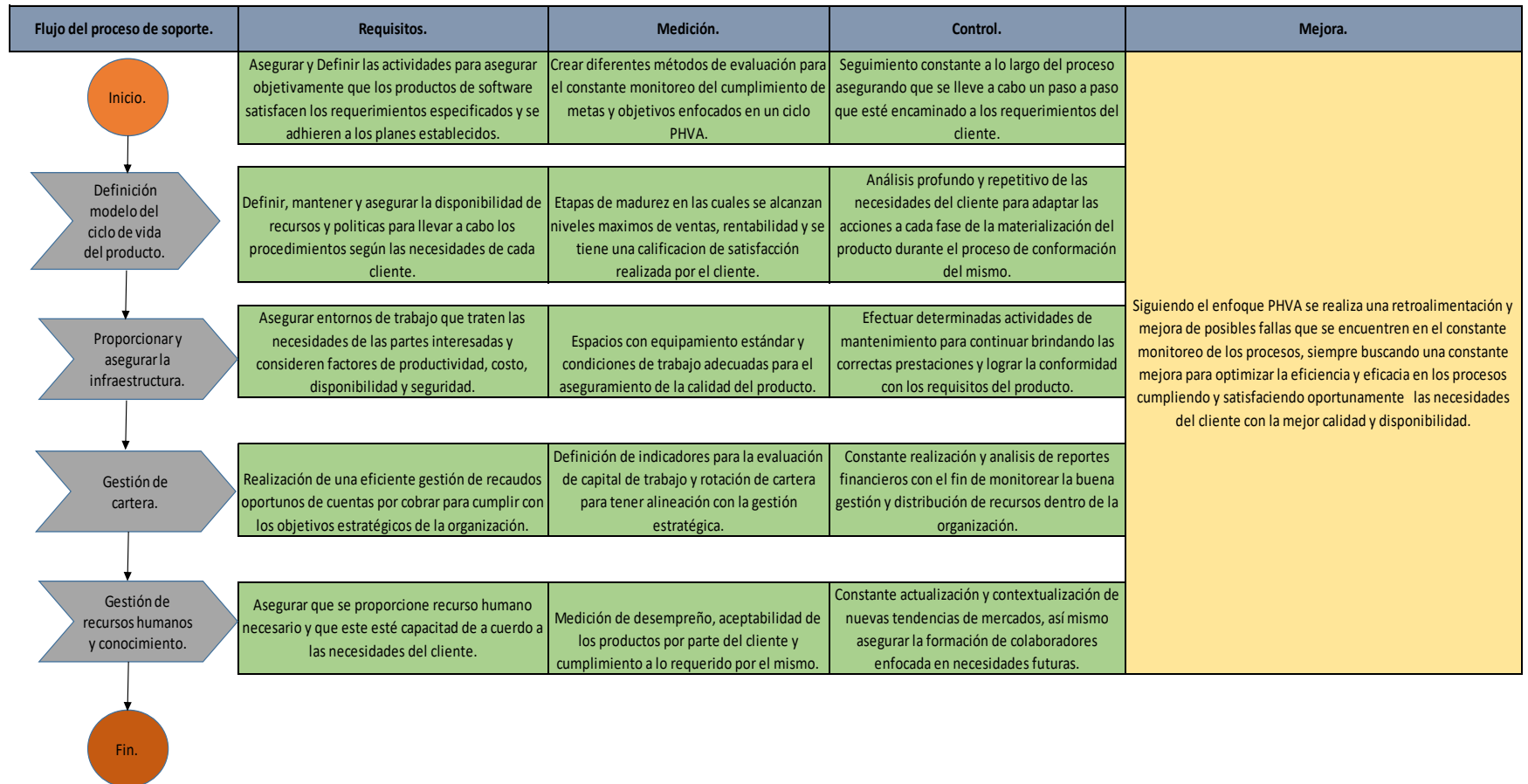
• Proceso tecnico.



• Proceso de gestión técnica.

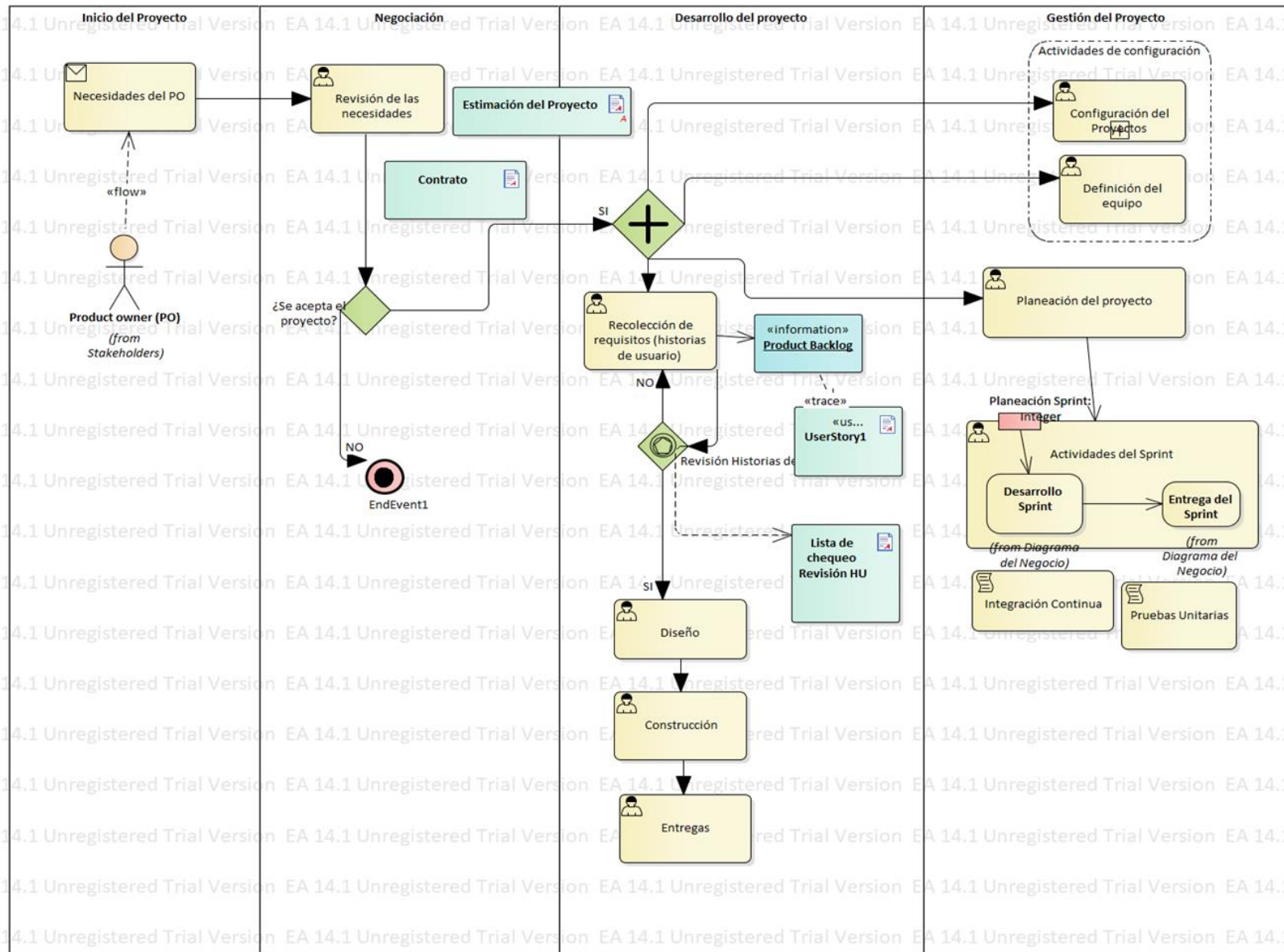
Flujo del proceso de gestión técnica	Requisitos.	Medición.	Control.	Mejora.
	<p>Previa definición del proyecto usando sistemas integrados con un enfoque en las necesidades analizadas con el cliente antes de comenzar el desarrollo del proyecto.</p>	<p>Constante retroalimentación del desarrollo de las actividades con el fin de realizar acciones correctivas cuando sea necesario, todo esto antes de poner el proyecto en funcionamiento.</p>	<p>Realizar periódicamente las acciones correctivas y su impacto positivo a la mejora del proceso.</p>	<p>Siguiendo el enfoque del ciclo PHVA se realiza una retroalimentación constante y mejora de posibles fallas que se encuentren durante todo el monitoreo de los procesos, siempre buscando una constante mejora para optimizar la eficiencia y eficacia en los procesos cumpliendo y satisfaciendo oportunamente las necesidades del cliente con la mejor calidad y disponibilidad.</p>
	<p>Establecer las prioridades del cliente y determinar un foco que va orientado al alcance de los objetivos.</p>	<p>Definición de un criterio de medición el cual lleve a la compañía a saber si se está trabajando con el enfoque que requieren las necesidades del cliente.</p>	<p>Seguimiento constante a lo largo del proceso asegurando que se lleve a cabo un paso a paso que esté encaminado a los requerimientos del cliente.</p>	
	<p>Determinación del tamaño óptimo del proyecto y establecer una serie de indicadores para tener monitoreo constante de las variables de interés.</p>	<p>En base a lo indicadores definidos se realiza un seguimiento para tener control de las actividades que se desarrollan en la organización.</p>	<p>Realizar monitoreo constante a los indicadores establecidos, y si es necesario desarrollar planes de mejora.</p>	
	<p>Establecimiento de un manual en el cual se muestra paso a paso el desarrollo de los proyectos según las necesidades de cada cliente.</p>	<p>Realización de una revisión antes de poner en marcha el proyecto con el fin de tener retroalimentación de posibles fallas en el proceso y poder corregir las mismas.</p>	<p>Identificar por medio de la retroalimentación posibles fallas del proyecto y generar mejoras al proyecto.</p>	
	<p>Revisión y planteamiento de posibles riesgos o falta de seguridad de la información para un posterior montaje de un plan de seguridad de la información.</p>	<p>Constante monitoreo de posibles fugas de información que pueda poner en riesgo la propiedad intelectual de las organizaciones y los procesos que se desarrollan con estas.</p>	<p>Desarrollar una matriz de riesgos, donde se pueda identificar y mitigar por medio de planes de mejora.</p>	
	<p>Asegurar la existencia de un manual de configuraciones de los procesos el cual presenta diversas configuraciones de sistemas según las necesidades que se quieran contrarrestar.</p>	<p>Ayudados de un manual de configuraciones se evalúa el seguimiento de las instrucciones para el correcto desarrollo de las actividades dentro del proyecto.</p>	<p>Actualización constante del manual de configuraciones para estar al día con las instrucciones.</p>	
	<p>Recolección y clasificación de información necesaria para el desarrollo del proyecto con el fin de tener sectorizada la información importante para cada fase del desarrollo del proyecto.</p>	<p>Espacios con equipamiento estándar y condiciones de trabajo adecuadas para el aseguramiento de la información.</p>	<p>Mantener controlada la información y contar con las condiciones óptimas para el desarrollo del mismo.</p>	
	<p>Recopilar, analizar e informar datos e información de objetivos para respaldar una gestión eficaz.</p>	<p>Constante monitoreo de la información y actualización de la base de datos.</p>	<p>Medir el impacto de la gestión eficaz y mejora continua del proceso.</p>	
	<p>Organizar la aplicación efectiva del proceso.</p>	<p>mejorar continuamente la gestión basada en indicadores de calidad en el proceso.</p>	<p>Análisis y control a los indicadores establecidos de calidad.</p>	
				

- Proceso de acuerdo.



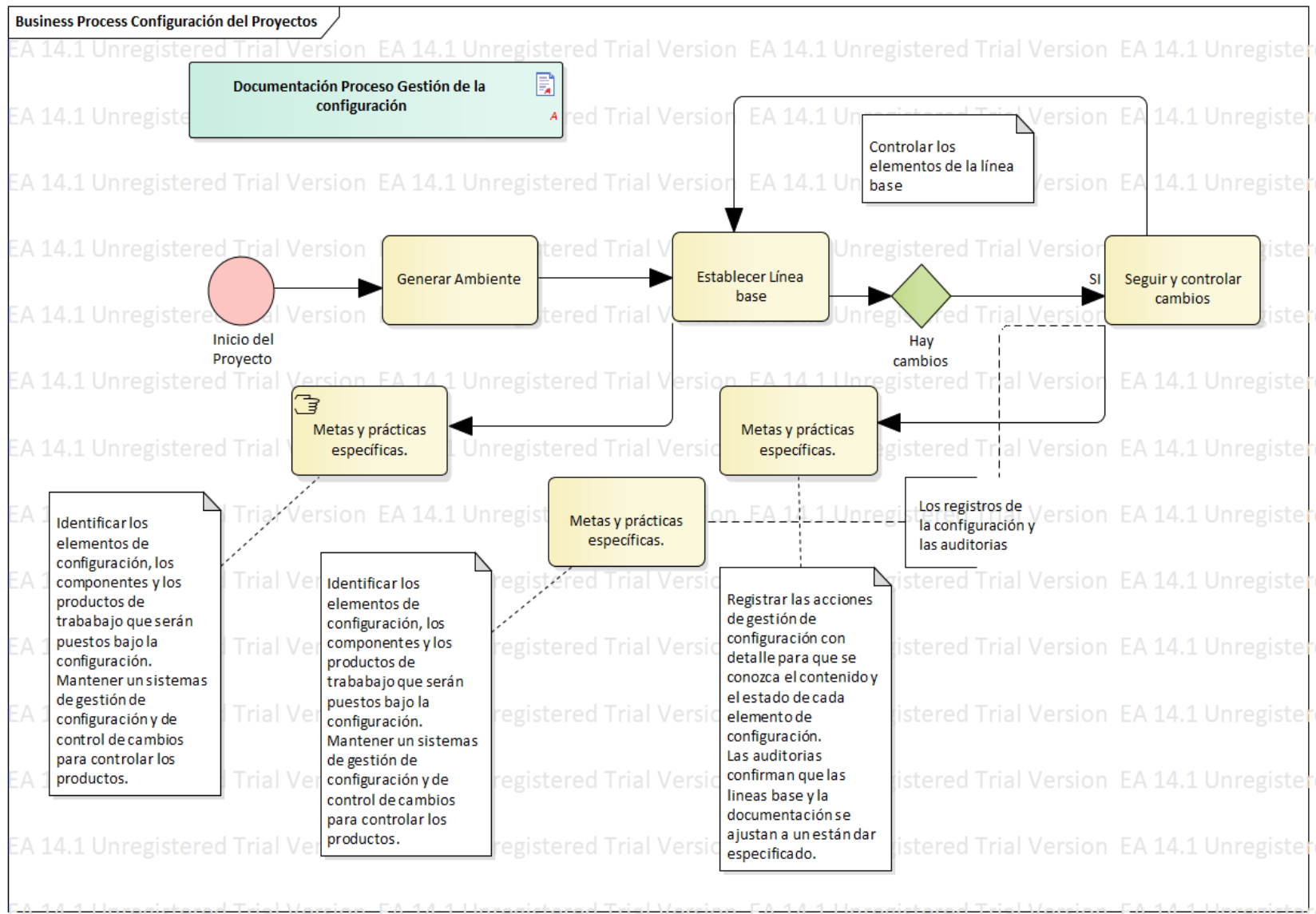
En el siguiente diagrama se presenta el ciclo de vida del producto, dentro de esto se definen cuatro fases que son el inicio del proyecto, la negociación, el desarrollo del proyecto y la gestión del proyecto; cada una de las fases se describen a continuación:





- Fase de inicio del proyecto; dentro de esta fase se tiene la identificación y evaluación de necesidades por parte del cliente.
- Fase de negociación; en esta se realiza una profunda revisión de cada una de las necesidades presentadas por el cliente, en base a esas necesidades se realiza una proyección y estimación del proyecto para posteriormente analizar la viabilidad del proyecto y determina si se hace efectivo o no, un contrato firmado por la partes interesadas.
- Fase de desarrollo del proyecto; es una de las fases más delicadas del proyecto, pues en esta se realiza toda la recolección de los requisitos que debe cumplir el Software para un posterior diseño, construcción y finalmente la entrega del proyecto; dentro de esta fase constantemente se debe realizar una retroalimentación de casa paso que se va dando dentro del desarrollo del proyecto.
- Fase de gestión del proyecto; en esta fase se realiza una configuración del proyecto y una completa definición del equipo mediante una planeación del proyecto que tiene como resultado un desarrollo y entrega de sprint y un producto con una funcionalidad de un 100% en términos de calidad, respaldo y confiabilidad.

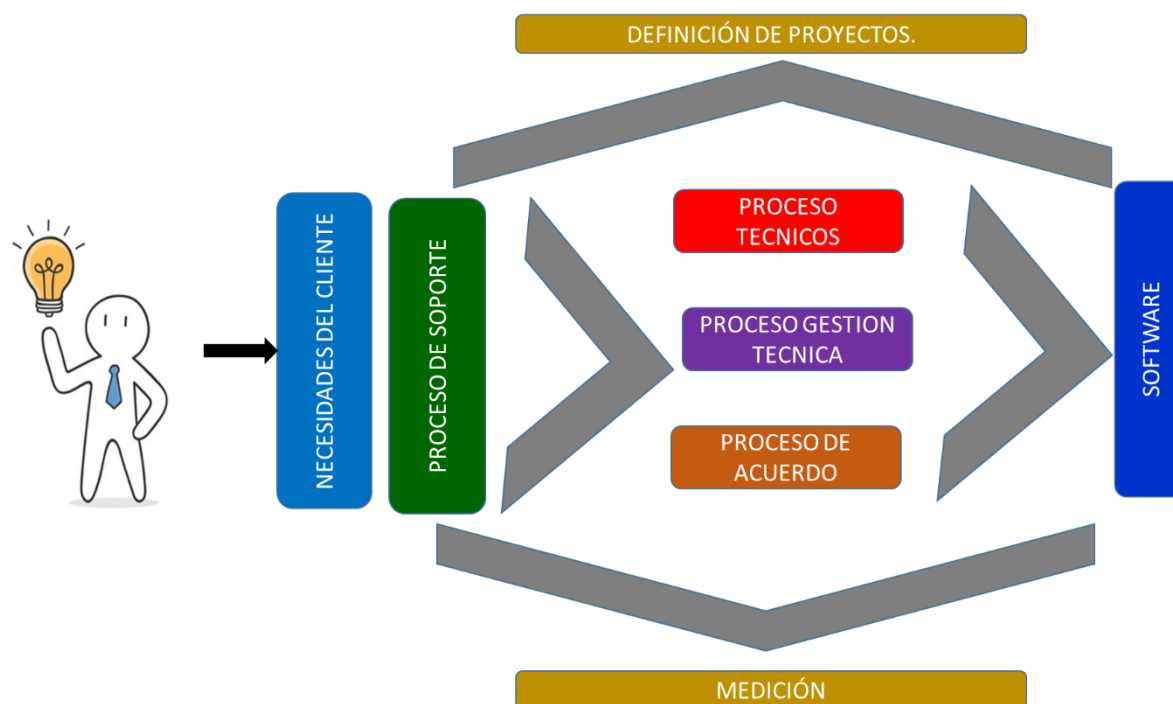
A continuación, se representa todo el proceso de configuración de proyectos teniendo en cada fase ciertas prácticas y metas específicas para asegurar el buen desarrollo de Software. Inicialmente se realiza toda una documentación en la cual se establecen los procedimientos para el desarrollo de cada proyecto, luego se genera todo un ambiente del software donde se determina como se desarrollará el proyecto definiendo unas líneas bases para enfocados en eso controlar posibles cambios que se realicen durante la realización del software.



### 7.3. Objetivo 3

Plantear el modelo de gestión de procesos para el laboratorio de la industria academica en desarrollo de software para la facultad de ingenierias de la Universidad Catolica de Oriente.

Basados en el desarrollo investigativo y propositivo de los objetivos anteriores se logró determinar el planteamiento del modelo de gestión de procesos que comprende este objetivo, lo anterior se plasma de la siguiente manera:



Si se observa el modelo propuesto, se puede evidenciar que se determinaron unas entradas, un proceso y una salida, dentro de las entradas se tienen en cuenta en primera instancia las necesidades del cliente y se define mediante los procesos de soporte todo lo relacionado al ciclo de vida del Software; luego se tiene todo el proceso de desarrollo bajo los lineamientos de los procesos técnicos, proceso de gestión técnica y los procesos de acuerdo, todos estos enfocados en la realización de un producto (Software) que se adecúe a las necesidades del cliente cumpliendo con la normatividad vigente y los más altos estándares de calidad en materia de desarrollo de software; finalmente se encuentra todo lo que son salidas donde básicamente se tiene un Software

completamente funcional que satisface las necesidades del cliente, y se plantea una medición realizada constantemente y analizada en determinados periodos de tiempo para asegurar el buen funcionamiento del Software y trabajar siempre bajo un ciclo PHVA.

A continuacion se definen los procedimientos referentes al modelo propuesto:

Propósito:

Con base a la propuesta del modelo propuesto dentro las instalaciones de la Universidad Católica de Oriente, logrando satisfacer las necesidades de la industria y la academia, se tiene como fin investigar sobre diferentes modelos de desarrollo de software de calidad los cuales estén dentro del alcance de este laboratorio, para con estos construir un modelo lógico y que se puedan implementar en la industria, para luego plantear propuestas coherentes a posibles clientes para que estos evalúen la viabilidad de los proyectos desarrollados con el enfoque de sus empresas.

Alcance:

Teniendo en cuenta los requisitos legales, reglamentario y de las partes interesadas pertinentes tanto internas y externas, SOFTLAB laboratorio de software, determina el alcance para este proyecto, teniendo como referente la norma ISO 9001:2015 e ISO 12207:2017, de la siguiente manera: solicitud del servicio del laboratorio, aceptación de la solicitud, ingreso al laboratorio en el horario acordado, recibimiento por parte del monitor encargado, socialización del manual del laboratorio, acompañamiento por parte del monitor durante la sesión, utilización del laboratorio del usuario, cumplimiento de los requerimientos del cliente, satisfacción del cliente por los servicios prestados, salida del usuario.

Responsabilidades:

Se determinó cuáles son los recursos humanos necesarios para el correcto funcionamiento de SOFTLAB, se contará con personal calificado y no calificado que tenga competencias

adecuadas para la plena interrelación de los usuarios y las personas prestadoras del servicio. Las competencias con las que contará el personal son las siguientes:

#### Personal Calificado

\* Desarrollar un pensamiento y un razonamiento crítico y saber Comunicarlos de manera efectiva.

\* Generar propuestas innovadoras, competitivas en la investigación y en la actividad profesional.

\* Gestionar el tiempo y los recursos disponibles.

\* Trabajar de forma organizada.

\* Capacidad para prevenir y solucionar problemas, adaptándose a situaciones imprevistas y tomando decisiones propias.

\*Comunicar eficientemente de forma oral y/o escrita conocimientos, resultados y habilidades, tanto en entornos profesionales como ante públicos no expertos.

#### Personal no calificado

\* Capacidad para aprender con facilidad y buena interpretación.

\* Adaptación al cambio.

\* Contribución a la mejora continua en la prestación del servicio.

\* Creatividad e innovación sobre cada uno de los procesos.

\* Trabajo en equipo y visión de futurista.

\* Habilidades en las relaciones interpersonales.

\* Reconocimiento a la diversidad y la multiculturalidad.

\*Razonamiento crítico y compromiso ético.

#### Procedimiento:

- Entradas y Salidas

## Entradas

Insumo	Origen
Necesidades del cliente	proceso de acuerdo
Proceso de soporte	proceso de gestión
Normatividad vigente	proceso de gestión técnica

## Salidas

Servicio o Producto	Destino
Necesidad satisfecha.	Empresas
Software	universidades
documentación y datos	centros de investigación

## Identificación y Trazabilidad:

- Trazabilidad.

Dentro de la medición del proceso de suministro se plantea un constante seguimiento del producto con el fin de tener conocimiento del funcionamiento en tiempo real del software para monitorear el correcto desempeño de este y tomar acciones correctivas a tiempo en caso de fallas.

Lo anterior con el fin de asegurar que durante etapa de adquisición el cliente haya recibido el producto que cumpla con los estándares de calidad contemplados por la norma ISO.

Esto se resume en la realización de diferentes registros los cuales serán diligenciados por el cliente cada cierto tiempo en los cuales se plasme toda la información necesaria para saber la adaptabilidad del cumplimiento de estándares del producto.

## Propiedad del Cliente:

El cliente tiene como propiedad las especificaciones que da de su producto, tanto la información personal y/o de la empresa, y el software terminado. En caso de daño o deterioro del software el cliente cuenta con seis meses de garantía que se hará efectiva presentando la factura del servicio. No cubre la garantía cuando se modifique la parte de arquitectura del software.

Proceso:

(#)	(Quien)	(Actividad)
1	Cliente	Especificar los requerimientos del producto., el cliente hace su solicitud de los componentes del software dependiendo de su necesidad.
2	PU Ingeniería de sistemas	Implementar el proceso de soporte. Donde se define el ciclo de vida del producto, se proporciona y se asegura la infraestructura del software, se gestiona la cartera, y se gestiona el personal competente para esta actividad.
3	PU Ingeniería de sistemas	Realizar el proceso técnico, donde se define el problema u oportunidad del proyecto a desarrollar, identificar las necesidades y expectativas de las partes interesadas, para establecer un conjunto de actividades con un cumplimiento en los requisitos, se establece el diseño, se analiza, se transforma los requisitos, se verifica, se integra, se valida y se opera el sistema.
4	PU Ingeniero de sistemas	Gestión técnica, se planifica, se evalúa y se genera puntos de control, se genera el proceso de decisión, se identifica y controla los riesgos, se realiza la configuración del proyecto, se ejecuta el proceso de información, medición donde se pueda asegurar la calidad.
5	PU Ingeniero de sistemas	Proceso de acuerdo, se debe realizar la evaluación y realización del proceso de acuerdo, proceso de adquisición, y proceso de suministro.
6	PU Ingeniero de sistemas	Se mide y se implementa las acciones de mejora.



## Vocabulario y siglas

Término	Definición
PU	Profesional universitario
Software	Hace referencia a un programa o conjunto de programas de cómputo que incluye datos, procedimientos y pautas que permiten realizar distintas tareas en un sistema informático.
Proceso	Es una secuencia de pasos dispuesta con algún tipo de lógica que se enfoca en lograr algún resultado específico.

## 8. CONCLUSIONES

El principal objetivo de esta tesis fue formular un modelo de gestión de procesos bajo el enfoque de aseguramiento de la calidad, basado en el ciclo de mejora continua PHVA para la implementación de un laboratorio de desarrollo de software, de la facultad de ingenierías de la Universidad Católica de Oriente, mediante este se realizó el análisis tanto de la ISO 12207 la cual está orientada al ciclo de vida del producto comparada con la norma CMMI, la cual se encontró gran similitud en la implementación de la calidad del software y las buenas prácticas para la implementación del mismo, donde mediante las diferentes herramientas se pudo desglosar y comparar las dos normas anteriormente mencionadas arrojando como resultado una mejor comprensión e identificación de los diferentes puntos de control de la calidad y así reforzar de manera directa los estándares de calidad que debe tener los diferentes desarrollos de software elaborados en el laboratorio.

Asimismo, mediante la encuesta realizada a personas que pertenecen a empresas y que desempeñan un rol específico dentro del sector de desarrollo de software de la ciudad de Medellín, donde lo que se pretendió era conocer los diferentes desarrollos y herramientas utilizadas para el desarrollo e implementación del software, en el que dejó ver claramente la diversidad que existe entre los proyectos de desarrollo de software, cada uno aplica metodologías e implementa herramientas de diferentes maneras. A pesar de esto se pueden identificar los principales medios para una construcción exitosa de Software. Por lo cual se propone un modelo de gestión de procesos estandarizado.

De igual forma se realizó el diseño y planteamiento del modelo de gestión de procesos, definiendo un mapa de procesos, en el cual se muestra desglosado cada uno de los procesos de soporte, técnico, gestión técnica y de acuerdo donde se contaba con flujogramas de procesos, identificando así los diferentes requisitos, medición y control de cada uno, dando como resultado

el insumo para la creación del procedimiento de desarrollo y mantenimiento del software establecido, formando así una estructura lógica que debe tener el laboratorio de software, contando con las pautas de calidad necesarias para la aplicación del mismo en la Universidad Católica de Oriente.

## REFERENCIAS BIBLIOGRÁFICAS

- Alur, D., Crupi, J., y Malks, D. (2003). *Core J2EE Patterns: Best Practices and Design Strategies (Second edition)*. Prentice Hall.
- Asato España, J. A. (2015). E. R. G. J. G. F. M. G. M. R. V. M. España, «Pistas Educativas, No. 114,» diciembre. Recuperado de:  
file:///C:/Users/Dell%2014%20%203458/Downloads/295-569-2-PB.pdf.
- B. B. Agarwal, M. G. (2010). *Software Engineering & testing: An Introduction*. Canada: Jones and Bartlett Publishers, LLC.
- Bass, L., Clements, P., y Kazman, R. (2003). *Software Architecture in Practice (Second Edition ed.)*. Addison-Wesley.
- Beck, K. (2000). *Extreme programming explained: embrace change*. Boston: Addison-Wesley Longman Publishing Co., Inc.
- Bentley, W. (2008). *Análisis de Sistemas, Diseños y Métodos (7 ed.)*. México: McGraw-Hill Interamericana.
- Boehm, B. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer* 21, 61-72.
- Braude, E. J. (2003). *Ingeniería de Software: Una perspectiva orientada a objetos*. México: ALFAOMEGA GRUPO EDITOR.
- Bunge, M. (2004). *How does it work? The search for explanatory mechanisms. Philosophy of the Social Sciences*.
- Burbeck, S. (2014). *Applications Programming in Smalltalk-80(TM)*:. Obtenido de How to use Model-View-Controller (MVC): <http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>

- Clements, P., y Northrop, L. (1996). *Software Architecture: An Executive Overview*. CMU Tech Report.
- Coad, P., Lefebvre, E., y De Luca, J. (1999). *Java Modeling In Color With UML: Enterprise Components and Process*. Prentice Hall.
- Cockburn, A. (2001). *Agile Software Development*. Boston, MA, U.S.A.: Addison Wesley Professional.
- Código Civil Colombiano*. (3 de Julio de 2013). Obtenido de [http://200.75.47.49/senado/basedoc/codigo/codigo\\_civil\\_pr046.html#1501](http://200.75.47.49/senado/basedoc/codigo/codigo_civil_pr046.html#1501)
- Constantine, L., y Yourdon, E. (1975). *Structured design* (2 ed.). Prentice Hall.
- Contrato*. (2014). Obtenido de <http://claudiaachevarria.galeon.com/cvitae1808035.html>
- DeMarco, T. (1979). *Structured analysis and systems specifications*. Prentice Hall.
- ECURED*. (16 de 04 de 2014). Obtenido de Conocimiento con todos y para todos: <http://www.ecured.cu/index.php/Crystal>
- Fenton, N. (1995). *Software quality Assurance and Measurement: a worldwide perspective*. Boston, USA: International Thompson Computer Press.
- Fisher, M. (1979). *Definitions in Software Quality management*. New York, USA: Petrocelli Books.
- Fuggetta, A. (2000). *Software Process: A Roadmap*. 22nd International Conference on Software Engineering (ICSE '2000), future of SoftwareEngineering Track. Limerick (Irlanda): ACM.
- Gane, C., y Sarson, T. (1977). *Análisis Estructurado de Sistemas*. (Quinta reimpresión. ed.). El Ateneo.
- García, P. (2003). *Calidad en el desarrollo y mantenimiento del software*. Madrid: RA-MA.

Gonzalez, M. (19 de Julio de 2014). *La letras al Derecho*. Obtenido de Para conocer y negociar sus derechos de autos: <http://letrasalderecho.info/derechos.html>

Goyal, S. (2008). *Agile Techniques for Project Management and Software Engineering*.

*HER Consultores Jurídicos*. (2012). Obtenido de

<http://www.despachoabogados.com.mx/blog/que-es-un-contrato-legal.html>

Highsmith, J. (2000). *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Dorset House.

Humphrey, W. (1995). *A discipline for Software Engineering*. USA: Addison-Wesley Publishing Company.

IEEE. (2000). *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*. IEEE Std 1471-2000. .

Kan, S. (1995). *Metrics and Models in Software Quality Engineering*. USA: Addison-Wesley Publishing Company.

Kruchten, P. (2000). *The Rational Unified Process : An Introduction*. Addison-Wesley Professional.

*Manifesto for Agile Software Development*. (16 de Abril de 2001). Obtenido de Agile Alliance: <http://guide.agilealliance.org>

Martin, R. C.; Beedle, M.; Bennekum, A. V.; Cockburn, A.; Cunningham, W.; M. Fowler ,W.; Highsmith , J.; Jeffries, R.; Kern, J.; Marick, B; . Schwaber, K; . Sutherland, J. y

Thomas, D. (S.F.). «Manifesto for Agile Software Development,» 01 04 2019.

Recuperado de : <http://agilemanifesto.org/iso/es/manifesto.html>.

*Metodologías ágiles de desarrollo*. (16 de Abril de 2014). Obtenido de

<http://agilidadeldesarrollo.wordpress.com/2012/12/02/adaptive-software-development-asd/>

- Modelos de Contrato .net.* (2014). Obtenido de <http://www.modelocontrato.net/modalidades-de-contrato-de-compraventa.html>
- Myers, G. (1975). *Reliable Systems through Composite design* (1 ed.). Petrocelli Charter.
- Nielsen, J. (1999). . *Designing Web Usability: The Practice of Simplicity*. Indianapolis: New Riders Publishing.
- Norman Fenton et all; Software quality Assurance and Measurement: a worldwide perspective; International Thompson Computer Press; Boston, USA; 1995.* (s.f.).
- Page-Jones, M. (1980). *The Practical Guide to Structured Systems Design*. Yourdon Press.
- Piattini, M. (1996). *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*. Madrid: Rama.
- Pressman, R. (2010). *Ingenieria del Software Un Enfoque Práctico* (7 ed.). MCGRAW-HILL.
- Que Aprendemos hoy.com.* (2012). Obtenido de <http://queaprendemoshoy.com/elementos-esenciales-y-efectos-de-un-contrato/>
- Reifer, D. (1985). *State of the art in Software Quality Management*. Torrance: Reifer Consultants.
- Royce, W. (1970). "Managing the Development of Large Software Systems". *Proceedings of IEEE WESCON 26*, 1-9.
- Rumbaugh, J. (2007). *EL LENGUAJE UNIFICADO DE MODELADO. MANUAL DE REFERENCIA (2ª ED.) (EN PAPEL)*. (ADDISON-WESLEY, Ed.)
- Rumbaugh, J., Booch, G., y Jacobson, I. (1999). *The Unified Modeling Language Reference*. Addison-Wesley.
- R. S., Jiménez. (s.f.). «GRUPO DE LABORATORIO DE INVESTIGACIÓN PARA EL,» 2015-2016. Recuperado de:

<https://www.usbcali.edu.co/sites/default/files/planestrategicogrupolaboratoriodeinvestigacionparaeldesarrollodelaingenieriadesoftwarelidis.pdf>.

Sabino, C. (1992). *EL PROCESO DE INVESTIGACIÓN*. Caracas: Panapo.

Schwaber, R., Beedle, K., y Martin, M. (2001). *Agile Software Development with SCRUM*. Prentice Hall.

SEI. (2010). *CMMI® para Desarrollo, Versión 1.3. CMMI-DEV, V1.3. Mejora de los procesos para el desarrollo de mejores productos y servicios*.

Sommerville, I. (2005). *Ingeniería de Software*. Madrid: PEARSON EDUCACIÓN. S.A.

Stapleton, J., & Constable, P. (1997). *DSDM: Dynamic Systems Development Method: The Method in Practice*. Addison-Wesley Professional.

Taylor, R., Medvidovic, N., y Anderson, K. (Junio de 1996). A Component- and Message-Based Architec



## ANEXOS

### Anexo 1. Procesos de desarrollo de software hoy

#### *Introducción*

Con el propósito de enriquecer el proceso de aprendizaje de los estudiantes de Ingeniería de Sistemas de la Universidad Católica de Oriente se toma la iniciativa de realizar un proyecto orientado a la creación de un *Laboratorio de Software* para el apoyo en el desarrollo de proyectos.

Se realizaron varios frentes, a saber: Revisión de fuentes primarias y secundarias. Revisión de que se está haciendo en el medio

Para responder la pregunta ¿Qué están haciendo las empresas de desarrollo de software en Medellín?, se realiza una encuesta que contempla 3(tres) partes fundamentales dentro del proceso de desarrollo de software, se identificaron de la siguiente manera: Se inicia con la información metodológica, seguida del desarrollo y ejecución de procesos, terminando con los factores de documentación.

La muestra de la encuesta es de 40(cuarenta) personas que desempeñan diferentes roles clasificados en las siguientes 6(seis) categorías:

- Desarrollador(a) de Software
- Analista de Calidad
- Agile Coach
- Analista TI - Infraestructura
- Consultor(a)
- Ingeniero(a) DevOps

## Roles

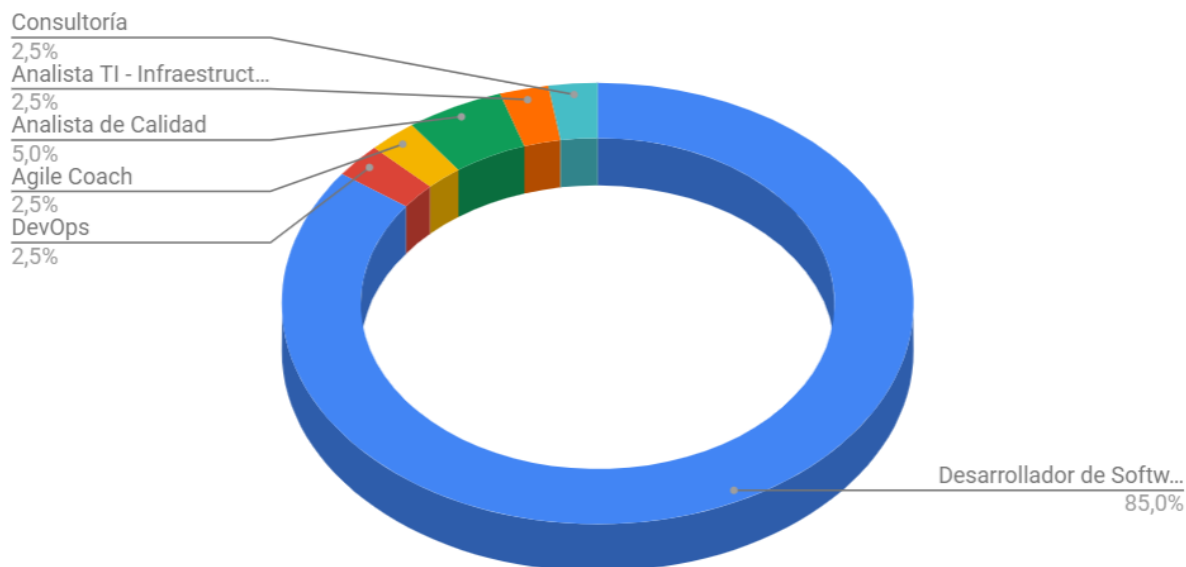


Ilustración 25: Roles de los desarrolladores de Software en la actualidad. Fuente: Elaboración propia.

De la figura anterior, con base a la encuesta realizada a las empresas desarrolladoras de software de Medellín, en el cual se describen los diferentes roles que se tienen se encuentra que el desarrollador de software cuenta con un 85%, siendo el más predominante frente a las otras opciones.

### *Metodologías*

La organización y planeación son dos factores fundamentales dentro el desarrollo de cualquier proceso, por tal motivo se recolectó información acerca de los enfoques de trabajo bajo los cuales se ejecutan los desarrollos de software.

Bajo esta instancia se presentan las siguientes opciones:

- Metodologías Ágiles
- Metodología en Cascada
- Método de Prototipos

- Modelo Incremental o Iterativo y Creciente
- Modelo en Espiral
- RAD: Rapid Application Development (Desarrollo Rápido de Aplicaciones)
- Otras

Se tiene el conocimiento de que las metodologías ágiles implementan diversos marcos de trabajo, por lo cual también se indaga acerca de estos marcos si la respuesta del enfoque utilizado era *Metodologías Ágiles*.

Estos fueron los resultados:

### Enfoque bajo el cuál se desarrollan los proyectos de Software

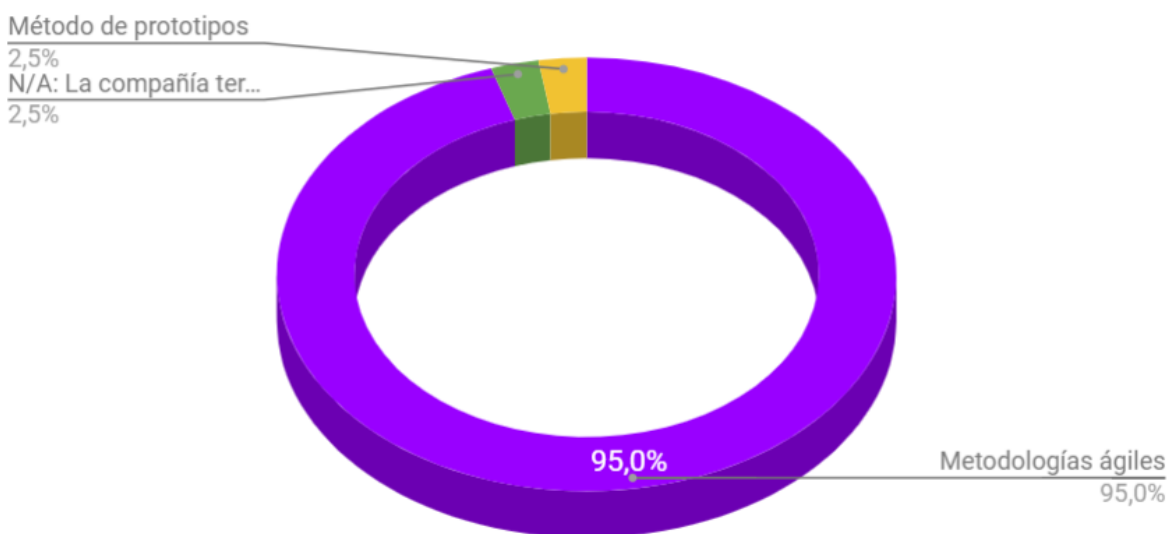


Ilustración 26: Enfoque de los desarrollos de Software en la actualidad. Fuente: Elaboración propia.

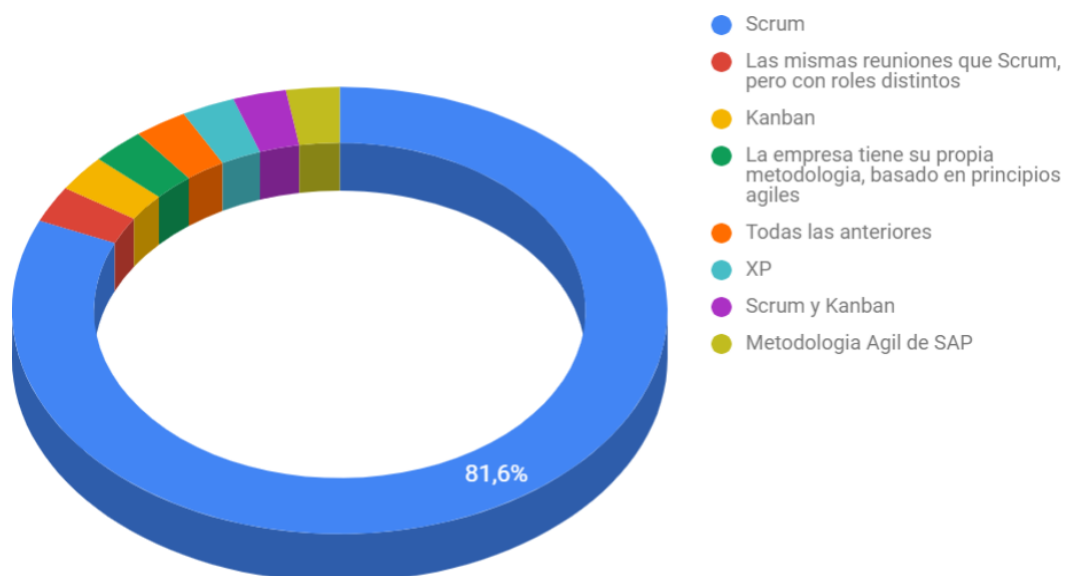


Ilustración 27: Metodologías implementadas en la actualidad. Fuente: Elaboración propia.

En este segmento de la investigación también se pudo concluir que un 45% (cuarenta y cinco por ciento) de las personas partícipes de la encuesta consideran que la metodología que implementan ha llevado a resultados exitosos de los proyectos que han desarrollado.

Pero si se habla de éxito se considera que existen parámetros y mediciones que permiten validar y afirmar cuando un proyecto es exitoso o no, por lo que también se hizo necesario investigar sobre ello lo cual arrojó el siguiente resultado,

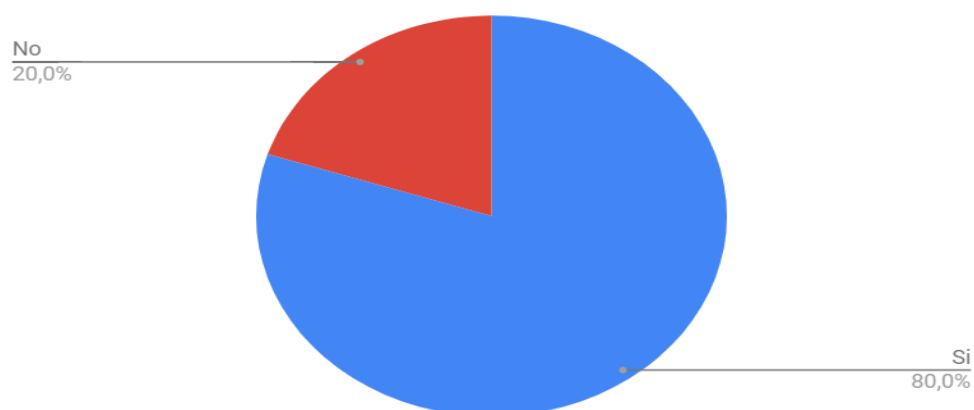


Ilustración 28: Uso de validaciones sobre la metodología usada dentro del desarrollo de proyectos. Fuente: Elaboración propia.

Con base a estas respuestas, se realizó un análisis sobre los métodos en los cuales se basan estas validaciones y se agruparon en los siguientes puntos:

- Durante la Sprint Review Meeting
- Cumplimiento de objetivos
- Entrega de valor durante los sprints
- Satisfacción del cliente
- Curva de alcance
- Feedback temprano
- Métricas del flujo de trabajo

Para cerrar esta primera etapa de análisis se presenta una serie de elementos que los encuestados consideran se deben tener presentes si se quiere desarrollar un modelo de trabajo que sea satisfactorio tanto para los desarrolladores como para el cliente:

- Adopción de integración continua.
- No a las ceremonias estrictas, ajustarlas a las necesidades del equipo.
- Realizar tasking acorde a las capacidades de cada miembro del equipo.
- Entregas continuas.
- Cambio cultural: compromiso, motivación y voluntad.
- Adecuado manejo de dependencias externas: que no afecten en gran medida el trabajo del equipo.
- Flexibilidad en estimaciones: sub o sobreestimación.
- Posibilidad de trabajo remoto.

## Herramientas.

Esta segunda parte de la investigación se centra en el desarrollo, para esto se examinan aspectos basados en las herramientas utilizadas dentro del proceso de desarrollo: Modelado, Gestión, Programación, Pruebas, Integración.

Existen cientos de herramientas para cada proceso, por esto, los resultados son agrupados en tablas, con el objetivo de sintetizar las respuestas obtenidas con las herramientas más utilizadas por cada proceso.

Herramientas para la creación de diagramas y modelado de Software	Enterprise Architect
	Google Drawings
	Guidewire
	ArgoUML
	UML Graph
	StarUML
	Toad Data Modeler
	JDeveloper
	Visio
	Lucidchart
	Drawio
	UMLetino
	Bizagi

Herramientas para la gestión de proyectos	Team Foundation Server
	Trello
	Jira
	Visual Studio Team Server
	Tuleap
	Azure Devops
	Microsoft Planner
	Stories On Board

Herramientas para mockups y patrones de diseño	Balsamiq
	Mockflow
	WireframePro
	Evolous Pencil

Herramientas de apoyo para despliegue de aplicaciones	Docker
	Artifactory
	Kubernetes
	Jenkins
	GCloud
	Azure

	AWS
	Serverless Framework
	Ruxit
	Control Room de Automation Anywhere
	UrbanCode Deploy
	GlassFish
Herramientas para el proceso de integración, entrega y despliegue continuo (Aplicaciones autocontenidas)	Jenkins
	BickBucket
	Concourse
	Gitlab
	VSTS
	Urban Code
	Docker
Herramientas de versionamiento de código	Git
	Bitbucket
	SVN
	VSTS
Herramientas de análisis de código estático	Fortify



Google CodePro
Sonar
PMD

	Consideraciones	Tipos de arquitecturas
Parámetros de definición de arquitectura	Necesidades de la compañía	Arquitectura por capas
	Tendencias tecnológicas	Monolítica, Servicios o Microservicios
	Bajo requerimientos no funcionales	Hexagonal
	En base al rendimiento	
	De acuerdo a necesidades particulares	
	Principios SOLID	

Frameworks	React
	Angular
	Spring
	Mockito
	Hibernate
	JUnit
	Entity

	Bootstrap
	Serenity
	Selenium
	PlayFramework

Bases de datos	Oracle
	SQL Server
	DynamoDB
	Mongo
	Cassandra
	MySQL
	Postgress
	H2

Herramientas para pruebas automatizadas	Selenide
	Cucumber
	Serenity
	Protractor
	JMeter
	Kalaton

	Appium
Librerías	Akka
	Phantom
	Vavr
	ReactJs
	Gson
	Redux
	Sagas
	Lodash
	Log4j
	jQuery
	iText
Herramientas y patrones para el aseguramiento de la calidad del código	DOD
	Pruebas Unitarias
	Pruebas de Integración
	Análisis de código estático
	Revisión de PRs
	TDD

Pipelines de Integración
ATDD
BDD

## IDE

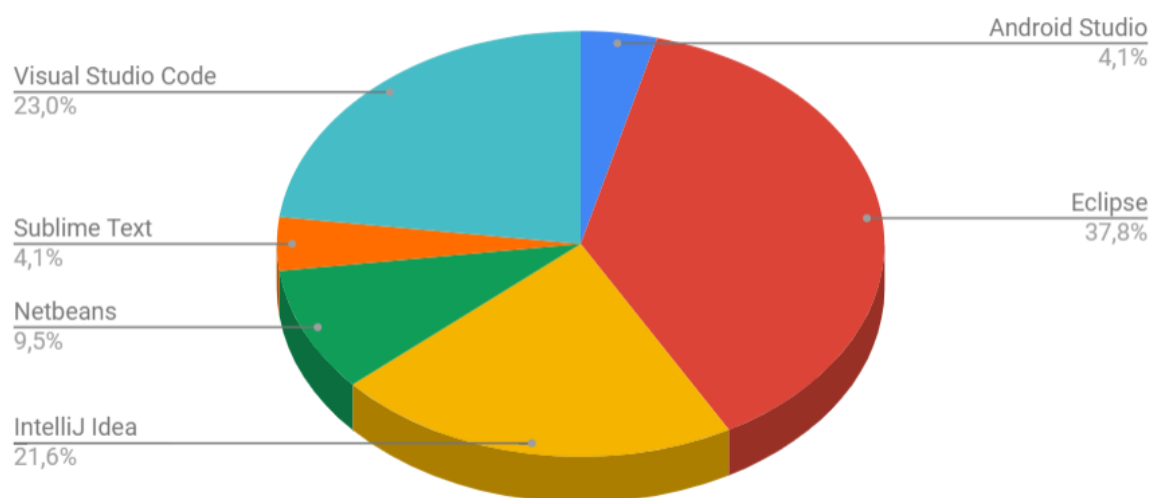


Ilustración 29: Programas para desarrollo de Software. Fuente: Elaboración propia.

Según la gráfica anterior, se puede observar que el programa más usado en la actualidad es el Eclipse con un porcentaje de 37.8 y el menos porcentaje lo tiene la interfaz sublime text; de lo anterior se pueden obtener tendencias de tecnologías que se emplean en la actualidad para desarrollo de software sobre todo en Antioquia.

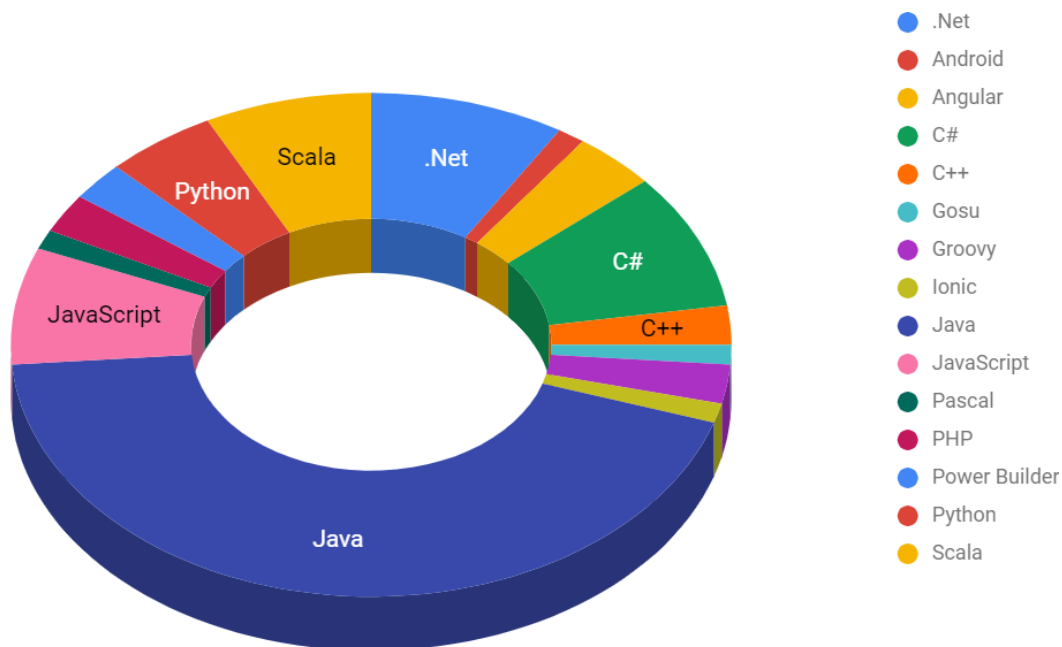


Ilustración 30: Lenguajes de programación. Fuente: Elaboración propia.

En el diagrama presentado anteriormente se plasman los porcentajes de participación en el mercado de diferentes lenguajes de programación usados en la actualidad por empresas desarrolladoras de Software, de lo anterior, se puede afirmar que el lenguaje de programación más usado hoy día es Java, esto por sus diferentes características como accesibilidad y facilidad de manejo de este lenguaje; además se observa que los lenguajes más poco usados son C# y Ionic.

#### *Documentación.*

La última sección de este trabajo fue dedicada a la documentación dentro de los proyectos, dentro de las respuestas fueron hallados los siguientes documentos que son realizados en los procesos de desarrollo implementados por los participantes:

- Diagramas de flujo
- Especificación de Requisitos
- Historias de Usuario

- Modelo 4+1 (arquitectura)
- Documentación basada en el Inception
- Diagramas de entidad relación
- Diagramas de estado
- Diagramas de despliegue
- Diagramas de componentes, clases, secuencias
- Manuales operativos y manuales de usuario

Toda la documentación se ubica en diferentes medios de almacenamiento, entre los cuales se tienen herramientas como lo son

- Gitlab
- Confluence
- Drive
- Wikis
- Cloud
- VSTS
- Swagger
- Teams
- WorkManager
- Servidores Locales

#### CMMI y Modelos de calidad de software

Los procesos de medición y análisis son importantes a la hora de conocer los indicadores y análisis respectivamente, con los cuáles se podrá determinar la satisfacción del producto o servicio y, a su vez, identificar los riesgos para prevenir errores y garantizar la calidad.

Se debe tener en cuenta que el modelo para el desarrollo de software, al igual que los cientos de modelos que se encuentran en el mercado, son una guía para implementar un modelo de organización. Cada proyecto es particular, si bien se puede usar lecciones aprendidas de los desarrollos previos, debe considerarse que siempre se encontrarán con nuevos retos y características. El uso de un modelo es muy importante, sobre todo para el aseguramiento de resultados exitosos y con estándares de calidad.

1. Nivel Inicial: en este nivel de madurez los procesos suelen ser ad hoc y caóticos, esto debido a la falta de organización en materia metodológica. Si bien los productos realizados en este nivel pueden ser funcionales frecuentemente exceden el tiempo de planeación y el presupuesto. Los equipos y organizaciones en este nivel se caracterizan por tendencia a abandonar los proyectos y no están en la capacidad de reincidir procesos exitosos.

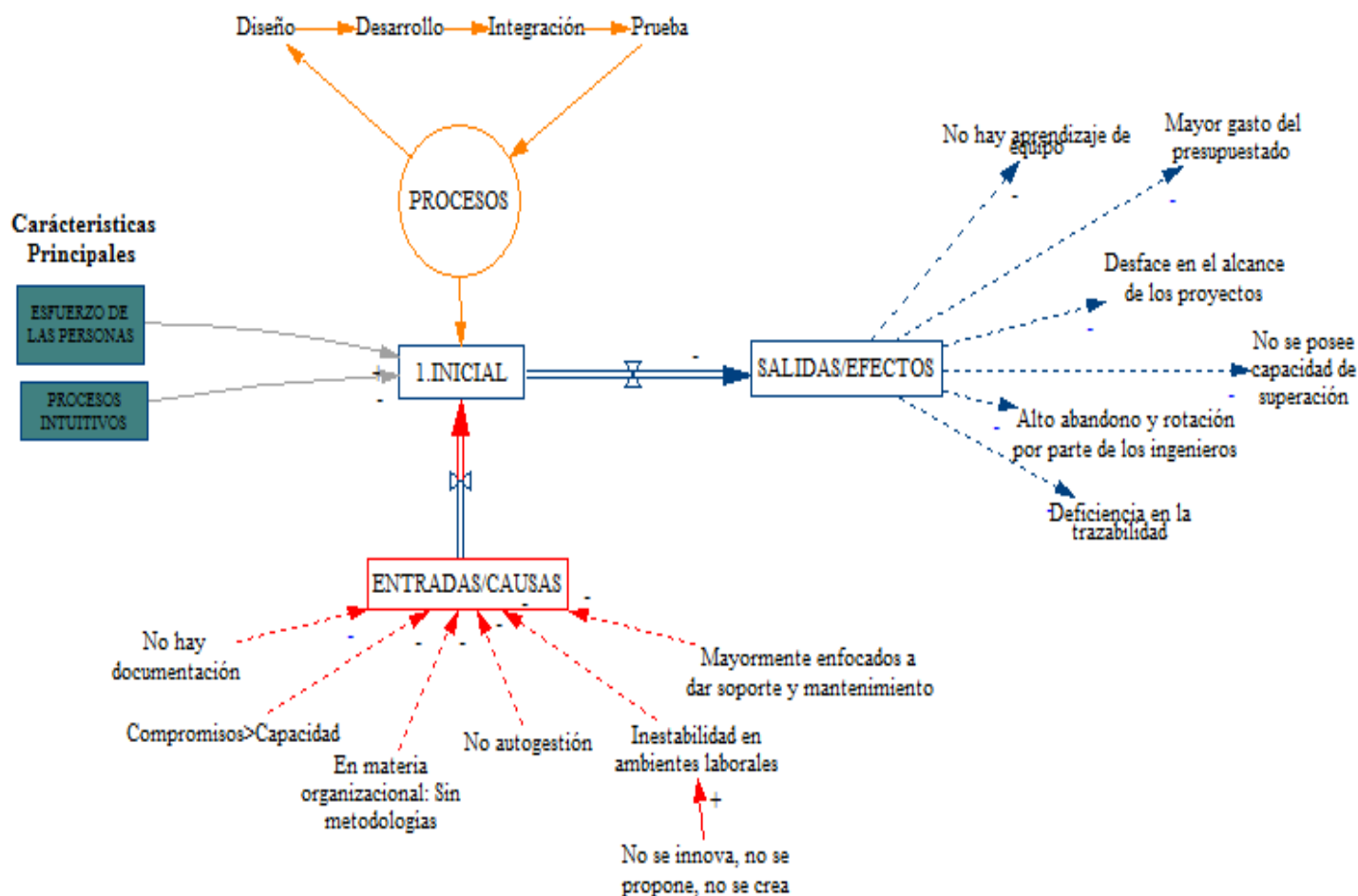


Ilustración 31: Nivel inicial de modelos de calidad de Software. Fuente: Elaboración propia.

2. Nivel Administrado: en este nivel los proyectos aseguran procesos de planificación medidos y controlados esto promueve a la satisfacción de las partes interesadas frente a los resultados finales.



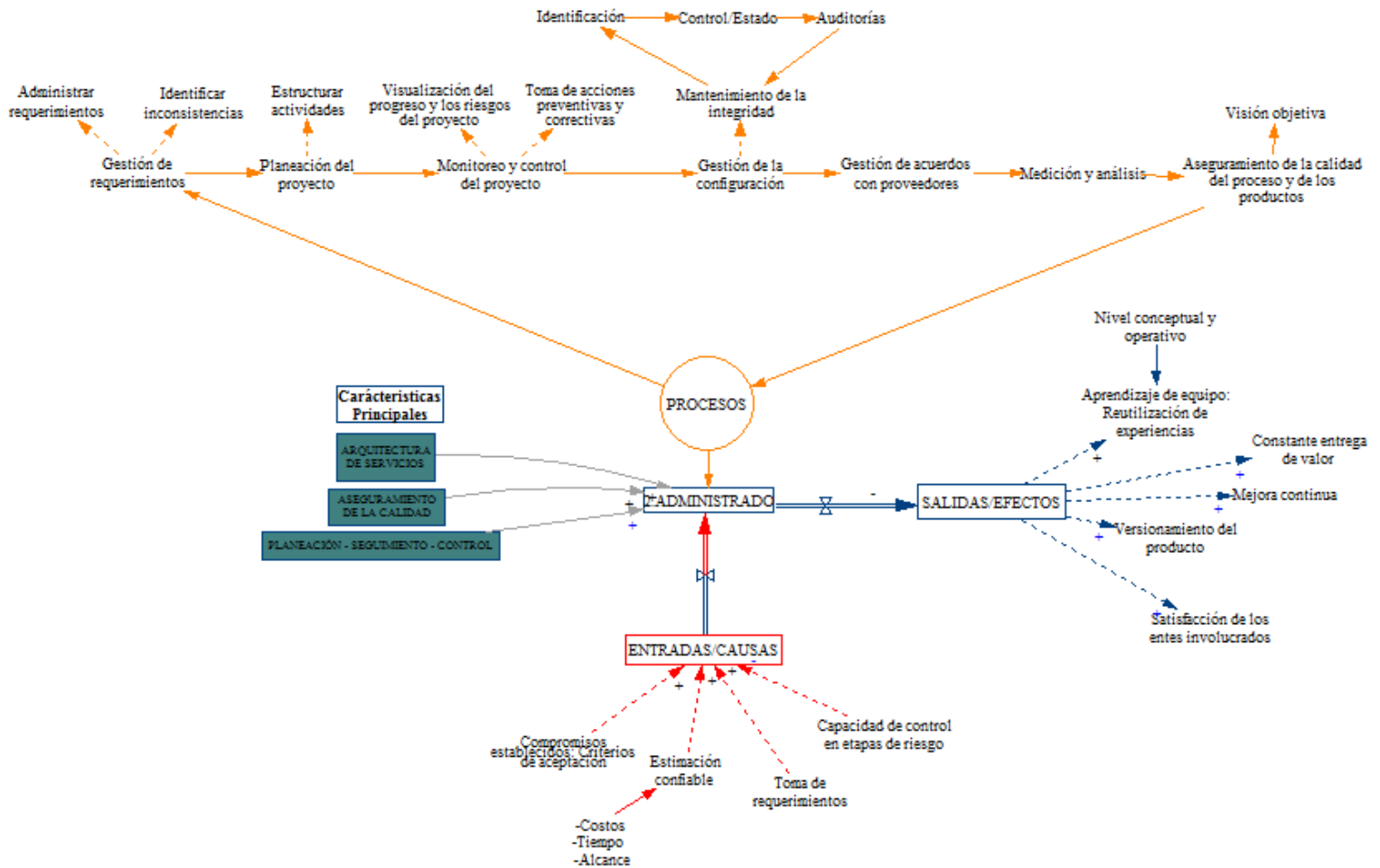


Ilustración 32: Nivel administrativo de modelos de calidad de Software. Fuente: Elaboración propia.

3. Nivel Definido: en este nivel los procesos están definidos y entendidos, gracias a que se establecen normas, procedimientos, métodos y herramientas. A diferencia del nivel 2 los procesos son descritos más detallada y rigurosamente y gestionados de manera más proactiva.

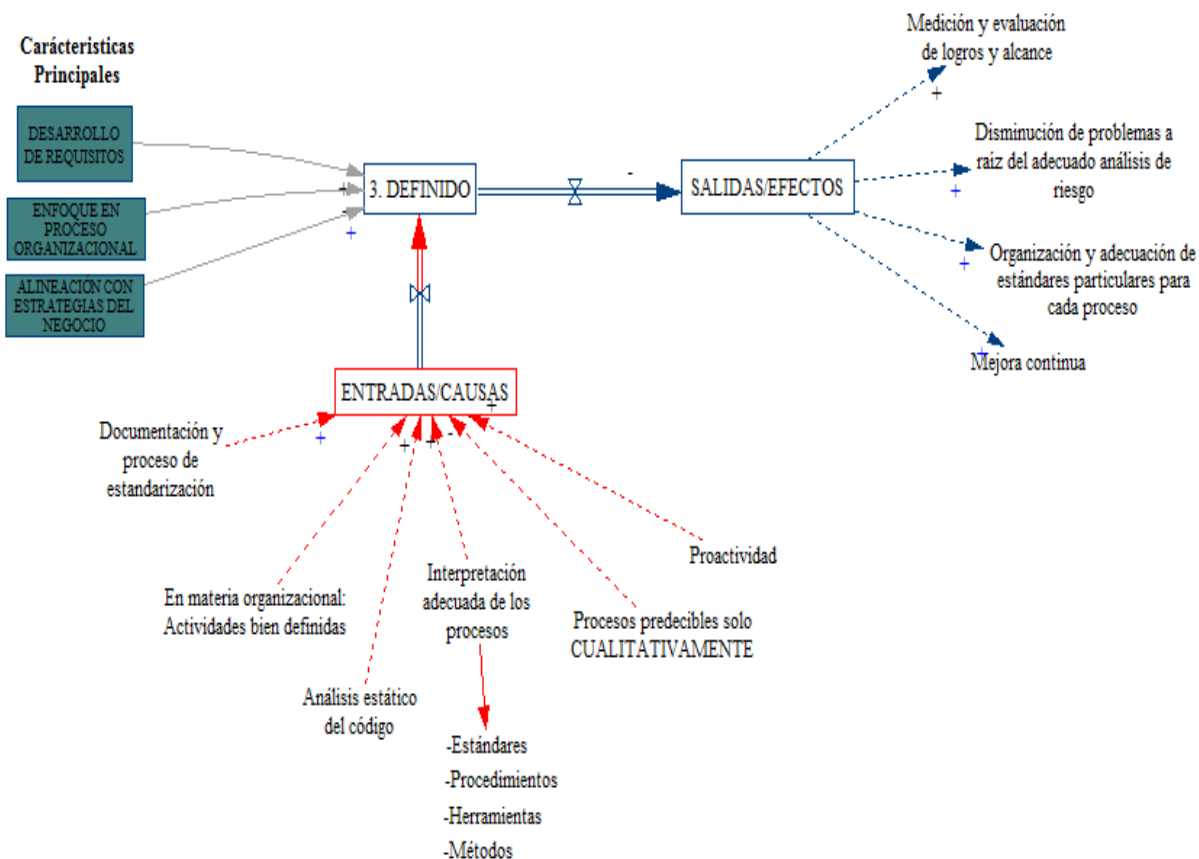


Ilustración 33: Nivel definido de modelos de calidad de Software. Fuente: Elaboración propia.

4. Nivel Administrado Cuantitativamente: en este nivel se consideran los objetivos cuantitativos que son aquellos que se basan en las necesidades del cliente y usuarios finales. La calidad se representa en estadísticas, esto permite visualizar riesgos futuros.

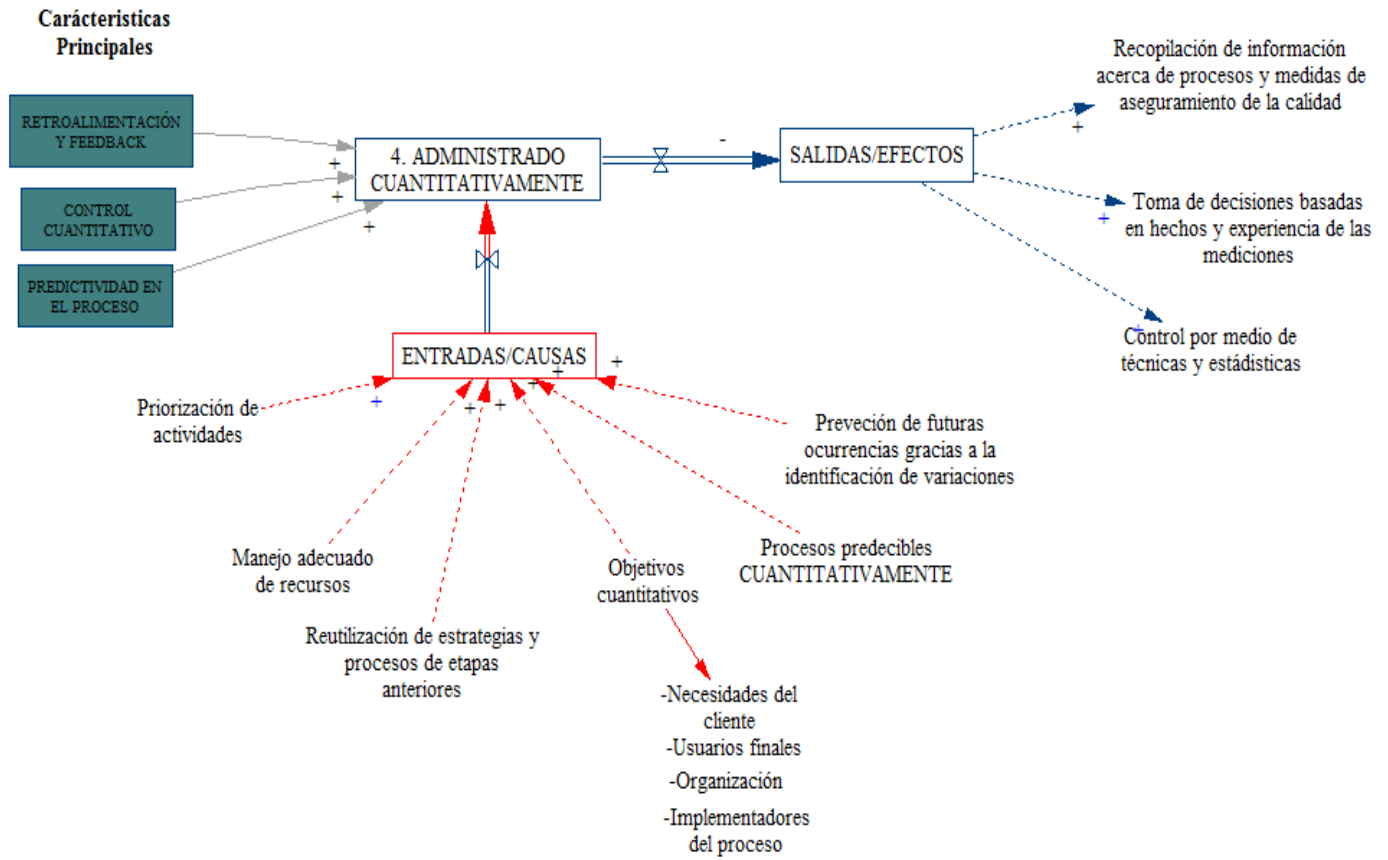


Ilustración 34: Nivel administrados cuantitativamente de modelos de calidad de Software.  
Fuente: Elaboración propia.

Analizando los niveles de madurez se entiende la necesidad de desarrollar software con calidad y constante, para que el cliente obtenga valor de manera evolutiva e incremental, y de esta manera el producto sea ajustado a las necesidades finales de los usuarios.

Por estas razones se trabajará bajo el marco de trabajo SCRUM, con el cual se asegura lo siguiente:

Se entienden los objetivos del proyecto y se asegura de que cada acción que se realice sea encaminada al cumplimiento de ellos.

- Desde el primer sprint se entrega software y esta entrega se realiza en un ambiente controlado.
- Sprint tras sprint el producto crece en cuanto a las nuevas características ofrecidas que habilitan a los usuarios finales a realizar nuevas tareas.
- Se habilitan las herramientas necesarias que permiten liberar software de forma automática.
- Desde el primer Sprint el equipo debe entregar software sin errores, software real, compuesto por unidades funcionales y no por mocks. Una nueva entrega deberá agregar funcionalidad al producto generado en la entrega anterior, y se deberá continuar esta dinámica hasta llegar al Release, en donde se tendrá un producto potencialmente instalable en un ambiente productivo.

### *Fase de Análisis*

#### *Identificación del problema o necesidad*

Bajo el marco de trabajo SCRUM la actividad por medio de la cual se da inicio a un proyecto es conocida como *Inception*, en esta, se incluyen todas las partes involucradas, desde stakeholders, clientes, usuarios finales, equipo de desarrollo y líderes del proyecto. El objetivo es definir la visión, objetivos, necesidades y tareas del proyecto. Tiene una duración de 1 a 2 días, dependiendo de lo robusto que puedan llegar a ser las necesidades del cliente, y del trabajo que se deba realizar para ayudarles a entender lo que en realidad necesitan como corporación.

Previa a esta actividad deben haberse realizado las siguientes tareas:

- Definir los compromisos (con fecha esperada de cierre y responsable con nombre y apellido) asociados a las actividades / acciones / recursos / personas de las cuales el cliente deberá disponer para iniciar la prestación de servicio. Dentro de estos compromisos, deben incluirse:

- La firma del contrato, la cual debe efectuarse idealmente antes de la fecha de inicio del proyecto.

- Las sesiones de capacitación que se realizan con la persona del cliente que ejercerá el rol de Product Owner. Para el inception, es ideal que el Product Owner ya esté capacitado, para que pueda aportar de manera activa en ese espacio; para el inicio del proyecto, es requisito indispensable que ya se haya realizado dicha capacitación.

- La sesión de sensibilización sobre agilismo que se realiza con las personas que harán parte del equipo extendido del cliente (involucrados / infraestructura, calidad), y que serán claves para que se puedan acordar los temas resultantes del inception.

- La fecha de realización del inception.

- La fecha de inicio del proyecto.

#### Definición del equipo

En un Equipo Scrum, se espera que intervengan tres roles: Product Owner, DevTeam y ScrumMaster

- Product Owner: Es la persona encargada del éxito del producto desde el punto de vista de los stakeholders. Sus principales responsabilidades son:

- Determinación de la visión del producto, hacia dónde va el equipo de desarrollo.

- Gestionar las diferentes expectativas de los stakeholders.

- Recolectar los requerimientos del proyecto.
- Determinar y conocer en detalle las diferentes características funcionales de alto y de bajo nivel.
- Generar y mantener el plan de entregas (release plan): fechas de entrega y contenidos de cada una.
- Maximizar la rentabilidad del producto.
- Determinar las prioridades de cada una de las características por sobre el resto.
- Cambiar las prioridades de las características según avanza el proyecto, acompañando así los cambios en el negocio.
- Aceptar/rechazar el producto construido durante el Sprint y proveer feedback valioso para su evolución.
- Participar de la revisión del Sprint junto a los miembros del Equipo de Desarrollo para obtener feedback de los stakeholders.
- DevTeam: El equipo de desarrollo está conformado por todos los individuos necesarios para la realización del producto. Es el único responsable por la ejecución y calidad del producto.

El equipo de desarrollo tiene principalmente tres responsabilidades que son tan fundamentales como indelegables. La primera es proveer las estimaciones de cuánto esfuerzo será necesario para cada una de las características que debe tener el producto. La segunda responsabilidad es, comprometerse al inicio de cada Sprint a construir un conjunto determinado de características en el tiempo que dura el mismo. Y finalmente, también, es responsable por la entrega del producto terminado inmediatamente finalice cada Sprint.

- ScrumMaster: El ScrumMaster es el Coach del equipo, y es quien lo ayuda a alcanzar su máximo nivel de productividad como sea posible.

#### Recolección de requisitos (historias de usuario)

Las historias de usuario serán el medio por el cual se evidencien las necesidades del producto, estas se componen de tres elementos:

- Card (Ficha): Toda historia de usuario, debe describirse en una ficha de papel pequeña. Si una Historia de Usuario no puede describirse en ese tamaño, esto quiere decir, de que se están traspasando las fronteras y comunicando demasiada información que debería compartirse frente a frente.
- Conversación: Toda historia de usuario debe mantener conversación con el Product Owner. Una comunicación cara a cara que intercambia no solo información sino también pensamientos, opiniones y sentimientos.
- Confirmación: Todo historial de usuario debe estar lo suficientemente explicada, con el fin que el equipo de desarrollo tenga conocimiento de qué es lo que debe construir y qué es lo que el Product Owner espera. Esto se conoce también como Criterios de Aceptación.

*Como (rol) Necesito (funcionalidad) Para (beneficio)<sup>24</sup>*

#### *Redacción de una historia de usuario*

Una historia real podría observarse como: “Como usuario necesito poder loguearme en la aplicación para acceder a mi perfil” o “Como administrador del sistema necesito poder crear, modificar y eliminar usuarios para la administración de los participantes de la aplicación”.

Existen diversas herramientas para la administración de historias de usuario, lo más importante es elegir una que le permita al equipo llevar un control de su tablero Kanban o Backlog. (Ver sección herramientas: *Herramientas para la gestión de proyectos*)



Anexo 2. Definiciones de calidad de varios autores.

1. “La composición de atributos que describen el grado de excelencia de un sistema de computadoras.” (Fisher, 1979)
2. “La medida en la que un producto de software posee un conjunto específico de atributos necesarios para cumplir con un propósito establecido. (Reifer, 1985)
3. La totalidad de funciones y características de un producto de software que le dan la habilidad de satisfacer las necesidades implícitas o explícitas. (Fenton, 1995)
4. Conformidad con los requerimientos del cliente. (Kan, 1995)
5. Se debe reconocer la naturaleza jerárquica de la calidad de software. Primero, un producto de software debe proveer las funciones del tipo y en el tiempo requerido por el usuario. Si no lo hace, nada más importa. Segundo, el producto debe funcionar. Si tiene tantos defectos de tal manera que no se comporta de una manera razonablemente consistente, el usuario no lo usará sin importar sus atributos. (Humphrey, 1995)
6. Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario (Pressman, 2010)

De todas las definiciones se debe tener cuenta que hay elementos esenciales como los requisitos a cumplir y el manejo de las expectativas del usuario.

En (BRAUDE, 2003) Un producto de calidad debe cumplir con lo siguiente:

- Satisface cada uno de los requerimientos establecidos con claridad.
- Verifica todas sus entradas; reacciona de manera predecible a entradas ilegales.
- Se ha inspeccionado, de manera íntegra por ingenieros que no son el autor.
- Se ha probado de modo exhaustivo, en varias formas independientes.
- Está bien documentado y soportado.
- Tiene una tasa de defectos confiable.

### Anexo 3. Metodologías ágiles

Esta metodología fue desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle.

Define un marco para lo relacionado con la gestión de proyectos, Está fundamentalmente indicada para algunos proyectos con un rápido cambio de requisitos. (Schwaber, Beedle, & Martin, 2001. Schwaber, R., Beedle, K., & Martin, M. (2001). Agile Software Development with SCRUM. Prentice Hall.)

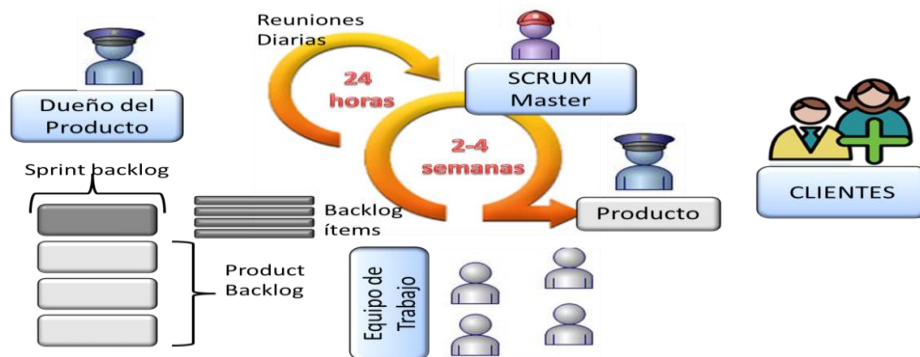


Ilustración 35. Proceso metodología SCRUM. Fuente: Diseño propio.

El desarrollo por la metodología SCRUM mostrada en la **¡Error! No se encuentra el origen de la referencia.**, se realiza mediante iteraciones que son llamadas sprints con una permanencia de aproximadamente dos a cuatro semanas, el resultado de cada uno de los sprint es un incremento que se expone al cliente. La otra particularidad son las reuniones que se tienen a lo largo del desarrollo del proyecto, entre ellas, sobresale la reunión diaria de 15 minutos del equipo de desarrollo para la coordinación e integración.

En cada sprint se trabaja sobre un juego de necesidades del cliente llamadas historias de usuarios y las funcionalidades que se desarrollan primero son las de mayor prioridad para el cliente.

## Crystal methodologies

Esta metodología se define como un conjunto de actividades para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo de trabajo (pues de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos o errores producidos. Estas han sido desarrolladas, por Alistair Cockburn.

El desarrollo de software se piensa como un juego cooperativo de invención y comunicación, restringido por cada uno de los recursos a utilizar. (Cockburn, 2001). Cockburn, A. (2001). Agile Software Development. Boston, MA, U.S.A.: Addison Wesley Professional.)

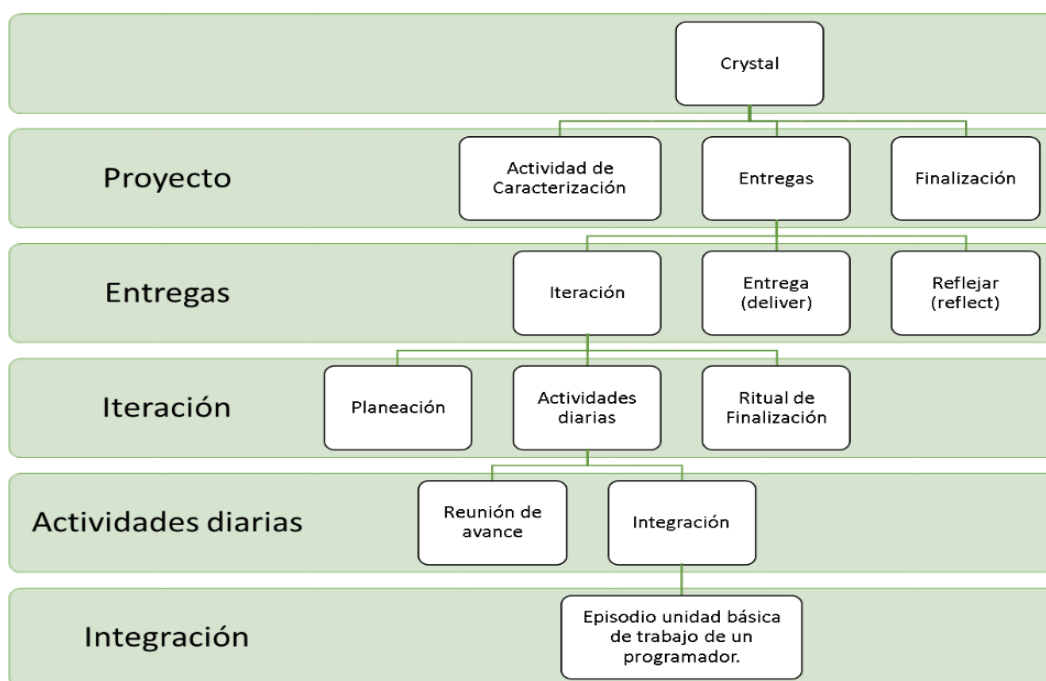


Ilustración 36: Proceso metodología Crystal. Fuente: Diseño propio.

La implementación de la metodología Crystal, mostrada en la Ilustración 36, define un conjunto de actividades a ejecutar las cuales proponen la ejecución de actividades diarias y una integración de las mismas, para posteriormente tener una retroalimentación más constante y al

mismo tiempo ir realizando una serie de mejoras en el desarrollo, con el fin de obtener un óptimo producto, que para este caso se habla de un Software completamente desarrollado.

### Dynamic Systems Development method (DSDM)

Este método, define el marco para realizar el desarrollo de un proceso de producción de software. Da origen en 1994 con el fin de crear una metodología RAD unificada. Las principales características son: es un proceso iterativo e incremental, el equipo de desarrollo y el usuario trabajan en conjunto. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas mencionadas son iterativas, además de existir realimentación en cada una de las fases. (Stapleton y Constable, 1997).

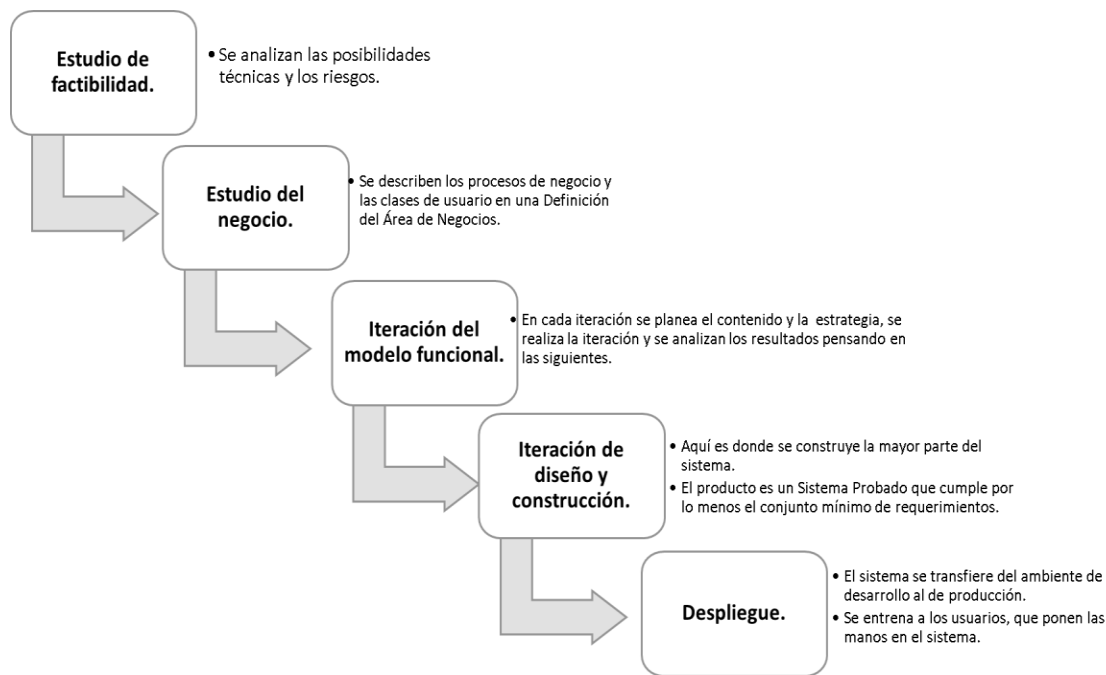


Ilustración 37: Dynamic Systems Development method (DSDM). Fuente: Diseño propio

Según lo plasmado en la Ilustración 37, se puede afirmar que es una de las metodologías más completas para lo que contempla el desarrollo de Software, pues por medio de esta en cada

una de sus etapas realiza un profundo análisis de diferentes aspectos para lo que es la planeación y posterior ejecución de proyectos de desarrollos que se puedan llevar a cabo.

### Adaptive Software Development (ASD)

Su promotor es Jim Highsmith, sus principales características son: iterativo, enfocado a los componentes software más que a las tareas, y tolerante a los cambios. El ciclo de vida que plantea tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas, se inicia el proyecto y se planifican las características del software; en la segunda, se desarrollan las características, y finalmente en la tercera, se revisa su calidad, y se entrega al cliente. La revisión de cada uno de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo. (Highsmith, 2000)

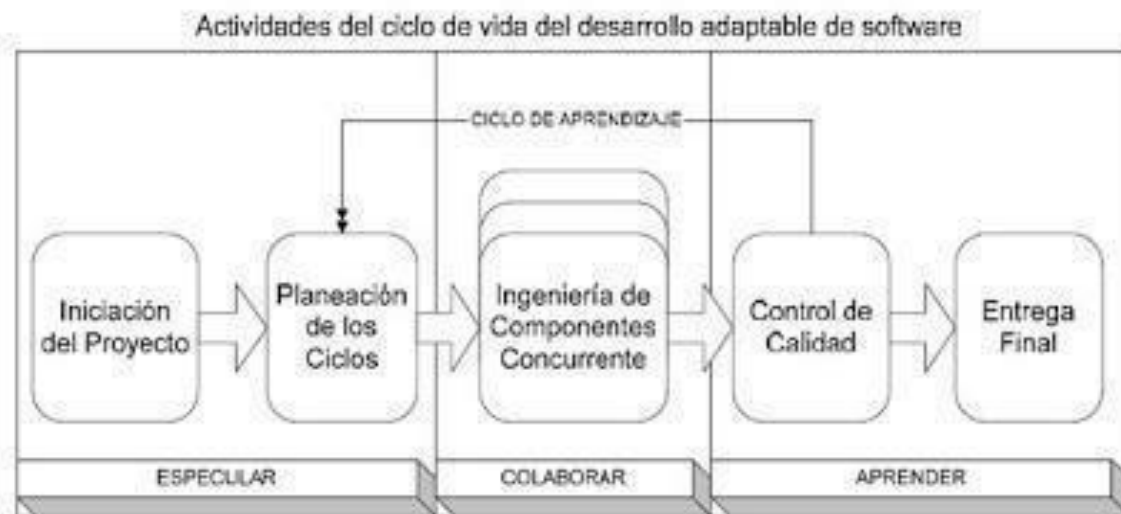


Ilustración 38: Adaptive Software Development (ASD). Fuente: Diseño propio.

La fase mostrada en la ilustración 10, tiene dentro de su proceso unas fases de especulación dentro de la que se tienen especulaciones que consisten en un inicio en el cual se define la misión del proyecto, la fijación de un maro temporal de este y la asignación de funcionalidades de cada una de las iteraciones determinadas; la colaboración que tiene como enfoque en el desarrollo del trabajo y la gestión del producto; y finalmente un aprendizaje basado

en la calidad del Software desarrollado en lo que se refiere a su funcionalidad y cumplimiento de las expectativas del cliente final.

### Feature- Driven Development (FDD)

Define un proceso, que consta de 5 pasos. Las iteraciones son cortas (con duración de hasta 2 semanas). Se enfoca en cada una de las fases de diseño e implementación del sistema, partiendo de una lista de características que debe tener el software. Sus impulsores son Jeff De Luca y Peter Coad. (Coad, Lefebvre, & De Luca, 1999)

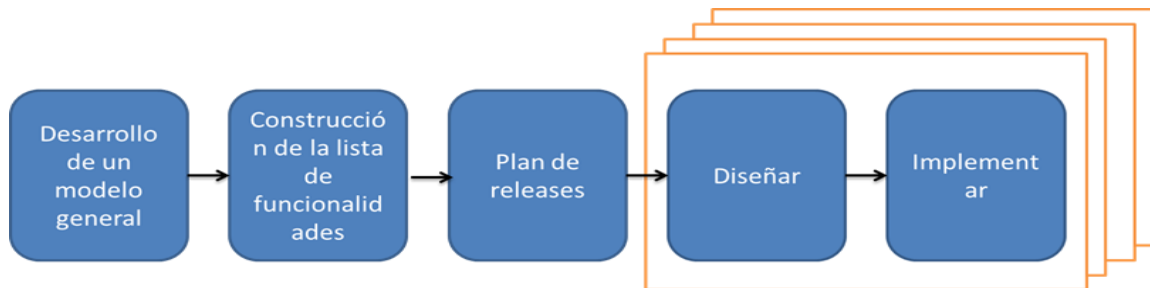


Ilustración 39: Feature- Driven Development (FDD). Fuente: Diseño propio.

La metodología descrita en la Ilustración 39, define el FDD como un modelo de desarrollo de períodos cortos que se enfoca en la fase de diseño y construcción.

En la primera fase, el modelo global de dominio es realizado por expertos del dominio y desarrolladores; el modelo de dominio consta de diferentes diagramas de clases, relaciones, métodos y atributos.

Los métodos no reflejan conveniencias de programación, sino por el contrario, estos muestran rasgos funcionales.

Sus características principales son las siguientes:

- Énfasis en las fases de diseño y construcción, más que en la toma de requisitos.
- Monitoreo constante del proyecto.
- Cierre de iteraciones cada dos semanas.

- Resultados periódicos y prácticos.

#### Anexo 4. Norma ISO/ IEC 12207.

La Norma ISO 12207 es la encargada de todo lo que compone el ciclo de vida del software, desde su conceptualización de ideas, hasta la retirada y consta de procesos para la adquisición y suministro de proyectos y servicios del software, estableciendo pautas para su control y mantenimiento.

En la siguiente ilustración se pueden apreciar algunas características de esta norma:



Ilustración 40: Características norma ISO/ IEC 12207. Fuente: Diseño propio.

Por otro lado, la norma contiene unos procesos de acuerdo, de soporte, técnicos y de gestión técnica. Cada uno de estos procesos se describe en términos de los siguiente:

- El título, se encarga de transmitir el alcance del proceso como un todo.
- El propósito, describe los objetivos de realizar el proceso.
- Los resultados, expresan lo esperado de la ejecución exitosa del proceso.
- Las actividades son conjuntos de tareas cohesivas de un proceso.
- Las tareas son repeticiones, recomendaciones o acciones permisibles destinadas a apoyar el logro de los resultados.



A continuación, se presenta una ilustración en la cual se puede apreciar más detalladamente los procesos que contiene esta norma y sus respectivos subprocesos:



Ilustración 41: Procesos de la norma ISO/ IEC 12207. Fuente: Diseño propio.

Resumiendo, la información plasmada en la Ilustración 41, se puede afirmar que esta norma es aplicable a la adquisición de sistemas, productos y servicios software, al suministro, desarrollo, operación y mantenimiento de productos software. Además, está realizada para ser utilizada tanto por personas adquirientes de sistemas, productos y servicios de software, como para desarrolladores, operadores, responsables de mantenimiento, administradores, responsables de aseguramiento de calidad y usuarios finales.

La Norma ISO 12207, está dirigida principalmente a la calidad en términos del ciclo de vida y puede ser utilizada por:

- Una organización de software: Con el objetivo de apoyar el establecimiento de un entorno de trabajo.
- Un proyecto: Con el fin de ayudar a la selección de una infraestructura adecuada, y emplear todos los elementos que conforman un ciclo de vida de software determinado.

- Un comprador o proveedor: Para contribuir en el desarrollo de un acuerdo sobre los procesos y actividades que se vayan a manejar.
- Asesores: Con el objetivo de realizar ciertas evaluaciones que puedan servir de apoyo para mejorar los procesos que se tienen dentro de la organización.

La norma ISO 12207, no está dirigida a productos de software preelaborados a menos de que formen parte de un producto entregable.

Como se mencionó anteriormente esta norma (ISO/IEC 12207) funciona basada en la ISO 9000 buscando un aseguramiento de la calidad interna y externa, por esto en el siguiente diagrama se puede apreciar algunos objetivos y finalidades de esta norma en cuanto al desarrollo y ciclo de vida del software.

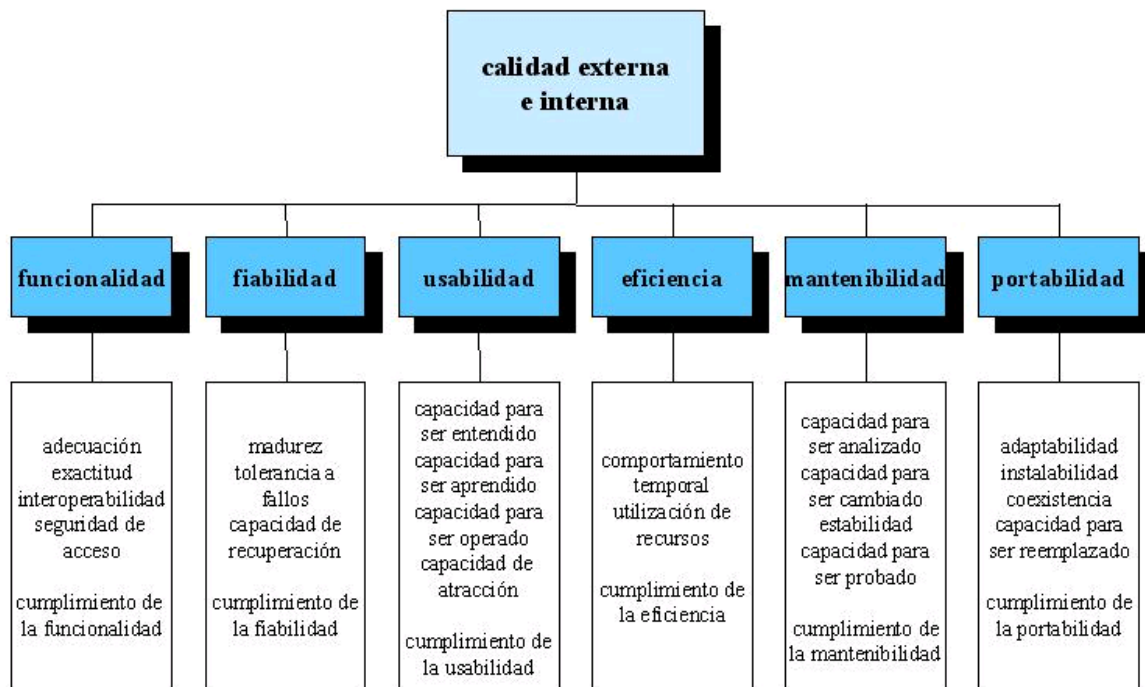


Ilustración 42: Factores de calidad que se aseguran bajo la ISO/ IEC 12207. Fuente: Diseño propio.

En la Ilustración 42, se presentan una serie de factores los cuales son indispensables que esta norma las cubra para un eficiente desempeño de los Software desarrollados en los cuales se describen desde la funcionalidad hasta la portabilidad al cliente final quien será el manipulador de los Software desarrollados dentro de laboratorio.

#### *Anexo 5: CMMI*

CMMI (SEI, 2010) Desarrollo consta de emplear una serie de buenas prácticas que manejan las actividades de desarrollo aplicadas a productos y servicios. Además, aborda las prácticas que cubren el ciclo de vida del producto desde la concepción hasta su entrega y el mantenimiento una vez este entra en operación.

CMMI-DEV contiene 22 áreas de proceso, de esas áreas de proceso, 16 son áreas de proceso base, 1 es de proceso compartida, y 5 son áreas de proceso específicas de desarrollo.

Todas las prácticas que contiene el modelo CMMI-DEV, se enfocan en cada una de las actividades de la organización desarrolladora. Cinco áreas de proceso se centran en las prácticas específicas del desarrollo: Tratando el desarrollo de requisitos, solución técnica, integración del producto, verificación y validación.

Los componentes del modelo se asocian en tres categorías –requeridos, esperados e informativos– que enseñan cómo interpretarlos.

Los componentes requeridos son CMMI que son importantes para alcanzar la mejora de procesos en un área de proceso dada. Los componentes requeridos en CMMI, son las metas y buenas prácticas específicas. La satisfacción de las metas se usa en las evaluaciones como base para establecer si un área de proceso ha sido satisfecha durante el proceso.

Los componentes esperados son componentes CMMI, que representan las actividades que son esenciales para alcanzar un componente CMMI requerido. Los componentes esperados,

orientan a quienes efectúan mejoras o realizan evaluaciones. Los componentes esperados en CMMI, son las prácticas específicas y genéricas. Antes de que las metas puedan considerarse satisfechas, sus prácticas tal y como se describen, o prácticas alternativas aceptables, deben estar presentes en todos los procesos planificados e implementados dentro de la organización.

Los componentes informativos son componentes CMMI, los cuales ayudan a todos los usuarios del modelo a comprender cada uno de los componentes CMMI requeridos y esperados. Estos componentes, pueden ser por ejemplo en un recuadro, explicaciones brevemente detalladas u otras informaciones útiles. Las sub prácticas, las notas, las referencias, los títulos de metas, los títulos de prácticas, las fuentes, los ejemplos de productos de trabajo y las elaboraciones de prácticas genéricas son algunos de los componentes informativos del modelo.

Las áreas de Proceso son: “Un conjunto de prácticas relacionadas en un área que, cuando se realizan de forma conjunta, satisface un conjunto de metas consideradas importantes para realizar mejoras en esa área”.