



UNIVERSIDAD NACIONAL DE INGENIERIA
DIRECCION DE ESTUDIOS DE POSGRADO
MAESTRIA INFORMÁTICA EMPRESARIAL

*Tesis para la obtención del grado de
Master
Informática Empresarial.*

***Propuesta para la aplicación de una metodología de desarrollo e
implementación de software en la División de Informática y
Sistemas de la Dirección General de Ingresos.***

Elaborado por:

✓ Lic. Gabriel Ernesto Alonso González.

Tutor de tesis:

✓ Msc. Juan Bosco Ordoñez.

Managua Nicaragua Septiembre, 2017

Dedicatoria

A Dios por permitirme disfrutar cada segundo de mi existencia, por ser mi fuerza y apoyo día a día. A mis padres por demostrarme que con esfuerzo y dedicación todo es posible. A mi esposa Jessenia Ordoñez por ser el motor que me impulsa a seguir adelante. Y a mis hijos Alexander y Allan por ser mi razón de ser.

Gabriel Alonso González.

Agradecimiento

A mi tutor Juan Bosco Ordoñez, que sin su ayuda y conocimientos no hubiese sido posible realizar este proyecto.

A mis padres, por haberme proporcionado la mejor educación y lecciones de vida.

En especial a la persona que durante un tiempo jugó el rol de padre, Roger Vásquez Berrios (q.e.p.d), por haberme enseñado que con esfuerzo, trabajo y paciencia todo se consigue, y que en esta vida nadie regala nada.

En especial a mi madre Nubia Quintero, por su apoyo incondicional y confiar en mis decisiones.

A mis compañeros de trabajo con los que he compartido grandes momentos durante el curso de la Maestría de Informática Empresarial.

A todos mis familiares por su apoyo moral.

En especial a la persona que durante mucho tiempo jugó el rol de segunda madre, Ileana Araya, por haberme aportado todo desde pequeño.

A todos aquellos que siguen estando cerca de mí y que le regalan a mi vida algo de ellos.

Resumen de la Tesis

Los cambios económicos junto con la dinámica competitiva que enfrentan actualmente las organizaciones a nivel global, obliga a sus directivos a afrontar grandes desafíos, oportunidades y amenazas en los distintos escenarios operativos. Por ende para cumplir exitosamente con las metas deben reorientar sus prácticas tradicionales y centrar su atención en los procesos estratégicos y críticos de la organización con el apoyo de herramientas tecnológicas de información y comunicación.

El objetivo principal de la presente tesis es una propuesta de una metodología dirigida a la gestión eficiente de proyectos de desarrollo e implementación de software en la División de Informática y Sistemas de la Dirección General de Ingresos.

Para lograr este objetivo se llevó a cabo un diagnóstico previo de los procesos de negocios involucrados para identificar las principales debilidades y oportunidades según el dominio del problema así como realizar un modelado de los procesos de negocios basado en la metodología BPM.

En base a estos resultados se logra identificar definir una metodología de desarrollo e implementación de software ágil que se adapte al entorno de trabajo de la Unidad de Sistemas Informáticos de la División de Informática y Sistemas.

Índice

INTRODUCCIÓN	1
ANTECEDENTES	2
PLANTEAMIENTO DE LA SITUACIÓN	3
OBJETIVOS	6
JUSTIFICACIÓN	7
MARCO TEÓRICO.....	9
I ETAPA PRELIMINAR: DIAGNÓSTICO DEL NEGOCIO	10
1.1 Conceptualización de Procesos.....	10
1.2 Gestión de los Procesos del Negocio.	11
1.3 Vinculación del Diagnóstico previo a la Automatización.	12
1.4 Modelado de Procesos de Negocios.	13
II ETAPA: LEVANTAMIENTO DE REQUERIMIENTOS	18
2.1 Ingeniería de Requisitos.....	20
III ETAPA. METODOLOGÍAS DE DESARROLLO E IMPLEMENTACIÓN DE SOFTWARE.	23
3.1 Clasificación de los modelos de ciclo de vida del software ^[5]	24
3.2 Comparativo entre metodologías de desarrollo e implementación de software.....	26
3.3 Metodologías Ágiles (Ventajas/Desventajas).	27
3.3 Factores de éxitos en la implementación de metodologías de SW ágiles....	28
CONCLUSIONES DEL MARCO TEÓRICO.....	29
Capítulo I DIAGNÓSTICO	30
1.1 DIAGNÓSTICO ORGANIZACIONAL.....	31
Etapa I: Inducción a la Reingeniería de Procesos.....	31
Etapa II: Definición del Negocio.	33
Etapa III. Modelado del Proceso de Negocio.	58

1.2 DEFINICIÓN DE LA METODOLOGIA AGIL DE DESARROLLO DE SOFTWARE.	73
CONCLUSIONES DEL CAPITULO I	78
Capítulo II PROPUESTA.....	79
2.1 RUTA DE IMPLEMENTACIÓN DEL MODELO DEL PROCESO AGIL SCRUM [1].	80
2.1.1 Etapa I: Diagnóstico Situacional.....	80
2.1.2 Etapa II: Evaluación Preliminar.	83
2.1.3 Etapa III. Definición final del modelo ágil SCRUM.....	84
2.2 BENEFICIOS.	88
2.3 CONCLUSIONES DEL CAPITULO II.	89
Conclusiones.....	90
Recomendaciones.....	91
Bibliografía	92
Anexos	95
Anexo 1. Estructura Organizativa de la Dirección General de Ingresos.	95
Anexo 2. Estructura Organizativa de la División de Informática y Sistemas.	96
Anexo 3. Figura 1. Ciclo de vida de los Procesos de Negocio.....	97
Anexo 4. Tabla 1. Clasificación de los diagramas estándar UML.	98
Anexo 5. Tabla 2. Clasificación de técnicas para el levantado de requerimientos.	99
Anexo 6. Figura 2. Estructura de la Ingeniería de Requisitos	101
Anexo 7. Tabla 3. Cuadro descriptivo de las fases del ciclo de vida del software.	102
Anexo 8. Tabla 4. Cuadro descriptivo de las etapas del ciclo de vida del software.	103
Anexo 9. Figura 3-4. Etapas de los Ciclos de Vida del Software Tradicional y Ligerito.....	104
Anexo 10 Figura 5. Mapa estratégico TIC de la División de Informática y sistemas.	105

Anexo 11 Figura 6. Ruta de Implementación del Modelo de Procesos Ágil SCRUM.	106
Anexo 13. Diagrama Causa y Efecto.....	114

INTRODUCCIÓN

La Dirección General de Ingresos, es una institución descentralizada con autonomía administrativa y financiera, cuyo objetivo es aplicar y hacer cumplir las Leyes, actos y disposiciones que establecen o regulan ingresos a favor del Estado nicaragüense, que están bajo la jurisdicción de la Administración Tributaria. También se encarga de administrar las leyes fiscales y recaudar los diferentes impuestos del país.

La institución está estructurada ^[1] por una dirección superior más tres divisiones: que incluye Informática y sistemas, siete direcciones que se subdividen en departamentos, además de las distintas administraciones de rentas distribuidas en gran parte del territorio nacional.

La División de Informática y Sistemas como una dependencia de la institución, tiene como objetivos: organizar, dirigir y controlar el diseño, desarrollo, implementación y mantenimiento de los sistemas informáticos, operativos, bases de datos, redes, telecomunicaciones y equipos para garantizar un servicio efectivo de apoyo tecnológico a todos los niveles de la institución en base a los requerimientos de cada una de las áreas administrativas. Su estructura ^[2] organizativa está formada por unidades tales como: Sistemas Tributarios (UST), Base de Datos (UDBA), Control de Calidad (UCC), Sistemas de Apoyo (USA) y Comunicaciones.

Como parte de su misión, la división de informática y sistemas debe dar respuesta inmediata según las diferentes soluciones informáticas que la institución requiera.

Actualmente existen graves deficiencias en la gestión de proyectos de software, careciendo de una metodología formal, que esté dirigida al desarrollo e implantación de sistemas de información en tiempos relativamente cortos. Esto trae como consecuencia el despliegue de aplicaciones informáticas deficientes resultando en la insatisfacción de los usuarios finales.

1/ Véase Anexo 1. Estructura Organizativa de la Dirección General de Ingresos.

2/ Véase Anexo 2. Estructura Organizativa de la División de Informática y Sistemas.

ANTECEDENTES

Inicialmente entre el año de 1990 al 2005, en la División de Informática y Sistemas de la Dirección General de Ingresos, solo se gestionaba dentro de su cadena de valor primaria un único sistema de información tributaria, que está basado en un lenguaje estructurado lineal y se desarrolló bajo la metodología tradicional SDLC (Software Development Life Cycle) por las características propias del sistema. Pero nunca se estableció formalmente una metodología de desarrollo e implementación de software estándar, solo se aplicaba como norma debido a las sugerencias que consultores externos sugerían utilizar como la metodología espiral ascendente con carácter de innovación.

A medida que las tecnologías de información y comunicación fueron tomando mayor impulso y relevancia en el país sumado a la necesidad surgida en la Dirección General de Ingresos por automatizar sus servicios de cara a los contribuyentes a fin de lograr una mejor atención a sus clientes, surgen nuevos proyectos de sistemas de información entre el año 2006 al 2011 que consistían en una Ventanilla Electrónica Tributaria (VET) y un sistema de registro único de contribuyentes (ERUC). Ambas aplicaciones se empezaron a desarrollar apoyándose de la metodología de desarrollo e implantación de Software de Proceso Relacional Unificado (RUP), pero no obstante no se adopta como metodología estándar para futuros proyectos de software. Posteriormente surgen nuevos sistemas de información críticos el cual manejan procesos sensibles como son Exoneraciones y Exenciones de impuestos en línea, Devoluciones Fiscales y el Sistema de Control de Facturación Electrónica. Aquí es donde se pierde de vista la formalidad de una metodología estricta de desarrollo e implementación de software que permita gestionar eficazmente los requerimientos funcionales que cambian constantemente en periodos relativamente cortos.

Actualmente la gerencia de la División de Informática y Sistemas en un común acuerdo con la dirección superior de la Dirección General de Ingresos, están impulsando la implementación de COBIT 5, que según sus fundamentos provee un marco de trabajo integral que tiene como propósito ayudar a las organizaciones a alcanzar sus objetivos para la gobernabilidad empresarial o institucional y la gestión de las TI corporativas.

PLANTEAMIENTO DE LA SITUACIÓN

Debido a los cambios de requisitos constantes sobre la marcha de los sistemas de información tributarios, que son un aspecto fundamental e inevitable en sus ciclos de vida, en la actualidad existe una fuerte debilidad en la elaboración de los requerimientos por parte de las distintas dependencias administrativas relacionadas al ciclo tributario de la institución quienes solicitan la automatización de un determinado proceso a través de una aplicación informática.

En el transcurso de levantado de requerimientos para nuevos sistemas de información tributaria o mejoras a los módulos, por lo general en la etapa inicial no existe el recurso humano capacitado que oriente a las áreas sustantivas, la secuencia y estructura que debe tener el documento. El mismo personal de las dependencias administrativas con pocos conocimientos técnicos informáticos, son quienes se encargan de diseñar a su percepción las interfaces y validaciones de determinado sistema.

Por parte de la Unidad de Sistemas Tributarios de la División de Informática y Sistemas, se reciben los requerimientos para su estricta ejecución en un lapso de tiempo corto que no superen más de dos meses, en donde el líder de la unidad es quien se encarga de asignar el trabajo al responsable del proyecto de software. Este a su vez con ayuda de su equipo analizan e interpretan el alcance que se quiere en el desarrollo del requerimiento. Si todo está claro y conforme lo requerido en la documentación, se procede a obtener las tareas a las cuales se les estima un período determinado no mayor a tres días hasta completar dos semanas o un mes según la complejidad de la solicitud.

El problema radica en la falta de una estricta metodología de desarrollo e implementación de software que permita gestionar de manera eficaz los proyectos que designa la dirección superior y la deficiente organización operativa que existe en la Oficina de Sistemas Informáticos Tributarios, en donde no están bien definidos los roles que ejercen los analistas de sistemas, y no hay flexibilidad de tiempo en los cambios repentinos de los requerimientos. A continuación se hace mención de las fallas agrupándolas en las más comunes:

❖ **Levantado de los requerimientos funcionales de los sistemas de información.**

- Personal no capacitado para el levantado de requerimientos funcionales.
- Cambios repentinos de los requerimientos.
- Acumulación excesiva de requerimientos como prioridad por parte de las áreas administrativas.
- Los planteamientos en el contenido de los requerimientos por parte de las áreas sustantivas administrativas no son claros.

❖ **Recursos (Humanos y Técnicos).**

- Falta de herramientas informáticas dentro de la oficina de sistemas tributaria que permitan la gestión de los proyectos de software en desarrollo.

❖ **Roles.**

- No están bien definidos los roles que ejercen los analistas de sistemas, sobre todo en área de control de calidad.

❖ **Comunicación con otras dependencias administrativas.**

- Se lleva una comunicación informal en todo el ciclo de vida de los proyectos de software.

❖ **Factor Tiempo.**

- Los tiempos de respuestas por parte de las oficinas administrativas ante la duda que se les presentan a los analistas para resolver un problema en los procesos de los sistemas de información tributaria, es muy tardía.
- Ajustados tiempos de ejecución dejando poca flexibilidad.

- Sobrecarga de trabajo para los analistas de la Unidad de Sistemas Tributarios. Además se atienden las incidencias de errores que se presentan en el día a día.
- Dificil percepción para el líder de la Unidad de Sistemas Tributarios y la dirección IT en los tiempos de los avances y condiciones de los requerimientos.
- Casi siempre se termina automatizando el desorden debido a la mala planificación que existe en las áreas normativas producto del mandato de la dirección superior por implementar un determinado sistema de información en el menor tiempo posible. Esta mala práctica incide directamente en la calidad del producto final, manifestándose en las incidencias de errores de procedimientos lógicos en las operaciones de las aplicaciones en tiempo de ejecución en producción.

➤ **Enunciado del Problema**

Falta de una metodología formal que permita la gestión efectiva sobre los proyectos de desarrollo de software en la oficina de sistemas tributarios de la división de informática y sistemas.

➤ **Objeto de Estudio**

División de Informática y Sistemas – Oficina de Sistemas Tributarios – Organización Operativa – Metodología de desarrollo e implantación de sistemas de información – Roles.

➤ **Campo de Acción**

Coordinación de las áreas sustantivas de la institución con la oficina de informática y sistemas tributarios para la planificación de los requerimientos funcionales y no funcionales.

OBJETIVOS

➤ **General:**

- Proponer una metodología dirigida a la gestión eficiente de proyectos de desarrollo e implementación de software en la División de Informática y Sistemas de la Dirección General de Ingresos.

➤ **Específicos:**

- Elaborar un diagnóstico de la situación actual mediante la evaluación de los procedimientos asociados a la gestión de proyectos de desarrollo e implementación de software en la división de informática y sistemas de la dirección general de ingresos.
- Presentar un análisis descriptivo y comparativo de metodologías de desarrollo e implementación de software cuyas fortalezas brinden oportunidades de mejora en la gestión actual de los proyectos de sistemas de información para la división de informática y sistemas de la dirección general de ingresos.
- Seleccionar una metodología específica que permita la eficiente gestión de los proyectos de desarrollo e implementación de software en la división de informática y sistemas de la dirección general de ingresos.
- Describir un procedimiento detallado que precise los procesos y recursos requeridos para lograr una adecuada implantación de la metodología orientada a la gestión eficiente de proyectos de desarrollo e implementación de software en la división de informática y sistemas de la dirección general de ingresos.

JUSTIFICACIÓN

Debido al enfoque estratégico actual, de la Dirección General de Ingresos, orientada a la modernización de sus servicios electrónicos con ayuda de las TIC, el que traen consigo, la simplificación y agilización de los diferentes procesos sustantivos y de apoyo que se ejecutan en la institución, ofreciendo una oportunidad importante para reducir los costos de procesamiento de datos y permitiendo el uso efectivo de la información, se ve la necesidad de reorientar el esquema funcional de la Oficina de Sistemas Tributarios de la División de Informática y Sistemas que esta de cara con las distintas áreas administrativas, contribuyentes y en algunos casos con otras instituciones públicas gubernamentales para garantizar la calidad y el funcionamiento óptimo de las aplicaciones.

Tomando como modelo las metodologías ágiles de desarrollo, que se según su teoría, se centran en el factor humano, en la colaboración y participación del cliente en el proceso de desarrollo y un incesante incremento de software de valor con iteraciones muy cortas de tiempo, se presenta una propuesta de estudio en el protocolo de tesis para seleccionar una metodología ágil que sea adaptable dentro de la organización de la oficina de sistemas tributarios para la gestión de proyectos con distintas particularidades y a su vez permita cumplir con los objetivos siguientes:

- **Alinear** los proyectos estratégicos con la ejecución en los programas, priorizando los desarrollos en función de la institución.
- Estructurar los programas teniendo en cuenta las cadenas de **valor** institucional que garanticen versiones periódicas incrementales de los sistemas y subsistemas involucrados.
- Organizar equipos desarrolladores pequeños, entregando funcionalidad cada 2 semanas, orientados a la **calidad** y a demostrar públicamente su trabajo para la satisfacción de los clientes internos y externos de la institución.

De igual modo se pretende demostrar que las metodologías ágiles son aplicables dentro del entorno de trabajo de la institución puesto que con el tiempo pueden ser escalables en toda la organización, permitiendo los siguientes beneficios:

- Satisfacción de los usuarios finales en el uso de los sistemas.
- Entrega temprana y continua de los requerimientos.
- Facilidad de responder a los cambios de diseño, contenido o funcionalidad de los sistemas tributarios.
- Autorregulación, organización y motivación del equipo de trabajo.
- Calidad de los sistemas de información.

MARCO TEÓRICO

Los cambios económicos junto con la dinámica competitiva que enfrentan actualmente las organizaciones a nivel global, obliga a sus directivos a afrontar grandes desafíos, oportunidades y amenazas en los distintos escenarios operativos. Por ende para cumplir exitosamente con las metas deben reorientar sus prácticas tradicionales y centrar su atención en los procesos estratégicos y críticos de la organización con el apoyo de herramientas tecnológicas de información y comunicación. A como indica (Peter Drucker (1993) citado por Sojo William. (2015, 09/22) que la evolución se ha visto impulsada por las nuevas tecnologías de Información, que han permitido incluso a los especialistas de la economía ver como sus hipótesis teóricas se acercan a la realidad al mejorar el proceso y la cantidad de información disponible. Esa información confirma como las empresas se ven en la necesidad imperiosa de reajustar sus estructuras organizacionales pasando de modelos tradicionales jerárquicos orientados al mando vertical, propios de la sociedad industrial, a estructuras que tienden cada vez más a la línea de mando horizontal por su eficiencia en el manejo de los procesos y la información, siendo la tecnología la herramienta fundamental y la información el insumo necesario propios de la sociedad del conocimiento. También (Benjamin Blunt (1992) citado por Sojo William. (2015, 09/22) afirma en sus estudios que el surgimiento de las nuevas configuraciones organizativas ha coincidido y han sido coadyuvadas por el desarrollo de las TIC, a las que se les suele atribuir una función fundamentalmente flexible y un carácter dinamizador de la organización, quedando en su investigación planteada la necesidad de contrastar en qué medida y en qué dirección las TIC afectan los rasgos del diseño organizativo y cómo se ajustan entre sí ambos elementos para dar respuesta a la cuestión crucial del cambio organizativo.

Sin embargo, esto depende de diferentes factores, tales como: tener un amplio conocimiento de los procesos de la organización y la planificación detallada de las necesidades de tecnología de la información que permitan una evaluación, selección e implementación de estos recursos en su estructura organizacional a fin de obtener resultados superiores.

Lo anterior sintetiza el contenido del presente Capítulo del marco teórico, definido como una investigación documental que recupera y trasciende reflexivamente el conocimiento acumulado descrito y contextualizado a través de distintos hallazgos de estudios, experiencias previas, referencias teóricas y perspectivas metodológicas relacionados al tema de estudio: “Propuesta para la aplicación de una metodología de desarrollo e implementación de software en la División de Informática y Sistemas de la Dirección General de Ingresos”.

I ETAPA PRELIMINAR: DIAGNÓSTICO DEL NEGOCIO

1.1 Conceptualización de Procesos.

Según, (Bravo 2004, p.27): “Proceso, es una unidad en sí que cumple un objetivo completo y que agrega valor para el cliente. Esta unidad es un sistema de creación de riqueza que inicia y termina transacciones con los clientes en un determinado período de tiempo”. Dicho de otra manera es una sucesión lógica de acciones diseñadas para producir en un tiempo determinado, un resultado específico para un cliente/usuario o negocio particular.

Estos procesos deben ser medidos, como indica: (Ortega, et al., 2010, p.15): “En todo proceso hay medidas o requisitos indispensables: Efectividad, Eficiencia, Adaptabilidad, Competitividad y Eficacia”. Los procesos se categorizan en primarios (críticos) y secundarios (apoyo). En el estudio (CIDEM, 2005, p. 9); refiere que: “Los procesos críticos de la empresa son aquéllos en los que se genera el valor principal

En conclusión: Podemos decir que un proceso es un conjunto de fases sucesivas de operaciones a la que se somete una cosa para obtener productividad eficiente y de calidad hacia los clientes.

para el cliente”.

1.2 Gestión de los Procesos del Negocio.

Definidos los términos anteriores, se conceptualiza según (Laudon 2004, p.51) el termino Procesos de Negocios ^[3] como: “flujos de trabajo concretos de material, información y conocimientos: conjuntos de actividades que permiten elaborar un producto servicio valioso. También tienen que ver con las maneras particulares de las organizaciones de coordinar el trabajo, la información y el conocimiento y con las formas que la administración elige para coordinar el trabajo.”. Los procesos de negocios agrupan recursos humanos, materiales y procedimientos de negocios con una finalidad.

Otra definición por (Hammer & Champy, 1993) indica que un proceso de negocio, “es una colección de actividades que tomando una o varias clases de entradas crean una salida que tiene valor para un cliente.”, de aquí denota la clave que representa para las organizaciones alcanzar una gestión efectiva sobre estos procesos.

De acuerdo a (Dupuy y Roland (1992:7) citado por Martínez, N. G. C.; Bravo, P. D.; Rodríguez, V. M. M.; Pineda, L. G): “gestión se define como el conjunto de procesos de recogida y utilización de la información que tiene por objeto supervisar y dominar la evolución de la organización en todos sus niveles”. La gestión involucra

En conclusión podemos decir que la gestión de procesos de negocios les ayuda a las organizaciones, lograr competitividad combinando la satisfacción de sus clientes con la eficiencia económica de sus actividades y la productividad.

cuatro funciones primordiales: Planificación, Organización, Dirección y Control.

1.3 Vinculación del Diagnóstico previo a la Automatización.

Se aborda como elemento de partida en este título la definición de lo que es una metodología, que según (Manuel E. Cortes Cortes y Miriam Iglesias León (2004)) “es la ciencia que nos enseña a dirigir determinado proceso de manera eficiente y eficaz para alcanzar los resultados deseados y tiene como objetivo darnos la estrategia a seguir en el proceso”. Es decir que parte del análisis previo para la poner en ejecución un determinado proceso en una organización.

Otra definición por (LASES, Franyutti Ma. Angélica (2006)), se refiere a una serie o conjunto de pasos ordenados y sistematizados que tienen como fin llegar a la obtención del conocimiento. Ambas definiciones coinciden en que la metodología permite el ordenamiento sistematizado en la gestión de los procesos y fija la ruta a seguir para la mejora continua de las organizaciones.

Finalmente en el caso de estudio (Implementación y Debugging. Cap 1) se define metodología para el desarrollo de software como los procesos a seguir sistemáticamente para idear, implementar y mantener un producto, desde que surge la necesidad hasta que se cumplen con los objetivos por el cual fue creado.

(Saroka, v1. (2002)) Indica que las metodologías de desarrollo de software proveen una cantidad de herramientas, técnicas y modelos que ayudan a representar determinado software en las diferentes etapas de su ciclo de vida. Posterior en el contenido 1.3 (etapa: Metodologías de desarrollo e implementación de software) se amplía más sobre la temática.

Las organizaciones antes de dar inicio a un proyecto de automatización de sus procesos deben tener clara la importancia que conlleva realizar un estudio previo al levantado de requerimientos funcionales.

Para ello (**Saroka, v1. (2002)**) en su estudio establece que se deben tener claras las metas y objetivos de la empresa ya que en la medida que disminuye el grado de abstracción, los enunciados acerca de meta y objetivo serán mejor definidos y más operativos. A su vez las mediciones de eficacia, que representan el valor de un sistema, regulan el grado en que se satisface el alcance del mismo.

En conclusión. Se puede deducir entonces que para poder realizar una automatización de procesos eficaz es importante elaborar un diagnóstico situacional previo de la organización para lograr cumplir con los objetivos de mejora continua.

1.4 Modelado de Procesos de Negocios.

Según (Darniame (1999) citado por Gómez, O. T., López, P. P. R., & Bacalla, J. S. (2014)), modelo de procesos es una representación del mundo real, que captura el estado actual de las actividades para guiar, reforzar o automatizar partes de la producción de los procesos.

Para (Vásquez, P. M., Giandini, R. S., & Bazán, P. (2010)) El modelado de procesos es una actividad importante en donde se representa la estructura y el comportamiento deseado de un sistema permitiendo así identificar con facilidad las interrelaciones existentes entre las actividades, analizar cada uno de los elementos y sus relaciones, identificar oportunidades de simplificación y reutilización o sacar a la luz problemas existentes dando oportunidad al inicio de acciones correctivas.

Se puede decir entonces que un modelo de procesos es una actividad mediante el cual un proceso es representado o descrito usando lenguajes apropiados que faciliten: la comunicación de la representación, su documentación y la comprensión del proceso.

El documento también resalta que las organizaciones deben tener claro que no basta con poseer un modelo de procesos elaborado sino también que es necesario contar con las herramientas adecuadas para definirlo, modificarlo y analizarlo.

Con el modelado de los procesos es posible según el estudio:

- Mediante la representación gráfica se pueden visualizar las interrelaciones existentes entre las distintas actividades que la conforman.
- Puntos de conexión con otros procesos o subprocesos.
- Roles o participantes encargados en las diferentes actividades.
- Permite identificar posibles problemas existentes así como oportunidades de mejora.

En conclusión. Con lo dicho anteriormente, el modelado de procesos conlleva a que las organizaciones puedan automatizar, integrar, monitorear y optimizar de forma continua los procesos que se administran.

1.4.1 Metodologías dirigidas a modelar procesos.

Para apoyar el modelado de procesos existen una serie de herramientas conocidos también como lenguajes notacionales de interpretación grafica que son muy útiles de emplear en el diseño de un modelo de proceso. En este epígrafe solo se harán mención de los más comunes como referencia para el capítulo II.

El modelo de negocios o e-Business se ha convertido en el factor más determinante en el mundo empresarial desde la revolución industrial (Burlton, 2001), dando lugar a la nueva economía (SeBW, 2008^a). Según el estudio realizado por Sectorial e-Business Wath a más de 5.500 empresas de diferentes sectores la adopción de modelos de negocios electrónicos como estrategia empresarial se ha incrementado de manera sustancial en los últimos años.

Actualmente implica una nueva forma de hacer negocios que afecta tanto la estructura como el comportamiento de las organizaciones. En el estudio destacan que entre 55% y el 70% de las empresas de todos los sectores que se entrevistaron consideraron que las TIC tienen una incidencia media o alta en sus negocios. Y subraya como motivos fundamentales de su implantación (Gilart, V. (2010)):

- Mejora del posicionamiento en el mercado con respecto a la competencia.
- La Mejora de la relación con los clientes y proveedores.
- Mejora en la automatización de los procesos.

Como definición, las metodologías de procesos de negocios tienen como único objetivo, modelar una vista completa del negocio, utilizando lenguajes que faciliten la comunicación entre los Stakeholders (partes interesadas) e ingenieros de requisitos.

A continuación se hace mención de las metodologías para la modelación de procesos más comunes:

➤ **RUP (Rational Unified Process) – UML (Unified Modeling Language):**

Para poder entrar en el contexto del UML (Unified Modeling Language en sus siglas en inglés) se debe conocer lo que es el RUP (Unified Modeling Language), un producto comercial desarrollado y comercializado por Rational Software, una compañía de IBM.

Es un proceso de ingeniería de software que según (Martínez, A., & Martínez, R. (2014)) proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades de una organización de desarrollo. También se le define como una metodología de software que intenta integrar todos los aspectos a tener en cuenta durante todo el ciclo de vida del software y proporciona herramientas para todos los pasos de desarrollo así como documentación en línea para los clientes.

Para llevar a cabo todas las tareas, existe el complemento estándar comúnmente conocido como UML (Lenguaje de modelado unificado).

El cual fue diseñado por OMG para visualizar, especificar, construir y documentar software orientado a objetos (Booch, G., Rumbaugh, J., Jacobson, I., Martínez, J. S., & Molina, J. J. G. (1999)).

Ventaja Principal del UML:

- Unifica distintas notaciones previas.

Desventajas

- Falta de integración con otras técnicas (p.ej. diseño de interfaces de usuario)
- UML es excesivamente complejo (y no está del todo libre de ambigüedades): “el 80% de los problemas puede modelarse usando alrededor del 20% de UML”.

UML contiene una serie de estándar de diagramas ^[4] para representar gráficamente un sistema desde diferentes puntos de vistas, son 9 en total.

➤ **BPM (Business Process Management) - BPMN (Business Process Modeling Notation)**

La Globalización en el mercado, la apertura de nuevas economías, los tratados de libre comercio sumados al entorno de competencia, exigen a las organizaciones a ser capaces de enfrentar su futuro con parámetros de eficiencia y eficacia. Por lo tanto soluciones tecnológicas como el BPM (Business process modeling notation por sus siglas en inglés) hacen posible el apoyo de estas condiciones, convirtiendo la gestión de procesos de negocios en una técnica estratégica, que permite generar y controlar cambios de forma “ágil”, oportuna, confiable y de calidad con miras al logro de los objetivos estratégicos establecidos por las empresas.

Como definición del Business Process Management (BPM) (Díaz Piraquive, F. N. (2008)) lo define como una metodología que orienta los esfuerzos para la optimización de los procesos de la empresa, en busca de mejorar la eficiencia y la eficacia por medio de la gestión sistemática de los mismos.

También Díaz hace énfasis a la filosofía del BMP como un sistema completo de información y comunicación, a través de un marco documental que permite publicar, almacenar, crear, modificar y gestionar procesos, así como acceder a ellos en cualquier momento y lugar.

Otra definición por (Garimella, (2008), citado por Díaz Piraquive, F. N. (2008)), un BPM es un conjunto de métodos, herramientas y tecnologías utilizados para diseñar, representar, analizar y controlar procesos de negocios operacionales permitiendo la colaboración entre personas de negocios y tecnólogos a fomentar procesos efectivos, ágiles y transparentes. Abarca personas, sistemas, funciones, negocios, clientes, proveedores y socios.

De acuerdo con (Laurentiis, (2008), citado por Díaz Piraquive, F. N. (2008)), es una estructura cohesionada coordinada adecuadamente para la acción.

En conclusión. Se puede decir que el enfoque de las tecnologías BPM es el análisis de la administración de los procesos de una empresa, desde que comienzan hasta que terminan; es decir, es la convergencia de plataformas de gestión, tecnologías y aplicativos de colaboración y gestión, y de metodologías de gestión empresarial existentes en la organización, que tiene como objetivo mejorar la productividad y la eficacia de la organización a través de la optimización de sus procesos de negocio.

4/ Véase Anexo 4. Tabla 1. Clasificación de los diagramas estándar UML.

II ETAPA: LEVANTAMIENTO DE REQUERIMIENTOS

La complejidad de los proyectos de software, así como el carácter multidisciplinar de la ingeniería de requisitos, obliga a los desarrolladores a seleccionar cuidadosamente las técnicas y prácticas de especificación de requerimientos durante el proceso de desarrollo de software. Uno de los mayores problemas de las empresas en el proceso de levantado de los requerimientos, es describir en un lenguaje natural una especificación, en la que el requerimiento tenga aplicación en el dominio, esta descripción se ve comúnmente afectado por la mala comprensión de la aplicación, el mayor trabajo para el ingeniero de software es explorar sobre ámbitos desconocidos y trabajar sobre un lenguaje en el que, tanto el usuario y el analista, mantienen en un grado de abstracción diferente. Dicho lo anterior el levantamiento de requerimientos es la actividad más crítica en el desarrollo de software por lo tanto en este epígrafe se dan a conocer las técnicas necesarias para el éxito de tener una buena documentación que permita definir los pasos a ejecutar.

De acuerdo al estudio de (Torres, M. & Aristizábal, N. (2009)) recalcan la importancia de selección de una buena práctica de levantamiento de requerimientos donde inician definiendo la “técnica” como una serie de pasos documentados que van de la mano con unas reglas para su uso y criterios para verificar su corrección, usualmente aplica a un proceso en el modelo de procesos. Algunas veces, dicha técnica incluye una notación y/o una herramienta asociada.

El levantamiento de requerimientos generalmente se realiza usando una metodología o varias técnicas. Muchas de esas metodologías y técnicas ya existen y tienen como objetivo asistir a los analistas en la tarea de entender las necesidades del cliente. A pesar de que algunos analistas consideran que la selección de una única técnica aplica para todas las situaciones, una metodología o técnica no puede ser suficiente para todas las condiciones del proyecto de software.

Es por ello que la selección de las técnicas ^[5] apropiadas para el levantamiento de requerimientos entre las técnicas disponibles, afecta enormemente el éxito o fracaso de todo el proceso de elaboración de requisitos de un sistema.

En conclusión. El levantamiento de requerimientos es un punto determinante para el desarrollo de cualquier proyecto de software. Puesto que se pone en evidencia que una buena selección de una técnica es fundamental en el proceso completo de ingeniería de software. A parte se hará acompañar de la metodología de desarrollo e implementación de software que al final de la tesis se propondrá.

5/ Véase Anexo 5. Tabla 2. Clasificación de técnicas para el levantado de requerimientos

2.1 Ingeniería de Requisitos

Cita Filosófica. *“La parte más difícil al construir un sistema de software es decidir que construir. Ninguna parte del trabajo invalida tanto al sistema resultante si ésta se hace mal. Nada es más difícil que corregir después”.*

Haciendo una reflexión a la cita filosófica de Fred Brooks, de que realmente dentro de todo proceso de ingeniería, lo ideal es tener claro desde un comienzo lo que se quiere para tener una base sólida en el diseño y la construcción de un sistema, porque de no ser así existe una alta probabilidad de que el resultado sea no satisfacer con el software las necesidades del cliente. Es aquí donde entra en juego la ingeniería de requisitos.

Dicho lo anterior (Pressman, R. S., & Troya, J. M. (1988)) nos da una estructura general de la ingeniería de requisitos a modo de preguntas básicas:

- **¿Qué es?**

Para comenzar cualquier trabajo técnico se debe de aplicar un conjunto de tareas de ingeniería a los requerimientos que llevarán a la comprensión de cuál será el efecto que tendrá el software en el negocio, que es lo quiere el cliente y como interactuarán los usuarios finales con el sistema.

- **¿Quién lo hace?**

Los ingenieros de software (comúnmente conocidos como ingenieros de sistemas o analistas) y los demás participantes del proyecto de software (Clientes, Gerentes y usuarios) intervienen en la ingeniería de requerimientos [6].

- **¿Por qué es importante?**

Porque no basta con diseñar y construir un sistema con estética que al final solo resuelve los problemas equivocados y no genera satisfacción a nadie. Por ello es importante comprender lo que el cliente desea antes de comenzar con el diseño y el desarrollo del producto.

- **¿Cuáles son los pasos?**

Se comienza con la *concepción*, tarea que define el alcance y la naturaleza del problema que se va a resolver. Seguido de la *indagación* que es la labor que ayuda a los participantes a definir lo que se quiere. Posterior la *elaboración*, donde se refinan y modifican los requerimientos básicos. Finalmente la *negociación*, donde los participantes definen el problema (prioridades de los que se quiere).

- **¿Cuál es el producto final?**

Esto se logra por medio de varios productos del trabajo: escenarios de uso, listas de funciones y de características, modelos de requerimientos o especificaciones.

- **¿Cómo me aseguro que lo hice bien?**

Revisando con los participantes los productos de trabajo de la ingeniería de requerimientos a fin de asegurar que lo que se aprendió es lo que ellos quieren decir en realidad. Tomando en cuenta que las cosas cambiarán aun después de que todas las partes estén de acuerdo y seguirán cambiando durante todo el proyecto.

Algo muy importante a lo que hace mención Pressman y que llama mucho la atención en el documento y que es muy cotidiano en las organizaciones que se dan a la tarea de automatizar sus procesos, es que se vuelve una batalla cuando se quieren obtener los requerimientos de los clientes, se tienen serios problemas al querer entender la información que se obtiene y es muy frecuente que se registren los requerimientos de manera desorganizada dedicando muy poco tiempo a verificar lo que se registra. Al final como ingenieros de sistemas o responsables de proyectos de software dejamos que el cambio desordenado nos controle en lugar de establecer mecanismos estandarizados para controlarlo a él.

Por otro lado (Méndez, G. (2008-2009)) nos da una definición sintetizada de lo que es la ingeniería de requisitos, como un conjunto de principios, métodos, técnicas y herramientas que permiten descubrir, documentar y mantener los requerimientos de sistemas basados en computadoras de forma sistemática y repetida.

(Zapata, Giraldo, & Mesa, 2010 citado por González, R, (2015) considera cuatro procesos fundamentales en la IR: educación, análisis, especificación y validación de requerimientos.

Dando fin con este apartado se presentan una serie de **casos de estudios** el cual hace reflexionar sobre la importancia de la ingeniería de requisitos durante el proceso de elaboración de requerimientos. Tal es el caso que presenta (González, R, (2015)) de un estudio conocido en la General Accounting Office (GAO) de EEUU 1979, el cual reveló que 47% del dinero empleado en proyectos de software se destinó a sistemas que no llegaron a utilizarse.

- Un 29% se empleó en proyectos que no llegaron a finalizar.
- Otro 19% se empleó en software que tuvo que ser profundamente modificado tras su entrega.
- Finalmente, tan sólo un 2% del dinero se empleó en proyectos software que sí cumplieron con sus requisitos, pero se trataba de proyectos más bien pequeños o de poca envergadura.

Otro caso analizado por Victor Basili ^[1] en 1981, es que él encontró cerca de 88 errores en una ERS de 400 páginas para el proyecto “A-7E Operational Flight Program”. La ERS había sido escrita por un grupo de expertos es especificaciones de requisitos.

En conclusión. Queda claro que la ingeniería de requisitos es la encargada de proporcionar a los desarrolladores un entendimiento de los requisitos y los límites del sistema permitiéndoles resultados satisfactorios.

1. Victor Basili. Es profesor emérito del departamento de ciencias de la computación en la Universidad de Maryland.

6/ Véase Anexo 6. Figura 2. Estructura de la Ingeniería de Requisitos

III ETAPA. METODOLOGIAS DE DESARROLLO E IMPLEMENTACIÓN DE SOFTWARE.

(Avison y FitzGerald (1995) citado por Gómez, O. T., López, P. P. R., & Bacalla, J. S. (2014)) nos da una definición de las metodologías de desarrollo muy clara, destacando sus principales componentes, fases, herramientas y técnicas. Definiéndola como “una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información”.

Otra definición que cita (Yelmo, C) es un marco de común de fases, actividades, métodos, herramientas y roles bien establecidos y conocidos que se aplican a todos los proyectos en una determinada organización.

En este apartado cabe destacar que no existe una metodología de desarrollo de software universal el cual sea adoptado por las organizaciones, cada una de las que existen se acoplan de acuerdo al entorno y el tipo de trabajo que desempeñan. *Yelmo* recalca la existencia de esquemas generales conocidos como modelos de ciclos de vida y recomienda que cada empresa que desarrolla software debe tener su modelo particular o un subconjunto de modelos en función de la tipología de proyectos.

La **ISO**^[1], International Organization for Standardization, en su norma **12207**, define el ciclo de vida de un software como un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando desde la definición hasta la finalización de su uso.

Desde un punto de vista general de acuerdo al análisis que se hace en el documento de (Implementación y debugging, Cap. 1) considera que el ciclo de vida del software tiene tres etapas claramente definidas y diferenciadas más dos complementarias a las cuales hace mención:

- Planificación ^[2].
- Implementación ^[2].
- Puesta en Producción ^[2].
- Inicio ^[2].
- Control en Producción ^[2].

Por otro lado también se hace mención a las etapas ^[8] de un modelo de ciclo de vida del software el cual establecen una serie de objetivos, tareas y actividades que lo caracterizan.

3.1 Clasificación de los modelos de ciclo de vida del software ^[5].

- **Ciclos de vida del software tradicionales** ^[3].

Según (**Yelmo C, 2014**) los ciclos de vida de software convencionales están más orientados a aquellos proyectos que sus requisitos no cambian con frecuencia y que por lo general se acoplan más a sistemas críticos (Defensa, Aeronáutica Civil, Espacial, Software enlatado) por lo tanto son estables.

De igual modo se caracterizan por ser:

- Procesos dirigidos por planes (planificación predictiva).
- Las fases han de concluir con la elaboración completa de todos los artefactos antes de que comience la siguiente.
- Contienen muchos artefactos y planificación a largo plazo.
- Complejidad, falta de flexibilidad y burocracia.

- **Ciclos de vida por procesos ágiles o ligeros** ^[3].

(Pressman, R. S., & Troya, J. M. (1988)) define a las metodologías ágiles como “una combinación filosófica de un conjunto de lineamientos de proceso, en el cual se pone énfasis en la satisfacción del cliente y la entrega rápida del software incremental con equipos pequeños y motivados para efectuar determinado proyecto de software así como la sencillez general en el desarrollo”. Dicho de otra manera las metodologías ágiles son una contraposición a las metodologías tradicionales, dado que aportan como novedad, nuevos métodos de trabajo que apuestan por una cantidad apropiada de procesos. Es decir que no se pierden en una cantidad excesiva de cuestiones burocráticas ni tampoco defienden la falta total del proceso. Buscan la relación entre Proceso/Esfuerzo.

Por tanto (Fowler (2001) citado por Cáceres, P., Marcos, E., & Kybele, G. (2001)) las caracteriza por ser:

- Adaptativas más que descriptivas.
- Están más orientadas al personal que a los procesos.
- Promueven un desarrollo sostenible.
- El software que funciona es la medida principal del progreso.

En conclusión. Podemos deducir entonces que las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo de software con el objetivo de conseguir un producto más eficiente y predecible. Por lo tanto hace mucho hincapié en la planificación de todo el trabajo a realizar y una vez que todo está detallado comienza el ciclo de desarrollo del software. En cambio los procesos ágiles son más flexibles ante cambios repentinos en la estructura de los requerimientos funcionales.

3.2 Comparativo entre metodologías de desarrollo e implementación de software.

A continuación se describe de modo general la comparación de las metodologías ágiles con respecto a las tradicionales. Estas diferencias suelen afectar no solo al proceso en sí, sino también al contexto del equipo junto con la organización (Véase Tabla 5).

Tabla 5: Comparativo de las metodologías de desarrollo e implantación de software.

Metodologías Ágiles ^[4]	Metodologías Tradicionales ^[4]
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles	Mas roles
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.
Fuente: Canós, J. H., Letelier, P., & Penadés, M. C. (2003).	

3.3 Metodologías Ágiles (Ventajas/Desventajas).

Según estudios, existen criterios, universalmente aceptados acerca del desarrollo de software. (Jacobson et al. (2000), Citado por Cáceres, P., Marcos, E., & Kybele, G. (2001)), afirma que el modelo de procesos más aceptado, es el iterativo e incremental (SCRUM, XP, KANBAN, CRYSTAL, ETC.) puesto que a diferencia de otros permiten la obtención de diversas versiones del producto antes de la entrega final del mismo, así como la validación y depuración progresiva lo que sin duda redundará en un software más satisfactorio para usuarios y clientes. Por otro lado (Conallen (2000), citado por Cáceres, P., Marcos, E., & Kybele, G. (2001)) indica que con este tipo de proceso es posible añadir o modificar requisitos que no han sido detectados con anterioridad.

Por tanto (Lowe and Hall (1999), Citado por Cáceres, P., Marcos, E., & Kybele, G. (2001)), afirma que los métodos clásicos no son adecuados para el desarrollo de aplicaciones Web, puesto que no contemplan determinadas características específicas para este tipo de aplicaciones. Finalmente otra afirmación que hace (Fowler (2001), Citado por Citado por Cáceres, P., Marcos, E., & Kybele, G. (2001)), las metodologías tradicionales generalmente imponen un proceso de desarrollo demasiado pesado y burocrático, lo que impide un desarrollo ágil y rápido para este tipo de aplicaciones.

Lo cual (Figuerola, R. G., Solís, C. J., & Cabrera, A. A. (2008)) describe una serie de ventajas que son predominantes en las metodologías ágiles y destaca las siguientes:

- Capacidad de respuesta a cambios de requisitos a lo largo del desarrollo.
- Entrega continua y en plazos breves de software funcional.
- Trabajo conjunto entre el cliente y el equipo de desarrollo.
- Importancia de la simplicidad, eliminado el trabajo innecesario.
- Atención continúa a la excelencia técnica y al buen diseño.
- Mejora continua de los procesos y el equipo de desarrollo.

También (Dolado, J., & RODRÍGUEZ, D. (2011)), concluye que no existen resultados definitivos sobre la superioridad de los métodos ágiles y tampoco se evidencian desventajas con respecto a los métodos clásicos en los aspectos críticos del desarrollo de software.

En conclusión. Podemos deducir entonces que las metodologías ágiles ofrecen una solución casi a la medida para una gran cantidad de proyectos de desarrollo de software que cambian constantemente, en particular para aquellos que se trabajan en entornos web y móvil. En este apartado para el trabajo de tesis nos centraremos en SCRUM el cual será abordado su ciclo de vida en el capítulo III del trabajo.

3.3 Factores de éxitos en la implementación de metodologías de SW ágiles.

(Dolado, J., & RODRÍGUEZ, D. (2011)), indica que los factores de éxitos con evidencias cuantitativas en la implementación de este tipo de proceso recae en la utilización de una estrategia correcta de entrega de los productos, el uso de una práctica adecuada de las técnicas de ingeniería del software ágiles, la integración de un grupo de trabajo de alta cualificación junto con una alta implicación del cliente y un proceso de gestión adecuado. Otro factor al que hace mención (Pressman, R, 2010) es la competencia individual (miembro del equipo), acoplada con la colaboración grupal del equipo.

En conclusión. Queda claro que la ingeniería de requisitos es la encargada de proporcionar a los desarrolladores un entendimiento de los requisitos y los límites del sistema permitiéndoles resultados satisfactorios.

-
1. Fuente: Implementación y debugging, Cap. 1. Ciclo de Vida del Software.
 2. 2/Véase Anexo 7. Tabla 3. Cuadro descriptivo de las fases del ciclo de vida del software.
 3. 3/Véase Anexo 8. Tabla 4. Cuadro descriptivo de las etapas del ciclo de vida del software.
 4. 4/Véase Anexo 9. Figura 3-4. Etapas de los Ciclos de Vida del Software Tradicional y Ligero.
 5. 5/Véase Anexo 12. Tabla 14. Clasificación de las Metodologías de Desarrollo de Software

CONCLUSIONES DEL MARCO TEÓRICO

A continuación se extraen las principales conclusiones de este capítulo:

- Se destaca la importancia de conocer a profundidad los procesos de negocios según el campo de acción delimitado, a partir de la ejecución de una fase de diagnóstico que permita entender lo que está sucediendo, según la situación problemática.
- Se define la aplicación de un enfoque (Business process reengineering, BPR) como marco de referencia para la elaboración del capítulo II en lo que refiere al diagnóstico situacional.
- Se define el modelado de procesos de negocios bajo el enfoque BPM con la utilización de BPMN, siendo necesario la designación posterior de una herramienta BPA.
- A partir del modelado de procesos de negocios se identificarán los problemas actuales de la organización.
- De acuerdo a los estudios analizados se pretende definir una metodología de desarrollo e implementación de software que permita la gestión de proyectos eficiente dentro de la unidad de sistemas tributarios.
- Tomando en cuenta los factores de éxito de implementación descritos en los estudios analizados se pretende definir una metodología ágil que sirva como un marco de referencia para el desarrollo eficiente de sistemas de información que sus requerimientos cambian constantemente.

Capítulo I DIAGNÓSTICO

En el capítulo se ampliará en los objetivos siguientes:

- Elaborar un diagnóstico de la situación actual mediante la evaluación de los procedimientos asociados a la gestión de proyectos de desarrollo e implementación de software en la división de informática y sistemas de la dirección general de ingresos.
- Seleccionar una metodología específica que permita la eficiente gestión de los proyectos de desarrollo e implementación de software en la división de informática y sistemas de la dirección general de ingresos.

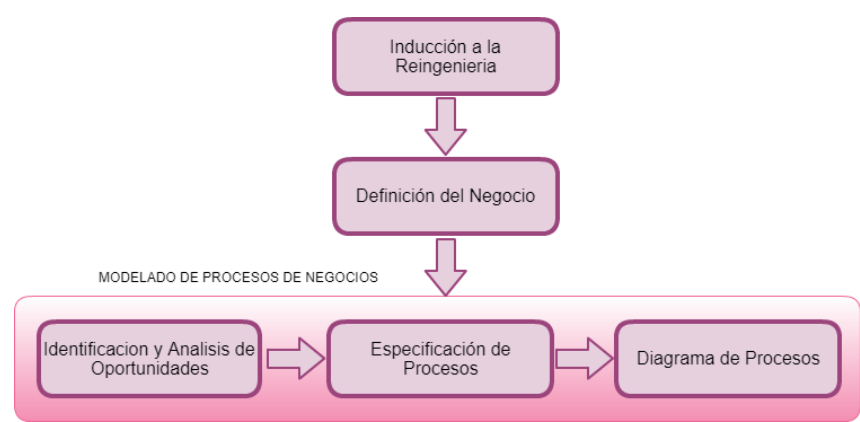
1.1 DIAGNÓSTICO ORGANIZACIONAL

Etapas I: Inducción a la Reingeniería de Procesos.

Según la bibliografía analizada (Laudon 2004, p.51) esta fase tiene como propósito involucrar y convencer a todo el personal relacionado sobre la necesidad de emprender cambios en los procesos actuales en la elaboración de requerimientos y la gestión de los proyectos de desarrollo de software, logrando una participación activa en todas las actividades. Además es clave la conjunción de factores tales como el compromiso asumido por la alta dirección superior, la comunicación entre el personal y la conformación acertada de los equipos de trabajo.

A continuación se definen los actores vinculados al dominio de problema, que representan a su vez las fuentes primarias de información (Tabla 5) en la aplicación de las herramientas de diagnóstico y levantado de requerimientos.

Para lograr estos objetivos se utilizará un enfoque de reingeniería de procesos de negocios (Business process reengineering, BPR), siguiendo el modelo propuesto:



Modelo propuesto bajo un enfoque BPM. Fuente: Elaboración propia.

Véase Tabla 5: Fuentes de Información.

DELIMITACIÓN	FUENTES DE INFORMACIÓN	POBLACIÓN	MUESTRA
División de Informática y Sistemas de la Dirección General de Ingresos.	Director IT	1	Población Total
	Sub Director IT	1	
	Jefes de Oficinas	3	
	Líderes de Unidades	7	
	Responsables de Módulos de Sistemas	5	
	Analistas de Sistemas desarrolladores	20	
	Analistas de Sistemas QA	10	
	Analistas de Sistemas DBA	10	

Fuente: Elaboración Propia

Tabla 6: Medios de Recopilación.

FUENTES DE INFORMACION (ACTORES)	POBLACION
Director IT	Entrevistas
Sub Director IT	
Jefes de Oficinas	
Líderes de Unidades	
Responsables de Módulos	
Analistas de Sistemas desarrolladores	
Analistas de Sistemas QA	
Analistas de Sistemas DBA	
Otras dependencias administrativas de la institución	Cuestionario de Preguntas
Fuente: Elaboración propia.	

Etapla II: Definición del Negocio.

En esta etapa del trabajo se tiene como objetivo realizar una caracterización del proceso de negocios en el ámbito de acción, en la coordinación de las áreas sustantivas de la institución con la Unidad de Sistemas Tributarios para la planificación de los requerimientos funcionales y no funcionales.

A). Pregunta de Análisis I

[Coordinación de las áreas sustantivas de la institución con la oficina de informática y sistemas tributarios para la planificación y ejecución de los requerimientos funcionales y no funcionales].

- **¿Cuál es la misión esencial o propósito de la organización en general?**

• **Dirección General de Ingresos.**

MISIÓN: Recaudar los tributos internos con equidad, transparencia y eficiencia, promoviendo la cultura Tributaria y cumpliendo con el Marco Legal, aportando al Gobierno recursos para el desarrollo económico y social del país.

VISIÓN: Ser una Administración Tributaria profesional, ágil y sencilla al servicio del pueblo Nicaragüense.

• **División de Informática y Sistemas de la Dirección General de Ingresos.**

MISIÓN: Asegurar el diseño, desarrollo, implantación, configuración y funcionamiento de los sistemas informáticos, operativos, bases de datos, redes, telecomunicaciones, equipos y prestación de servicios requeridos por las máxima autoridad y áreas requirentes, a fin de garantizar un servicio efectivo de apoyo informático a todos los niveles de la Dirección General de Ingresos.

VISIÓN: Ser una división altamente profesional, ágil y comprometida al cambio.

- **¿Cuál es su estrategia general para lograr ese propósito?**

- | | |
|-----------------|--|
| - Pro activismo | - Honestidad y Responsabilidad |
| - Lealtad | - Compromiso |
| - Servicio | - La Excelencia y Calidad |
| - Innovación | - Responsabilidad social institucional |

- ¿Cuáles son las estrategias a nivel tecnológico?

Las líneas estratégicas que se desarrollarán para el plan estratégico de tecnologías de la información 2017-2020 son:



Fuente: Elaboración Propia.

Plan Estratégico TI 2017 - 2020

ESTRATEGIA		DESCRIPCIÓN
E1	Desarrollar e integrar de manera modular el SIT.	Esta estrategia persigue la integración de los distintos módulos que existen en un solo sistema para que estos se relacionen de manera integral.
E2	Optimizar y asegurar los sistemas y servicios de la institución.	Con esta estrategia se desea asegurar que los sistemas y servicios tecnológicos se encuentren debidamente resguardados y controlados por una nueva área de seguridad de la información.

E3	Implementar un sistema de inteligencia de negocios.	Mediante la implementación de un sistema de inteligencia de negocio se desea incrementar la eficiencia, dar respuesta de manera más rápida a los problemas que puedan surgir, que se pueda tomar un decisión basada en información precisa, que se pueda dar un seguimiento al comportamiento de los contribuyentes y tener un mejor control sobre las áreas funcionales de la empresa
E4	Implementar un sistema de gestión de TIC's para asistir a los clientes internos de la institución.	La estrategia busca alcanzar un modelo de gestión de TI, mediante la implementación de estructuras, procesos y buenas prácticas, que aseguren la provisión de servicios tecnológicos de calidad.

- **¿Cuáles son los objetivos y lineamientos estratégicos TIC?**

Estrategia 1
Desarrollar e integrar de manera modular el SIT.
<u>Objetivo estratégico</u>
Integrar los distintos módulos a un nuevo sistema de gestión tributaria
<u>Iniciativas estratégicas</u>
✓ Construcción y transición de los módulos. ✓ Integración de los módulos existentes dentro de nuevos sistemas

Estrategia 2

Optimizar y asegurar los sistemas y servicios de la institución.

Objetivo estratégico

Optimizar y asegurar los sistemas y servicios tecnológicos que se brindan tanto a los contribuyentes como de uso interno, garantizando la calidad y seguridad de estos.

Iniciativas estratégicas

- ✓ Análisis del rendimiento de los servicios.
- ✓ Investigación y análisis de arquitecturas.
- ✓ Implementación de los cambios seleccionados.
- ✓ Ajustes a la estructura organizacional con la creación de un área de seguridad de la información.

Estrategia 3

Implementar un sistema de inteligencia de negocios.

Objetivo estratégico

Contar con un sistema de inteligencia de negocio que permita realizar análisis de los datos y de esta manera desarrollar planes de acción que contribuyan a mejorar la recaudación y fiscalización de los contribuyentes.

Iniciativas estratégicas

- ✓ Capacitación del personal desarrollador y usuarios finales.
- ✓ Definición de las necesidades de información.
- ✓ Implementación del Sistema de Inteligencia de Negocios.

Estrategia 4

Implementar un sistema de gestión de TIC's para asistir a los clientes internos de la institución.

<u>Objetivo estratégico</u>
Implementar un sistema de gestión de TIC's que optimice costos y apoye al cumplimiento de la gestión tributaria.
<u>Iniciativas estratégicas</u>
✓ Capacitación del personal. ✓ Definición de las necesidades de información. ✓ Implementación de controles de gestión.

- **Mapa estratégico TIC de la División de Informática y sistemas.**

Para el diseño del mapa estratégico ^[5], se consideran las iniciativas estratégicas resultado del análisis de la brecha entre el estado actual y el estado deseado en cuanto a competencias de personal, capacidades y escenarios de TI. En él se representa gráficamente las relaciones causa – efecto entre sus componentes estratégicos, para una mejor comprensión de su funcionamiento estratégico y operativo.

- **Cuadro de Mando Integral.**

PERSPECTIVA	OBJETIVOS ESTRATEGICOS	INDICADORES	METAS 2020	INICIATIVAS
Contribución a la institución	Mejorar los sistemas que contribuyen a la recaudación y fiscalización de los contribuyentes.	Nivel de satisfacción del usuario de la institución con la calidad y disponibilidad de la información de gestión.	70-80%	Realizar todas las medidas necesarias para el óptimo funcionamiento de los sistemas.
	Integrar los distintos módulos a un nuevo sistema de gestión	Nivel de integración de los módulos.	50-70%	Integración de los módulos existentes dentro de nuevos

	tributaria			sistemas.
Orientación al usuario	Optimizar y asegurar los sistemas y servicios de la institución.	Nivel de integración de la nueva área de seguridad de la información	80-90%	Ajustes a la estructura organizacional con la creación de un área de seguridad de la información.
Excelencia Operativa	Contar con un sistema de inteligencia de negocio que permita realizar análisis de los datos y de esta manera desarrollar planes de acción que contribuyan a mejorar la recaudación y fiscalización de los contribuyentes.	Nivel de satisfacción de los usuarios al momento de realizar el análisis de los datos de los sistemas tributarios.	80-90%	Definición de las necesidades de información
Orientación al futuro	Implementar un sistema de gestión de TIC's que optimice costos y apoye al cumplimiento de la gestión tributaria.	Nivel de avance respecto al acoplamiento del sistema de gestión de TIC's	70-80%	Implementación de controles de gestión.
Fuente: Elaboración Propia				

5/ Véase Anexo 10. Mapa estratégico TIC de la División de Informática y Sistemas

- **Portafolio de proyectos de la división de informática y sistemas**

PILARES	ESTRATEGIAS	OBJETIVOS	PROGRAMAS
Gestión Institucional	Desarrollar e integrar de manera modular el SIT	Integrar los distintos módulos a un nuevo sistema de gestión tributaria	Integración de sistemas
	Optimizar y asegurar los sistemas y servicios de la institución	Optimizar y asegurar los sistemas y servicios de la institución.	Implementación de área de Seguridad de la Información
	Implementar un sistema de inteligencia de negocios	Contar con un sistema de inteligencia de negocio que permita realizar análisis de los datos.	Bussiness Intelligence
	Implementar un sistema de gestión de TIC's	Implementar un sistema de gestión de TIC's que optimice costos y apoye al cumplimiento de la gestión tributaria.	Gobierno y gestión de TI (COBIT 5)

- **¿Cuáles son los servicios que ofrece la Organización?**

- **Dirección General de Ingresos**

- Inscripción de Contribuyentes en el registro único.
- Declaración y pagos de Impuestos.
- Otros pagos.
- Procesos de Fiscalización.
- Interposición de Recursos.
- Exenciones y Exoneraciones.
- Reembolsos o Devoluciones.
- Compensaciones.
- Categorización de regímenes simplificados.

- Cierre de obligaciones fiscales.
- Avalúos Catastrales.
- Asesorías tributarias por medio de capacitaciones.
- Solicitudes en línea 24/7.

- **División de Informática y Sistemas**

Véase Tabla 7: Servicios de la División de Informática y Sistemas

Clientes Internos		Clientes Externos
HARDWARE Y SISTEMAS OPERATIVOS	- Mantenimiento de equipos de cómputos. - Mantenimiento de Servidores. - Mantenimiento a las Redes. - Respaldo de Discos Duros - Actualización de Sistemas Operativos. - Actualización de Antivirus. - Actualización de Licencias.	Ventanilla Electrónica Tributaria 24/7: - Módulo de Declaración - Módulo de Solvencia Fiscal. - Módulo de Inscripción en línea. - Módulo de Devoluciones. - Módulo de Exoneraciones - Módulo de facturación e impresos. - Módulo de Avalúos catastrales.
COMUNICACIONES	- Monitoreo de redes. - Monitoreo en los enlaces de comunicación. - Monitoreo del ancho de banda.	
SISTEMAS INFORMATICOS	- Desarrollo de aplicaciones informáticas web/móvil tributarios y de RRHH.	

- **¿Cuáles son los canales para brindar los servicios?**

Véase Tabla 8: Canales para brindar los servicios	
Dirección General de Ingresos	División de Informática y Sistemas
Canales Directos:	
- Centro de asistencia al contribuyente (CAT)	- Mesa de Ayuda (GASI)
- Centro de recepción de causas	- Control de Calidad
- Administraciones de Rentas	- Coordinador de Sistemas
Canales Indirectos:	
- Ventanilla Electrónica Tributaria en línea	- Sistema de Bitácoras y Requerimientos

- **¿Qué procesos utiliza la organización para brindar los servicios a sus clientes?**

- Gestión del Negocio.
- Gestión de Soporte.
- Gestión Operativa.
- Gestión de Seguimiento.

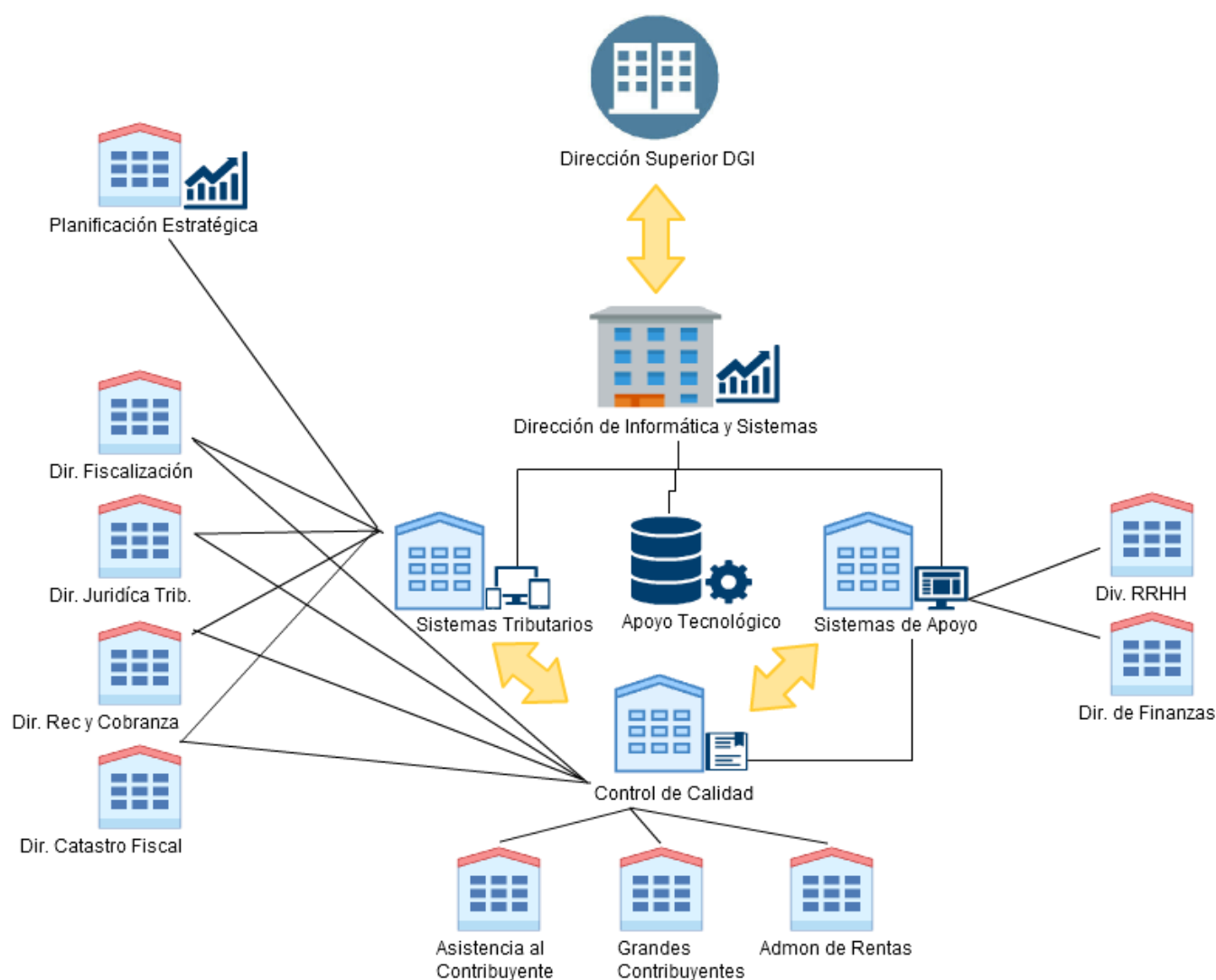
Los procesos involucrados se dividen de la siguiente forma:

Figura 6. Esquema sistémico de interacción de procesos y actores



Fuente: Elaboración Propia

Figura7. Esquema sistémico de interacción de dependencias con la División de Informática y Sistemas a través de la Unidad de Sistemas Tributarios.



Fuente: Elaboración Propia

Grupos de interés:

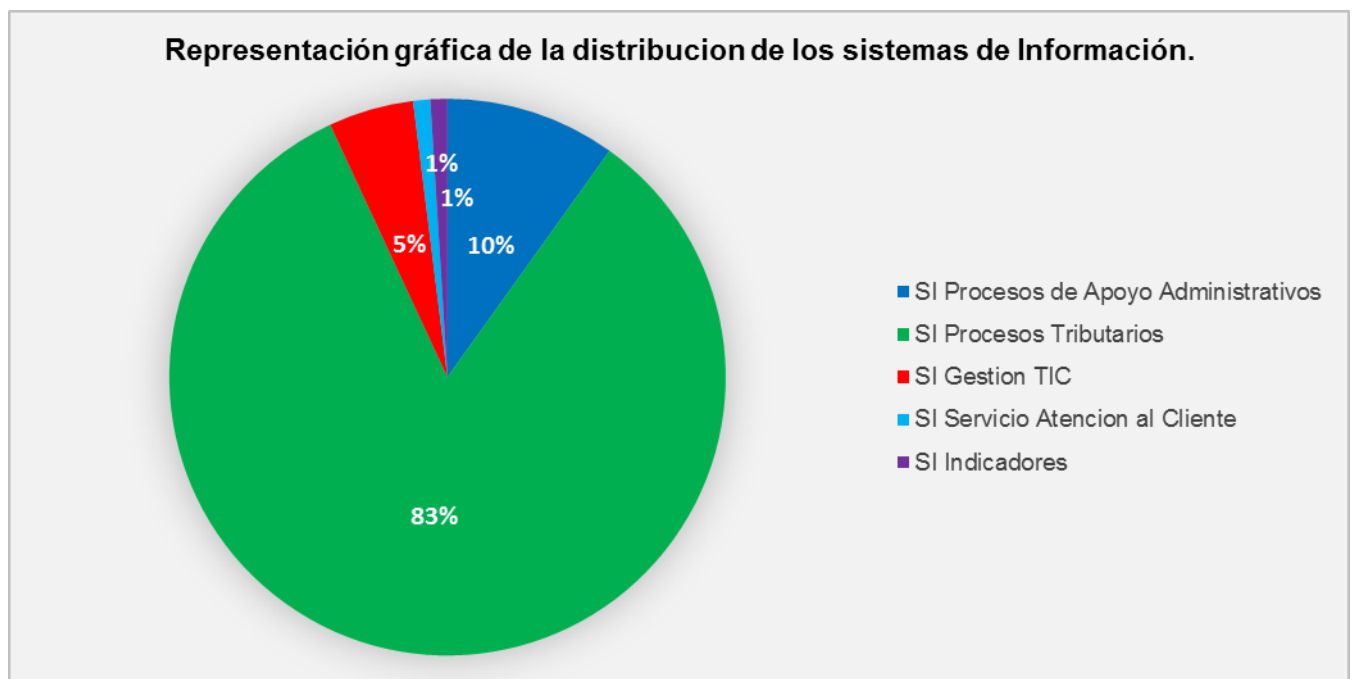
La división de informática y sistemas de la dirección general de ingresos a través de la unidad de sistemas tributarios interactúa con diversas entidades relacionadas con el proceso de tributación así como otros grupos de interés (público/privado).

- **¿Cómo están distribuidos los sistemas de información?**

Con la información obtenida del inventario de sistemas de información en producción y el catálogo de servicios recopilados en la División de Informática y sistemas de la Dirección General de Ingresos, se realizará el análisis de los mismos.

Para objeto de análisis se clasificaron a los sistemas de información de acuerdo al proceso al que brindan apoyo:

- Sistemas de Información para procesos tributarios.
- Sistemas de Información de apoyo al proceso administrativo
- Sistemas de Información de gestión TIC
- Sistemas de servicios para atención al cliente.
- Sistemas de información de planificación estratégica.



Grafica 1. Fuente: Elaboración Propia.

A como se observa en la gráfica 1, la División de Informática y Sistemas se encuentra en un proceso continuo de construcción y fortalecimiento de su Sistema Integral de Información^[1] en apoyo al plan estratégico de la Dirección General de Ingresos, el 83% de sus módulos apoya a los procesos tributarios sumándose a este porcentaje el 10% de módulos que ayudan a los procesos de apoyo administrativos, un 5% de módulos que soportan a los procesos de Gestión TIC más un 1% para apoyar el servicio de atención al cliente y otro 1% para indicadores y KPIS.

Por tanto con base a la información existente en cuanto a los sistemas de información administrados por la División de Informática y Sistemas de la Dirección General de Ingresos, se logra determinar la siguiente oportunidad de mejora:

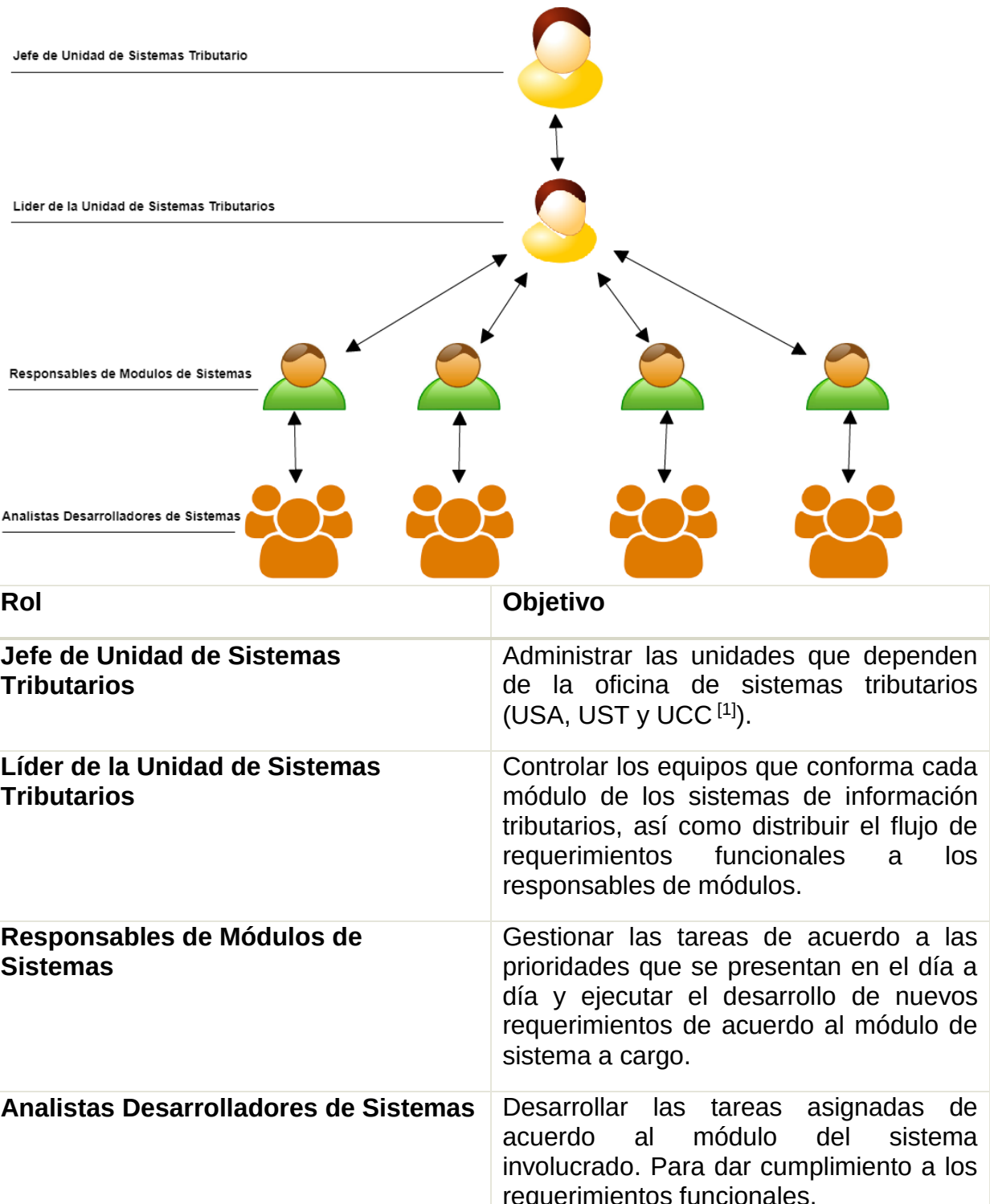
De acuerdo a las buenas prácticas de TI (COBIT 5), dominio BAI02.05, facilitar la operación y el uso, se debe planificar, implementar y documentar todos los aspectos técnicos, operativos y de modo de uso de tal manera que todos los involucrados puedan ejercer sus responsabilidades. Esto implica la transferencia de conocimiento, soporte antes y después de la implantación, la definición y registro de métricas de satisfacción del cliente y usuarios finales.

Véase Tabla 8. Detalle de los sistemas de información de la Dirección General de Ingresos.

TIPO	Clasificación
Sistemas de Información para procesos tributarios	- Ventanilla Electrónica Tributaria. - Registro Único. - Exoneraciones - Devoluciones - Facturación e Imprentas - Certificados de Solvencia - Pagos por Bancos - Sistema Tributario

	- Sitio oficial DGI - Avalúos muebles e inmuebles - Generación de Cédulas RUC - Aplicaciones Móviles
Sistemas de Información de apoyo al proceso administrativo	- Sistema Financiero - Control de Recursos Humanos - Nomina
Sistemas de Información de gestión TIC	- NEXUS - Sistema de Bitácoras - Mesa de Ayuda GASI
Sistemas de servicios para atención al cliente.	- ATENEO
Sistemas de información de planificación estratégica	- Sistema de Indicadores

Figura 6. Esquema de roles de la Unidad de Sistemas Tributarios de la División de Informática y Sistemas de la Dirección General de Ingresos.



1/véase Anexo 2. Estructura Organizativa de la División de Informática y Sistemas.

- ¿Quiénes son clientes/proveedores internos?



Fuente: Elaboración Propia

Pregunta de Análisis II

- **Clientes Internos**

Para medir el nivel de satisfacción de las necesidades y expectativas clientes internos con respecto a sus proveedores internos se aplicó una breve encuesta a una muestra total de 17 personas, con los siguientes resultados:

Tabla 9. Perspectiva general servicio recibido					
Proveedor interno	Malo	Regular	Bueno	Muy bueno	Excelente
Unidad de Bases de Datos	64%	56%	23%	3%	1%
Unidad de Control de Calidad	47%	52%	34%	19%	8%
Soporte Técnico	72%	65%	47%	2%	0%
Dependencias Administrativas	30%	27%	16%	4%	12%

Según los resultados obtenidos existe una valoración negativa en el nivel de servicios que reciben las partes involucradas en el área de unidad de sistemas tributarios, teniendo una opinión desfavorable soporte técnico (72%), unidad de bases de datos (64%) y unidad de control de calidad (47%). Para profundizar en los factores de la opinión desfavorable se realizó un análisis que permita calificar el nivel de importancia y nivel de satisfacción de los servicios ofrecidos por los proveedores internos, valorando los siguientes elementos.

a) Orientación al Cliente	Relación Proveedor – Unidad de Sistemas Tributarios	1. Predisposición para atender necesidades-solicitudes de la unidad de sistemas tributarios.
		2. Colaboración para la resolución de problemas
		3. Receptividad ante aportaciones de nuevos cambios de paradigmas.
		4. Preocupación por conocer la opinión de los equipos de trabajo de la unidad de sistemas tributarios.
	Conocimiento necesidades de la unidad de sistemas tributarios	1. Funcionamiento de los canales de comunicación
		2. Capacidad para prever y anticipar las necesidades del grupo de trabajo de la unidad de sistemas tributarios.
		3. Sistemas de valoración continúa de los involucrados de la unidad de sistemas tributarios y la calidad de los sistemas de información.
	Actuar con la unidad de sistemas tributarios	1. Método de coordinación de actividades con los involucrados en la unidad de sistemas tributarios.
		2. Sistemas de establecimiento de prioridades (proveedor)
		3. Desarrollo de sistemas dirigidos al apoyo y gestión de tareas del responsable de módulo de sistemas en la unidad de sistemas tributarios.
b) Procesos	Orientación objetivos y estrategias de los procesos	1. Actividad orientada según los objetivos de la organización.
		2. Actividad orientada según los objetivos del de la unidad de sistemas tributarios.

	Calidad en la gestión de los procesos	1. Concreción objetivos a conseguir/resultados previstos
		2. Los Servicios priorizan las necesidades de los involucrados en la unidad de sistemas tributarios.
	Funcionamiento procesos	1. Las solicitudes se atienden en el plazo acordado
		2. Los servicios se acompañan de método en su uso
		3. El número de solicitudes resuelto es aceptable.

Tabla 10: Escala de valores para medir nivel de importancia y de satisfacción.

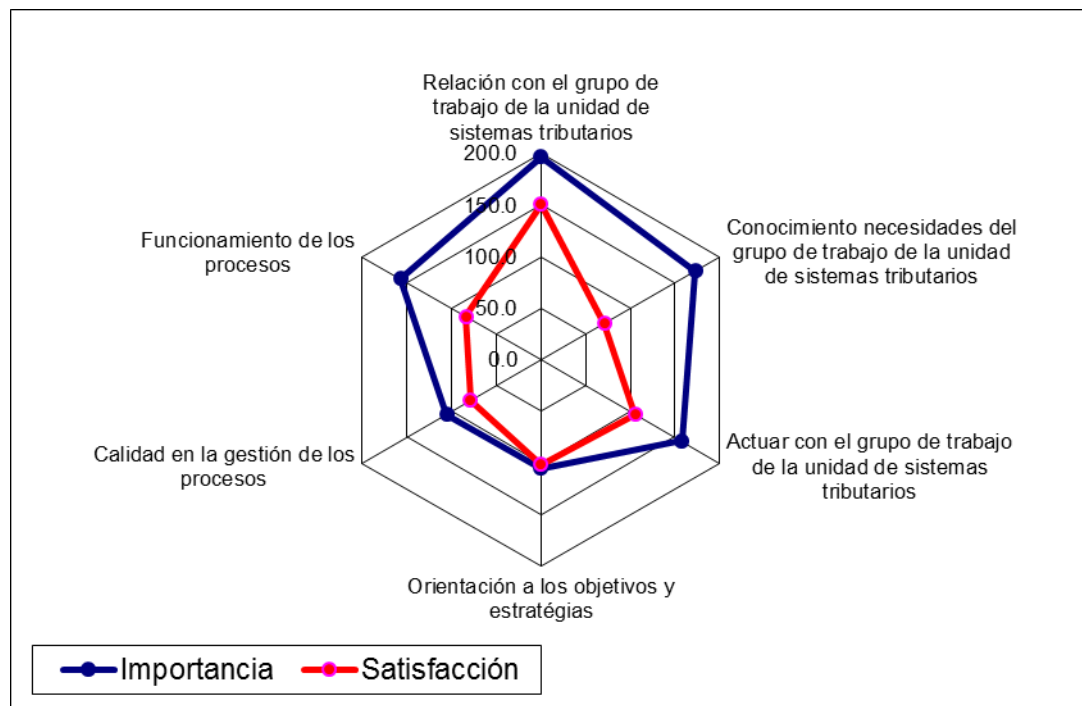
Grado de Importancia Concedida		Grado de Satisfacción Obtenida	
Muy Poca	1	Muy Mala	1
Poca	2	Mala	2
Mediana	3	Regular	3
Mucha	4	Buena	4
Muchísima	5	Muy Buena	5

Tabla 11. Brecha en la Satisfacción del Cliente Interno

Calificación del nivel de importancia y nivel de satisfacción de los servicios ofrecidos por los proveedores internos, valorando los siguientes elementos:		Importancia (Total)	Satisfacción (Total)	Brecha en la Satisfacción del Cliente Interno
Orientación al Cliente:	Relación con el grupo de trabajo de la unidad de sistemas tributarios	196.5	151.00	45.5
	Conocimiento necesidades del grupo de trabajo de la unidad de sistemas tributarios	172.5	71.0	101.50
	Actuar con el grupo de trabajo de la unidad de sistemas tributarios	157.5	105.5	52.00
Procesos:	Orientación a los objetivos y estrategias	105.00	101.5	3.50
	Calidad en la gestión de los procesos	105.5	79.00	26.50
	Funcionamiento de los procesos	157.00	84.00	73.00

De acuerdo a los resultados en la tabla anterior, nos indica que en la orientación hacia el cliente existe una brecha en el nivel de satisfacción del conocimiento de las necesidades del cliente interno y a cuanto a los procesos existe una brecha importante en aspectos relacionados a la gestión y funcionamientos de los procesos.

Grafica 2. Brecha en la satisfacción del cliente interno.



A continuación se identifican las causas de esta brecha en el nivel de satisfacción de los clientes internos respecto a la gestión con los proveedores internos.

A. Conocimiento procesado resultado de la aplicación de técnicas de recopilación de información

A partir de la aplicación y procesamiento de las técnicas de recopilación de información (Entrevistas y cuestionarios) en este inciso se pretende realizar una descripción de las funciones de los actores involucrados (clientes y proveedores internos) así como su papel en los procedimientos que conforman los procesos internos que interactúan en la gestión operativa de la división de informática y sistemas a través de la unidad de sistemas tributarios.

ACTOR: Unidad de Sistemas Tributarios		ROL: Cliente Interno
PROCESO	FUNCIÓN	FALLAS/DEBILIDADES
Gestión del proceso	Establecer los procesos de la unidad sistemas tributarios, en función de los términos requeridos identificados en el plan estratégico. Así como definir, planear, e implantar las actividades de mejora en los sistemas de información tributarios.	Deficiencia del control interno a nivel de la unidad de sistemas tributarios. Políticas y procedimientos cambiantes en relación con la autorización de modificaciones a programas y/o acceso a archivos de datos. Falta de políticas y estrategias para gestionar el conocimiento de la unidad.
Control y Gestión de Proyectos	Asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización.	No se aplican metodologías de desarrollo de software de ningún tipo. Herramientas de gestión de proyectos escasos o no acordes al ritmo de avance de los proyectos de software. Comunicación insuficiente con el equipo de trabajo y las partes interesadas. Falta de implicación efectiva del equipo directivo de las otras dependencias de la organización.
Administración de proyectos específicos	Establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en específico en tiempo y funcionalidad esperada.	No se está alineado a una metodología de desarrollo de software estándar que permita hacer frente a los cambios repentinos en los sistemas de información. Falta de compromisos de parte de las dependencias administrativas. Omisión de tareas de estimaciones necesarias para el cumplimiento de objetivos estratégicos. Añadir requisitos no solicitados por las dependencias administrativas.

Desarrollo y Mantenimiento de Software	Realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevos o modificados cumpliendo con los requerimientos especificados.	Insuficientes controles desde la dirección del proyecto. Los desarrolladores añaden características no solicitadas por las partes interesadas. Motivación inadecuada. Roces entre los desarrolladores y las partes interesadas (dependencias administrativas).
---	--	--

ACTOR: Unidad de Bases de Datos		ROL: Proveedor
PROCESO	FUNCION	FALLAS/DEBILIDADES
Gestion del proceso DBA	Administrar las tecnologías de información y comunicación de la organización así como velar por los aspectos técnicos, tecnológicos, inteligencia de negocios y que se dé cumplimiento de las políticas legales de bases de datos para garantizar la calidad de los datos.	Escasas políticas de seguridad para la integridad de las bases de datos. Las funciones que ejercen los analistas de apoyo a la unidad de bases de datos no van acorde al perfil que requiere el cargo. No existe una fusión que se encuentre estrechamente relacionada con el diseño de los sistemas de información, administración y la programación que permitan tener un buen rendimiento de las aplicaciones en tiempo de ejecución. Falta de políticas y estrategias para gestionar el conocimiento de la unidad.
Administración Bases de Datos	Definir esquemas de las bases de datos a través de diccionarios de datos. Definir la estructura de almacenamiento y los métodos de acceso a las bases de datos.	No se suele gestionar correctamente el creciente volumen de datos que generan los diferentes sistemas de información de la organización en particular aquello de tipo transaccional. La configuración del servidor MYSQL entre otros, el esquema de base de datos y otras facetas de las aplicaciones cambia repentinamente con el tiempo.

	Conceder autorización a usuarios para el acceso de las bases de datos.	La comunicación con los desarrolladores de las distintas aplicaciones es aislada, lo que conlleva a desconocer cómo están estructuradas las bases de datos de los sistemas de información. Demanda excesiva de los servicios de bases de datos para las diferentes áreas de la división de informática y sistemas. Carencia de metodologías que permitan la gestión eficiente de incidencias y control de riesgo.
--	---	--

ACTOR: Unidad de Control de Calidad		ROL: Proveedor
PROCESO	FUNCION	FALLAS/DEBILIDADES
Gestión del proceso de Calidad	Verificar que los objetivos que se han planteado en los requerimientos funcionales previos, se cumplan dentro de los plazos previstos y con los recursos que fueron asignados. Brindar control y seguimiento del funcionamiento de los sistemas de información en producción.	Serías debilidades en la gestión de requisitos que solicita auditoria interna. Falta de políticas de prevención de defectos. Falta de una metodología rigurosa en la gestión de calidad de los sistemas de información. Falta de herramientas que permitan a los analistas QA realizar test de rendimiento a los sistemas de información previo a ser instalados en producción. Poco personal QA con conocimientos en el proceso de control de calidad. Falta de una norma ISO que apoye la gestión de control de calidad.

ACTOR: Unidad de Soporte Técnico		ROL: Proveedor
PROCESO	FUNCION	FALLAS/DEBILIDADES
Gestión del proceso de Soporte Técnico	Brindar mantenimiento preventivo y correctivo de los equipos periféricos de cómputos adquiridos e instalados en la organización. Proporcionar asesoramiento para la adquisición y repotenciación de equipos y productos de software de acuerdo a las necesidades planteadas por la organización.	Falta de un plan de contingencia riguroso ante amenazas de incidencias externas e internas que puedan afectar directamente los servicios informáticos que brinda la división de informática y sistemas.

ACTOR: Dependencias Administrativas		ROL: Proveedor/Cliente
PROCESO	FUNCION	FALLAS/DEBILIDADES
Gestión del proceso administrativo/operativo de la Dirección General de Ingresos	Establecer los requisitos formales inherentes a la administración tributaria, que deben cumplir los sistemas de información tributarios por cada servicio brindado por la institución.	Ejecutivos poco involucrados en los procesos a automatizar. Falta de reuniones periódicas para atender los cambios en los procesos automatizados. Se proponen líneas de tiempos en los cambios demasiado cortas desconociendo el impacto que pueda repercutir en la calidad de los servicios que ofrecen los sistemas de información. Elaboración de requerimientos incompletos lo cual dificulta el avance de los mismos. Se concede un retraso en la planificación (asumiendo un error en la misma) pero a cambio se añaden nuevos requisitos (por lo que se comente un nuevo error).

Como actividad complementaria a las entrevistas se realizó seguimiento y observación de los procesos relacionados al manejo de los requerimientos funcionales y no funcionales y evaluación a la gestión de proyectos de software tradicional que actualmente se usa en la unidad de sistemas tributarios de la división de informática y sistemas así como la interrelación existente con clientes y proveedores internos. A partir de las fallas y debilidades identificadas en la gestión de procesos según el resultado de las entrevistas a los actores involucrados y anotaciones de la observación directa en el campo, se identifican una serie de problemas presentados en la etapa siguiente.

Etapas III. Modelado del Proceso de Negocio.

1.3.1 Identificación y análisis de oportunidades.

A través del diagnóstico situacional se ha detectado un número importante de problemas:

- Falta de herramientas informáticas dentro de la oficina de sistemas tributaria que permitan la gestión de los proyectos de software en desarrollo.
- No están bien definidos los roles que ejercen los analistas de sistemas, sobre todo en área de control de calidad.
- Canal de comunicación informal con las partes interesadas en todo el ciclo de vida de los proyectos de software.
- Los tiempos de respuestas por parte de las oficinas administrativas ante la duda que se les presentan a los analistas para resolver un problema en los procesos de los sistemas de información tributaria, es muy tardía.
- Ajustados tiempos de ejecución dejando poca flexibilidad.
- Sobrecarga de trabajo para los analistas de la Unidad de Sistemas Tributarios.
- Falta de una unidad que atiendan las incidencias de errores que se presentan en el día a día.

- Dificil percepción para el líder de la unidad de sistemas tributarios y la dirección IT en los tiempos de los avances y condiciones de los requerimientos.
- Automatización desordenada debido a la mala planificación que existe en las áreas normativas producto del mandato de la dirección superior por implementar un determinado sistema de información en el menor tiempo posible.
- Elaboración de requerimientos incompletos lo cual dificulta el avance de los mismos.
- Comunicación insuficiente dentro de los equipos de trabajo.
- Errores en los procesos de control de calidad.
- Metodologías para la gestión de proyectos de software estándar inexistentes dentro la unidad de sistemas informáticos de la división de informática y sistemas.

Calidad de los sistemas de información vulnerables a generar errores de excepción de código fuente y procesos.

A. Jerarquización de problemas.

Esta lista de fallas / problemas / debilidades es analizado por un grupo de trabajo de 6 personas conformados por el Líder la de la unidad de sistemas tributarios, Analistas de Sistemas (4), Jefe de sistemas responsable de la unidad de sistemas de información con el fin de identificar los problemas más significativos, a través de la aplicación de una herramienta de jerarquización con el siguiente procedimiento:

- Cada integrante del grupo debe seleccionar, de entre todos los problemas listados, los 10 problemas que considere más importantes y representativo, marcando con una "X" la casilla correspondiente a su número y a cada uno de los problemas seleccionados.

- Después que cada participante seleccione sus 10 problemas más importantes, se contarán las "X" o votos asignados por los participantes a cada problema del listado.
- Todo problema que obtenga 4 votos o más será considerado como prioridad.

Selección y Análisis de los Problemas **(DEBILIDADES / AMENAZAS)** más Importantes para la Organización:

- Falta de herramientas y metodologías para la gestión de proyectos de software.
- Roles mal definidos.
- Canales de comunicación informal con las partes interesadas.
- Tiempos de respuestas tardías (Dependencias Administrativas).
- Tiempos ajustados para el desarrollo de requerimientos.
- Sobrecarga de trabajo para los analistas de sistemas.
- Unidad de atención y corrección de incidencias de errores nula.
- Los responsables IT no logran percibir avances (Requerimientos).
- Automatización desordenada.
- Requerimientos incompletos y mal estructurado.
- Comunicación insuficiente dentro del equipo de trabajo.
- Errores en los procesos de control de calidad.
- Calidad de los sistemas de información vulnerables.

Tabla 12. Detalle de la valoración del equipo de la División de Informática y Sistemas

Selección y Análisis de los Problemas (DEBILIDADES / AMENAZAS) más Importantes para la Organización	INTEGRANTE DEL EQUIPO No.					TOTAL	
	1	2	3	4	5	6	
Falta de herramientas y metodologías para la gestión de proyectos de software	X	X	X	X	X	X	6
Roles mal definidos	X	X	X				3
Canales de comunicación informal con las partes interesadas	X	X	X	X		X	5
Tiempos de respuestas tardías (Dependencias Administrativas)		X	X	X			3
Tiempos ajustados para el desarrollo de requerimientos	X	X	X	X	X		5
Sobrecarga de trabajo para los analistas de sistemas	X	X	X		X	X	5
Unidad de atención y corrección de incidencias de errores nula	X	X	X	X	X	X	6
Los responsables IT no logran percibir avances (Requerimientos).	X	X		X	X	X	5
Automatización desordenada	X	X	X	X	X		5
Requerimientos incompletos y mal estructurados	X	X	X	X	X	X	6
Comunicación insuficiente dentro del equipo de trabajo		X	X			X	3
Errores en los procesos de control de calidad	X	X	X	X	X	X	6
Calidad de los sistemas de información vulnerables	X	X	X	X	X	X	6

Tabla 13. Selección de problemas según la votación obtenida.

Listado de los problemas preseleccionados como los más importantes	Votos
Falta de herramientas y metodologías para la gestión de proyectos de software	6
Roles mal definidos	3
Canales de comunicación informal con las partes interesadas	5
Tiempos de respuestas tardías (Dependencias Administrativas)	3
Tiempos ajustados para el desarrollo de requerimientos	5
Sobrecarga de trabajo para los analistas de sistemas	5
Unidad de atención y corrección de incidencias de errores nula	6
Automatización desordenada	5
Comunicación insuficiente dentro del equipo de trabajo	3
Calidad de los sistemas de información vulnerables	6

A continuación con la utilización de la Matriz SAATY se mide la importancia de los problemas, siguiendo el siguiente procedimiento:

1. Construir una Matriz cuyas filas y columnas contienen los problemas a comparar
2. Comparar cada fila con todas las columnas (seguir escala SAATY) y anotar valor resultante en la casilla correspondiente.
3. Sumar puntuaciones de columnas (Total).
4. Crear nuevas columnas, una para cada problema. Anotar en cada casilla nueva el cociente de los valores originalmente asignados y Total.
5. Calcular medias de cada fila de las nuevas casillas creadas (pesos en tanto por uno de cada estrategia o plan).
6. Transformar tantos por uno en tantos por ciento.

Priorizar cada problema según valores del paso anterior.

Tabla 14. Escala de comparación de la Matriz SAATY

5 - Mucho más Importante
3 - Más Importante
1 - Igual de Importante
1/3 - Menos Importante
1/5 - Mucho menos Importante

Tabla 15. Resultados de la Matriz SAATY

Selección y Análisis de los Problemas (DEBILIDADES / AMENAZAS) más Importantes para la Organización	Matriz SAATY para importancia											
		A	B	C	D	E	F	G	H	I	J	
Falta de herramientas y metodologías para la gestión de proyectos de software	A	1	5	3	1	1/3	1/5	3	1	3	3	11.94 %
Roles mal definidos	B	1/5	1	3	3	1/3	1/3	3	1	5	5	11.87 %
Canales de comunicación informal con las partes interesadas	C	1/3	1/3	1	1	1/3	5	1	5	1/3	3	9.03 %
Tiempos de respuestas tardías (Dependencias Administrativas)	D	1	1/3	1	1	3	5	1/3	1	1/3	5	11.16 %
Tiempos ajustados para el desarrollo de requerimientos	E	3	3	3	1/3	1	3	1	5	1/3	5	14.04 %
Sobrecarga de trabajo para los analistas de sistemas	F	5	3	1/5	1/5	1/3	1	3	3	1	3	11.62 %
Unidad de atención y corrección de incidencias de errores nula	G	1/3	1/3	1	3	1	1/3	1	5	1/3	1/5	7.27 %
Automatización desordenada	H	1	1	1/5	1	1/5	1/3	1/5	1	5	1/3	5.96 %
Comunicación insuficiente dentro del equipo de trabajo	I	1/3	1/5	3	3	3	1	3	1/5	1	1/3	10.37 %
Calidad de los sistemas de información vulnerables	J	1/3	1/5	1/3	1/5	1/5	1/3	5	3	3	1	6.73 %

De acuerdo a esta tabla 15 de escala de importancia, se presentan los resultados de los problemas ordenados según prioridad.

Tabla 16: Nivel de Importancia

Valor de la matriz			Clase	
0 %	a	25%	Alta	4
25%	a	50%	Media Alta	3
50%	a	75%	Media baja	2
75%	a	100%	Baja	1

Tabla 17. Resumen final de los resultados

Listado JERARQUIZADO de los Problemas (DEBILIDADES / AMENAZAS) más Importantes para la Organización	Importancia	Prioridad	Clase
Falta de herramientas y metodologías para la gestión de proyectos de software	11.94%	2	4
Tiempos ajustados para el desarrollo de requerimientos	14.04%	1	4
Roles mal definidos	11.87%	3	3
Tiempos de respuestas tardías (Dependencias Administrativas)	11.16%	5	3
Sobrecarga de trabajo para los analistas de sistemas	11.62%	4	3
Canales de comunicación informal con las partes interesadas	9.03%	7	2
Comunicación insuficiente dentro del equipo de trabajo	10.37%	6	2
Unidad de atención y corrección de incidencias de errores nula	7.27%	8	1
Automatización desordenada	5.96%	10	1
Calidad de los sistemas de información vulnerables	6.73%	9	1

Los primeros 7 problemas de mayor prioridad (80/20) tiene que ver con:

- Falta de herramientas y metodologías para la gestión de proyectos de software.
- Tiempos ajustados para el desarrollo de requerimientos.
- Roles mal definidos.
- Tiempos de respuestas tardías (Dependencias Administrativas)
- Sobrecarga de trabajo para los analistas de sistemas.
- Canales de comunicación informal con las partes interesadas.
- Comunicación insuficiente dentro del equipo de trabajo.

B. Diagrama Causa y Efecto ^[6]

Del listado anterior de problemas / fallas / debilidades /amenazas se va a realizar un análisis de relación de causa y efecto. A continuación se categorizan según las 6M:

RECURSO HUMANO	ELICITACION DE REQUERIMIENTOS	TIEMPO
• Dificultad cognitiva sobre los proceso a automatizar. • Malas prácticas metodológicas de los analistas. • Incremento de la Jornada laboral para cumplir objetivos. • Personal poco motivado y comprometido. • Poco personal en la unidad de control de calidad.	• Problemas de ámbito en los requerimientos. • Requerimientos incompletos. • Expectativas no realistas. • Requisitos cambiantes. • Dificultad de estructuración de los requerimientos.	• Tiempos de entrega no se ajustan a la realidad. • Problemas de volatilidad en los requerimientos.

Sobrecarga de trabajo hacia los analistas de sistemas		
ORGANIZACIÓN DE ROLES	RECURSOS	CANAL DE COMUNICACION
- Inexistencia del rol de arquitectura de software. - Inexistencia del rol de ingeniería de requisitos. - Deficiencia en los roles de control de calidad. - Inexistencia de un rol para atender y corregir incidencia de errores.	- Inexistencia de herramientas para gestionar tareas - Sistema de control de requerimientos incompleto. - Inexistencia de herramientas de medición de rendimiento en los sistemas.	- Dificultad de comunicación entre usuarios y analistas. - Reuniones aisladas. - Carencia de medios multimedios de comunicación .

6/véase Anexo 13. Diagrama Causa y Efecto.

1.3.2 Diagrama de Procesos.

De acuerdo a lo descrito en el Marco Teórico, para esta última etapa se utiliza como marco referencial la metodología BPM, cuyo desarrollo incluye la modelación de los procesos de negocios como primera fase apoyado en la notación BPMN, para lograr una representación gráfica de los procesos, mostrando principalmente cómo y por quién son llevadas a cabo las actividades.

Para dicho fin se utiliza como herramienta de modelado de procesos, BizAgi Process Modeler, por encontrarse ubicada en el 2010 como “Visionario” en el cuadrante mágico de Gartner para Suites de BPMS (Business Process Management Suites) por sus características dirigidas a optimizar el desempeño de los procesos de negocios, y que estos sean visibles y explícitos con un perfil ejecutivo y de tecnología (IT). Además de otra ventaja importante, la facilidad de contar con una versión libre gratuita que permita su uso para fines prácticos en la ejecución de esta etapa.

A partir de la literatura referente al BPM, se presentan los procesos necesarios para diseñar los diagramas (participantes, eventos, actividades, entradas/salidas, tiempo, sistemas de información involucrados, indicadores KPI y reglas de negocio). A continuación se describen los procesos de negocios identificados:

A. Proceso de Negocio I. Planificación de nuevos requerimientos de acuerdo al plan estratégico institucional.

ELEMENTOS	DESCRIPCIÓN
Participantes del proceso.	• Director General • Sub Director General • Gerente IT • Gerentes y personal de otras áreas administrativas • Personal de Informática.
Eventos que dirigen	• ¿Qué inicia el proceso?: Solicitud de

el proceso.	requerimientos de acuerdo al plan estratégico. • ¿Cuál es el resultado final?: Nuevo plan de requerimientos para cumplir los objetivos estratégicos.
Actividades.	1. Surgimiento de nuevo plan. 2. Reunión con las partes interesadas para revisar nuevo plan. 3. Definición de los objetivos y tiempo de ejecución del nuevo plan. 4. Coordinar plan de trabajo para elaborar requerimientos. 5. Verificar que el requerimiento cumple con los objetivos previstos. 6. Verificar validez técnica. 7. Definición de los tiempos de entrega Desarrollo/Pruebas de Calidad/Instalación en Producción.
Entradas y salidas de información generadas	• Nuevos requerimientos. • Respuesta inmediata ante alguna dificultad o sugerencia. • Notificación de invalidaciones legales. • Notificación de invalidaciones técnicas.
Tiempo para realizar cada actividad.	• Actividad 1: Medio día • Actividad 2: 1 día • Actividad 3: 4 días • Actividad 4: 1 día • Actividad 5: 2 día • Actividad 6: 1 hora • Actividad 7: 1 hora
Sistemas de información que intervienen	• Sistema para la gestión de requerimientos (NEXUS)
Indicadores KPI.	• % Cumplimiento de Requerimiento total

	• % Cumplimiento de Requerimiento por característica
Reglas del negocio	El requerimiento total debe de ser cumplido aunque no se cumplan ciertas características del mismo.

B. Proceso de Negocio II. Elaboración de los requerimientos y ejecución de desarrollo.

ELEMENTOS	DESCRIPCIÓN
Participantes del proceso.	• Dependencias administrativas. • Jefes de la unidad sistemas tributarios/apoyo tecnológico. • Líderes de unidad sistemas tributarios/control de calidad/Apoyo Tecnológico.
Eventos que dirigen el proceso.	• ¿Qué inicia el proceso?: Desarrollo de las características de los requerimientos (fase de desarrollo). • ¿Cuál es el resultado final?: Nueva versión de los sistemas de información probadas e instaladas en producción.
Actividades.	1. Distribución de tareas. 2. Fase de codificación. 3. Solicitud de nuevas funcionalidades o modificación al requerimiento inicial. 4. Verificar validez legal. 5. Verificar validez técnica. 6. Análisis de dificultades. 7. Fase de pruebas de control de calidad. 8. Fase de instalación en producción de nuevos cambios
Entradas y salidas de información generadas	• Notificación de invalidaciones legales. • Notificación de invalidaciones técnicas. • Respuesta inmediata ante alguna dificultad o

	sugerencia. • Actas de entrega para pruebas de control de calidad • Actas de entrega para instalación de nuevas características en producción.
Tiempo para realizar cada actividad.	• Actividad 1: 1 hora • Actividad 2: 1 semana* • Actividad 3: 1 día* • Actividad 4: 1 día • Actividad 5: 1 día • Actividad 6: medio día* • Actividad 7: tres días* • Actividad 8: 1 hora*
Sistemas de información que intervienen	• Correo electrónico • Sistema de gestión de bitácoras.
Indicadores KPI.	• % Cumplimiento de Requerimiento total • % Cumplimiento de Requerimiento por característica
Reglas del negocio	El requerimiento total debe de ser cumplido aunque no se cumplan ciertas características del mismo.

Diagrama 1. Planificación de nuevos requerimientos de acuerdo al plan estratégico institucional.

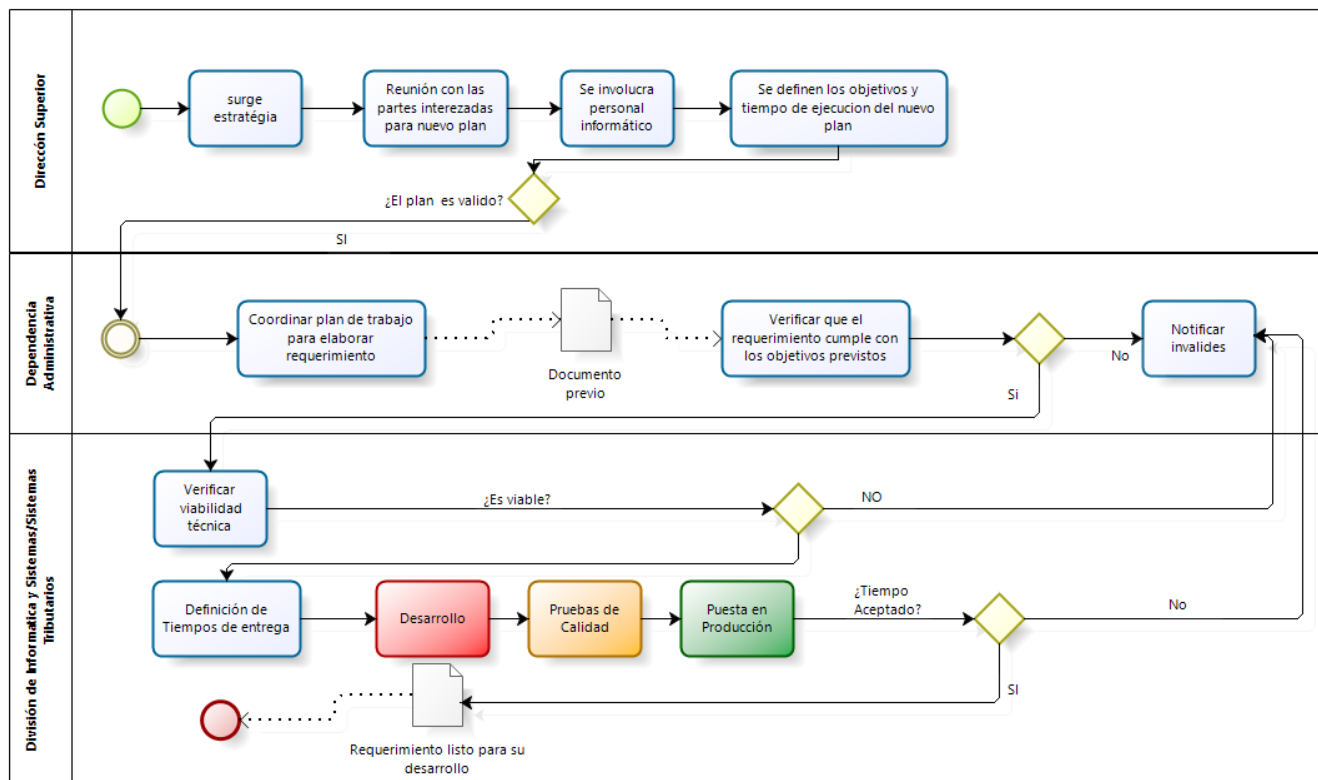
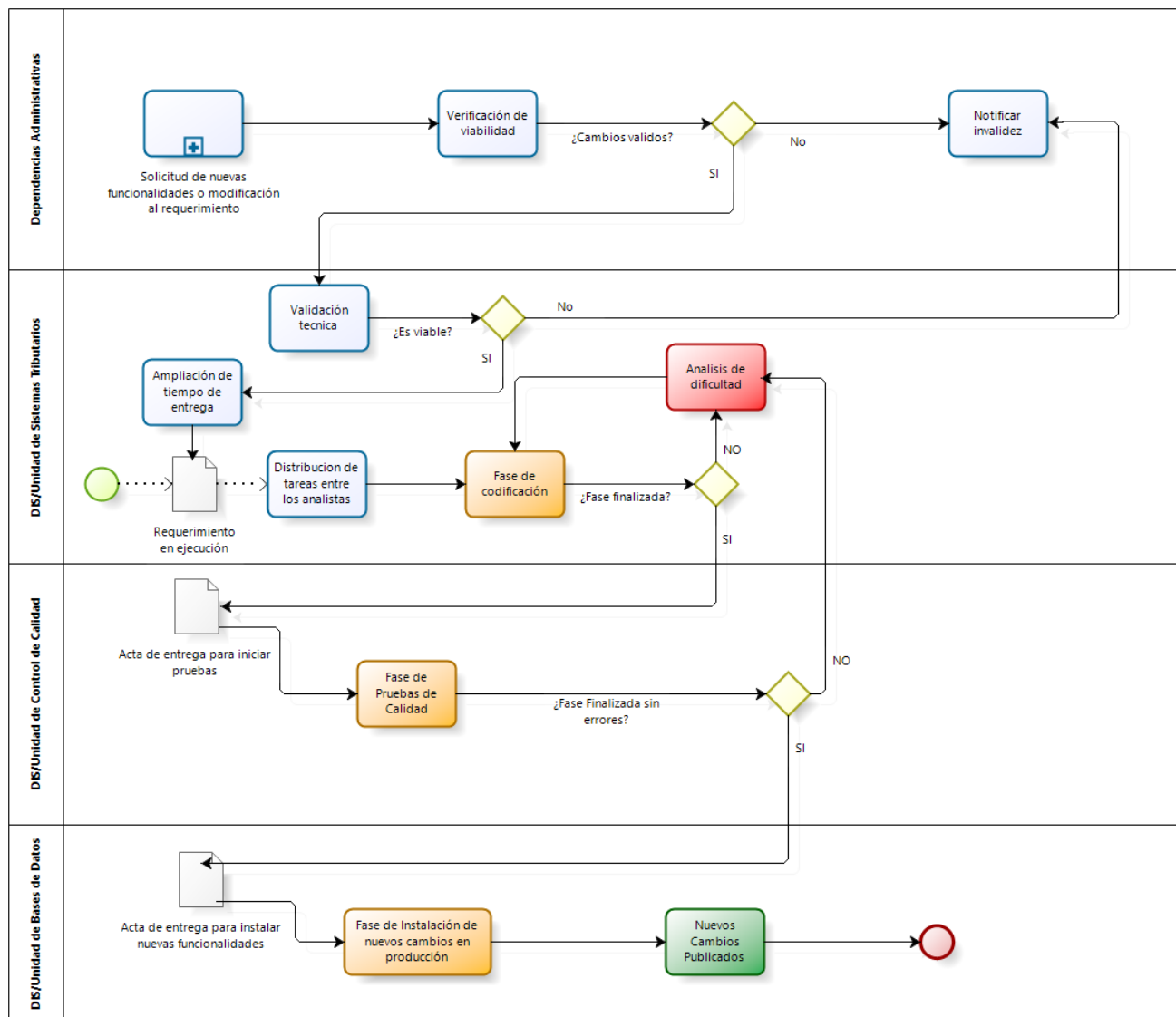


Diagrama 2. Elaboración de los requerimientos y ejecución del desarrollo.



1.2 DEFINICIÓN DE LA METODOLOGIA AGIL DE DESARROLLO DE SOFTWARE.

A partir de la aplicación del diagnóstico organizacional con el apoyo que brinda la modelación de los procesos se pudo definir un modelo de gestión de proyectos de tipo ágil que sea funcional y que ayude a los gestores de proyectos con el desarrollo de los sistemas de información, desde su planificación hasta la puesta en marcha del mismo, dentro del entorno de trabajo de la división de informática y sistemas de la Dirección General de Ingresos. En este caso se decide por SCRUM por contar con mayor documentación y tener mayor aceptación entre las organizaciones ^[1].

SCRUM se define, como una colección de procesos para la gestión de proyectos, que permite centrarse en la entrega de valor para el cliente y la potenciación del equipo para lograr su máxima eficiencia, dentro de un esquema de mejora continua. Y se deriva de mejores prácticas de negocios en empresas como Toyota, Fuji-Xerox, Honda y Canon.

Como características propiamente de una metodología ágil:

- Es un modo de desarrollo adaptable, antes que predictivo.
- Está orientado a las personas, más que a los procesos.
- Emplea el modelo de construcción incremental basado en iteraciones y revisiones.

Sus actividades para mantener un control ágil:

- Revisión de las iteraciones.
- Desarrollo incremental.
- Desarrollo evolutivo.
- Auto organización del equipo.
- Colaboración.

Sus Roles:

- **Scrum Master**, es la persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el Product Owner para maximizar el ROI.
- **Product Owner (PO)**, es el representante de los clientes que usan el software. Se focaliza en la parte de negocio y él es responsable del ROI del proyecto (entregar un valor superior). Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en el Product Backlog y las reprioriza de forma regular.
- **Team SCRUM**, es el grupo de profesionales (Ingenieros de Sistemas de Información) con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las historias a las que se comprometen al inicio de cada sprint.
- **Stakeholder**, son las partes interesadas del proyecto (Usuarios, Clientes).

Sus Artefactos:

- **Backlog de Producto (pila de requisitos)**, es la lista de todas las tareas que son necesarias de realizar sobre el producto. El Product Owner es el responsable de recopilar las nuevas funcionalidades que son requeridas y esperadas sobre el software manteniendo un continuo contacto con el usuario para priorizarlas.
- **Backlog del Sprint**, una vez definidos los objetivos deseados para un sprint, se incluyen en él las tareas que otorgan la funcionalidad que se pretende lograr.
- **Scrum Taskboard (Tablero kanban)**, refleja de forma rápida el estado y avance de todas las historias de usuarios (tarjetas de características del sistema) dentro del spring.

- **Grafica Burndown (Burndown charts)**, es un gráfico de trabajo pendiente a lo largo del tiempo, muestra la velocidad a la que se está completando los objetivos/requisitos. Permite extrapolar si el equipo podrá completar el trabajo en el tiempo estimado.

De acuerdo a las actividades a llevar a cabo en SCRUM serán las siguientes:

1. El gestor o responsable del proyecto deberá gestionar las tareas y detallar un plan de desarrollo de sistemas de información y comunicar las fecha objetivo a los autores involucrados en el proyecto.
2. **Planificación de la iteración**, en el primer día de la iteración se deberá realizar la reunión de planificación el cual consta de dos partes:

a) Selección de los requisitos.

- El usuario deberá presentar al equipo la lista de requisitos priorizadas del producto o proyecto.
- El equipo de desarrollo deberá preguntar al usuario las dudas que surgen y seleccionar los requisitos más prioritarios que se comprometerán a completar en la iteración, de manera que puedan ser entregados si el usuario lo solicita.

b) Planificación de la iteración.

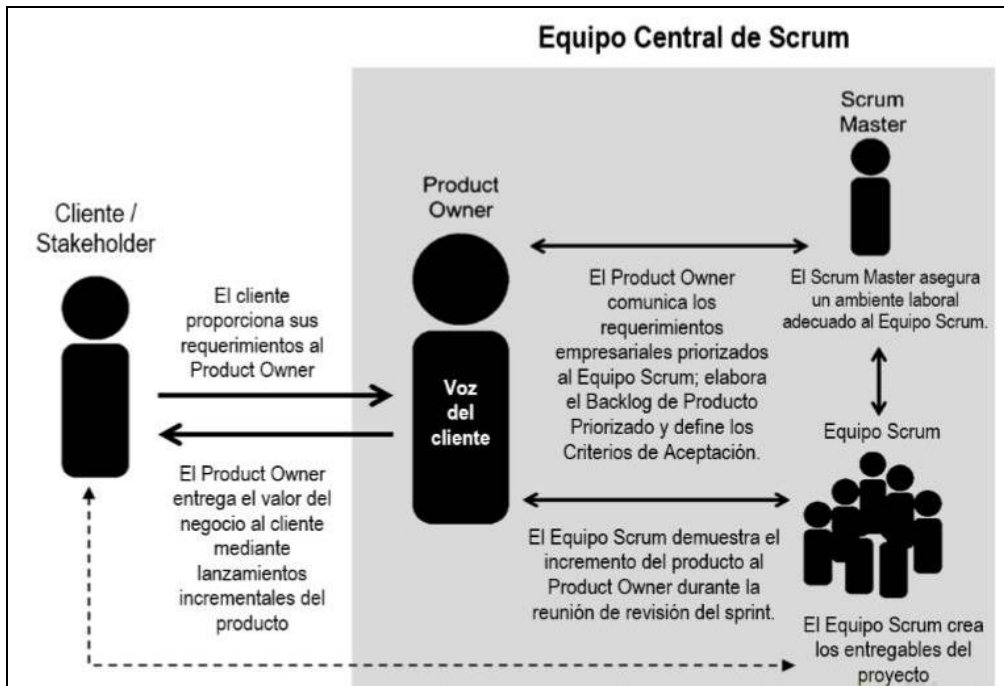
- El equipo deberá elaborar la lista de tareas de la iteración necesarias para desarrollar los requisitos al cual se han comprometido.
- La estimación de esfuerzo se hará de manera conjunta y los miembros del equipo de desarrollo se auto asignarán las tareas.

3. Ejecución de la iteración.

- Cada día el equipo de desarrollo deberá realizar una reunión de sincronización. En donde se inspeccionará el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido.

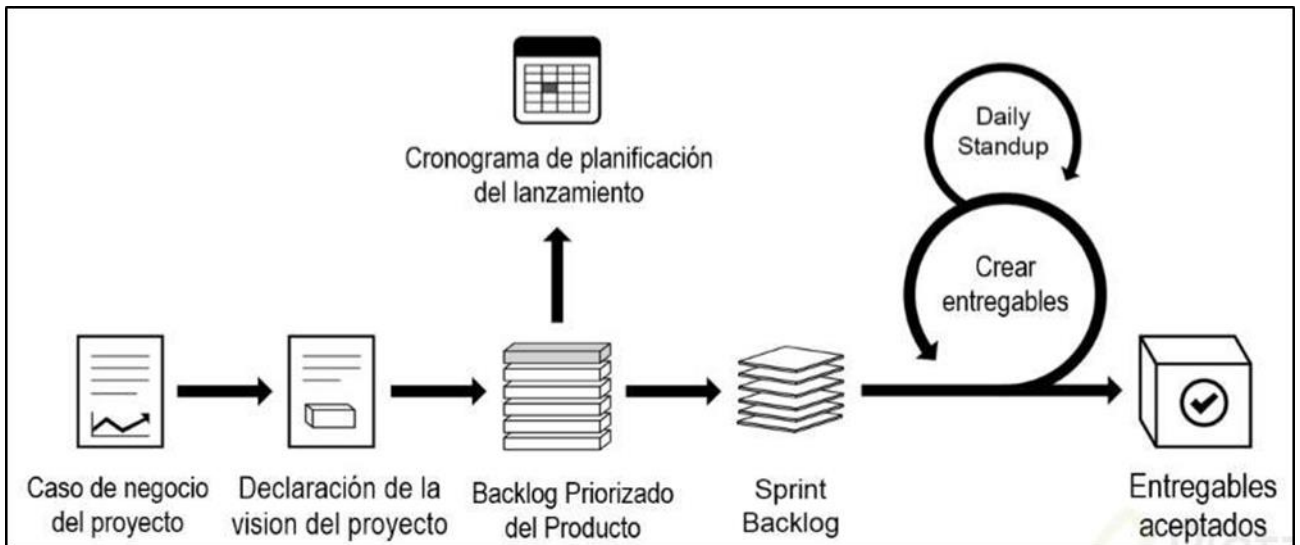
- En la reunión cada miembro del equipo deberá responder a tres preguntas:
 - ¿Qué he hecho desde la última reunión de sincronización?
 - ¿Qué voy a hacer a partir de este momento?
 - ¿Qué impedimentos tengo o voy a tener?
 - Durante la iteración el Facilitador (Scrum Master) se encargará de que el equipo pueda cumplir con sus compromisos y de que no se merme su productividad.
 - ✓ Eliminar los obstáculos que el equipo no puede resolver por sí mismo.
 - ✓ Proteger al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.
4. **Inspección y adaptación.** En el último día de la iteración se deberá realizar la reunión de revisión de la iteración, el cual consta de dos partes:
- a). Demostración:**
- El equipo presentará al usuario los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo.
 - En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el usuario realizará las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, re planificando el proyecto.
- b) Retrospectiva.** El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados.

Figura 7. Gráfica de relación de roles centrales del equipo Scrum.



Fuente: tomado de: <https://platzi.com/blog/que-es-scrum-y-los-roles-en-scrum/>

Figura 8. Gráfica del proceso SCRUM.



Fuente: tomado de: <https://platzi.com/blog/que-es-scrum-y-los-roles-en-scrum/>

CONCLUSIONES DEL CAPITULO I

Las principales conclusiones que se pueden extraer de este capítulo son las siguientes:

- A partir de la aplicación del enfoque metodológico de reingeniería de procesos de negocios (Business process reengineering, BPR), se realiza un diagnóstico actual claro del negocio según el dominio del problema, se describe la interacción de procesos de negocios, actores involucrados y el papel que tienen los mismos.
- La utilización de estas técnicas permitió identificar, jerarquizar y agrupar toda una serie de fallas / debilidades que conforma la situación problemática actual descrita a través de un diagrama Causa / Efecto de los problemas.
- Se identificaron los procesos actuales, y basado en la metodología BPM a través de la herramienta Bizagi Process Modeler, se presenta una descripción y diagramas de los procesos de negocios que conforman la gestión operativa de los proyectos de software de la división de informática y sistemas a través de la unidad de sistemas tributarios y otras entidades relacionadas de apoyo técnico y para la elaboración de los diferentes requerimientos funcionales y no funcionales.
- Finalmente al cumplir el objetivo del diagnóstico organizacional sirve como base importante para lograr una acertada propuesta de implementación de un modelo de proceso de desarrollo de software de tipo ágil, el cual ayudará a los gestores de proyectos con la planificación de los requerimientos funcionales hasta la puesta en marcha del mismo, logrando así mayor productividad, calidad y satisfacción de los clientes/usuarios en el manejo de los sistemas de información tributarios de la Dirección General de Ingresos.

Capítulo II PROPUESTA

Finalmente este en este capítulo se abordará el último objetivo propuesto para el trabajo de tesis final de maestría de informática empresarial.

- Describir un procedimiento detallado que precise los procesos y recursos requeridos para lograr una adecuada implantación de la metodología orientada a la gestión eficiente de proyectos de desarrollo e implementación de software en la división de informática y sistemas de la dirección general de ingresos.

2.1 RUTA DE IMPLEMENTACIÓN DEL MODELO DEL PROCESO AGIL SCRUM ^[1].

Para proponer el procedimiento que permita la implementación del modelo de procesos ágil “SCRUM” en apoyo a la coordinación de las áreas sustantivas de la institución con la oficina de informática y sistemas tributarios para la planificación de los requerimientos funcionales y no funcionales y ejecución de los mismos, es importante retomar los factores de éxito que menciona la literatura de (Dolado, J., & RODRÍGUEZ, D. (2011) y Pressman), abordado en el Marco Teórico: el cual se refieren al éxito organizacional, técnico y personal. Por tanto se describe el proceso a seguir en tres etapas:

- ✓ Etapa I: Diagnostico Situacional.
- ✓ Etapa II: Evaluación Preliminar.
- ✓ Etapa III: Definición final del modelo ágil SCRUM

2.1.1 Etapa I: Diagnóstico Situacional.

Esta fase consiste en un estudio preliminar de la situación, llevado a cabo en el Capítulo II, que va desde la selección y conformación de los equipos de trabajo, definición del negocio, identificación de problemas y oportunidades, evaluación de los procesos de negocios actuales hasta finalizar con la valoración de los beneficios de la metodología de desarrollo e implementación de software propuesta, aspecto que representa una base sólida como punto de partida en el transcurso del proceso de decisión de proponer un modelo de proceso de gestión de proyectos ágil que se acople a la realidad de la organización.

Definición del Proceso de licitación de requerimientos.

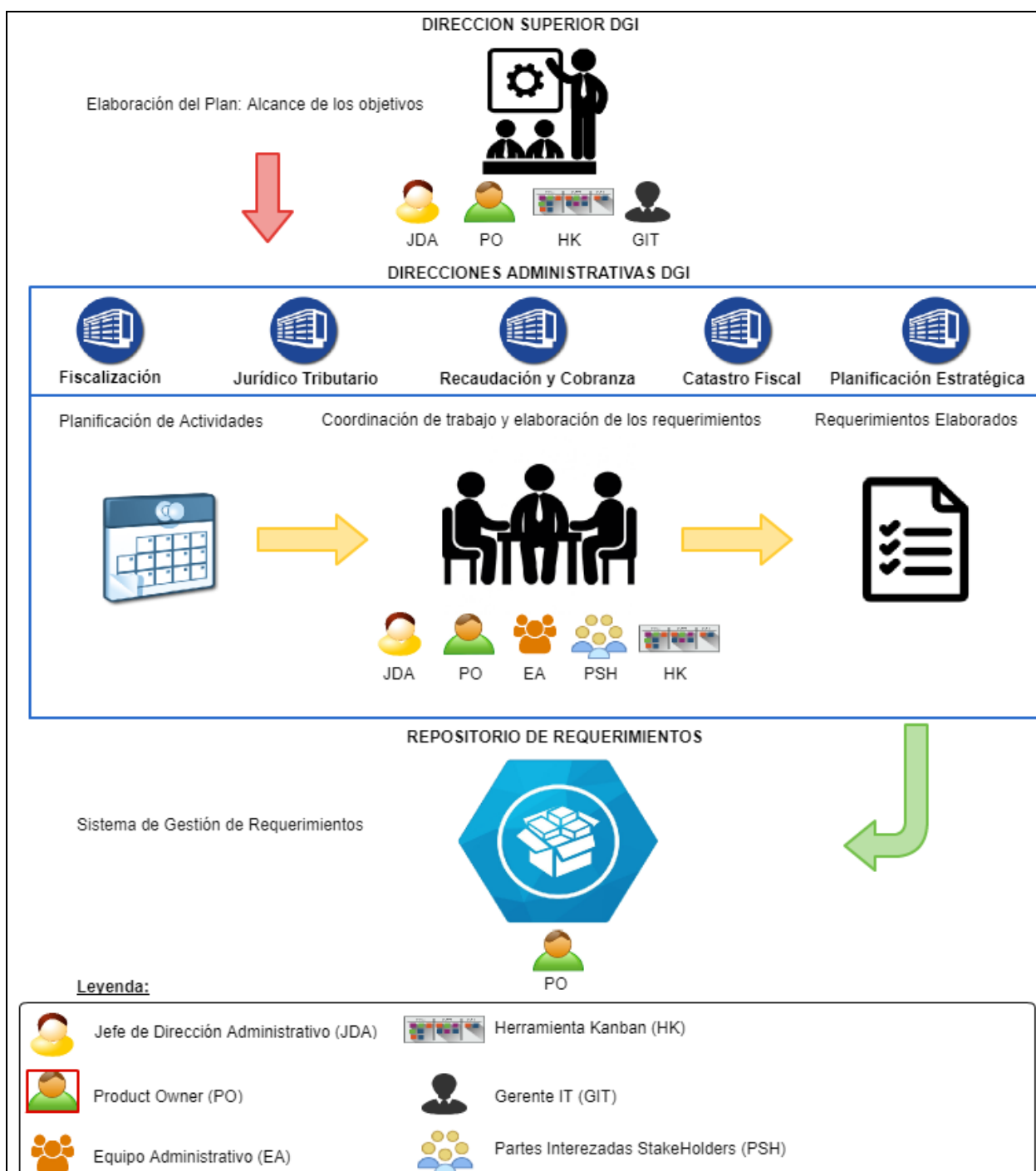
Habiendo identificado a los actores involucrados de acuerdo a sus capacidades para cada dirección administrativa de la Dirección General de Ingresos, que jugarán el papel de product owner en el modelo propuesto para la elicitación de requerimientos, se deja definida las actividades que se deberán seguir, tales como: reuniones con la dirección superior de la institución de donde surge elabora el plan con sus objetivos. En esta etapa estarían involucrados (Director, Sub Director, Jefes de Direcciones administrativas, el product owner y el gerente de IT). También se harán acompañar de la herramienta KANBAN (herramienta de apoyo ágil) para el seguimiento de las metas y objetivos.

En las siguientes etapas de planificación de actividades y coordinación de trabajo y elaboración de requerimientos se involucran: Jefes de Direcciones administrativas, el product owner, los StakeHolders y parte del equipo administrativo con las mejores capacidades en el manejo de las reglas del negocio apoyándose de la herramienta ágil KANBAN para que puedan tener un panorama claro del movimiento de los requerimientos en el cumplimiento de los objetivos propuestos.

Finalmente los Product Owner tendrán la obligación de gestionar los requerimientos en un repositorio o sistema de gestión de los documentos elicitados. De acuerdo a lo descrito en el capítulo II, en los roles de SCRUM. El Product Owner es quien representa al cliente y se convierte en el dueño del producto, por tanto en el modelo propuesto se hará necesario integrar a este nuevo cargo dentro de cada dirección administrativa de la Dirección General de Ingresos. De igual modo tendrá la responsabilidad de e licitador de requerimientos funcionales y no funcionales y estar en constante comunicación directa con el equipo de trabajo SCRUM de la división de informática y sistemas. Los StakeHolders en referencia a la documentación investigada, son las partes interesadas en los proyectos, estos pueden ser agentes externos (Consultores externos, otras instituciones público/privado) e internos (Consultores internos).

De este modo se estaría escalando la metodología ágil SCRUM en parte de la organización.

Figura 9. Modelo Propuesto donde si integra el rol de e licitación de requerimientos (Product Owner).

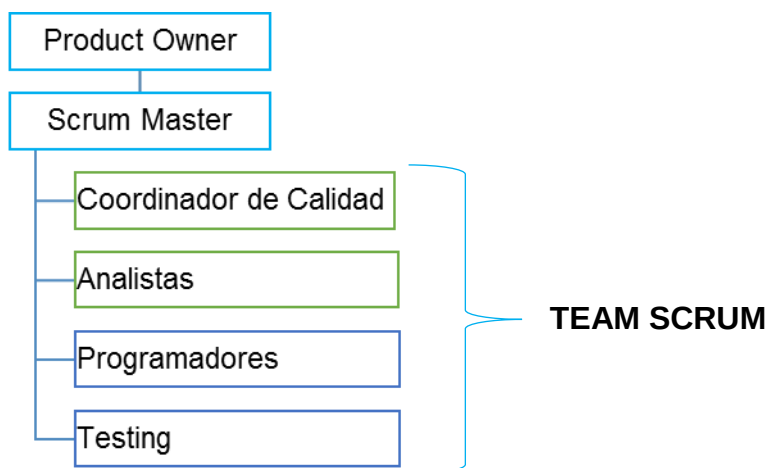


Fuente: Elaboración Propia

2.1.2 Etapa II: Evaluación Preliminar.

Esta etapa se subdivide en tres procesos:

- Identificación de los actores que conformaran el marco de trabajo de SCRUM.



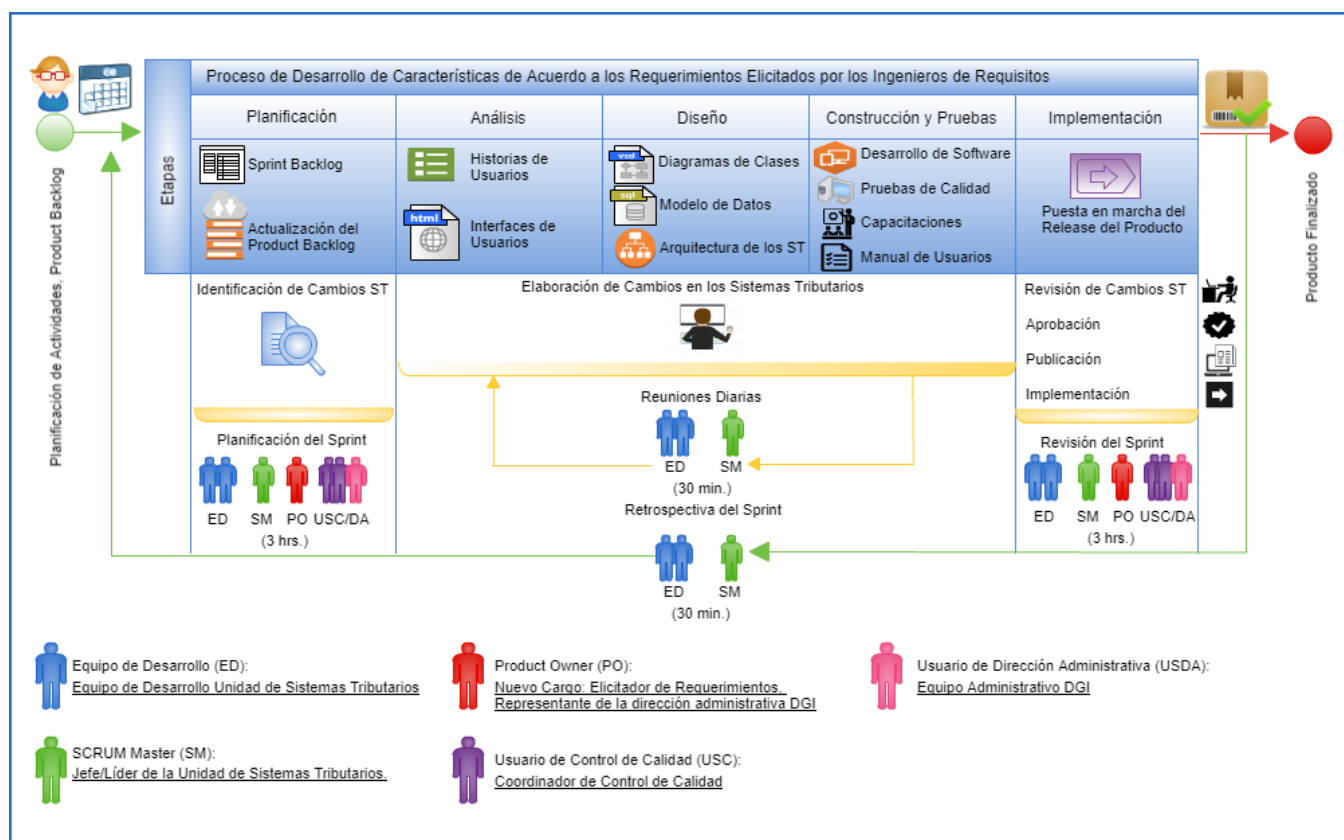
- Evaluación y preselección de los actores que conformarán el marco de trabajo de SCRUM.

Rol	Test de Habilidades
PRODUCT OWNER	- Habilidad de comunicación. - Amplios conocimientos de los procesos de la organización. - Capacidad tomar decisiones inmediatas. - Capacidad de negociación. - Alto desempeño en su rol de planificador. - Amplios conocimientos en elicitación de requerimientos.
SCRUM MASTER	- Habilidad de comunicación - Nivel de experiencia en gestión de proyectos. - Habilidad de negociación. - Habilidad de coordinación de trabajo. - Habilidad de motivación hacia su equipo de trabajo.
TEAM SCRUM	- Habilidad de trabajar en equipo - Amplios conocimientos informáticos - Pro activos - Motivados - Innovadores - Comprometidos con el proyecto - Disponibilidad 100%

2.1.3 Etapa III. Definición final del modelo ágil SCRUM.

En esta última etapa se definen los roles y artefactos que serán necesarios en la metodología ágil propuesta “SCRUM”, como entorno de trabajo dentro de la división de informática y sistemas de la Dirección General de Ingresos para la gestión de proyectos de software con requerimientos cambiantes en tiempos cortos y el cumplimiento de los objetivos propuestos en ellos. (Ver Figura 3.1.3.1)

Figura 3.1.3.1 Propuesta para implementar el modelo de gestión de proyectos ágil SCRUM dentro de los procesos operativos de la División de Informática y Sistemas.



a) Definición de los Roles:

- Product Owner: Representa al cliente, aunque también puede ser un miembro del equipo con suficiente conocimiento en las reglas del negocio. En el caso particular de esta propuesta, se deberá integrar un a un experto con habilidades en la elicitación de requerimientos en cada dirección administrativa de la Dirección General de Ingresos y serán los que representaran a este rol, dado que conceptualizaran el modelo del negocio en base a las necesidades de los procesos de las diferentes direcciones de la institución y brindarán la información necesaria para el desarrollo de los distintos proyectos TIC. Además, deberán ser los encargados de aprobar los cambios realizados en los Sistemas de Información Tributarios que han sido propuestos al final de cada Sprint.
- Scrum Master: Representa al líder del proyecto y es el encargado de asegurarse que el plan se lleve a cabo de acuerdo con las prácticas, valores y reglas de Scrum. Claramente, la persona a la que le corresponde este cargo es al Jefe/Líder de la Unidad de Sistemas Tributarios para transforma las necesidades de las diferentes direcciones administrativas de la institución en requerimientos del proyecto, revisando los cambios elaborados por el equipo de desarrollo y asegurándose que estos a su vez se encuentren con los lineamientos que la Unidad de Control de Calidad exige.
- Equipo de Desarrollo: Equipo auto organizado encargado tanto del análisis, diseño, desarrollo, pruebas y documentación de los Sistemas de Información Tributarios. Este cargo deberá ser asumido en su totalidad por el Equipo de la Unidad de Sistemas Tributarios, tomando los roles particulares que sean necesarios en cada caso. Serán quienes lleven a cabo la implementación total del proyecto de cara a las diferentes direcciones administrativa de la institución, quienes mantendrán reuniones continuas con el Jefe/Líder de la Unidad de Sistemas Tributarios para asegurar que el progreso del desarrollo, cubra los requisitos exigidos por las direcciones administrativas y superior. Además, serán los encargados de elaborar la documentación necesaria para el manejo de los sistemas de información tributarios.

b) Elementos en la gestión de los proyectos de software:

- Iteraciones: cada proyecto se deberá llevar a cabo en 8 Sprints de 3 semanas cada uno como máximo,
- Actividades:
 - Planificación del Sprint: El Jefe/Líder de la Unidad de Sistemas Tributarios deberá reunirse con el equipo de desarrollo y usuarios para determinar que requerimientos se van a implementar durante la iteración y definir las tareas a realizar. En esta etapa se identifican los posibles cambios que se producirán en el Sistema de Gestión de Requerimientos.
 - Duración: 3 horas como máximo.
 - Entradas:
 1. Cronograma de actividades.
 2. Product Backlog.
 3. Release del Producto del Sprint anterior.
 4. Documentación del Sistema de Gestión de Requerimientos
 - Salidas:
 1. Sprint Backlog.
 2. Actualización del Product Backlog.
 3. Identificación de los cambios a realizar en la documentación del Sistema de Gestión de Requerimientos.
 - Reuniones diarias: Se realizan reuniones breves diariamente al iniciar el día para medir el avance de los proyectos y documentar los cambios que se actualizarán, posteriormente, en la Documentación del Sistema de Gestión de Requerimientos. En esta reunión, deberá participar el Equipo de desarrollo y el Jefe/Líder de la Unidad de Sistemas Tributarios.
 - Duración: 30 minutos máximo.
 - Entradas:
 1. Sprint Backlog.
 2. Burndown Chart.
 3. Historias de usuario.
 4. Descripción de las tareas realizadas por el equipo de desarrollo.
 - Salidas:
 1. Burndown Chart actualizado.

2. Elaboración de la documentación de los cambios identificados en el Sistema de Gestión de Requerimientos.

- Revisión del Sprint: Se realiza al finalizar el Sprint, donde el equipo de desarrollo con el Jefe/Líder de la Unidad de Sistemas Tributarios deberán presentar a los Jefes de dirección de la institución, coordinador de calidad y usuarios, una demo de la versión del Release del producto construido en el Sprint. El Coordinador de Control de Calidad deberá revisar la documentación elaborada y los jefes de dirección deberán aprobar la versión del Release del Producto para ser liberados en producción.
- Duración: 3 horas.

➤ Entradas:

1. Release del Producto construido en el Sprint
2. Documentación de cambios a actualizar en producción.

➤ Salidas:

1. Revisión, Aprobación, Publicación e Implementación de los cambios en la documentación del Sistema de Gestión de Requerimientos.
 - Retrospectiva del Sprint: Se deberá realizar luego de finalizar la Revisión del Sprint, donde el Equipo de Desarrollo se reúne con Jefe/Líder de la Unidad de Sistemas Tributarios para analizar la forma de trabajo y proponer mejoras en base a la experiencia del último Sprint.
 - Duración: 30 minutos máximo.

c) Recursos

- Personal altamente cualificado en la metodología ágil SCRUM.
- Certificación del personal que se involucrará en el entorno de trabajo ágil SCRUM.
- Disponibilidad de herramientas como software de gestión de proyectos que permitan a los involucrados organizar las tareas de acuerdo a la metodología SCRUM (JIRA, ICESCRUM).
- Condiciones para reuniones (Pizarras, equipos multimedios).

2.2 BENEFICIOS.

De acuerdo a la problemática presentada en el capítulo II del diagnóstico situacional se plasman los problemas de mayor relevancia que afectan directamente la gestión eficiente de los proyectos de software en la división de informática y sistemas a través de la unidad de sistemas tributarios y se dejan claros los beneficios que tendría la implementación de una metodología ágil propuesta en el capítulo II “SCRUM”, para el correcto funcionamiento en los procesos operativos de la organización.

- **Variaciones constantes en las necesidades de la Oficina de Admisión:** La metodología ágil Scrum permite adaptar y redirigir el proyecto según las nuevas prioridades o los cambios que surjan en los requerimientos debido a la constante variación de las necesidades de las distintas direcciones administrativas de la institución, así mismo de la dirección superior para dar cumplimiento al plan estratégico institucional.
- **Plazos cortos de avances en el desarrollo de los requerimientos y entregas regulares del producto final:** Con esta metodología se pueden presentar partes del producto final que son completamente funcionales en cada iteración, lo cual permitirá un avance progresivo durante las fechas establecidas por las direcciones administrativas de la institución.
- **Equipo de desarrollo mejor organizado:** La naturaleza ágil de la metodología SCRUM hace fácil que los integrantes del equipo colaboren entre sí, ya que si se realiza de manera abierta, independientemente de sus responsabilidades en los proyectos de software, se enriquece y asegura su desarrollo para tener un software de calidad que sea eficiente hacia los usuarios finales.

2.3 CONCLUSIONES DEL CAPITULO II.

Las principales conclusiones que se pueden extraer de este capítulo son las siguientes:

- Se presentó un modelo de e licitación de requerimientos de acuerdo a la ruta de implementación de la metodología ágil SCRUM de previo, en el cual se detalla el rol del product owner que deberá desempeñar en el modelo de proceso propuesto.
- Se presentó una propuesta en donde se describe y detalla los procesos y recursos requeridos para para lograr la gestión eficiente de proyectos de desarrollo e implementación de software de acuerdo al modelo de proceso ágil SCRUM en el entorno de trabajo de la división de informática y sistemas de la dirección general de ingresos a través de la unidad de sistemas tributarios.
- Se detallan los beneficios alcanzables que traerían a la organización operativa y administrativa de la Dirección General de ingresos la implementación de una metodología ágil SCRUM en los procesos de desarrollo de software ejecutados por la división de informática y sistemas a través de la unidad de sistemas tributarios.

Conclusiones

- Se elaboró un diagnóstico de la situación actual mediante la evaluación de los procedimientos asociados a la gestión de proyectos de desarrollo e implementación de software en la división de informática y sistemas de la dirección general de ingresos.
- Se presenta un análisis descriptivo y comparativo de metodologías de desarrollo e implementación de software cuyas fortalezas brinden oportunidades de mejora en la gestión actual de los proyectos de sistemas de información para la División de Informática y Sistemas de la Dirección General de Ingresos.
- Se seleccionó una metodología específica que permita la eficiente gestión de los proyectos de desarrollo e implementación de software en la división de informática y sistemas de la dirección general de ingresos.
- Se describe un procedimiento detallado que precise los procesos y recursos requeridos para lograr una adecuada implantación de la metodología orientada a la gestión eficiente de proyectos de desarrollo e implementación de software en la división de informática y sistemas de la dirección general de ingresos.

Recomendaciones

- Se propone esta tesis como guía de referencia para la implementación de un modelo de procesos que incluye la elicitación de requerimientos y gestión de proyectos e implementación de software ágil “SCRUM”.
- Utilizar un enfoque de metodologías de desarrollo e implementación de software ágil que fomente la aplicación de las mejores prácticas de acuerdo a las recomendaciones que hace COBIT 5 en sus catalizadores, permitiendo un alineamiento integral entre los procesos operativos de la División de Informática y Sistemas a través de la Unidad de Sistemas Tributarios y las diferentes dependencias administrativa de la Institución que satisfagan las necesidades de toda la organización.
- Se recalca la importancia de involucrar y lograr la participación de todo el personal involucrados en el cumplimiento de los objetivos propuestos en el plan estratégico institucional permitiendo vencer la resistencia al cambio representando un factor de éxito organizacional para la adopción de la metodología ágil SCRUM.
- Es importante para la organización identificar los roles correspondientes en el modelo de proceso ágil SCRUM.

Bibliografía

- Bazan, P., Giandini, R., & Diaz, J. (2009). *Modelos de procesos y modelos de servicios: una visión unificada*. Jornadas Chilenas de Computación. Recuperado de https://www.researchgate.net/profile/Javier_Diaz25/publication/291344921_Modelos_de_procesos_y_modelos_de_servicios_una_vision_unificada/links/56a10d1408ae984c4498c89a.pdf
- Gómez, O. T., López, P. P. R., & Bacalla, J. S. (2014). *Criterios de selección de metodologías de desarrollo de software*. Industrial Data, 13(2), 070-074. Recuperado de <http://revistasinvestigacion.unmsm.edu.pe/index.php/idata/article/view/6191/5386>
- Hammer & Champy (1993). “*Manifiesto para la Revolución de los negocios*”. Recuperado de <http://dit.upm.es/~fsaez/intl/capitulos/5%20-Reingenier%EDa%20I.pdf>
- LASES, Franyutti Ma. Angélica. “*Metodología de la Investigación: Un nuevo enfoque*”. Ed. Lases Print. Primera edición. Hidalgo, 2006.
- Laudon, Kenneth C y Laudon, Jane p (2004). *Sistemas de Información Gerencial. Octava Edición*. Pearson Educación, México (2004). ISBN 970-26-0528-8. Recuperado de https://books.google.com.ni/books?id=KD8ZZ66PFgC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Martínez, N. G. C.; Bravo, P. D.; Rodríguez, V. M. M.; Pineda, L. G (2012, Octubre). *La Comunicación estratégica en la empresa intercultural dentro del mundo digital*. Revista Electrónica de Investigación y Docencia (REID), Número Monográfico 141-155. ISSN: 1989 2446. Recuperado de <http://www.ujaen.es/revista/reid/monografico/n2/REIDM2art7.pdf>
- Oktaba, H., Esquivel, C. A., Ramos, A. S., Martínez, A. M., Osorio, G. Q., López, M. R.,... & Lemus, M. Á. F. (2005). *Modelo de procesos para la industria de software Moprosoft*. Instituto de Investigaciones en Matemáticas

Aplicadas y en Sistemas (IIMAS), Distrito Federal, México. Recuperado de <https://www.uv.mx/rrojo/mis/desarrollo1/material/moprosoft-v1.1.pdf>

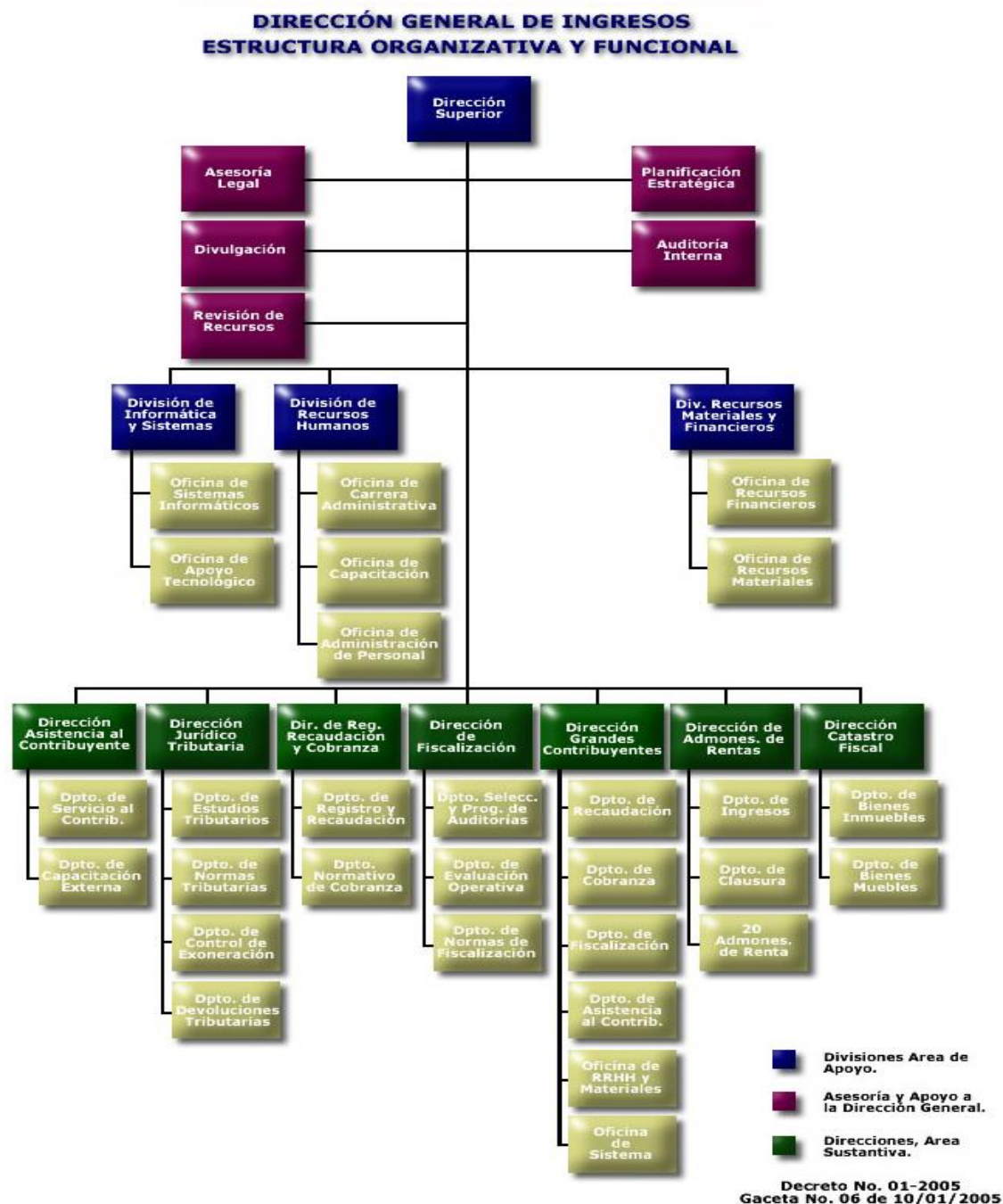
- Saroka (2002). *Sistemas de información en la era digital*. Fundación OSDE, Argentina. ISBN: 987-9358-08-2.
- Sojo William. (2015, Septiembre 22). *Influencia e importancia de las TIC en las organizaciones*. Recuperado de <https://www.gestiopolis.com/influencia-e-importancia-de-las-tic-en-las-organizaciones/>
- Vasquez, P. M., Giandini, R. S., & Bazán, P. (2010). *Lenguajes Notacionales para Modelado de Procesos: un análisis comparativo*. In XII Workshop de Investigadores en Ciencias de la Computación. Recuperado de http://sedici.unlp.edu.ar/bitstream/handle/10915/19526/Documento_completo.pdf?sequence=2
- Gilart, V. (2010). *Metodología para la gestión integral de los procesos de producción: modelado de la maquinaria industrial como un sistema de gestión de procesos de negocio*. Universidad de Alicante. Recuperado de www.rua.ua.es
- Martínez, A., & Martínez, R. (2014). *Guía a rational unified process*. Escuela Politécnica Superior de Albacete–Universidad de Castilla la Mancha.
- Booch, G., Rumbaugh, J., Jacobson, I., Martínez, J. S., & Molina, J. J. G. (1999). *El lenguaje unificado de modelado (Vol. 1)*. Addison-Wesley. Recuperado de <http://elvex.ugr.es/decsai/java..pdf/3E-UML.pdf>
- Díaz Piraquive, F. N. (2008). *Gestión de procesos de negocio BPM (Business Process Management), TICs y crecimiento empresarial. ¿Qué es BPM y cómo se articula con el crecimiento empresarial?* Universidad & Empresa, 7(15). Recuperado de <http://www.redalyc.org/pdf/1872/187214457007.pdf>
- Torres, M. & Aristizábal, N. (2009). *Técnicas de Levantamiento de Requerimientos con Innovación*. [online] www.researchgate.net. Recuperado de https://www.researchgate.net/profile/Miguel_Torres_Moreno/publication/244724135_Tecnicas_de_Levantamiento_de_Requerimientos_con_Innovacion/links/

[00b4951f0312e1cb80000000/Tecnicas-de-Levantamiento-de-Requerimientos-con-Innovacion.pdf](https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/03-Requisitos.pdf)

- Pressman, R. S., & Troya, J. M. (1988). *Ingeniería del software*.
- Méndez, G. (2008-2009) *Ingeniería de Requisitos*. Dpto. de Ingeniería de Software e Inteligencia Artificial Facultad de Informática Universidad Complutense de Madrid. Recuperado de <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/03-Requisitos.pdf>
- González, R, (2015). *Ingeniería de Requisitos*. Recuperado de <https://ingenieriadesoftware4u.wordpress.com/author/trjames91/page/2/>
- Cáceres, P., Marcos, E., & Kybele, G. (2001). Procesos ágiles para el desarrollo de aplicaciones Web. Taller de Web Engineering de las Jornadas de Ingeniería del Software y Bases de Datos de, 2001. Recuperado de <http://www.dlsi.ua.es/~jaime/webe/articulos/s112.pdf>
- Canós, J. H., Letelier, P., & Penadés, M. C. (2003). Metodologías ágiles en el desarrollo de software. *Metodologías Ágiles en el Desarrollo de Software*, 1(10), 1-8.

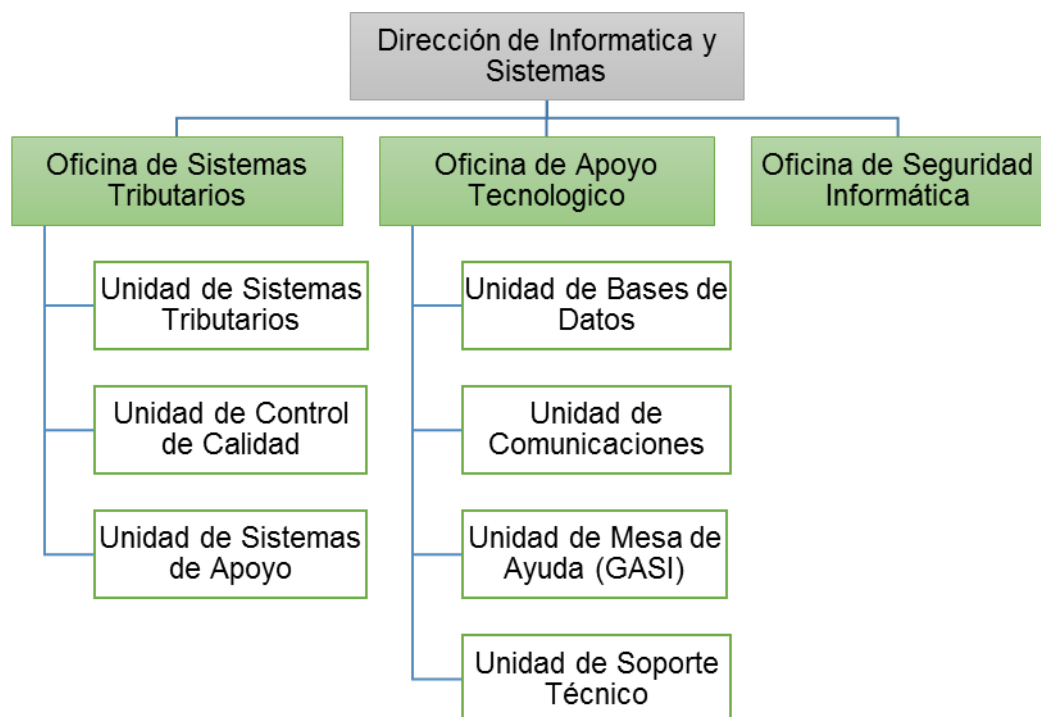
Anexos

Anexo 1. Estructura Organizativa de la Dirección General de Ingresos.

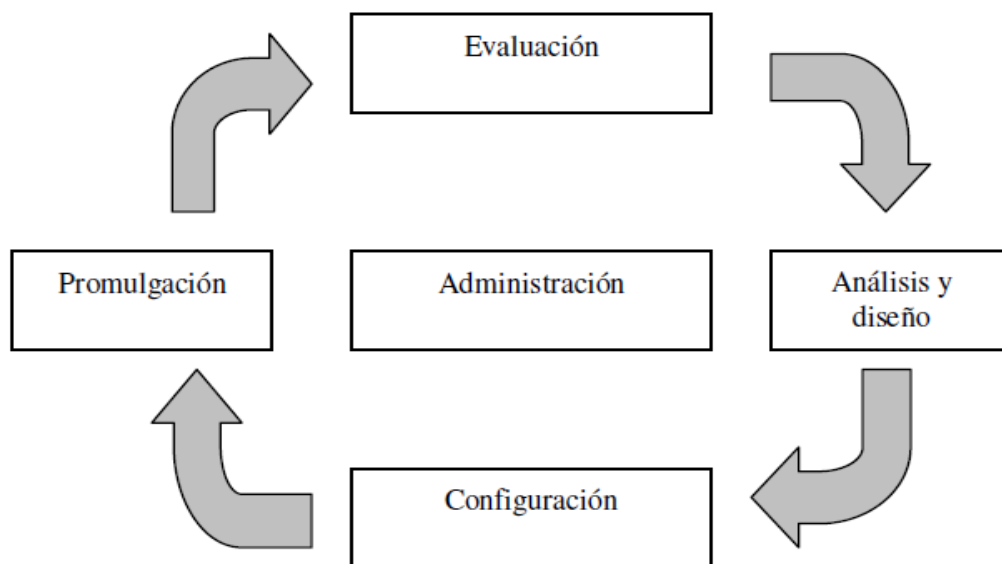


Fuente: www.dgi.gob.ni

Anexo 2. Estructura Organizativa de la División de Informática y Sistemas.



Anexo 3. Figura 1. Ciclo de vida de los Procesos de Negocio



Fuente: Bazán, P., Giandini, R., & Diaz, J. (2009). Modelos de procesos y modelos de servicios: una visión unificada. Jornadas Chilenas de Computación.

El Ciclo de Vida de los procesos de negocios se componen de fases cíclicas que no implican necesariamente un orden temporal, pero si una dependencia lógica, según lo define M.Weske.

Anexo 4. Tabla 1. Clasificación de los diagramas estándar UML.

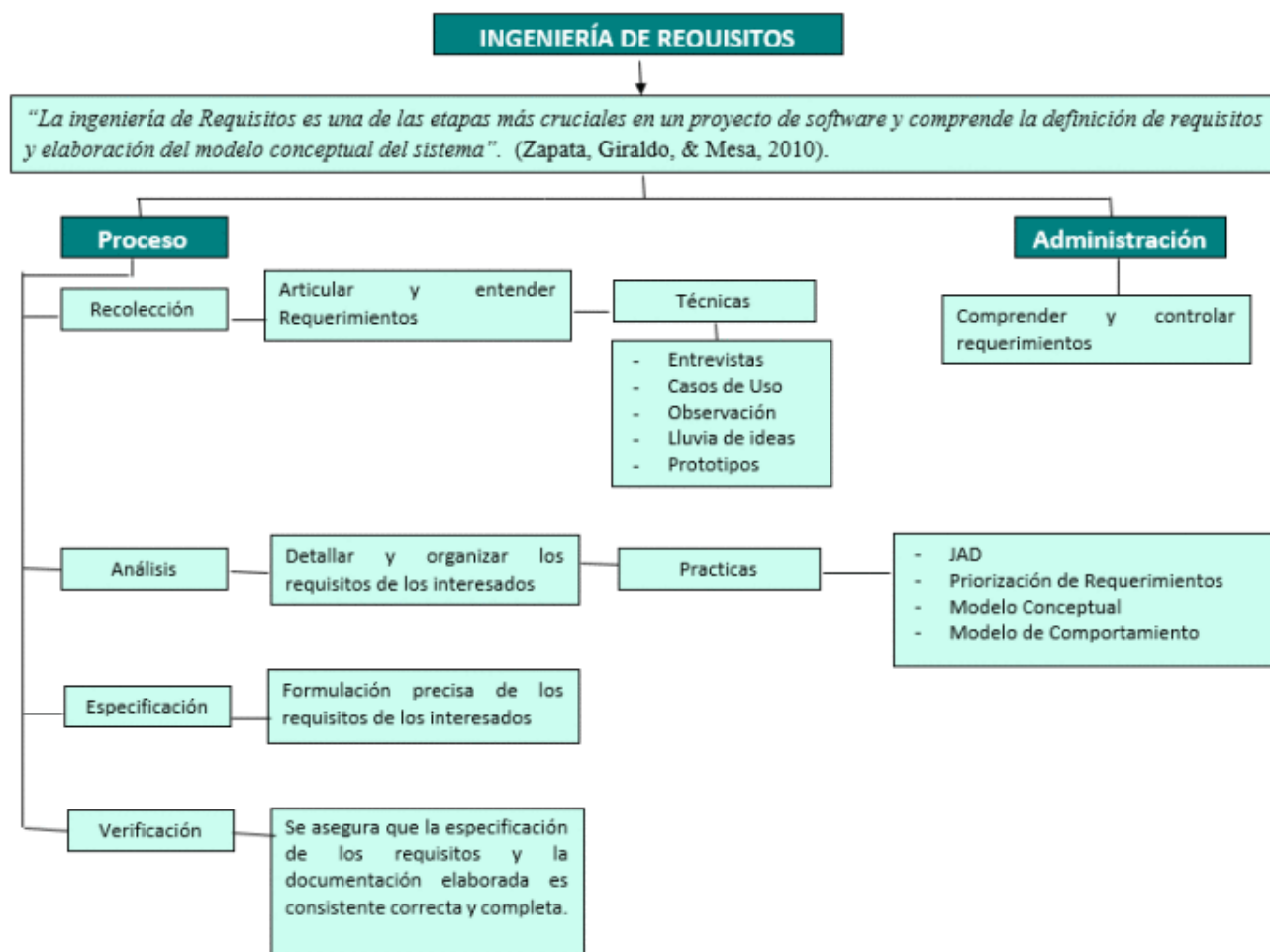
Tabla1. Clasificación de los diagramas estándar UML.	
Tipo de Diagrama	Características
Diagrama de clases	Muestran un conjunto de clases y sus relaciones. Proporcionan una perspectiva estática del sistema.
Diagrama de objetos	Muestran un conjunto de objetos y sus relaciones.
Diagrama de instancias	Muestran una interacción concreta, un conjunto de objetos y sus relaciones juntos con los mensajes que se envían entre ellos.
Diagramas de secuencia	Resaltan la ordenación temporal de los mensajes que se intercambian.
Diagramas de comunicación (UML2.0)	Resaltan la organización estructural de los objetos que intercambian mensajes.
Diagramas de casos de usos	Utilizan el modelado de sistemas desde el punto de vista de usuarios para representar acciones de cada usuario.
Diagramas de estado	Describen el comportamiento de un sistema reactivo cuyo comportamiento está dirigido por eventos.
Diagrama de actividades	Muestran el orden en el que se van realizando las tareas dentro de un sistema.
Diagrama de componentes	Organización lógica de la implementación de un sistema.
Fuente: (Booch, G., Rumbaugh, J., Jacobson, I., Martínez, J. S., & Molina, J. J. G. (1999)).	

Anexo 5. Tabla 2. Clasificación de técnicas para el levantado de requerimientos.

Tabla 2: Clasificación de técnicas para el levantado de requerimientos.	
Tipo	Características
Entrevistas	- El equipo de ingeniería de requerimientos hacen preguntas a las partes interesadas sobre el sistema que utilizan y el que se va a desarrollar. - Pueden ser de dos tipos. a) Cerradas: Donde las partes interesadas responden a un conjunto predefinido de preguntas. b) Abiertas: No existe un programa definido. - Son utilizadas para obtener información preliminar de la organización, entendimientos de nuevos dominios para identificar conflictos entre las partes interesadas, comprender lo que hacen, como interactúan con el sistema y que dificultades existen con el actual.
Grupos focales y sesiones colaborativas	- Se reúnen grupos entre 6 a 10 personas para discutir experiencias y opiniones alrededor de un tema introducido por algún moderador. - Cada sesión dura alrededor de 1 a 2 horas y es una manera rápida para entender la percepción de los usuarios acerca de un tema o concepto en particular. - Son muy útiles para descubrir requerimientos conflictivos y cuando existen partes interesadas distribuidas geográficamente.
Etnografía u observación	- Es una técnica utilizada para entender los requerimientos sociales y organizacionales. - Permite a los analistas sumergirse en el entorno laboral donde se utilizará el sistema, observar el trabajo diario y anotar las tareas reales en la que los participantes

	estarán involucrados. - Los grupos de trabajo por lo general son de composición interdisciplinaria. - Se basa de la observación y entrevistas. - Por lo general son relativamente costosas pero altamente instructivas, debido a que ilustran muchas tareas y los procesos organizacionales.
Modelado	- Se basan en diagramas de flujos de datos, escenarios, casos de usos, diagramas de estado, modelo entidad relación, UML entre otros. - Juegan un papel de suma importancia en el levantado de requerimientos. - Ayudan a mejorar el entendimiento entre los equipos multidisciplinarios previendo una estructura a grandes cantidades de información necesaria para expresar y llevar a cabo los requerimientos. - Reducen la ambigüedad y simplifican la detección de conflictos.
Fuente: Elaboración propia	

Anexo 6. Figura 2. Estructura de la Ingeniería de Requisitos



Fuente: González, R, (2015).

Anexo 7. Tabla 3. Cuadro descriptivo de las fases del ciclo de vida del software.

Cuadro descriptivo de las fases del ciclo de vida del software.	
Fase	Descripción
Planificación	Donde se idea un planeamiento detallado que guía la gestión del proyecto temporal y económicamente.
Implementación	Donde se acuerda el conjunto de actividades que componen la realización del producto.
Puesta en producción	Donde el proyecto entra en la etapa de definición, se le presenta al cliente sabiendo que funciona correctamente y responde a los requerimientos solicitados en su momento. Se considera la etapa más importante no solo por representar la aceptación del usuario final sino por las múltiples dificultades que suele presentar en la práctica, alargándose excesivamente y provocando costos no previstos.
Inicio	Se definen los objetivos del proyecto y los recursos necesarios para su ejecución. Un alto porcentaje del éxito del proyecto se define en esta fase.
Control en producción	Donde se controla el producto, analizando como el proceso difiere o no de los requerimientos originales que puedan llegar a surgir en el ambiente productivo.
Fuente: Implementación y debugging, Cap. 1. Ciclo de Vida del Software.	

Anexo 8. Tabla 4. Cuadro descriptivo de las etapas del ciclo de vida del software.

Cuadro descriptivo de las etapas del ciclo de vida del software.	
Etapas	Objetivo
Expresión de necesidades	Armar un documento en cual se reflejan los requerimientos y funcionalidades que ofrecerá al usuario el sistema a implementar.
Especificaciones	Formalizar los requerimientos; el documento obtenido en la etapa anterior se toma como punto de partida para esta etapa.
Análisis	Determinar los elementos que intervienen en el sistema a desarrollar, su estructura, relaciones, evolución temporal, funcionalidades, se tiene una descripción clara del producto que se va a construir, que funcionalidades aportará y que comportamiento tendrá.
Diseño	Definir en detalle entidades y relaciones de las bases de datos y seleccionar el lenguaje de programación a utilizar junto con el sistema gestor de bases de datos.
Implementación	Empezar a codificar algoritmos y estructuras de datos definidos en las etapas anteriores en el correspondiente lenguaje de programación o determinado gestor de bases de datos. En muchos proyectos se pasa directamente a esta etapa; son proyectos muy arriesgados que adoptan un modelo de ciclo de vida code & fix (codificar y corregir) donde se eliminan las etapas de especificaciones, análisis y diseño con la consiguiente pérdida de control sobre la gestión de proyecto.
Debugging	Garantizar que el software no contiene errores de diseño y codificación.
Validación	Verificar que el sistema cumple con los requerimientos expresados inicialmente por el cliente y que dan lugar al presente proyecto.
Evolución	Esta etapa es considerada como mantenimiento y evaluación asignando no solo el agregado de nuevas funcionalidades (evolución); sino la corrección de errores que surgen (mantenimiento).
Fuente: Implementación y debugging, Cap. 1. Ciclo de Vida del Software.	

Anexo 9. Figura 3-4. Etapas de los Ciclos de Vida del Software Tradicional y Liger.



Figura 2. Etapas del ciclo de vida del software tradicional.

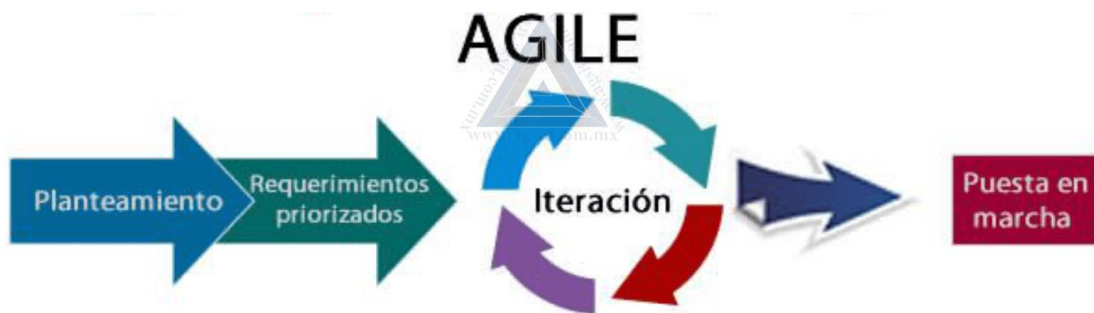
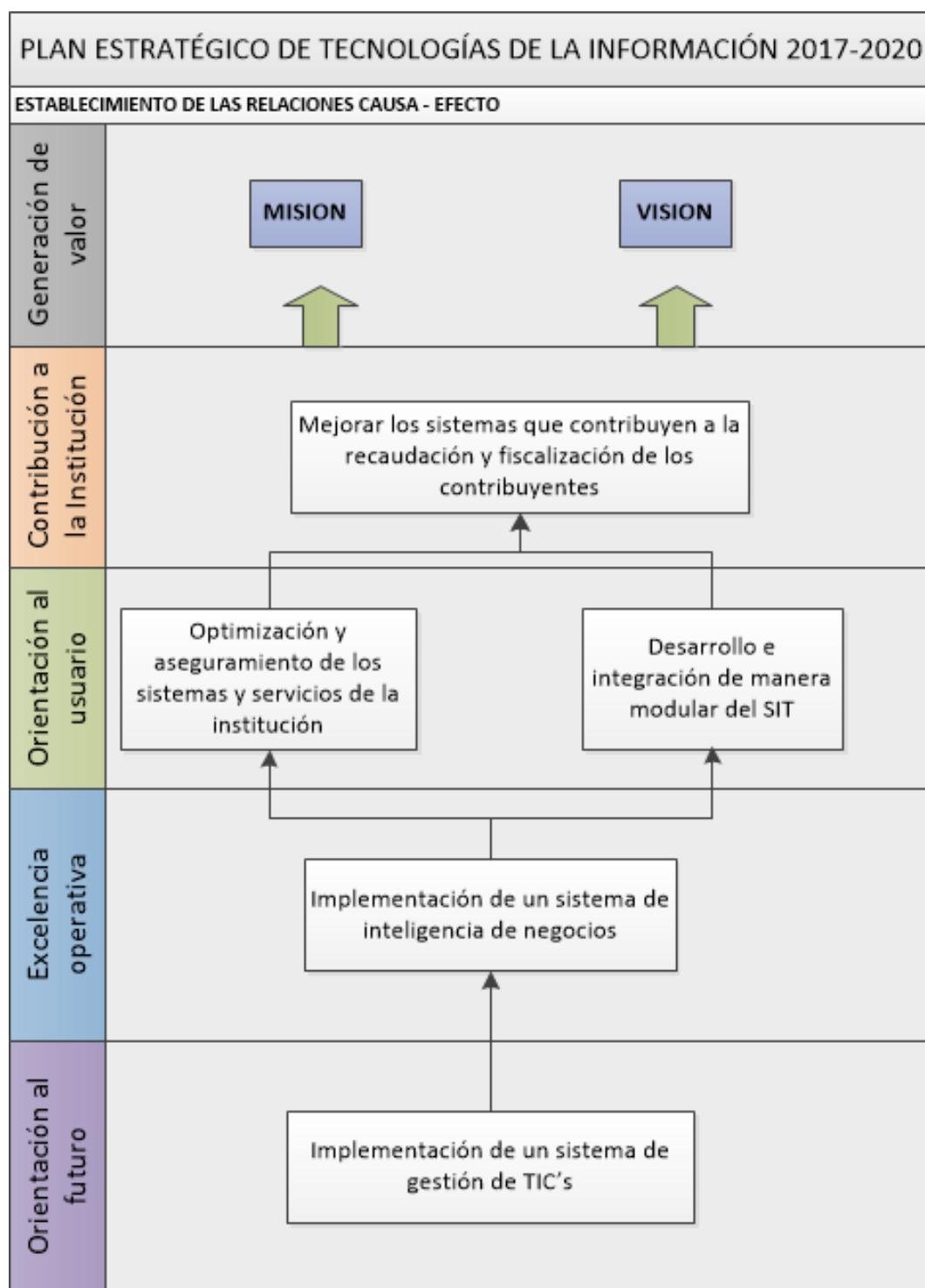


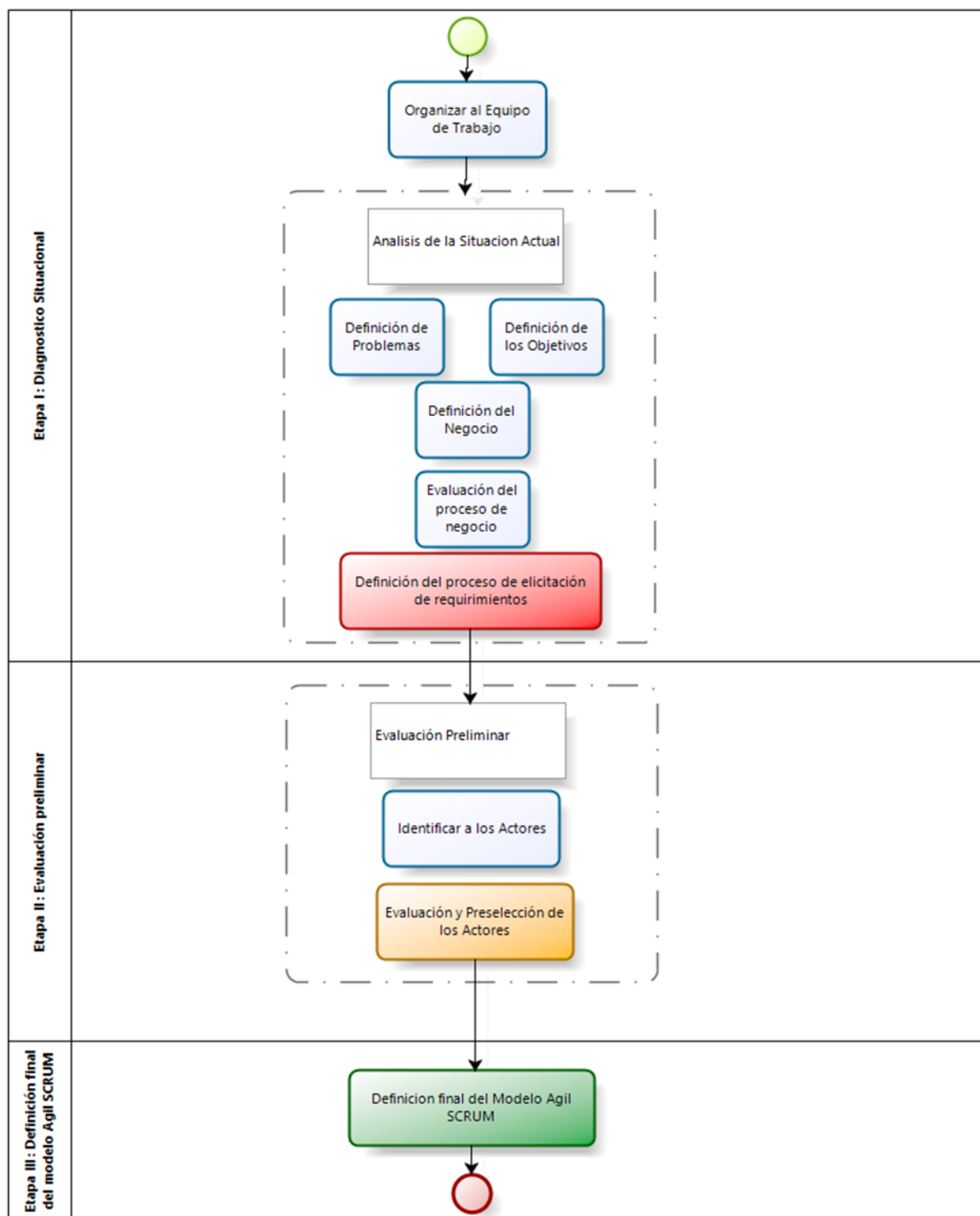
Figura 3. Etapas del ciclo de vida del software ligero.

Anexo 10 Figura 5. Mapa estratégico TIC de la División de Informática y sistemas.



Fuente: División de Informática y Sistemas de la Dirección General de Ingresos.

Anexo 11 Figura 6. Ruta de Implementación del Modelo de Procesos Ágil SCRUM.



Fuente: Elaboración Propia

5/Véase Tabla 6. Descripción de las diferentes metodologías tradicionales.

MODELOS	ETAPAS	VENTAJAS	DESVENTAJAS	APLICACIONES
CASCADA	1.-Diseño del Sistema 2.-Diseño del Programa 3.-Codificación 4.-Pruebas 5.-Implantación	• Es un modelo sencillo y disciplinado • Es fácil aprender a utilizarlo y comprender su funcionamiento • Está dirigido por los tipos de documentos y resultados que deben obtenerse al final de cada etapa	• Los proyectos raramente siguen el proceso lineal tal como se definía originalmente el ciclo de vida • Es difícil que el cliente exponga explícitamente todos los requisitos al principio • El cliente debe tener paciencia pues obtendrá el producto al final del ciclo de vida	El modelo de cascada es ampliamente utilizado por ejemplo de desarrollo de software grandes casas como los empleados por el Departamento de Defensa de EE.UU. y la NASA , y para muchos proyectos gubernamentales de gran tamaño
	NIVEL 1 está orientado al cliente. El inicio del proyecto y el fin del proyecto constituyen los dos extremos del ciclo. Se compone del análisis de requisitos y especificaciones, se traduce en un documento de requisitos y especificaciones. NIVEL 2 se dedica a las características	La relación entre las etapas de desarrollo y los distintos tipos de pruebas facilitan la localización de fallos • Es un modelo sencillo y de fácil	• Es difícil que el cliente exponga explícitamente todos los requisitos • El cliente debe tener paciencia pues obtendrá el producto al	Es utilizado por grandes empresas algunas crean sus propios modelos V

MODELO V	funcionales del sistema propuesto. NIVEL 3 define los componentes hardware y software del sistema final, a cuyo conjunto se denomina arquitectura del sistema. NIVEL 4 es la fase de implementación, en la que se desarrollan los elementos unitarios o módulos del programa.	aprendizaje • Hace explícito parte de la iteración y trabajo que hay que revisar. • Especifica bien los roles de los distintos tipos de pruebas a realizar • Involucra al usuario en las pruebas	final del ciclo de vida • Las pruebas pueden ser caras y, a veces, no lo suficientemente efectivas • El producto final obtenido puede que no refleje todos los requisitos del usuario	
MODELO ITERATIVO	• Etapa de inicialización Se crea una versión del sistema. La meta de esta etapa es crear un producto con el que el usuario pueda interactuar, y por ende retroalimentar el proceso. • Etapa de iteración Esta etapa involucra el rediseño e implementación de una tarea de la lista de control de proyecto, y el análisis de la versión más reciente del sistema. • Listas de control del proyecto	• Disminuyen riesgos • Fácilmente cambian los requerimientos • Reduce costos • Al usuario se le entrega parte del producto es decir una versión con la cual él puede trabajar	• Hasta el momento se puede decir que no existen grandes desventajas, pero si hay puntos a manejar con sumo cuidado porque su proceso de desarrollo es largo ya que en cada iteración se va actualizando.	

MODELO EN ESPIRAL	•Comunicación con el cliente: las tareas requeridas para establecer comunicación entre el desarrollador y el cliente. •Planificación: las tareas requeridas para definir recursos, el tiempo y otras informaciones relacionadas con el proyecto. Son todos los requerimientos. •Análisis de riesgos: las tareas requeridas para evaluar riesgos técnicos y otras informaciones relacionadas con el proyecto. •Ingeniería: las tareas requeridas para construir una o más representaciones de la aplicación. •Construcción y adaptación: las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.	• Incorpora muchas de las ventajas de los otros ciclos de vida • Conjugla la naturaleza iterativa de los prototipos con los aspectos controlados y sistemáticos del modelo clásico • Proporciona el potencial para el desarrollo rápido de versiones incrementales • Puede adaptarse y aplicarse a lo largo de la vida del software. • Es un enfoque realista del desarrollo del software • Permite aplicar el enfoque de construcción de prototipos en cualquier momento	• Puede resultar difícil convencer a algunos clientes de que el enfoque evolutivo es controlable • Solo resulta aplicable para proyectos de gran tamaño • Supone una carga de trabajo adicional, no presente en otros ciclos de vida • Requiere una considerable habilidad para la evaluación y resolución del riesgo, y se basa en esta habilidad para el éxito. • Si un riesgo importante no es descubierto y gestionado, indudablemente surgirán problemas • Es bastante complicado de realizar	Es el más utilizado para la creación de grandes sistemas
--------------------------	---	--	---	---

	•Evaluación del cliente: las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementación durante la etapa de instalación.	para reducir riesgos • Reduce los riesgos antes de que se conviertan en problemáticos	y su complejidad puede incrementarse hasta hacerlo impracticable	
MODELO EN PROTOTIPOS	1.- Investigación preliminar. 2.- Definición de los requerimientos del sistema. 3.- Diseño técnico 4.- Programación y prueba 5.- Operación y mantención	• Permite la construcción del sistema con requisitos poco claros o cambiantes • El cliente recibe una versión del sistema en muy poco tiempo, por lo que lo puede evaluar, probar e, incluso, empezar a utilizarlo • Se pueden introducir cambios en las funcionalidades del sistema en cualquier	• El cliente puede quedar convencido con las primeras versiones y, quizás, no vea la necesidad de completar el sistema o rediseñarlo con la calidad necesaria • Requiere trabajo del cliente para evaluar los distintos prototipos y traducirlo en nuevos requisitos • Requiere un tiempo adicional para definir adecuadamente el	Este modelo es útil cuando: El cliente no identifica los requisitos detallados. El responsable del desarrollo no está seguro de la eficiencia de un algoritmo, sistema operativo o de la interface hombre-máquina.

		momento • Involucra al usuario en la evaluación de la interfaz de usuario	sistema	
--	--	--	---------	--

6/Véase Tabla 7. Descripción de las diferentes metodologías Ágiles

MODELOS	ETAPAS	VENTAJAS	DESVENTAJAS	APLICACIONES
RAP	•Modelado de gestión •Modelado de dato •Modelado de proceso •Generación de aplicaciones •Pruebas de entrega	Comprar puede ahorrar dinero en comparación con construir. Los entregables pueden ser fácilmente trasladados a otra plataforma. El desarrollo se realiza a un nivel de abstracción mayor. Visibilidad temprana.	Comprar puede ser más caro que construir . Costo de herramientas integradas y equipo necesario. Progreso más difícil de medir. Menos eficiente. Menor precisión científica.	Lo utilizan los desarrolladores más cuando son equipos híbridos.

		Mayor flexibilidad. Menor codificación manual.	Riesgo de revertirse a las prácticas sin control de antaño.	
RUD	1.-Fase de inicio 2-Fase de elaboración 3.-Fase de construcción 4.-Fase de transición	La ventaja principal es que se basa todo en las mejores prácticas que se han implementado y se han probado en el campo.	Por su grado de complejidad puede no resultar muy adecuado.	
XP	1. El cliente define el valor de negocio a implementar. 2. El programador estima el esfuerzo necesario para su implementación. 3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo. 4. El programador construye ese valor de negocio. 5. Vuelve al paso 1.	Programación organizada. Menor tasa de errores. Satisfacción del programador.	Es recomendable emplearlo solo en proyectos a corto plazo. Altas comisiones en caso de fallar.	Adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.
SCRUM	El desarrollo de software se realiza Mediante iteraciones, denominadas Sprint, con una duración de 30 días. El resultado	Las Entregas Periódicas. Flexibilidad a	Requiere delegar responsabilidades al equipo, incluso permite fallar si es necesario.	Es óptima para equipos de trabajo de hasta 8 personas, aunque hay empresas que han

	de cada Sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 Minutos del equipo de desarrollo para coordinación e integración.	cambios Predicciones de tiempo Mayor productividad	• Es una metodología que difiere del resto, y esto causa cierta resistencia en su aplicación para algunas personas.	utilizado Scrum con éxito con equipos más grandes.
--	---	---	---	---

Fuentes: _____.

- Cáceres, P., Marcos, E., & Kybele, G. (2001). Procesos ágiles para el desarrollo de aplicaciones Web.
- Dolado, J., & Rodríguez, D. (2011). Utilidad de los procesos ágiles en el desarrollo de software.
- Gómez, O. T., López, P. P. R., & Bacalla, J. S. (2014). Criterios de selección de metodologías de desarrollo de software.
- Nuño, C., & Fraquoso, H. (2014). Adopción de metodologías ágiles de desarrollo.

Anexo 13. Diagrama Causa y Efecto

