



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

Tesis para la obtención del grado de

Maister

Informática Empresarial

Estudio del impacto de la aplicación de metodologías de software en el proceso de análisis, diseño, codificación e implementación en los Sistemas Tributarios de la División de Informática y Sistemas de la Dirección General de Ingresos

Elaborado por: Carrillo Torres, Yaritza Cristina

Tutora de la Tesis: Gómez Guardián, Mariela Ninette

Managua, Nicaragua 2019

Contenido

Dedicatoria	i
Agradecimientos	ii
Resumen	iii
Introducción.....	3
Antecedentes.....	5
Planteamiento de la situación	7
Objetivos.....	9
General	9
Específicos.....	9
Justificación.....	10
Capitulo I. Marco Teórico.	11
1. Proyecto.....	11
1. Software.....	12
2. Sistemas de información (SI)	13
3. Proceso Software (PS).....	14
4. Ciclo de vida.....	14
5. Metodologías de desarrollo	15
5.1 Tipos de metodologías.....	17
5.2 Estudio comparativo de metodología	18
5.3 Selección de una metodología.	19
5.4 Ingeniería de Software.....	20
8. Calidad de software	20
8.1 Clasificación de la Calidad en el Producto de Software	21
8.2 Estándares para la calidad del software.	22
8.3 Evaluación de la Calidad en el Software	22
8.4 Métricas de calidad del software	23
8.5 Relación de la calidad con la eficiencia, la efectividad y la eficacia	23
8.6 Estándares para el Ciclo de Vida del Software	24
8.7 Aseguramiento de la Calidad de Software	24
9. Diseño metodológico del trabajo	25
9.1 Diseño de la investigación.....	25
9.2 Población y muestra	25

9.3 Diseñar y aplicar instrumentos	26
9.4. Metodología MIGME-RCC.....	28
Capítulo II. Diagnóstico del estado actual de la DIS-DGI	32
2.1 Situación actual.	32
2.1.1 Estructura organizacional.	32
2.1.2 Análisis de los involucrados	32
2.1.3 Descripción del sistema de negocio.	33
2.1.4 Diagramas de procesos	33
2.1.6 El árbol de problemas	37
2.1.7 Análisis del impacto	38
Capítulo III. Propuesta de solución para la mejora del proceso de software.	70
3.1 Análisis de alternativas	70
3.2 Diagramas del proceso propuesto	73
Conclusiones.....	76
Recomendaciones	78
Glosario de términos.....	79
Bibliografía	83
Apéndice	89
A: Evaluaciones	89
B: Estructura organizacional	94
C: Otros	97
Anexos	103
A: Ingeniería de software	103

Dedicatoria

Mi tesis la dedico con todo mi amor y cariño, a ti DIOS que me diste la oportunidad de vivir y de regalarme una familia maravillosa.

A mis padres naturales y espirituales por siempre creer en mí y empujarme a ser mejor persona y siempre luchar por mis sueños.

A mis mejores amigos por su apoyo incondicional, en todo momento.

A la Dirección General de Ingresos que me dio el privilegio de realizar este estudio.

“Lo que no se define, no se puede medir.
Lo que no se mide, no se puede mejorar.
Lo que no se mejora se degrada siempre”
Lord Kevin: William Thomson.

El corazón humano genera muchos proyectos,
pero al final prevalecen los designios del Señor.
(Proverbios 19:21 | NVI)

Agradecimientos

Dios, tú amor y bondad no tienen fin, me permites sonreír ante todos mis logros que son resultado de tu ayuda, y de cada prueba que he atravesado me doy cuenta de cuán infinita es tu misericordia, que la vida es un milagro y si respiro es con un propósito; gracias, porque en cada paso que doy tu mano me sostiene y no podría ser de otro modo, eres tú quien me permites alcanzar victorias.

Gracias a mi madre por estar dispuesta a acompañarme cada larga y agotadora noche de estudio, en las que su compañía y sus cafés era para mí como agua en el desierto; gracias a mi padre por siempre desear y anhelar lo mejor para mí, por cada consejo y por cada una de sus palabras.

Este trabajo se ha realizado con el apoyo de la Dirección General de Ingresos - División de informática y Sistemas quien ha creído en mí como profesional y ha edificado tantas cosas en mí.

Quiero agradecerle a mi tutora por cada detalle y momento dedicado para aclarar cualquier tipo de duda, por la ayuda y exactitud de sus respuestas.

Resumen

Según la - ley de simplificación de trámites y servicios en la administración pública de la República de Nicaragua, LEY No. 691, Aprobada el 30 de Julio del 2009. La economía del país requiere mayor dinamismo en sus actividades productivas, lo que implica mayor agilización de parte de la Administración Pública que regula y participa en los trámites solicitados por los usuarios.

La Dirección General de Ingresos, DGI, con el objetivo de modernizar los servicios de gobierno electrónico y que haya mejor seguridad y rapidez en los Trámites Fiscales, de forma continua avanza en dicha modernización y rediseño de sus procesos tecnológicos para potenciar la gestión administrativa del Estado.

El estudio realizado analiza el proceso de desarrollo de software que se lleva a cabo en la División de Informática y Sistemas DIS de la DGI específicamente en la Unidad de Sistemas Tributarios UST, con la finalidad de identificar el impacto del uso de una metodología de software dentro del ciclo de vida de los sistemas desde la concepción del requerimiento hasta su implementación.

El desarrollo de software seguro y de calidad es de suma importancia para la institución como tal, dado que al automatizar los procesos se crea una dependencia en las aplicaciones para la operación normal del giro del negocio requiriendo una alta disponibilidad y seguridad de la información para garantizar un mejor servicio.

Es por esto que se hace necesario implementar efectivamente metodologías de desarrollo de software aplicadas en cada fase del ciclo de vida: requisitos, diseño, codificación, pruebas e implementación y que a la vez tenga incluida la parte de seguridad. Por lo que para este estudio se compararon las diferentes metodologías de desarrollo de software utilizadas o no dentro de la DIS, versus los resultados obtenidos y se definió pautas para evaluar la calidad de los sistemas basándose en este estudio y estándares existentes.

Introducción

La Dirección General de Ingresos (DGI) de la República de Nicaragua, es una institución descentralizada con autonomía administrativa y financiera, cuyo objeto es aplicar y hacer cumplir las leyes, actos y disposiciones que establecen o regulan los ingresos a favor del Estado, que están bajo su jurisdicción.

Su visión es ser una Administración Tributaria moderna, respetada y calificada, al servicio del pueblo nicaragüense.

Con el apoyo de las tecnologías de información, la Dirección General de Ingresos ha avanzado automatizando sus trámites y gestiones tributarias siendo su principal interés brindar un mejor servicio a través de los diversos sistemas, invirtiendo en nuevas plataformas de software y hardware para lograrlo.

La presente investigación se llevará a cabo en la División de Informática y Sistemas (DIS) que se encarga de administrar las tecnologías de información, está conformada por la Oficina de Sistemas Informáticos (OSI) y la Oficina de Apoyo Tecnológico (OAT); la OSI a su vez, en la Unidad de Sistemas Tributarios (UST), Unidad de Sistemas de Apoyo (USA), Unidad de Contacto y Control de Calidad (UCCC); la OAT en Unidad de Bases de Datos y Sistemas Operativos, Unidad de Comunicaciones, Unidad de Soporte Técnico, Unidad de Apoyo de los Sistemas Informáticos (GASI).

La Unidad de Sistemas Tributarios es la encargada del desarrollo de aquellos sistemas relacionados a registrar y procesar información referente a los tributos del Estado y de dar respuesta a las solicitudes de las áreas administrativas relacionadas a dicha información y tareas afines.

La Unidad de Bases de Datos es la encargada de configurar los diversos ambientes para los sistemas desarrollados, sus funciones abarcan monitoreo del servidor y cuestiones de bases de datos como la ejecución de script y procedimientos en los

ambientes del cliente como producción y cuando es requerido se apoya la unidad de sistemas operativos y redes.

La UCCC, por su parte, es la encargada de garantizar el funcionamiento de los sistemas y las correcciones de inconsistencias a través de la realización de diversas pruebas y recepción de bitácoras y casos reportados por las rentas o usuarios del sistema.

Para el desarrollo de la investigación se utilizó el método análisis y síntesis, posterior un estudio holístico del problema. Como técnica de recolección de datos se utilizaron encuestas y entrevistas para describir los procesos realizados en el desarrollo de sistemas informáticos también listas de verificación basada en estándares para evaluar el proceso y el producto de software.

El propósito de este estudio es analizar el impacto que tiene utilizar metodologías de software en el ciclo de vida del producto y los resultados obtenidos en los Sistemas Tributarios evaluando las partes medulares de dicho proceso en la UST y UCCC desde la concepción del proyecto hasta la implementación del producto en los diversos entornos o ambientes; esto desde la perspectiva del proceso y producto de software en la UST y UCCC.

Se pretende primeramente, abordar el contexto actual de la DIS, conocer su proceso de desarrollo actual, sus mejores prácticas, las metodologías que utilizan y los resultados obtenidos; para luego valorar el tipo de metodología más conveniente, según la naturaleza de los proyectos, realizar una comparativa entre éstas y luego de definir qué es la calidad del software desde la perspectiva de la UCCC como un usuario que evalúa el trabajo de la UST, encontrar el grado de calidad de los sistemas tributarios, para finalmente proponer una lista de verificación para el área de control de calidad que le permita llevar un mejor control de aspectos de calidad de los sistemas.

Antecedentes

En los últimos años, Nicaragua ha avanzado sustancialmente en el ámbito de la ciencia y la tecnología. (Brahima Sanou, 2018); Esto gracias al impulso de programas y políticas de gobierno y cooperación internacional (DGI, 2012-2018) que han sido aplicadas en sectores importantes como la DGI la cual ha avanzado con respecto al 2010, según (DGI, 2016) y (GRUN, 2014) en cuanto a modernización de la administración pública utilizando TIC.

El Estado está mejor estructurado y modernizado; este esfuerzo va acompañado de la aplicación apropiada de las TIC en todas sus actividades, sobre todo en lo referente a su uso en la implementación de la Estrategia Nacional de Desarrollo.

Según (DGI, 2016) *“La... (DGI)... aplicando estrategias para controlar a los contribuyentes mediante el uso de las Tic, permite altos niveles de efectividad con el uso de la información en la fiscalización y la cobranza, garantiza de la sensación de riesgo, disminución de la evasión fiscal y del aumento del cumplimiento voluntario en función de una mayor recaudación”.*

El desarrollo de software, es uno de los sectores tecnológicos más competitivos que ha tenido una evolución constante en lo que se refiere a las metodologías o bien, las formas en las cuales se realiza la planeación para el diseño del software, básicamente con el objetivo de mejorar, optimizar procesos y ofrecer una mejor calidad.

“La ciencia, la tecnología y la innovación (CTI) son reconocidas comúnmente como motores centrales de crecimiento económico sostenido a largo plazo. Las innovaciones tecnológicas, es decir, el surgimiento de productos, servicios y procesos, nuevos o mejorados, se traducen en un incremento del nivel de vida de los individuos y una mayor competitividad de las empresas y los países. Con la

evidencia empírica micro y macroeconómica se muestra que la CTI tiene un efecto positivo en el desempeño económico” (CONICYT, 2010)

En el contexto actual de la gestión de proyectos de desarrollo de software, es necesario disponer de herramientas para monitorear y evaluar el avance y el desempeño de los proyectos, así como sus resultados y el impacto.

La División de Informática y Sistemas de la Dirección General de Ingresos ha encaminado su plan estratégico (2017-2021) en brindar soluciones que sostengan de forma automatizada el giro del negocio y la alta disponibilidad de sus servicios a la población y usuarios en general, por lo que ha ido incluyendo nuevos desarrollos, nuevas tecnologías y mejora continua de los procesos. Se han implementado sistemas para llevar un control de procesos internos, como el de bitácoras y el de requerimientos como partes de una metodología de trabajo, además se han impulsado diversos programas de capacitación, cursos y maestrías relacionados a informática, buenas prácticas de programación, seguridad, gestores y administración de bases de datos y servidores, entre otros.

Planteamiento de la situación

La incorporación de tecnologías de la información en la Dirección General de Ingresos nace de la visión de ser una administración ágil e innovadora tomando una parte muy importante en el plan estratégico de la DIS la cual está preocupada por lograr las metas, automatizar los procesos con calidad en sus sistemas y servicios proporcionando tanto a contribuyentes como a la administración tributaria la disponibilidad y buen funcionamiento de los mismos.

El uso de una metodología depende de las características propias de cada proyecto y de su complejidad actualmente se desarrolla en diversos lenguajes tanto a nivel de programación (C#, .NET, JavaScript, Java, PHP, Natural de software AG, HTML) como bases de datos (SQL, MYSQL, PostgreSQL). El uso de un estándar, metodología o práctica va al juicio del analista de desarrollo, el cual elige el nombre de las tablas, clases, diseño del sitio; es decir, no se encuentra definido un estándar a seguir, dejándose abierto a los hábitos y experiencia del personal de TI

Los tiempos de entrega generalmente no se ajustan al planificado por diversos motivos entre los cuales se pueden observar: carga individual de trabajo, configuración de los ambientes, la experiencia del personal en determinado lenguaje de programación, metodologías, estándares, herramientas de desarrollo y su adaptación a un sistema existente; así como por el levantamiento de requerimientos realizado por diversos usuarios no especializados resultando documentos con diversas estructura a veces ambiguo, incompleto y cambiante, sin embargo hacen su mayor esfuerzo conforme identifican una necesidad, lo que repercute en su respectivo análisis, codificación y prueba por parte del analista de desarrollo y de control de calidad, llevando a un planteamiento irreal del tiempo de entrega y posibles fallos por los cambios cercanos a su entrega o escenarios no contemplados, mientras otros proyectos se estancan o no se implementan en producción por diversos motivos como cambios de leyes, prioridades u otros.

El personal tiene muchas asignaciones de tareas referidas al desarrollo de nuevos sistemas de tipo tributario y/o administrativo; así como el apoyo tecnológico y resolución de casos de las diferentes áreas solicitantes de la DGI, por lo que se asignan prioridades a determinados procesos y se interrumpe una tarea antes de que concluya, para comenzar otra de mayor prioridad lo cual afecta ciertas etapas del proyecto y al querer cumplir con lo planificado no se culminan algunas etapas como el análisis detallado del sistema, aumentando las horas de trabajo, reduciendo el periodo de pruebas y descuidando la documentación de los sistemas.

Algunos sistemas poseen poca documentación técnica, funcional y metodológica y por su nivel de complejidad a veces se dificulta encontrar un error. Aunque el manejo de errores, control de versiones y respaldos son un excelente apoyo para monitorearlos, el corregirlos depende del conocimiento y experiencia del analista desarrollador ya que un mínimo cambio sin conocer el sistema de forma integral puede provocar nuevos y mayores errores.

En cuanto a la problemática asociada a la adquisición de software es de necesidad que dicho proceso tome en consideración el riesgo de calidad del software, así como el real soporte a la funcionalidad de las áreas que éste pretende apoyar. De no ser así, se estará pagando por un servicio que no reporta beneficio y, por el contrario, podrá representar el inicio de una experiencia onerosa, no funcional y peligrosa para la viabilidad de la institución en términos de pérdidas por cuestiones que se salen del presupuesto.

Objetivos

General

- Estudiar el impacto de la aplicación de metodologías de software en el proceso de análisis, diseño, codificación e implementación en los Sistemas Tributarios de la División de Informática y Sistemas de la Dirección General de Ingresos.

Específicos

- Conocer el grado de cumplimiento y calidad de los sistemas tributarios en comparación a los estándares y metodología formales con el apoyo de herramientas de evaluación de calidad.
- Determinar los factores que influyeron en el éxito o fracaso en la implementación de sistemas.
- Determinar la importancia de utilizar una metodología de desarrollo.

Justificación

La DGI en su visión, que incluye el desarrollo tecnológico, pone especial énfasis en los sistemas tributarios debido al aporte de estos a la recaudación, agilidad de procesos y buen servicio; por lo que la calidad y uso eficiente de los recursos son necesarios estén presentes en cada desarrollo.

Actualmente la forma de desarrollo de software es individual o en equipo en dependencia del tiempo de entrega, carga laboral y la complejidad del proyecto. Cada desarrollador según su nivel de conocimiento utiliza o no una metodología de desarrollo lo que aporta valores diferentes a cada proyecto, sin embargo su evaluación de calidad está basada en funcionalidades del sistema más que en su estructura.

Debido a la naturaleza del negocio el tiempo de respuesta es fundamental y no se queda atrás la calidad de la información y de los sistemas; por lo que otro aspecto importante es el rendimiento, la fiabilidad, integridad, seguridad y la capacidad de mantenimiento del software.

Con el resultado de este estudio se pretende determinar los factores que influyeron en el éxito o fracaso en la implementación de Sistemas de Información y a su vez determinar la importancia de utilizar una metodología de desarrollo definida y funcional para incrementar la calidad de los sistemas desarrollados.

El análisis de todo proyecto debe incluir indicadores básicos de control como eficiencia de costos, productividad, entregas, tiempo de ciclo, tasas de error, etc. y tratar que los indicadores estén ligados a los objetivos del negocio. Es también importante incorporar modelos de escenarios en el análisis inicial para que pueda evaluarse diferentes opciones y extrapolar las consecuencias de diferentes cursos de acción en el tiempo, y luego seleccionar la mejor opción.

Capítulo I. Marco Teórico.

1. Proyecto.

Según la definición que nos proporciona PMI en su guía PMBOK (Garcíatorres, 2008) un proyecto se podría definir como *“un servicio temporal que se lleva a cabo para crear un producto, servicio o resultado único”*.

Las actividades son las tareas que deben ejecutarse para llegar en conjunto a un fin preestablecido (objetivo deseado). *Toda actividad debe estar basada en una metodología y en principio, cualquier metodología es mejor que ninguna.*

Un aspecto fundamental en todo proyecto es el orden en el cual se realizan las actividades. Y para determinar la secuencia lógica de las actividades se debe establecer el **método**, el **tiempo** y el **costo** de cada operación.

Ver anexos [Figura. N° 1].

Independientemente de su complejidad, característicamente todo proyecto reúne la mayoría de los siguientes criterios:

- Tener un principio y un fin.
- Tener un calendario definido de ejecución.
- Plantearse de una sola vez.
- Constar de una sucesión de actividades o de fases.
- Agrupar personas en función de las necesidades específicas de cada actividad.
- Contar con los recursos necesarios para desenvolver las actividades.

Factores de éxito

Según el Informe de (Lynch, 2015) los principales factores de éxito en la gestión de proyectos son los siguientes, Ver anexos [Figura. N°2].

- Apoyo a la dirección.
- Madurez emocional.
- Optimización.
- Personal calificado.
- Arquitecturas y marcos estándar.
- Procesos ágiles.
- Ejecución moderada y limitada.
- Experiencia en la gestión de proyectos.
- Objetivos claros del negocio

1. Software.

Según la (IEE, 1990) se define como “programas, procedimientos, documentación y datos asociados, relacionados con la operación de un sistema informático.

El software se puede definir como el conjunto de tres componentes (INTECO, 2009):



[Figura. N° 3] Componentes del software por: (INTECO, 2009).

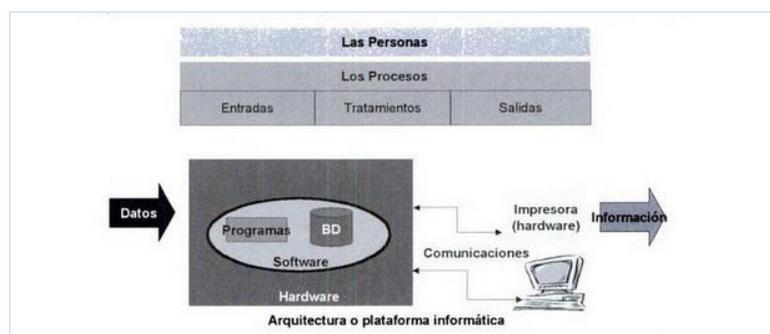
- **Programas:** Los programas son conjuntos de instrucciones que proporcionan la funcionalidad deseada cuando son ejecutadas por el ordenador. Están escritos usando lenguajes específicos que los ordenadores pueden leer y ejecutar.

- **Datos:** Los programas proporcionan la funcionalidad requerida manipulando datos. Usan datos para ejercer el control apropiado en lo que hacen. El mantenimiento y las pruebas de los programas también necesitan datos. El diseño del programa asume la disponibilidad de las estructuras de datos tales como bases de datos y archivos que contienen datos.
- **Documentos:** Además de los programas y los datos, los usuarios necesitan también una explicación de cómo usar el programa. Documentos como manuales de usuario y de operación son necesarios para permitir a los usuarios operar con el sistema. Los documentos también son requeridos por las personas encargadas de mantener el software para entender el interior del software y modificarlo, en el caso en que sea necesario (INTECO, 2009).

2. Sistemas de información (SI)

Un SI es un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan, elaboran y distribuyen la información (o parte de ella) necesaria para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes (decisiones) para desempeñar su actividad de acuerdo a su estrategia de negocio (CUARTAS, 2008).

Los elementos que forman un SI son:



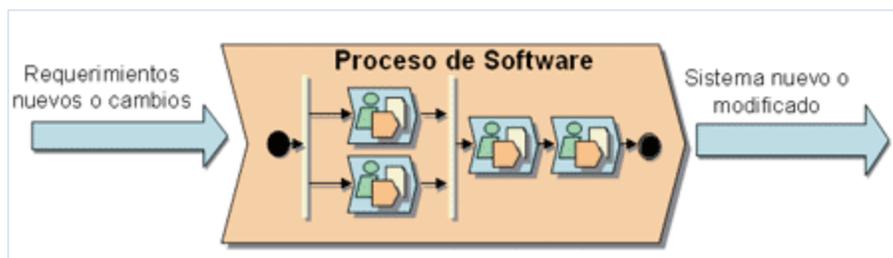
[Figura. N°4] Estructura o plataforma informática. Por: (Fuente, 2004).

Forman parte de la estructura los siguientes elementos (Fuente, 2004)

- Los datos
- La información
- El hardware
- El software
- Las bases de datos
- Las redes de comunicaciones
- Las personas
- Internet, intranet y extranet
- Los procesos para construir sistemas de información.
- Las metodologías para la construcción de sistemas de información.

3. Proceso Software (PS)

Es un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software. Ver también anexos [Figura. N°28] donde se verán los elementos relacionados al proceso.



[Figura. N° 5] Proceso de software. Por: (Ruvalcaba, 2019)

4. Ciclo de vida

La norma (ISO/IEC/IEEE 12207:2008(E), 2008) entiende por ciclo de vida "un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto software,

abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso". Ver Anexos [Figura. N°6].

5. Metodologías de desarrollo

Una definición representativa del concepto podría ser la de (R.N, 1983), que define Metodología como "un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentos y aspectos de formación para los desarrolladores de sistemas de información".

Según esto, una metodología es un conjunto de componentes que especifican:

- Cómo se debe dividir un proyecto en etapas.
- Qué tareas se llevan a cabo en cada etapa.
- Qué salidas producen y cuándo se deben producir.
- Qué restricciones se aplican.
- Qué herramientas se van a utilizar.
- Cómo se gestiona y controla un proyecto.

Esta descomposición del proceso de desarrollo guía a los desarrolladores en la elección de las técnicas que debe elegir para cada estado del proyecto, y facilita la planificación, gestión, control y evaluación de los proyectos. Una metodología, por tanto, representa el camino para desarrollar software de una manera sistemática.

Un proceso estándar en la organización aporta los siguientes beneficios:

- Registrar los requisitos de un sistema de información de forma acertada.
- Proporcionar un método sistemático de desarrollo de forma que se pueda controlar su progreso.
- Construir un sistema de información dentro de un tiempo apropiado y unos costos aceptables.
- Construir un sistema que esté bien documentado y que sea fácil de mantener.

- Ayudar a identificar lo más pronto posible cualquier cambio que sea necesario realizar dentro del proceso de desarrollo.
- Proporcionar un sistema que satisfaga a todas las personas afectadas por el mismo, ya sean clientes, directivos, auditores o usuarios.

Una metodología puede seguir uno o varios modelos de ciclo de vida. El ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto. La metodología indicaría el cómo.

Diversos autores coinciden en señalar algunos requisitos que deben tener las metodologías de desarrollo:

- Visión del producto.
- Vinculación con el cliente.
- Establecer un modelo de ciclo de vida.
- Gestión de los requisitos.
- Plan de desarrollo.
- Integración del proyecto.
- Medidas de progreso del proyecto.
- Métricas para evaluar la calidad.
- Maneras de medir el riesgo.
- Cómo gestionar los cambios.
- Establecer una línea de meta.

5.1 Tipos de metodologías

(Cantone, 2006) menciona que existen dos clasificaciones de metodologías que tienen analogía en la práctica con los paradigmas de programación. Estas son:

➤ Metodologías Estructuradas

Estas metodologías están orientadas hacia los procesos que intervienen propiamente en el sistema a desarrollar, es decir, se dirige a cada función a realizar por el sistema y que por lo general se descompone o se subdivide en módulos individuales más pequeños. Es importante recordar que es más sencillo atacar problemas pequeños haciendo referencia al concepto de divide y vencerás, para luego tomar esas soluciones y resolver el problema general.

➤ Metodología Orientada a Objetos.

Estas metodologías a diferencia de las anteriores, no se basan en los procesos como funciones sino que considera módulos basados en componentes (elemento de un sistema que ofrece un servicio predefinido, y es capaz de comunicarse con otros componentes), es decir, cada componente es independiente del otro. Esta separación por componentes permite que el código sea reutilizable, o sea, que se pueda usar en más de un sistema; permitiendo además que el mantenimiento del mismo sea más fácil debido a que los cambios están localizados en cada uno de los componentes.



[Figura. N°7]. Paradigmas de programación. Por: Cantone, Implementación y Debugging, 2006, p.20.

5.2 Estudio comparativo de metodología

Según las fuentes consultadas, se exponen distintos cuadros de estudio comparativo de las dos principales metodologías, las tradicionales y las ágiles.

Según lo expuesto hasta el momento, no se puede afirmar categóricamente que una metodología sea mejor que otra, ya que los resultados del éxito o fracaso del proyecto dependerán de si se ha elegido una metodología adecuada para su desarrollo, si se han empleado correctamente las herramientas de gestión, si se ha realizado una buena gestión por parte del jefe de proyectos o si el equipo ha estado motivado y concentrado en el trabajo.

4.2.1 Agiles y tradicionales

Según (Navarro Cadavid, Fernández Martínez, & Morales Vélez, 2013) las metodologías tradicionales buscan imponer disciplina al proceso de desarrollo de software y de esa forma volverlo predecible y eficiente. Para conseguirlo se soportan en un proceso detallado con énfasis en planeación propio de otras ingenierías. El principal problema de este enfoque es que hay muchas actividades que hacer para seguir la metodología y esto retrasa la etapa de desarrollo. Las metodologías ágiles tienen dos diferencias fundamentales con las metodologías tradicionales; la primera es que los métodos ágiles son adaptativos no predictivos. La segunda diferencia es que las metodologías ágiles son orientadas a las personas no orientadas a los procesos. *Ver anexos [Tabla N°19] y anexos [Tabla N°20] y [Figura N° 6].*

4.2.1.1 Metodologías tradicionales

Las metodologías tradicionales de desarrollo de software son orientadas por planeación. Inician el desarrollo de un proyecto con un riguroso proceso de

obtención de requerimientos, previo a etapas de análisis y diseño. Con esto tratan de asegurar resultados con alta calidad circunscritos a un calendario. En las metodologías tradicionales se concibe un solo proyecto, de grandes dimensiones y estructura definida; se sigue un proceso secuencial en una sola dirección y sin marcha atrás; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto, demandando grandes plazos de planeación previa y poca comunicación con el cliente una vez ha terminado ésta.

4. 2.1.2 Metodologías Ágiles

Las metodologías ágiles son flexibles, pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto.

Los proyectos ágiles se subdividen en proyectos más pequeños mediante una lista ordenada de características.

Las metodologías ágiles son flexibles, sus proyectos son subdivididos en proyectos más pequeños, incluyen comunicación constante con el cliente, son altamente colaborativos y se adaptan mejor a los cambios. De hecho, el cambio en los requerimientos es una característica esperada al igual que las entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente.

5.3 Selección de una metodología.

Utilizar una metodología de desarrollo que implemente uno y otro modelo depende en gran medida de la complejidad del proyecto de software a desarrollar.

Si el proyecto es muy simple, una metodología funcional puede resolverlo adecuadamente, pero si es complejo, habría que utilizar una metodología orientada a objetos (Alonso Amo, 2005).

Normalmente, la complejidad de un proyecto informático está íntimamente relacionada con la dimensión del código fuente producido.

A este respecto, E. N Yourdon realiza la siguiente clasificación:

Dimensión del Proyecto	Nivel de Complejidad
< 1.000 líneas de código fuente	Trivial
< 10.000 líneas de código fuente	Simple
< 100.000 líneas de código fuente	Difícil
< 1.000.000 líneas de código fuente	Complejo
< 10.000.000 líneas de código fuente	Casi imposible
< 100.000.000 líneas de código fuente	Absurdo

[Figura. N° 8] Dimensiones del proyecto. Por (Alonso Amo, Loïc, & Segovia Pérez, 2005).

5.4 Ingeniería de Software

La Ingeniería del Software es, *ver anexos [Figura N°9]:*

- El estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas de software (Zelkovitz, 1978).
- La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software: es decir, la aplicación de ingeniería al software (IEEE, 1993).
- Una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema hasta el mantenimiento de éste después que se utiliza (Sommerville, 2004).
- Una disciplina que integra el proceso, los métodos, y las herramientas para el desarrollo de software de computadora (Pressman, 2005).

8. Calidad de software

Según (Pressman, 2002) la calidad de software como “la concordancia del software con los requisitos explícitamente establecidos, con los estándares de desarrollo expresamente fijados y con los requisitos implícitos, no establecidos formalmente pero que desea el usuario”.

De acuerdo a la definición del Instituto de Ingenieros Eléctricos y Electrónicos (IEE, 1990) “La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”.

Los factores que determinan la Calidad en el Software son:

- Corrección. ¿Hace lo que quiero?
- Fiabilidad. ¿Lo hace de forma fiable todo el tiempo?
- Eficiencia. ¿Se ejecutará en mi hardware lo mejor que pueda?
- Seguridad (Integridad). ¿Es seguro?
- Facilidad de uso. ¿Está diseñado para ser usado

8.1 Clasificación de la Calidad en el Producto de Software

La norma (IRAM-ISO/IEC9126-1) Modelo de Calidad, clasifica la calidad en interna, externa y calidad en uso:

- La **Calidad Interna** se define como la totalidad de los atributos de un producto que determinan su habilidad para satisfacer los requerimientos y necesidades especificados explícita e implícitamente, en un determinado contexto de uso. Se define/mide/logra cuando el software es implementado, utilizando métricas internas.
- La **Calidad Externa** se define como el grado en que un producto satisface los requerimientos y necesidades especificados explícita e implícitamente en un determinado contexto de uso. Se mide cuando el software es ejecutado, en un ambiente simulado, utilizando métricas externas. *Ver anexos [Figura. N°10].*
- La **Calidad en Uso** es la capacidad del producto de software de permitir que los usuarios logren sus objetivos con eficacia, productividad, satisfacción y seguridad, en los contextos de uso correspondientes. Esta

calidad es el objetivo general de calidad desde la perspectiva del usuario.

8.2 Estándares para la calidad del software.

Un estándar (Acosta, Espinel, & García, 2017) es una serie de recomendaciones a seguir para la entrega de un producto, dentro de la calidad del software es una metodología para que los entregables cumplan con las expectativas del negocio.

8.3 Evaluación de la Calidad en el Software

Según ISO/IEC25000 (2005), la calidad del producto no solamente debe ser vista como una actividad exclusiva de la fase de pruebas de software, sino también debe considerar la evaluación de los entregables del producto durante su ciclo de vida.

8.3.1 Elementos que permiten evaluar la calidad en el software

(McCall, 1977) Propuso una clasificación de los factores que afectan directamente a la calidad del software. Estos factores se muestran en *anexos [Figura. N°11] de los anexos*. En ella se concentran tres aspectos importantes de un software:

- Características operativas.
- Capacidad para experimentar cambios.
- Capacidad para adaptarse a nuevos entornos.

8.3.2 Verificación y Validación de Software

Un aspecto importante a tener en cuenta son las verificaciones y validaciones como parte del ciclo de vida en el proceso de desarrollo de software, estas se utilizan para detectar fallas durante las etapas que conforman este ciclo.

8.4 Métricas de calidad del software

El uso de las métricas establece cómo se debe ajustar el software a los requisitos implícitos y explícitos del cliente. Es decir, la medición para que el sistema se adapte a los requisitos establecidos, las métricas de calidad de sistemas de información utilizados para evaluar y controlar el proceso de desarrollo del software deben permitir:

8.5 Relación de la calidad con la eficiencia, la efectividad y la eficacia

La definición de calidad sistémica en el desarrollo de los sistemas de software contempla la Matriz de Calidad Global Sistémica. Ver anexos [Figura. N°12].

Se definen cada uno de los componentes de la matriz de la siguiente manera:

- Eficiencia del Producto: Es determinada por actividades de diseño interno y programación, ya que un producto eficiente es conseguido cuando se aplican las prácticas correctas de diseño físico y programación.
- Efectividad del Producto: Es determinada por las actividades de identificación de requerimientos, diseño de interfaces y diseño general de la red (ubicación de puntos), debido a que la misma está relacionada con la adecuación y confort del usuario.
- Eficiencia del Proceso: Está asociada con las actividades de gerencia de proyectos, las cuales incluyen el cumplimiento de fechas de entrega, aumento de la productividad y ahorro de recursos.
- Efectividad del Proceso: Se relaciona con las actividades generales de gerencia, tales como liderazgo, administración de cambio, relaciones humanas y grupales, ya que las mismas conducen a establecer buenas relaciones entre los integrantes del equipo responsable del desarrollo de Sistemas de Información.

8.6 Estándares para el Ciclo de Vida del Software

A continuación se mencionan y sintetizan estándares internacionales de evaluación del producto y mejora de procesos:

REQUERIMIENTOS	IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications IEEE Std 1063-2001 IEEE Standard for Software User Documentation
ANÁLISIS Y DISEÑO	IEEE Std 1471-2000 IEEE Recommended Practice for Architectural Description of Software-Intensive Systems
CONSTRUCCIÓN DEL SOFTWARE	ANSI/IEEE 1008-1987 IEEE Standard for Software Unit Testing NTP ISO/IEC 12207
PRUEBAS DEL SOFTWARE	NTP ISO/IEC 9126-1:2004 NTP ISO/IEC 9126-2:2004 NTP ISO/IEC 9126-3:2005
MANTENIMIENTO DEL SOFTWARE	IEEE Std 219-1998 IEEE Standard for Software Maintenance

[Figura. N°13]. Estándares para el ciclo de vida del software. (Morales, 2010)

8.7 Aseguramiento de la Calidad de Software

Según la norma (ISO 9000:2000), el aseguramiento de la calidad es la parte de la gestión de la calidad orientada a proporcionar confianza en que se cumplirán los requisitos de calidad.

El aseguramiento de la Calidad del Software engloba:

- Un enfoque de gestión de calidad.
- Métodos y herramientas de Ingeniería del Software.
- Revisiones técnicas formales en el proceso del software.
- Una estrategia de prueba multiescala.
- El control de la documentación del software y de los cambios realizados.
- Procedimientos para ajustarse a los estándares de desarrollo del software.
- Mecanismos de medición y de generación de informes.

9. Diseño metodológico del trabajo

A continuación se desarrollan los pasos requeridos para definir el diseño metodológico de la presente investigación.

9.1 Diseño de la investigación

El enfoque de esta investigación es mixto (cuantitativo, cualitativo) debido a que se requiere determinar variables para medir el impacto y a su vez categorías de análisis; Con estudio de casos múltiples dado que hay varios tipos desarrollo (lenguajes de programación: C#, PHP, .NET, Java, etc.). Y el desarrollador es el que realiza el análisis, diseño y desarrollo del proyecto según las especificaciones del área tributaria solicitante por lo que se requiere un estudio de forma integral.

- Investigación bibliográfica de trabajos que correspondan con estudios similares.
- Determinación y procesamiento de datos recopilados de las diferentes normas estándares.
- Establecimiento de análisis comparativo por las diferentes métricas.
- Análisis de los datos obtenidos y verificación de los mismos
- Elaboración de los resultados, discusión y la conclusión.

9.2 Población y muestra

Si la población es menor a cincuenta (50) individuos, la población es igual a la muestra” (Castro, 2003) . Dado que el área de desarrollo UST de la DIS está conformada por 29 personas de las cuales 2 son líder de equipo, 1 jefe de área y 26 analistas de sistemas tributarios, se toma como muestra el universo de la población, sin embargo dos analistas no estuvieron presentes por lo que se aplicó la encuesta a 24 analistas; con respecto a los de control de calidad se aplicó a un total de 9 analistas.

Para determinar el grado de calidad del software como parte del proceso de evaluación se utilizó diferentes estándares de trabajo, y en particular los que estén más orientados a la DIS.

Se evaluará un total de 17 proyectos que por motivos de confidencialidad denominaremos con pseudónimo: Proyecto (n+1).

Como resultado se propusieron un conjunto de cinco indicadores y cuatro criterios de calidad (Muy buena, Buena, Aceptable y Deficiente) en función de los resultados de los indicadores para cada medida, Subcaracterísticas y característica de calidad, con el objetivo de brindar una valoración más completa y más clara de los resultados alcanzados

9.3 Diseñar y aplicar instrumentos

A continuación se muestran las fuentes de información utilizadas para recolección de la información del presente estudio.

Fuente Primaria:

- Encuestas
- Entrevistas
- Listas de verificación
- Análisis de los sistemas utilizando herramientas de software.

Población	Descripción	Fuente	Tipo de información	Procedimiento
Analista-Desarrollador	Es la persona parte del proceso de análisis, diseño, Codificación	Encuesta	Administración de requerimientos Metodologías, prácticas o estándares que utilizan para el ciclo de vida del software. Principales dificultades.	Elaborar contenido de la encuesta. La definición de objetivos Selección de la muestra

			Experiencia en el proyecto. Gestión de documentación	Elección del método de encuesta: recogida de datos Análisis e interpretación de los datos.
Jefe de sistemas tributarios, de unidad y subdirector DIS.	Son las personas encargadas de dirigir la DIS.	Entrevistas	Proyectos planificados vs ejecutados. Principales dificultades Políticas y normas internas establecidas con relación a este proceso.	Desarrollar entrevista. Planificar entrevista. Aplicar entrevista Análisis e interpretación de los datos.
Investigador (maestrante)	Se trata de persona como investigadora.	Observación directa e investigación documental	Describir el proceso de desarrollo de software. Revisar documentación relacionada y producto de cada etapa del ciclo de vida del software.	Observar el proceso del ciclo de vida del desarrollo de software con el fin de realizar un análisis e identificar cada una de las etapas que se llevan a cabo. Revisar la documentación del proceso de desarrollo de software del área.

[Tabla N°1] Instrumentos (Elaboración propia)

Fuente secundaria:

- Requerimientos
- Plan operativo
- Manuales
- Informes
- Guías
- Estándares
- Leyes
- Plan Estratégico
- Artefactos del software

9.4. Metodología MIGME-RCC

Este método de evaluación fue desarrollado como parte de una metodología de trabajo de tesis doctoral. Esta permite la implementación gradual y continuas mejoras en los procesos de software (Muñoz, Mejía, Calvo, Cuevas, & San Felio, 2012). Incluye:

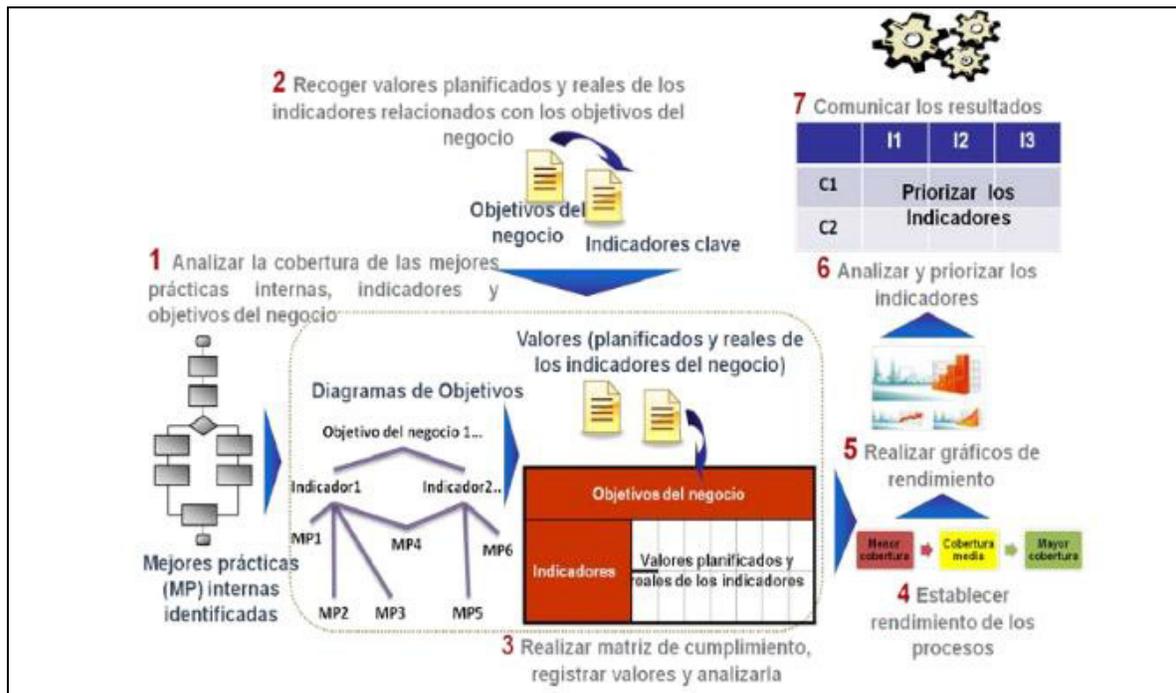
- Identificación de las mejores prácticas internas: actividades enfocadas en la extracción del conocimiento tácito y el análisis de la documentación formal de los procesos para identificar mejores prácticas.
- Evaluar el rendimiento de la organización: actividades enfocadas al rendimiento de los procesos a través del análisis de cobertura de mejores prácticas, los indicadores relacionados con los objetivos y los objetivos en sí mismo.
- Analizar las mejores prácticas externas: actividades enfocadas a establecer un entorno multimodelos que se utilizarán como referencia para la selección de mejores prácticas que se ajusten a la forma de trabajo y hagan procesos más eficientes.
- Implementar mejoras: actividades enfocadas al diseño de los nuevos procesos a través de la integración de mejores prácticas internas y externas seleccionados.



[Figura. N°14] Metodología MIGME-RCC. Por: (Muñoz, Mejía, Calvo, Cuevas, & San Felio, 2012)

9.4.1 Descripción de actividades del método de evaluación

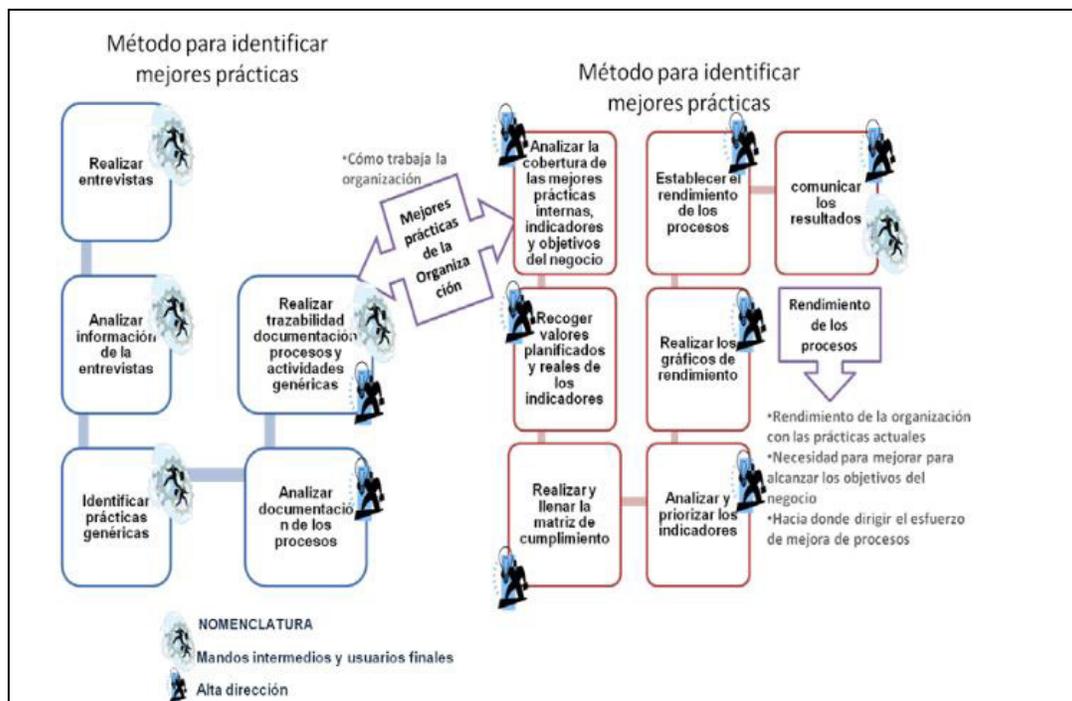
A continuación se describen brevemente las actividades del método de evaluación, incluyendo los productos a obtener, la técnica utilizada.



[Figura. N°15] Método de evaluación de rendimiento de procesos (Muñoz, Mejía, Calvo, Cuevas, & San Felio, 2012)

9.4.2 Analizar la cobertura de las mejores prácticas internas, indicadores y objetivos del negocio

El propósito de esta actividad es identificar las correspondencias entre las mejores prácticas identificadas, los indicadores y los objetivos del negocio mediante la realización de trazabilidad “bottom-up”. Cobertura por lo tanto, es el grado de correspondencias que existe entre mejores prácticas, indicadores y objetivos del negocio.



[Figura. N°16] Interacción de métodos e implicado. (Muñoz, Mejía, Calvo, Cuevas, & San Felio, 2012)

9.4.3 Recoger valores planificados y reales de los indicadores

- Valores planificados: la alta dirección tiene establecidos objetivos del negocio específicos a ser alcanzados.
- Valores reales: la fuente de esta información son las auditorías internas llevadas cabo en la organización.

9.4.4 Realizar y llenar la matriz de cumplimiento

El propósito de esta actividad es realizar y llenar la matriz de cumplimiento. La matriz registra los objetivos del negocio en las columnas e indicadores en las filas.

9.4.5 Establecer el rendimiento de los procesos

El propósito de esta actividad es establecer el cumplimiento de los indicadores mediante la comparación de los valores planificados y reales, mediante la aplicación de la fórmula: $\text{Cumplimiento} = \text{valor real} - \text{valor planificado}$

A través del análisis del porcentaje resultante se establece el grado de cumplimiento de los indicadores, y por lo tanto, de los objetivos del negocio. Además, para realizar este análisis se propone utilizar una escala de colores para resaltar los valores resultantes de acuerdo al grado de cobertura de tal manera que:

- Indicadores que han resultado con un mayor porcentaje son resaltados con color anaranjado. Esto indica que el objetivo del negocio tiene una menor cobertura.
- Indicadores que han resultado con un menor porcentaje son resaltados con verde. Esto indica que el objetivo del negocio tiene una mayor cobertura.
- Indicadores que han resultado con un porcentaje medio son resaltados con amarillo. Esto indica que el objetivo tiene una cobertura media

9.4.6 Realizar los gráficos de rendimiento

El propósito de esta actividad es realizar el o los gráficos que permitan una mejor comprensión del rendimiento de los procesos; estos gráficos permitirán resaltar la necesidad de la mejora de los procesos para el cumplimiento de los objetivos del negocio.

9.4.7 Analizar y priorizar los indicadores

El propósito de esta actividad es analizar los indicadores y elegir aquellos en los cuales se considere que existe una oportunidad de mejora y realizar una lista con estos indicadores.

9.4.8 Comunicar los resultados

El propósito de esta actividad es comunicar a los partes interesadas los resultados tanto del rendimiento de los procesos como de la priorización de los indicadores del negocio a ser mejorados.

Capítulo II. Diagnóstico del estado actual de la DIS-DGI

En este capítulo se realiza un análisis descriptivo de la situación actual de la División de Informática y Sistemas en cuanto a su organización, ciclo de desarrollo de software, identificación de problemas, diagnóstico del nivel de madurez del proceso y análisis de los resultados de encuesta y entrevistas realizadas.

2.1 Situación actual.

2.1.1 Estructura organizacional.

A continuación se muestra la estructura organizacional de la Dirección General de Ingresos; La cual está conformada por tres grupos: Divisiones de Apoyo, Asesoría y Apoyo a la Dirección general y Direcciones y Áreas Sustantivas. *Ver anexos [Figura. N°17] y [Figura. N°18].*

2.1.2 Análisis de los involucrados

Los interesados potenciales del proyecto, son aquellas personas o grupos que se ven afectados positiva o negativamente con la ejecución del mismo, en la siguiente tabla se listan los involucrados identificados con la finalidad de identificar las expectativas, intereses e influencia de cada uno de ellos.

Grupo	Intereses	Problemas percibidos	Mandatos y recursos
Contribuyente.	Disponibilidad de los servicios proporcionados por DGI.	Los sistemas presentan inconsistencias	Quejas y reclamos
Funcionarios usuarios de los sistemas	Realizar su trabajo haciendo uso de los sistemas de forma que se simplifique.	Atrasos en algunos procesos	Quejas y reclamos
Área Solicitante de software	Cumplir metas objetivos por lo cual creó un requerimiento de software. Facilitar su trabajo	Retrasos en la entrega de algunas solicitudes debido a la atención de otras de mayor prioridad	Quejas, reclamos, solicitud de cambios
Analista/Desarrollador	Culminar con sus metas	Alta carga laboral Ejecución de varias funciones.	

[Tabla N° 2] Involucrados (elaboración propia).

Áreas solicitantes de los servicios de informática.

A continuación se muestran otros involucrados que ocasionalmente solicitan uso de los servicios de la DIS:

- Usuarios DGI: son aquellos funcionarios que solicitan un cambio o simplemente hacen uso del sistema o servicios ofrecidos por la DIS.
- Dirección superior: Hay sistemas que nacen como estrategias de directivos, por lo que el requerimiento inicia como una idea que será madurada posteriormente.
- Otras instituciones públicas. Estas intervienen cuando se requiere interconexión de sistemas o servicios de información; ya sea de parte de ellos o de la parte de la DGI o por órdenes del Gobierno.

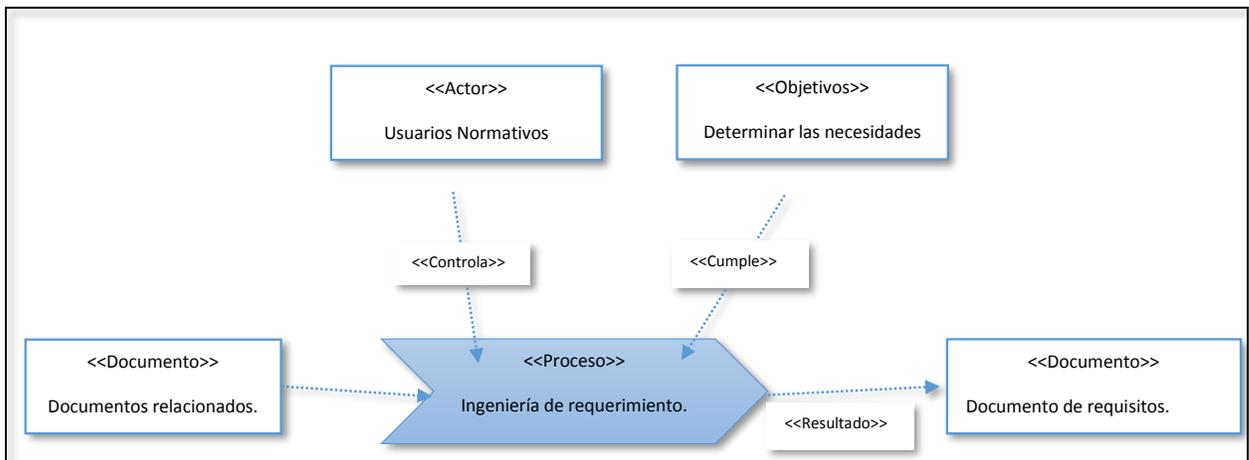
2.1.3 Descripción del sistema de negocio.

A continuación se muestran los diagramas del proceso actual del ciclo de vida del software en la DIS desde su concepción hasta su implantación en producción

2.1.4 Diagramas de procesos

Fase 1: Ingeniería de requerimientos:

En esta etapa el usuario realiza el requerimiento según sus necesidades y revisando la documentación requerida. Es realizada por usuarios normativos y el documento que entrega a DIS es el de requerimientos.



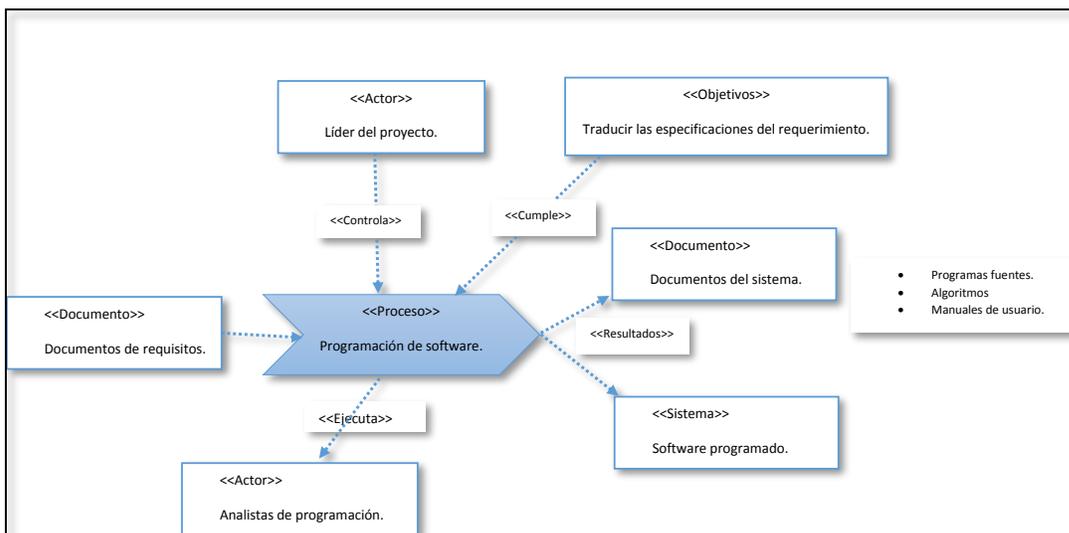
[Figura. N°19]. Ing. Requerimiento (Elaboración propia)

Fase 2: Diseño de software y rediseño de procesos.

Esta etapa es opcional, por tal motivo no se realiza diagrama dado que a veces está incluido dentro del proceso de programación de software.

Fase 3: Programación de software.

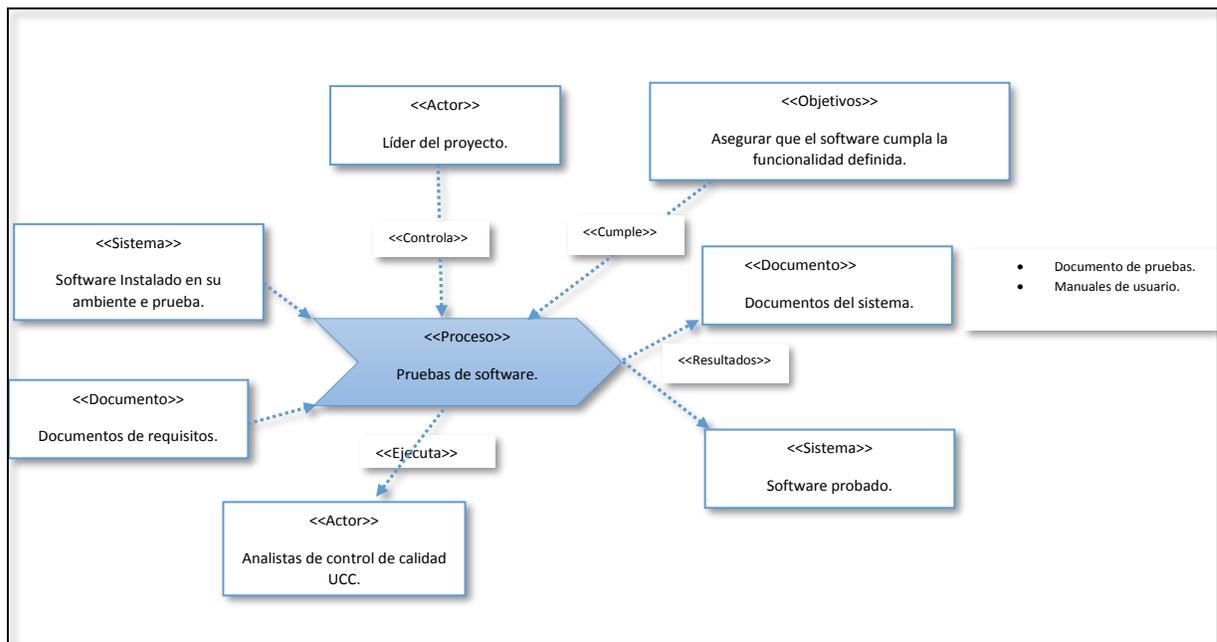
En esta fase el documento de entrada es el requerimiento, mejoras, memos o correos que hacen referencia a él, lo recibe el líder del proyecto y lo asigna a uno o varios analistas que la final entrega el sistema y la documentación de este.



[Figura. N°20] Programación de Software (Elaboración propia)

Fase 4: pruebas de software.

Esta fase describe el proceso de prueba desde que se culmina el desarrollo y se instala el software en los ambientes de prueba. Una vez se tienen listos los cambios se le notifica al usuario que lo solicitó inicialmente. En esta etapa generalmente se crea el manual de usuarios y surge el acta de aceptación de control de calidad para su puesta en marcha en producción.

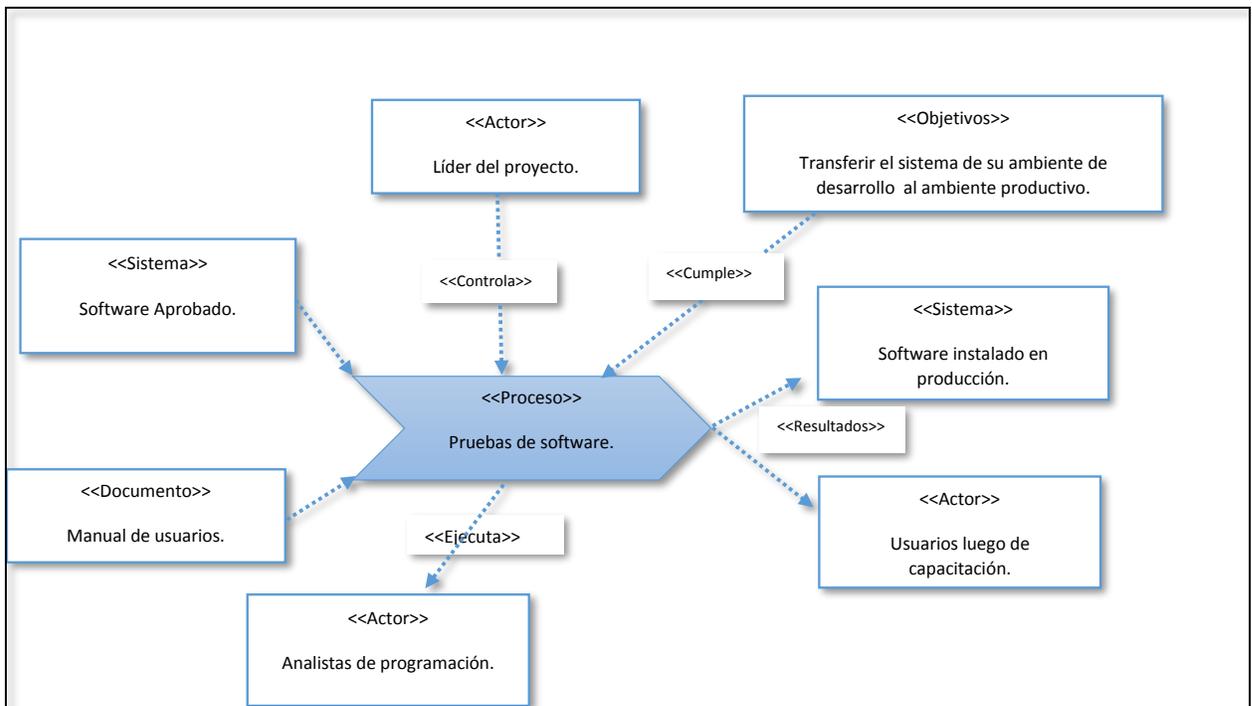


[Figura. N°21] Pruebas de Software. (Elaboración propia)

Fase 5: Implantación del sistema.

Una vez se obtiene la aceptación, se configura el software en un ambiente pre productivo, se realizan ajustes a nivel de base de datos o servidor antes de la instalación y puesta en marcha del sistema. Es decir, primero se realizan las configuraciones necesarias de permisos, de redes, etc.

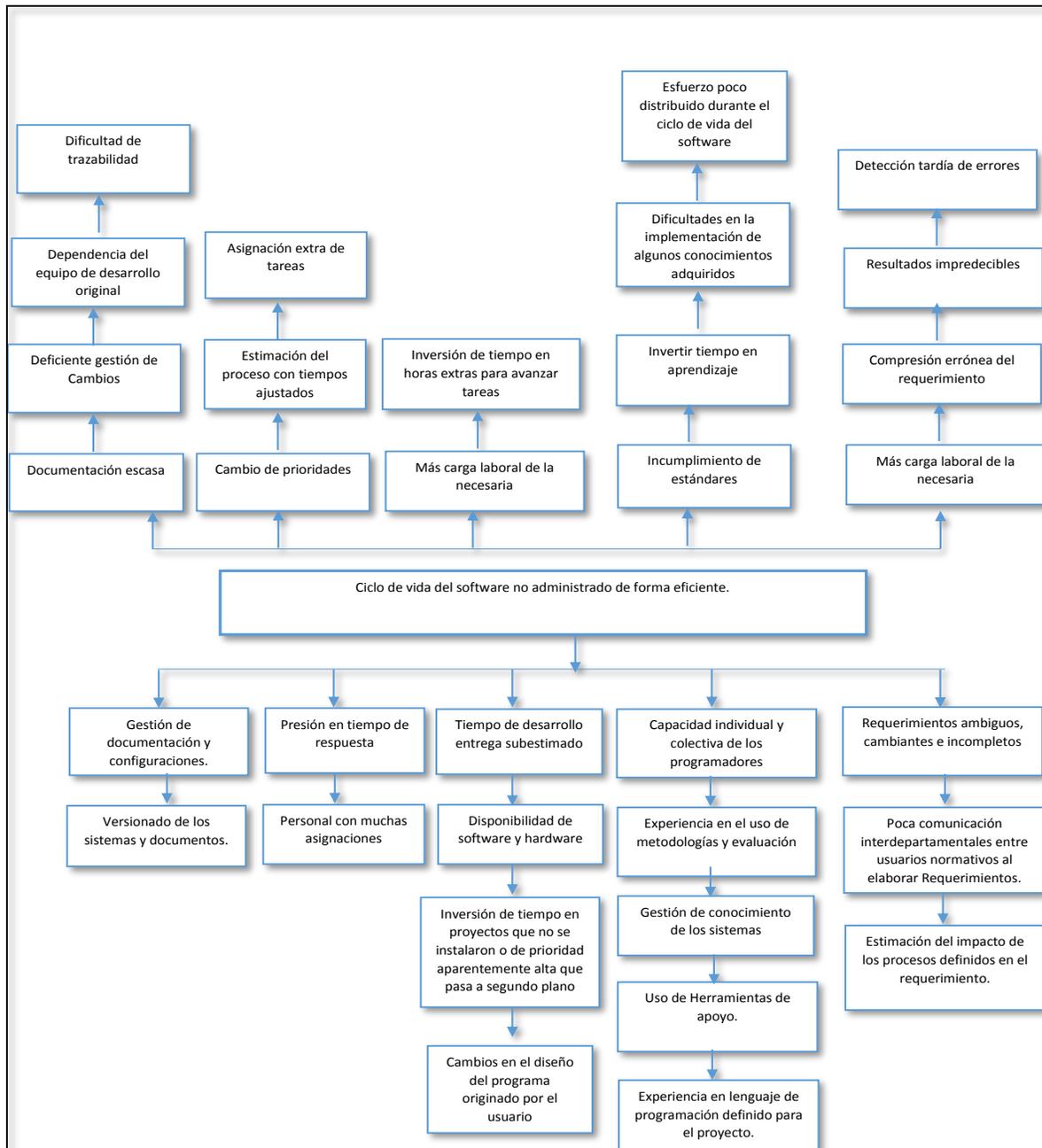
En cuanto a la capacitación, varía según la planificación del proyecto, a veces se da antes de la instalación en producción.



[Figura. N°22] Implantación del sistema (Elaboración propia)

2.1.6 El árbol de problemas

El árbol de problema permite organizar la problemática a investigar:



[Figura. N° 23] Árbol de problemas (elaboración propia)

2.1.7 Análisis del impacto

A continuación se aplica la metodología MIGME-RRC para evaluar el rendimiento de los procesos de software de la DIS utilizando como técnica alternativa a los cuestionarios.

2.1.7.1 Mejores prácticas identificadas

A continuación se muestra la lista de mejores prácticas identificadas, utilizando la escala de frecuencia, donde 1= Siempre, 2= Regularmente, 3= Algunas Veces, 4= Muy raro y 5= Nunca:

Ambito	Práctica	Grado
Proyecto	▪ Reuniones diarias para verificar el avance	3
	▪ Revisión continua del avance estimando el esfuerzo realizado.	5
	▪ Desarrollo continuo	4
	▪ Entregas en reunión con clientes	3
	▪ Presentación de los sistemas de forma visual	3
	▪ Reuniones de planificación frecuentes	2
	▪ Adecuada planificación de la construcción y mediciones del desempeño y calidad de los programadores	4
Organización	▪ Equipos auto organizados	2
	▪ Espacio de trabajo en común	1
	▪ Flexibilidad funcional: Las distintas técnicas y procedimientos aplicados en el desarrollo de un producto son conocidos y compartidos con todo el equipo. Así que una tarea puede ser realizada por cualquier integrante.	3
	▪ Participación activa de los clientes	3
Desarrollos (Ing. Software)	▪ Programación en parejas como mínimo.	2
	▪ Propiedad colectiva del código: Todo los integrantes del grupo son responsables de seguir la arquitectura definida para el proyecto y del código fuente almacenado en el control de versiones	1
	▪ Estándares de codificación	4
	▪ Documentar de manera efectiva los requisitos	4
	▪ Metodología de desarrollo de software	3
	▪ Visualización de los requerimientos organizados por prioridad	3
	▪ Análisis por riesgos	4

Producto	<ul style="list-style-type: none"> ▪ Utilización de herramientas de diseño 	4
	<ul style="list-style-type: none"> ▪ Definir la arquitectura del sistema 	1
	<ul style="list-style-type: none"> ▪ Optimizar el diseño de la aplicación 	2
	<ul style="list-style-type: none"> ▪ Pruebas rigurosas y validación de funcionalidades 	3
	<ul style="list-style-type: none"> ▪ Documentar continuamente 	3
	<ul style="list-style-type: none"> ▪ Revisión de código 	4
	<ul style="list-style-type: none"> ▪ Gestión del control de las fuentes del software 	3
	<ul style="list-style-type: none"> ▪ Gestiones de control de calidad 	2
	<ul style="list-style-type: none"> ▪ Instalación y despliegue eficaz 	2
	<ul style="list-style-type: none"> ▪ Comentar Código 	3
	<ul style="list-style-type: none"> ▪ Gestión de seguridad 	3
	<ul style="list-style-type: none"> ▪ Estrategia de soporte y mantenimiento 	4
	<ul style="list-style-type: none"> ▪ Gestión de diseño 	4
Análisis y Diseño del software	<ul style="list-style-type: none"> ▪ Control de cambios y gestión de requerimientos. 	4
	<ul style="list-style-type: none"> ▪ Aplicación de herramientas para evaluar 	4

[Tabla N° 3]. Prácticas y técnicas (Elaboración propia)

En la tabla a continuación se encuentra los valores dados al grado de las mejores prácticas:

Valoración del indicador de cumplimiento:

Esta escala se establece para asociar dicha valoración a las buenas prácticas relacionadas con las preguntas del cuestionario.

Pesos	Nombre	Referencia
0	No existe evidencia	Inexistente evidencia física o digital que certifique la implantación de acciones asociadas a este artefacto.
1	Se realizan acciones pero no hay evidencia	Aunque se afirme la existencia de acciones para conseguir la evidencia, no existe.
2	Existe evidencia de baja calidad	Existe evidencia se levanta con baja rigurosidad y no se verifica su calidad.
3	Existe la evidencia con deficiencias significativas	Existe pero está incompleta y tienen deficiencias significativas que bajan su calidad.
4	Existe la evidencia pero se debe mejorar	Existe pero hay elementos que se deben revisar y son susceptibles a mejorar
5	Existe y su calidad es excelente	Las mejoras que requieren con menores.

[Tabla N° 4]. Valoración cuantitativa para el indicador de cumplimiento (Gloria P. Gasca-Hurtado¹, 2017)

El resultado de la valoración de las buenas prácticas determina un porcentaje que define el nivel de madurez del proceso que agrupa las prácticas valoradas. Este

porcentaje se establece utilizando la escala que se muestra en la [Tabla N° 5]. Con el fin de tener una calificación del nivel de madurez por cada proceso seleccionado.

Referencia	Porcentaje (%)
Cumple	0-20
Cumple parcialmente	21-69
Cumplido en nivel alto	70-94
Completamente se cumple	95-100

[Tabla N° 5]. Valoración porcentaje para calificar el proceso. (Gloria P. Gasca-Hurtado1, 2017)

2.1.7.2 Analizar la cobertura de mejores prácticas, indicadores y objetivos del negocio.

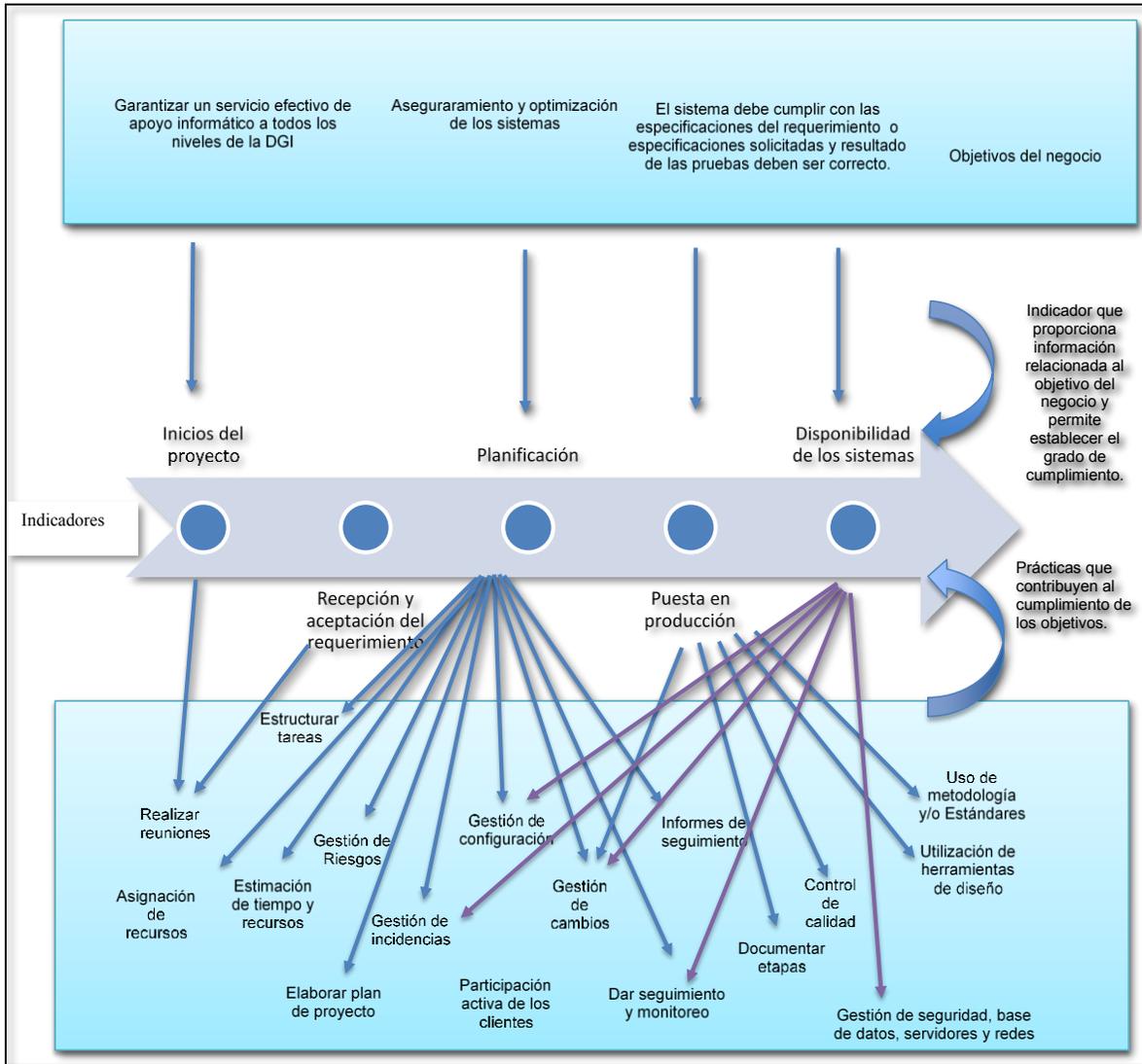
Se identifica la correspondencia entre las mejores prácticas y los objetivos asociados:

Política Institucional: Garantizar la modernización, estabilidad de la plataforma tecnológica y el desarrollo de los sistemas informáticos de apoyo a la gestión fiscal, para facilitar el cumplimiento voluntario de los contribuyentes, a fin de brindar las herramientas de control y seguimiento a la administración tributaria.

Objetivo DIS: 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 la 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.

- **Objetivo Estratégico (O1):** Optimizar y asegurar los sistemas y servicios tecnológicos que brinda la DIS a lo interno y a los contribuyentes para garantizar calidad y seguridad en los mismos.
- **Objetivo estratégico (O2):** Incorporar nuevos módulos a los sistemas definidos en el plan estratégico 2017-2021

- **Objetivo Estratégico (O3):** Implementar un sistema de gestión que permita administrar las tecnologías de información de acuerdo a las mejores prácticas, que optimice costos y apoye al cumplimiento de la gestión tributaria.



[Figura N° 24] Diagrama de objetivos del negocio (Elaboración propia)

2.1.7.3 Valores reales versus planificados

La información relacionada con los objetivos del negocio fue solicitada al jefe de unidad:

#	Indicador	Plan	Real
I1	Recepción y aceptación del requerimiento	≤50%	20%
I2	Planificación del proyecto	≤ 55%	20%
I3	Puesta en producción	≤ 65%	10%
I4	Disponibilidad de los sistemas	≤ 50%	20%

[Tabla N° 6]. Valores Planificados y reales de los indicadores (Elaboración propia)

Donde:

I1- Recepción y aceptación del requerimiento: su medida tiene que ver con la administración de requerimientos desde su entrega formal, la aceptación por la DIS y el análisis formal del analista de desarrollo. El plan es reducir el 50 % que normalmente no se registra de manera correcta en NEXUS el cual es una herramienta utilizada para registrar todo tipo de requerimientos. Actualmente de este 50% el 20% se han incorporado.

I2 - Planificación del proyecto: tiene que ver con la gestión de proyectos; donde, de los proyectos planificados para el periodo 2017-2021 se cumpla el 100%. A la fecha, está pendiente de ejecutarse un 55% del cual ya se ha alcanzado el 20%. Ver anexos [Figura N°. 29].

I3 - Puesta en producción: representa que los proyectos definidos en el plan estratégico y procesos prioritarios del día a día sean cumplidos al 100%. A la fecha, de estos proyectos el 65% están pendientes de finalizarse y de los cuales el 10% se ha instalado en producción en el tiempo planificado.

I4 - Disponibilidad de los sistemas: representa la disponibilidad 24/7 de los sistemas ya que esto es indispensable para lograr el cumplimiento de las obligaciones fiscales. Esta disponibilidad, para efectos de este estudio, se ha determinado un 50% como meta en lo que corresponde a infraestructura (servidores, redes, comunicaciones, etc.) y el otro 50% a incidencias de los propios sistemas (*Ver anexos [Grafica N°36]*). En lo que respecta a infraestructura, se cumple en un 99%; sin embargo, de acuerdo a las incidencias registradas y atendidas de los sistemas, actualmente se gestiona el 20%.

2.1.7.4 Establecer el rendimiento del proceso

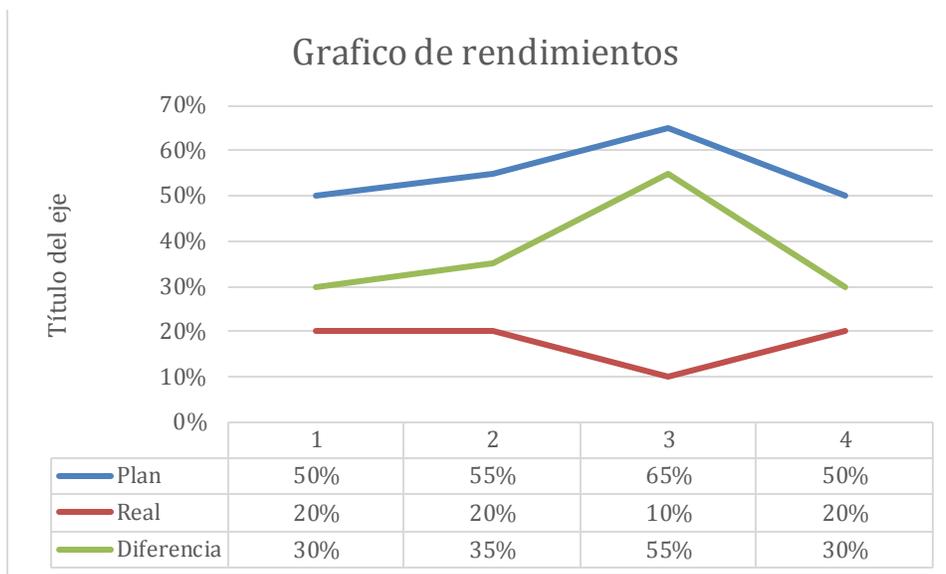
A continuación la matriz de cumplimiento resultante de la comparación de lo planificado con lo real, es decir el avance según los objetivos. *Ver anexos [Ficha Técnica N° 1]*.

Objetivos			
Indicadores	O1	O2	O3
I1	30%		
I2		35%	
I3			55%
I4			30%

[Tabla N° 7] Matriz de cumplimientos (Elaboración propia)

2.1.7.5 Gráficos del rendimiento

A continuación se presentan los gráficos de rendimiento para una mejor comprensión del rendimiento de los procesos:



[Grafico N° 1]Rendimiento (Elaboración propia).

2.1.7.6 Analizar y priorizar indicadores

En este caso, los indicadores de mayor prioridad serían aquellos cuya puntuación sea mayor según los pesos asignados a los criterios que éstos cumplen, donde 1 es el mínimo y 5 el máximo.

Indicadores	Pesos
I1	4
I2	5
I3	4
I4	3

[Tabla N° 8]. Priorización de indicadores (Elaboración Propia).

La participación y aportación de los usuarios solicitantes del software en la estructuración del requerimiento impacta directamente en un nivel de calidad de los requisitos y del producto software final. Consecuentemente, esto se ve reflejado en el producto software cómo errores, y para corregir estos, se debe realizar

correcciones en el producto, lo cual conlleva a invertir en recursos y tiempo, ocasionado sobre costos y sobre carga de trabajo.

La inadecuada gestión de las restricciones de los proyectos de software, se prioriza tiempos, alcance y costos de desarrollo en lugar de calidad de software este tipo de priorización siempre tienen un impacto negativo en la calidad del producto y por lo tanto debe ser gestionado y manejado durante el desarrollo del producto.

2.1.7.7 Evaluación de la calidad

Criterios de evaluación del producto

En la tabla a continuación presenta las características y sub-características como propuesta a considerarse al evaluarse la calidad del producto.

Característica/ Sub características		Medidas
Adecuación funcional	Complejidad funcional	Cobertura de la implementación funcional
Fiabilidad	Madurez	Eliminación de fallos Densidad de fallos Cobertura de las pruebas
Usabilidad	Facilidad de operación	Consistencia operacional
	Facilidad de aprendizaje	Complejidad de la documentación del usuario o facilidad de ayuda
	Protección contra errores del usuario	Chequeo de la validez de la entrada de datos. Evasión de operaciones incorrectas.
Eficiencia del comportamiento	Comportamiento temporal	Tiempo medio de respuesta. Tiempo medio de rendimiento
	Capacidad	Número máximo de solicitudes en línea. Número máximo de accesos simultáneos
Seguridad	Autenticidad	Métodos de autenticación
Facilidad de mantenimiento	Facilidad de prueba	Capacidad de reinicio de la prueba
	Facilidad de modificación	Localización del grado de impacto de la corrección

[Tabla N° 9]. Propuesta de características, sub características, medidas y herramientas de calidad externa del producto de software.

La selección se fundamenta en el hecho de que estas características coinciden con las priorizadas y más empleadas por las organizaciones analizadas en el diagnóstico. Se decide postergar a una segunda etapa la evaluación de las características Compatibilidad y Portabilidad por ser las menos priorizadas por las organizaciones y por el hecho de no contar con una herramienta informática que brinde soporte al cálculo de sus medidas.

Criterios de evaluación del proceso

A continuación se propone la siguiente guía que permite evaluar el Ciclo de Vida del Desarrollo de Software. Teniendo como base el híbrido de metodologías, debido a que cada una plantea diferentes estrategias para alcanzar altos niveles de calidad y seguridad, donde 1= No Aplica, 2= Raramente, 3= Algunas Veces, 4= Regularmente, 5= Siempre.

No.	Característica	Observación/Resultado	Valoración (1-5)
ISO 27001:2013			
1	¿Documenta constantemente los procedimientos que ejecuta su área, y los da a conocer los mismos?	<p>No todos los procedimientos están documentados.</p> <p>En caso de que un analista/desarrollador decida marcharse de la organización, con él se iría parte del conocimiento.</p> <p>Generalmente se procura estén dos o más personas en un proyecto y todos deben manejar lo mismo, sin embargo hay sistemas que solo lo conocen por completo una persona, los requisitos solicitados por correo a veces no se actualizan en el documento original, si ese recurso se va se pierde la información de su correo o no es accesible al nuevo encargado o aun el mismo desarrollador podría olvidar el porqué de ciertas funciones del código, funcionamiento o programación de base de datos, ya que esta es estructurada por el programador, sin estándar, documentación o descripción de funcionamiento.</p>	3
2	¿Cuenta con un procedimiento de control de cambios, es decir, si se requiere un cambio de un aplicativo	Algunos cambios no se documentan, quedan a nivel de correo.	2

No.	Característica	Observación/Resultado	Valoración (1-5)
	puesto en producción cuales son los pasos para aplicar dicho ajuste?		
3	¿Tiene registros para verificar y determinar que los recursos a nivel de servidores están asignados correctamente y que estos no requieren mayor capacidad?	<p>Se monitorean los servidores con un sistema NAGIOS, y hay un servidor de replica que entra en servicio si hay una caída del actual.</p> <p>También se revisa el espacio de los servidores y bases de datos.</p> <p>La unidad de bases de datos debe incluir un análisis de los proyectos en desarrollo, podría ser en paralelo a pruebas UCCC, para ver si hay capacidad para ellos.</p>	4
4	¿Realiza pruebas de seguridad en los aplicativos?	El desarrollador, basado en su conocimiento. No existe algo estipulado a seguir. Pero hay protocolos y otras medidas que ayudan a reducir ciertas debilidades en cuanto a esto.	3
5	<p>¿Realiza pruebas de aceptación de la aplicación?</p> <p>¿Se revisa el cumplimiento técnico de las aplicaciones, estas cumplen con normas de seguridad?</p> <p>¿Cómo proceden cuando hallan dichas vulnerabilidades?</p>	<p>Si se realiza prueba de aceptación.</p> <p>Se entra a evaluar el sistema y se aplican los ajustes que requieran.</p> <p>No existe norma establecida o es de desconocimiento al personal, solo se pide que tenga ciertos requisitos de seguridad básicos pero no está estipulado a nivel de división, y el cumplimiento de la arquitectura a seguir: en capas o en dominio, manejo de usuarios, control de errores y sesiones; pero no es revisado su cumplimiento.</p>	3
ISO 19011: 2011 (Base de datos)			
6	¿Define Modelo Entidad Relación (MER)?	No, la BD se va construyendo de acuerdo a medida que avanza el desarrollo; pero en algunos casos se trabaja primero en el diseño de la base de datos o prototipos o se ajustan sistemas que fueron desarrollados externamente y son retomados por el equipo DIS.	2
7	¿Cómo evita la redundancia de datos en su BD?	Se confía en el criterio de cada desarrollador	2

No.	Característica	Observación/Resultado	Valoración (1-5)
8	¿Cómo realiza la selección del motor de BD?	A criterio del desarrollador sin evaluaciones adicionales, generalmente se utilizan 3: MySQL, SQL, ADABAS y aunque existe PostgreSQL solo es utilizada por pocos sistemas.	3
ISO_IEC12207 (Procesos de desarrollo, Mantenimiento, soporte y Gestión de Configuración)			
9	¿Existe evidencias documentadas para el levantamiento de información?	A veces apoyándose de correos o ayudas memorias.	3
10	¿Dónde están ubicados los manuales de usuario y documentación general de los aplicativos?	Únicamente se realiza manual de usuario en algunos casos, se validó uno de ellos y este se encuentra incompleto, los tiene el área solicitante y a veces los desarrolladores y usuarios finales.	3
11	De qué manera, identifica que los recursos de infraestructura son suficientes	No se realiza dicho procedimiento, simplemente se tiene un servidor de aplicaciones y allí se carga cualquier aplicación que se desarrolle esto en los ambientes de prueba, ya que en producción se basan en los monitoreo de los servidores.	3
12	¿Los aplicativos contienen pruebas unitarias?	No, únicamente se realizan pruebas funcionales. Solo uno de los sistemas de muestra realiza prueba unitaria en algunas funcionalidades.	2
13	¿Dónde mantiene su versionamiento de código?	Se maneja el repositorio en Team Foundation, este permite hacer uso de las versiones según la necesidad	5
14	¿Tiene documentado un estándar de desarrollo?	No, cada quien programa según su experiencia	1
15	¿Dónde se puede observar la solicitud de cambios realizados a un sistema por parte del usuario funcional?	No se lleva control de cambios de todos los proyectos, solo algunos a través de un sistema de requerimientos.	2
16	¿Se deja evidencias del análisis que se realiza para validar la coherencia de este cambio?	No siempre;	2
17	¿Dónde se encuentra alojada la documentación de soporte de las aplicaciones?	Puede que a veces no existe documentación o esté incompleta. Si acaso existe, el desarrollador es el propietario y si desea	3

No.	Característica	Observación/Resultado	Valoración (1-5)
		respaldar o le solicitan dicha documentación es compartida entre el equipo de trabajo.	
18	¿Cómo realiza la liberación de versiones y cómo documenta dicho procedimiento?	Una vez el control de calidad o el usuario funcional da el visto bueno, a través de un acta "Acta de instalación" se realiza solicitud al área de bases de datos, se suben los cambios al control de versiones, el desarrollador se conecta a un servidor para descargar los cambios, se configura, se publica y dicha carpeta o archivos son trasladados a producción, se hacen las configuraciones productivas, ya puesto en marcha, se realizan pruebas de acceso o pruebas básicas que no afecten integridad de datos; pero parte de este proceso no se documenta, es decir, en el "Acta de remisión de pruebas" se describen lo que se entrega a control de calidad: objetos de aplicación o bases de datos, sin embargo en el transcurso de pruebas surgen cambios, se crean nuevos objetos a veces no se documentan. Pero no se documenta proceso de esto.	4
ISO 25000 2011 (Calidad)			
19	¿Los sistemas son mantenibles, es decir, es fácil adaptarse a las nuevas especificaciones y requisitos del software	Algunas veces; Hay cambios que requieren estructuración de varias cosas o no son viables.	3
20	¿Los sistemas elaborados son portables, es decir tiene la capacidad de ser transferidos de un entorno a otro?	La gran mayoría si dado que son aplicaciones web.	4
21	¿Los aplicativos cuentan con su código documentado?	Nadie más que el analista/Programador tiene acceso a revisar el código, se revisaron los sistemas descritos en la muestra y efectivamente existen funciones documentadas en código o en documento técnico u de otro tipo.	4
SCRUM (Requerimiento, Seguimiento)			
22	¿Se tiene un responsable por cada proyecto?	SI	5
23	¿Tiene algún procedimiento para estimación de tiempos para cada etapa?	No, todo se realiza con base a la experiencia del negocio.	3

No.	Característica	Observación/Resultado	Valoración (1-5)
24	¿Dónde se almacena el control de seguimiento de cada proyecto?	No se lleva control por proyecto de forma completa e integral, pero se realiza una reunión semanal con todo el equipo y allí se analiza cómo avanza cada actividad asignada al equipo en dependencia de la prioridad del tema se reúnen más veces.	3

[Tabla N° 10]. Evaluación del proceso del ciclo de desarrollo de software (Elaboración propia).

A continuación se muestra el resultado de la sumatoria de factores y su resultado:

Alto Cumplimiento = $5 \times 24 = 120$
Buen cumplimiento = $4 \times 24 = 96$
proceso Medio = $3 \times 24 = 72$
proceso Deficiente = $2 \times 24 = 48$
Altamente Deficiente = $1 \times 24 = 24$
Resultado de la ponderación = $\sum \text{Peso} = 72$;

[Tabla N° 11]: Ponderación de valores del proceso (Elaboración propia)

Artefactos propuestos para la evaluación de la calidad del producto

A continuación se definen una serie de artefactos propuestos a tomar al evaluar la calidad de software:

Artefacto	Propósito	¿Existe?
Solicitud de evaluación de calidad del producto de software	Registrar los datos del área solicitante y del producto de software. Además, se agregaron los detalles del entorno de prueba.	Sí ; Acta de entrega y de pruebas, pero suele enviarse una sola vez a UCCC, no se actualizan si durante pruebas hay objetos modificados sino hasta al final en el acta de instalación que no siempre define dichos objetos y la descripción de las pruebas son de forma generalizada, haciendo más referencia al requerimiento.

Artefacto	Propósito	¿Existe?
Listado de características de calidad definido por el cliente.	Definir el listado de características de calidad a evaluar según el cliente	No; pero solo se tomaría en cuenta al área solicitante del requerimiento o UCCC; dado que son quienes definen y quien prueba según el caso; el contribuyente que es el usuario final, solo sigue los procesos del sistema luego de una capacitación y a menos que tenga alguna falla la reporta y se ve el caso específico.
Producto de software	Manual de instalación, manual de configuración y manual de usuario	No existen; es recomendable incluirlos para que el DBA que se encarga de administrar los servidores tenga mayor conocimiento, control y conozca los requisitos de ambiente y objetos afectados. Así también el manual de usuarios al área de control de calidad para que evalúe este.
Especificación de requisitos	Definir el listado de requisitos funcionales y no funcionales del producto de software	No; pero es recomendable para comprender mejor y manejar la trazabilidad de los requisitos y verificar su cumplimiento.
Diseño de casos de prueba	Listar los casos de prueba definidos y registrar sus resultados.	No; pero es recomendable documentarlos para el control de pruebas.
Listado de características, Subcaracterísticas, medidas de calidad y tipos de prueba	Definir las características, Subcaracterísticas, medidas de calidad evaluar, así como los tipos de prueba a realizar en dependencia del tipo de producto y de las necesidades del cliente.	No; es necesario definir que es calidad para el cliente y para la institución basándose en normas de calidad que lo respalde.
Plan de pruebas	Establecer el periodo de pruebas, los recursos, actividades planificadas y herramientas de evaluación.	No; se realiza conforme a lo definido en la hoja de prueba proveniente de UST, y no hay otra herramienta definida de prueba más que el propio sistema a evaluar.
Herramientas de evaluación	Identificar el conjunto de herramientas que se utilizarán durante el desarrollo de las pruebas y en la evaluación de calidad.	No;

Artefacto	Propósito	¿Existe?
Listado de no conformidades	Describir las no conformidades generadas en el proyecto. Clasificadas y priorizadas según su importancia e impacto en la calidad del producto.	No;
Informe de resultados de la prueba	Realizar una valoración de la calidad de la prueba, basada en el cumplimiento del periodo de prueba, la utilización de los recursos, la cantidad de no conformidades y la cobertura de la prueba.	No; se realiza pero con enfoque al resultado de las pruebas funcionales e integrales realizadas a los sistemas, no incluye no conformidades ni la cobertura de pruebas.
Informe de calidad	Ofrecer una valoración de la calidad del producto a través del análisis de los resultados de las características, Subcaracterísticas y medidas evaluadas.	No; no se tiene métricas más que el control de incidencia de los sistemas en producción y la resolución de las bitácoras al cual le falta incluir el nuevo software.
Acta de liberación del producto	Documentar la liberación del producto de software.	SI; Actualmente luego de la aceptación del software se realiza el acta de instalación.

[Tabla N° 12]: Artefactos propuestos para Control de calidad (Elaboración propia)

2.1.7.8 Análisis de los resultados

En esta fase se analiza la información recolectada de las encuestas y entrevistas realizadas al Grupo 1, Grupo 2 y líderes de la unidad en estudio, con el fin de poder determinar las debilidades y fortalezas del proceso del ciclo de vida del software en la DIS.

Si bien el ciclo de vida involucra a muchas áreas en la DIS; las más involucradas son la UST y Control de Calidad, dado que tienen mayor interacción con el producto y son los responsables de asegurar la calidad y el control de calidad, por esta razón la encuesta se realizó únicamente en dichas áreas.

Entrevistas

Se realiza el análisis de la entrevista a los líderes de la Unidad de Sistemas Tributarios y Control de Calidad, encontrándose que en el ciclo de desarrollo de los sistemas utilizan metodologías pero no se encuentra institucionalizada, por lo que cada analista implementa prácticas de metodologías y buenas practicas basadas en el conocimiento adquirido.

El área de desarrollo cumple con las entregas de sus aplicaciones pero no siempre en el tiempo estipulado y se lo logra es con doble esfuerzo de tiempo y recursos, reduciendo el tiempo de pruebas.

El personal es constantemente capacitado en temas de desarrollo, seguridad y procesos del giro del negocio, es decir aspectos tributarios y de funcionamiento de los sistemas.

En cuanto a asignación de requerimiento hay algunos que no están cubiertos por la carga laboral del analista de programación, por tanto se priorizan por urgencia de respuesta, por lo que a veces se detiene un avance por priorizar otro.

En al cálculo de los tiempos no se estima el grado de avance de estos con facilidad, dando estimaciones de entrega a veces irreales.

La UST y UCCC están conscientes de la necesidad de mejorar los controles e implementar una metodología; sin embargo el factor costo y resistencia al cambio representan una limitante.

Por otro lado se encontró que hay software que son heredados, ya sea por desarrollo externo o por otros analistas que cambiaron de cargos o se retiraron de la institución, por lo que en ciertos proyectos hay limitantes de gestión del conocimiento dado que no se encontró documentación completa; generalmente solo manuales de usuarios y actas de procedimientos, pero no documentación técnica.

Análisis al área de la UST

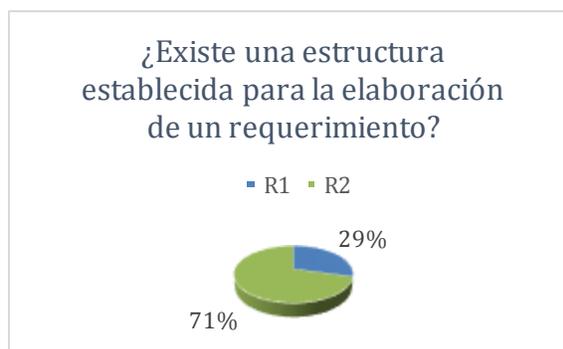
A continuación se presenta el resultado de la encuesta aplicada al equipo de desarrollo:

Fase de ingeniería de requerimientos:

La mayoría de requerimientos de los sistemas son definidos por un área ya sea normativa, administrativa u otras áreas de la institución representando el 91% de usuarios solicitante de cambios en los sistemas, los cuales poseen diversas especialidades pero pocos en ingeniería de requerimientos o conocimientos relacionados a ello, por lo que solicitan cambios basándose en necesidades, inconsistencias, mejora, entre otros; Dado que la estructura de un requerimiento varía según el usuario en un 100% dato encontrado al revisar los requerimientos físicos de los sistemas estudiados, se encontró omisión de aspectos importantes en la revisión general del requerimiento (análisis preliminar), otras veces durante el proceso de desarrollo, durante las pruebas o en el peor de los casos en producción provocando inconsistencias por escenarios no contemplados.



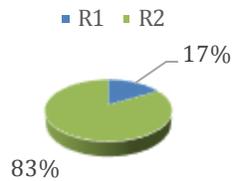
[Gráfico N° 2]



[Gráfico N° 3]

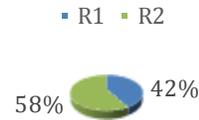
El 71% de los analistas de desarrollo afirman que no existe una estructura definida que deba tener un requerimiento, en esta fase se realizan reuniones, preguntas vía telefónica, confirmación por correo electrónico sobre las dudas o situaciones que van surgiendo y mejoras sobre los requerimientos.

¿Se entiende con facilidad los requerimientos recibidos?



[Gráfico N° 4]

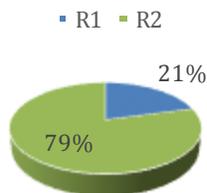
¿Se gestionan los cambios en los requerimientos?



[Gráfico N° 5]

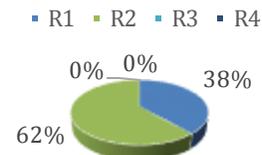
En un principio, el 83% de analistas de desarrollo tienen dificultades para entender dichos requerimientos, esto lleva a diferentes interpretaciones desde el punto de vista de los involucrados en el desarrollo de un sistema de software y un 58% confirman que los cambios desde esta fase hasta después de su implementación en producción no se documentan o se actualizan.

¿Existe alguna práctica definida para manejar la trazabilidad de los requisitos?



[Gráfico N° 6]

¿Qué metodología de desarrollo de software está usando actualmente?



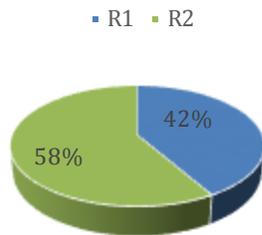
[Gráfico N° 7]

En cuanto al manejo de la trazabilidad de los requisitos, un 79% de los analistas afirman que no existe. La trazabilidad permite hacer una evaluación en el diseño y en la implementación, un requerimiento debe de ser rastreable desde que se define y durante todo el desarrollo del software, lo cual garantiza una adecuada administración del cambio con el fin de evaluar el impacto en el resto del sistema; así que para evaluar el impacto se basa en el conocimiento del desarrollador y el resultado de las pruebas.

Fase de análisis, Diseño y rediseño de procesos

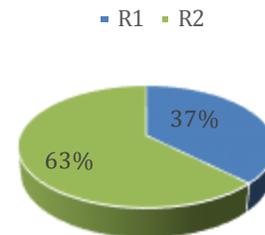
Un 58% afirma que esta fase no se realiza, el 63% no realiza diagramas ni modelado de los procesos; es opcional esta etapa.

¿Se realiza un análisis de los requerimientos versus los recursos disponibles y procesos existentes relacionados a lo que se solicita?



[Gráfico N° 8]

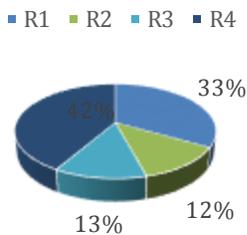
¿Se realiza modelado o diagramas de los proceso del sistema y de la base de datos relacionada?



[Gráfico N° 9]

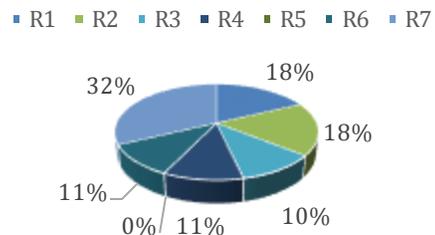
En primera instancia se evaluó si se utiliza alguna herramienta de apoyo en esta etapa dando como resultado que se utilizan al gusto del programador.

¿Se utilizan estándares de calidad, seguridad, de mejores prácticas en el durante el proceso de desarrollo?



[Gráfico N° 10]

¿Cuál de las siguientes herramientas utiliza para diseñar diagramas o modelos?

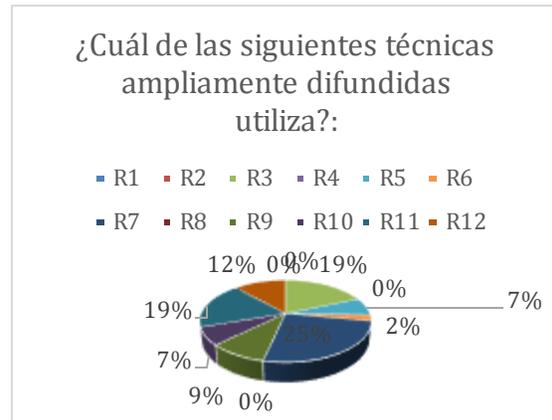


[Gráfico N° 11]

Se verificó que sí existen técnicas de diseño sin embargo se aplican según lo que elija el programador, es decir, el documento técnico y de diseño si existe no siempre tiene todos los diagramas o análisis. Entre los porcentajes con mayor puntaje se encuentran: 19% utiliza normalización y optimización, 19% modelado de procesos, 7% modelado de entidad/relación, 9% diagrama de flujos de datos y el 12% no realiza ningún tipo de diagrama.



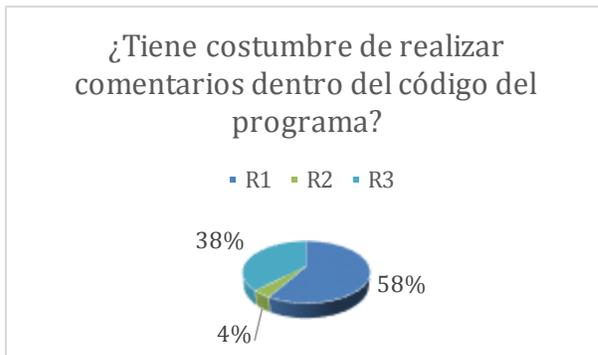
[Gráfico N° 12]



[Gráfico N° 13]

Fase de programación de software

En la institución hay ciertas políticas que se aplican en el área de bases de datos y a nivel de servidor, sin embargo en el área de desarrollo el 37% afirma que no existen políticas o estándares a seguir en el desarrollo, 38% desconocen si existe, el restante 25% indica que existen al menos una.



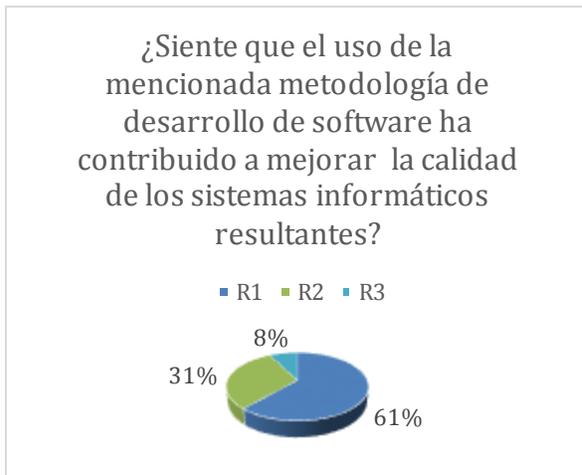
[Gráfico N° 14]



[Gráfico N° 15]

En cuanto a metodologías de desarrollo un 54% utilizan metodología, de los cuales 38% es ágil (25.64% SCRUM y 12.67% XP) y 62% tradicional (RUP), el 46% restante no utiliza una metodología o estándar que provea de lineamientos a seguir durante el ciclo de vida del software; de ese restante el 38.46% normaliza la base de datos, 42.30% el modelo de entidad –relación, 15.38% hacen uso de diagramas UML, 38.46% modela los procesos y el 26.92% no utiliza absolutamente nada de metodologías o diagramas.

Por lo que se concluye que esto es opcional; entre los motivos por el cual no utilizan una metodología surgen tres factores de mayor puntaje: 33% no la realizan porque es opcional, 24% desconoce sobre metodologías, 10% utiliza procedimientos propios a seguir basados en su experiencia.

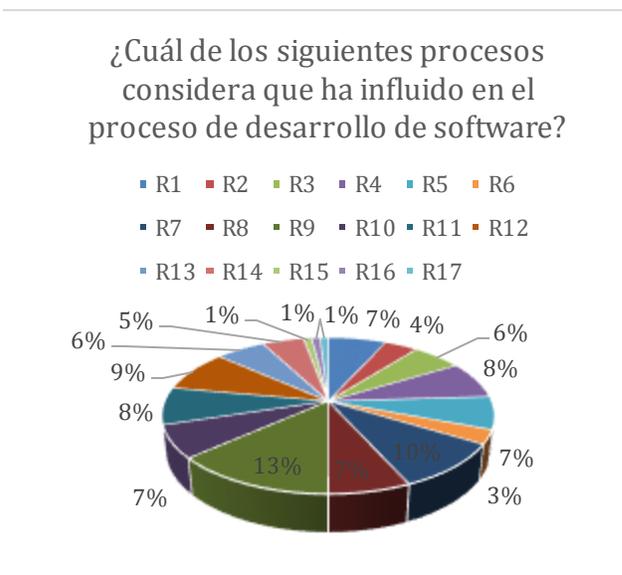


[Grafico N° 16]

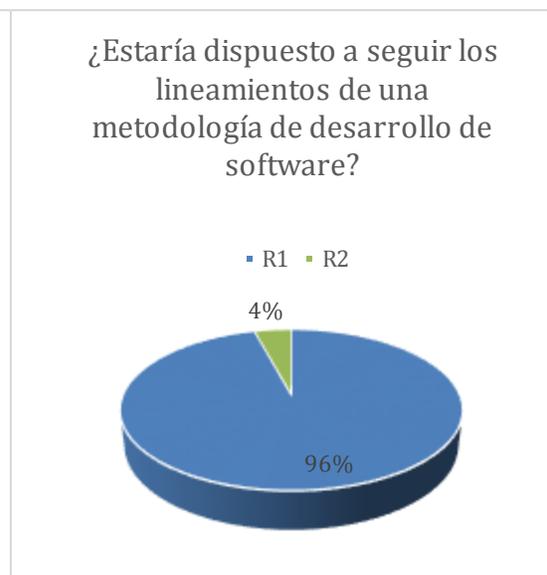


[Grafico N° 17]

Por otro lado, se analizó las razones que afectan al proceso de desarrollo en aspectos de entrega, calidad y respuesta y se distribuyen de la siguiente manera: 13% sobrecarga de asignaciones, 10% experiencia del desarrollador, 9% estimación de los tiempos, 8% cambios de prioridades, 8% metodología y estándares utilizados, 7% gestión de cambios, 7% gestión de requerimientos y el restante por otros motivos.



[Grafico N° 18]

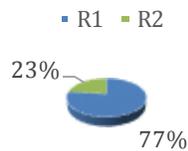


[Grafico N° 19]

Sin embargo el 96% de los analistas están abiertos a aplicar una metodología de software para mejorar sus procesos actuales.

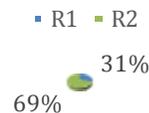
Para cumplir con los tiempos planificados en la medida de lo posible el 46% de los analistas realizan horas extras, 50% algunas veces y solo el 4% no realiza horas extras.

¿Se siente cómodo usando la mencionada metodología de desarrollo de Software?



[Gráfico N° 20]

¿Le resulta complicado seguir los procesos de la mencionada metodología de desarrollo de software?

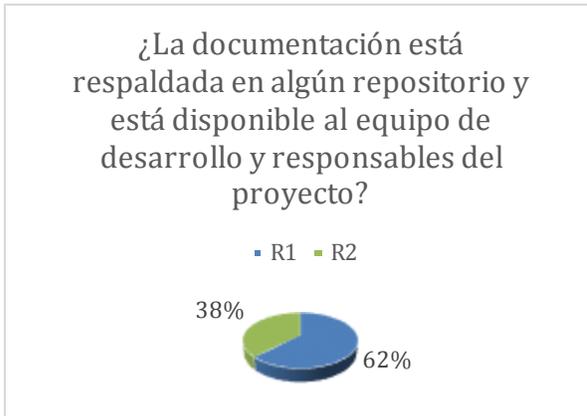


[Gráfico N° 21]

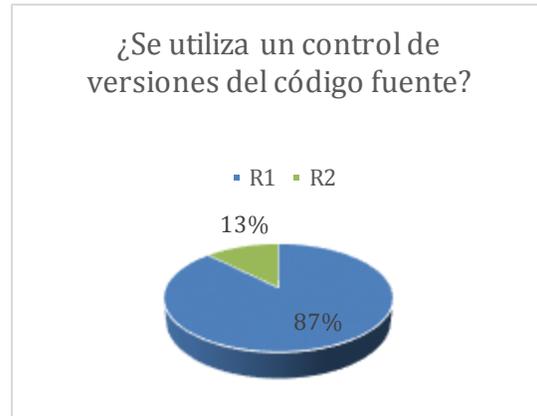
El rediseño de procesos surge por distintas necesidades; entre las cuales pueden ser: corregir una inconsistencia o mejorar sobre algún proceso, eliminar una restricción o crearla de forma temporal o permanente para resolver uno o más casos reportados a través de cualquier medio. Algunos de estos casos se resuelven con actualizaciones a nivel de base de datos pero otras requieren ajustes a los sistemas.

Implementación del sistema

Todos los sistemas utilizan un control de versiones, el 58% tiene la costumbre de dejar comentarios en su código, el 38% a veces deja sus comentarios.



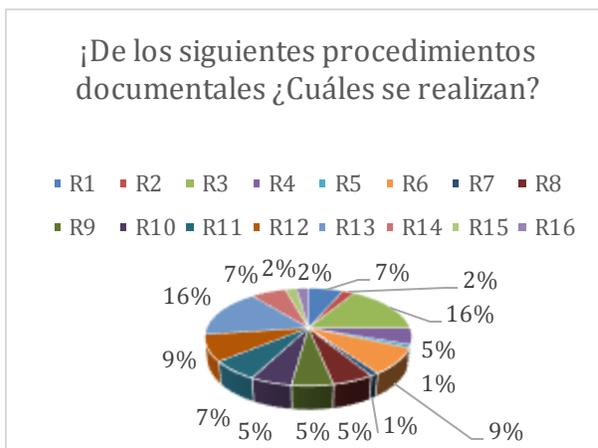
[Grafico N° 22]



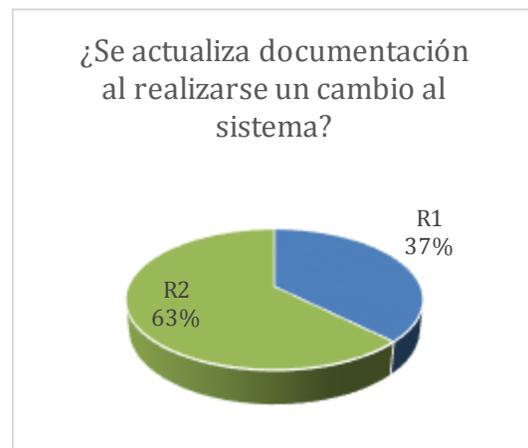
[Grafico N° 23]

En cuanto a la gestión de la documentación lo que más se utiliza son 16% actas de instalación, 16% plan de pruebas, 9% actas de entrega, 9% análisis del requerimiento. El plan de pruebas es de forma general, las actas de entrega especifican los formularios, tablas, objetos que se le entrega a control de calidad para que inicie sus pruebas, el acta de instalación se utiliza para instalar en producción el cambio una vez se obtiene el visto bueno de control de calidad.

Lo que son manuales de usuario generalmente los realiza el área que creó el requerimiento a veces con apoyo de la UST; y los manuales técnicos pocos sistemas lo tienen.

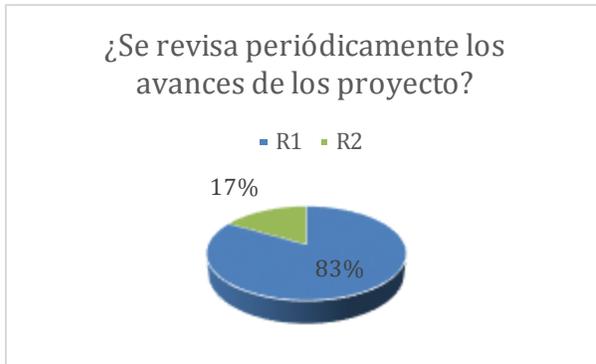


[Grafico N° 24]

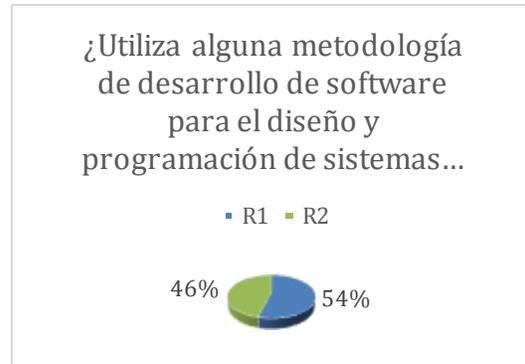


[Grafico N° 25]

Un 63% afirma que la documentación no se actualiza muchas veces al realizarse un cambio en el sistema.



[Gráfico N° 26]



[Gráfico N° 27]

Análisis al área de UCCC

A continuación se hace un resumen del resultado obtenido a la encuesta realizada a analistas de control de calidad de la UCCC compuesta por 10 funcionarios y se aplicó a 9 de ellos.



[Gráfico N° 28]



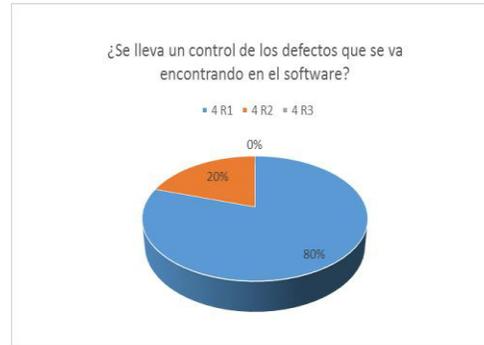
[Gráfico N° 29]

El 90% de los analistas realizan sus pruebas basadas en los requerimientos, de los cuales el 60% asegura que no existen parámetros establecido para medir la calidad de los productos desarrollados, mientras un 40% de los analistas lleva control de ciertos parámetros. En entrevista con la líder de la unidad se aclaró que se llevan

ciertos controles sobre las incidencias de los sistemas reportados por usuarios y de la cantidad de sistemas afectados e instalados.

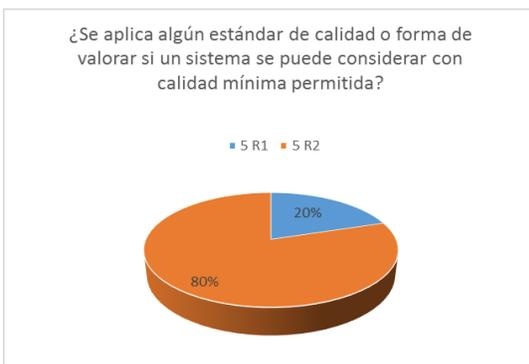


[Gráfico N° 30]

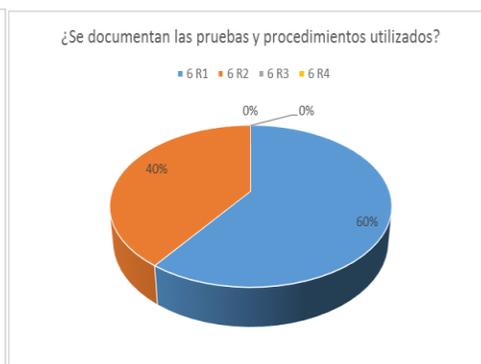


[Gráfico N° 31]

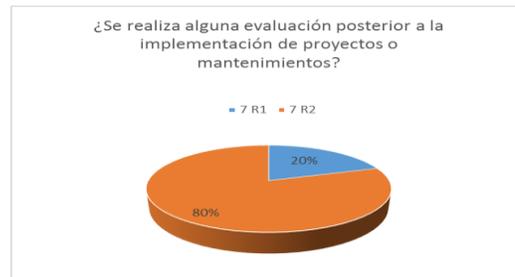
En cuanto al tipo de prueba realizada sobre los productos tanto de nuevo desarrollo como sobre los que se realizan algún ajuste, en un 70% son funcionales, un 40% de caja blanca y el líder de unidad anexaba el tipo de prueba integral que está dentro de la funcional que trata de probar otros modulo relacionados al cambio en el sistema. Se concluye pues, que el software es verificado externamente, no es revisado internamente para evaluarlo y determinar su calidad interna. De los trabajos realizados en el área el 80% documenta las acciones realizadas.



[Gráfico N° 32]



[Gráfico N° 33]



[Gráfico N°34]

El 80% afirma que no se aplica ningún estándar de calidad sobre el software como para determinar una calidad mínima; aunque un 20% utiliza alguna metodología es opcional. En entrevista con líder de unidad afirma que este tipo de control no existe como tal pero estaría dispuestos en su área se aplique para mejorar sus procesos. En este punto es donde mi objetivo específico número 2 contribuirá un poco ya que no es una metodología o un estándar pero si es un control de forma inicial.



[Gráfico 35] Incidencias

2.1.7.9 Importancias de implementar una metodología o estándar.

La DGI, en los últimos años ha avanzado, aminorando conductas de incumplimiento tributario; realizado a través del desarrollo de programas de fortalecimiento institucional, de mejora de infraestructura y de desempeño de sus actividades operativas básicas que incluyen la tecnología como apoyo a la gestión y el fortalecimiento institucional.

Según el plan operativo (2017-2021) para aumentar los niveles de cumplimiento voluntario, las Administraciones Tributarias deben contar con buenos servicios de información y asistencia, por tal motivo, esta tarea se ha transformado a lo largo de los años en un elemento clave para cualquier administración moderna.

Uno de los roles principales de la DGI es reducir la evasión y la elusión tributaria, elevando la percepción del riesgo (en los evasores) a ser fiscalizado; así como, facilitar el cumplimiento de las obligaciones tributarias, mejorar y simplificar el sistema tributario y elevar la cultura tributaria esto con nuevas formas de gestión e inclusión de nuevos sistemas y módulos a los sistemas existentes.

En base a las referencias mencionadas en el marco teórico se concluye que la calidad de los productos de software depende en gran medida de los procesos utilizados desde su creación y que la implementación de cierta metodología varía según las necesidades del proyecto pero que utilizar estándares y metodología con lleva a muchos beneficios, limitaciones e implicaciones. *Ver anexos [Tabla N°19] y [Tabla N°20].*

Para asegurar la buena implementación de cualquier norma o modelo se deben tomar en cuenta tres componentes:

- Las prácticas
- Las herramientas
- La gente

Por lo anterior expuesto, se nota la importancia que tienen los sistemas para la institución y según los planes y metas a alcanzar aparte de cumplir con el 100% de lo planificado una de las prioridades es dar una mejor atención al contribuyente y a sus funcionarios que utilizan los sistemas para facilitar el propósito fiscal, hablando en términos de calidad externa, pero para lograr esto, también es importante no olvidar otros aspectos de calidad de los sistemas y del proceso de desarrollo, para reducir la incidencia de fallos, plazos de entrega de proyectos y otros aspectos mencionados entre la problemática de estudio, por lo que sí es de vital importancia utilizar una metodología de análisis, diseño, codificación e implementación de los sistemas desarrollados para lograr obtener medidas y ver el avance en la mejora al proceso y al producto como tal.

En todo desarrollo de sistemas de software es de suma importancia el seguir alguna especificación que permita a los desarrolladores el tener una disciplina que haga que todas las etapas del desarrollo del sistema, desde la pesquisa inicial de requerimientos hasta las pruebas finales del sistema, sean no solo más coherentes sino también más formales.

Con respecto a esto se evaluó aspectos de la forma en que se realiza el proceso de desarrollo, entre esto: conocer si el personal aplica alguna metodología, si ésta es estándar para todos los proyectos y el nivel de conocimiento en este tema. El resultado es que sí conocen o han escuchado de las metodologías, estándares de calidad de software y procesos de desarrollo, pero no a profundidad, sino de forma autodidacta o basado en experiencias y al no existir algo definido con lo cual guiarse no se puede medir si se cumple o no esto. Por otro lado hay quienes no utilizan metodología o la utilizan de forma parcial por el corto plazo que tienen para entregar los productos. La cultura de hacer entregas periódicamente de funcionalidades al cliente, y durante el proceso de funcionamiento en entorno de producción se van localizando las fallas de la aplicación, que son reportadas en el periodo de garantía, para la respectiva solución.

Todos los encuestados coincidieron en que el levantamiento de información es importante y de mayor atención en el momento de construir una aplicación, seguido del análisis y diseño. Por otra parte, las de menos interés o relevancia fueron las actividades de documentación y pruebas.

Las buenas prácticas deben institucionalizarse. La gente debe de ser capaz y responsable de seguir cada una de las prácticas que están definidas para toda la organización. Para poder ayudar a la gente a dar seguimiento a las prácticas correspondientes se puede hacer uso de herramientas especializadas. Las herramientas harán que las personas no vean al proceso como algo hostil y fastidioso. Es necesario definir qué es lo que se va a hacer, por quién y cuándo. Otro aspecto importante es el ciclo de vida de los procesos. El hecho de haber definido, documentado, medido e institucionalizado los procesos no significan que sean los mejores. Todo proceso está sujeto a cambios. Tener un mal proceso que no evoluciona representa más un obstáculo que una ayuda.

A continuación se muestran algunos beneficios que se obtendrían al utilizar una metodología de desarrollo estándar:

- Estandarización del proceso.
- Mayor control en el proceso, lo que hace más sencillo realizar seguimiento y monitoreo de actividades y artefactos de cada fase.
- Mejores resultados en la calidad del software.
- Consolidación de la documentación e información obtenida durante todo el proceso de software.
- Tener una línea base para establecer estimaciones.
- Mejorar el producto, con las métricas se detectan defectos.
- Conocimiento cuantitativo de las características del proceso del producto.

2.1.7.10 FODA

<p>Fortalezas</p> <ul style="list-style-type: none"> ➤ Se establecen procedimientos para el desarrollo de software; se cuenta con una política de priorización de Proyecto. ➤ Se proveen los recursos por parte del Centro de Informática definidos para la ejecución de los diferentes procesos planificados para cada área de proceso de forma adecuada cada vez que se necesitan. ➤ Se revisa el estado y resultados del proyecto con el Director de Informática y la alta gerencia de forma periódica por medio de reuniones con los involucrados en el proyecto. ➤ Se realizan control de versiones del código fuente de cada proyecto. ➤ Se capacita constantemente al personal en grupos en temas relacionados al giro del negocio y tecnología (Seguridad, Bases de Datos, mejores Prácticas de desarrollo, programación). 	<p>Oportunidades</p> <ul style="list-style-type: none"> ➤ Obtener pautas de control y revisión del proceso de desarrollo. ➤ Mayor agilidad y eficiencia en desarrollo del software. ➤ Mejorar productividad del personal de desarrollo. ➤ Implementación de nuevas tecnologías para mejorar la eficiencia de los procesos en el ciclo de vida del software.
<p>Debilidades</p> <ul style="list-style-type: none"> ➤ Indefinido un plan de calidad en el que se establezcan políticas para la planificación y ejecución de los procesos de desarrollo de software. ➤ A veces no se documentan las desviaciones significativas ni las acciones correctivas de forma adecuada en general estas prácticas se realizan directamente. 	<p>Amenaza</p> <ul style="list-style-type: none"> ➤ Algunas veces los requerimientos que tienen una mayor prioridad son complejos, lo cual motiva que se inviertan esfuerzos para explorarlos antes de comenzar su desarrollo para reducir el riesgo general del desarrollo. ➤ Requerimientos cambiantes por leyes o por circunstancias no tomadas en cuenta. ➤ Retrasos por imprevistos

<ul style="list-style-type: none">➤ Indefinidos criterios para la evaluación y aceptación de los requerimientos.➤ Poca documentación de las versiones de los requerimientos➤ No se estiman las tareas que deben realizarse después de un cambio de requisitos.➤ Especificaciones erróneas o incompletas.➤ Mala interpretación de la comunicación con el cliente.➤ Violación de los estándares de programación sin embargo no hay uno establecido a seguir.➤ Documentación inexacta o incompleta.➤ El analista de sistemas aparte de analizar el requerimiento tiene otras cargas que podrían ser balanceadas con otras áreas; por ejemplo el diseño y creación de bases de datos, generación de estadísticas.	<ul style="list-style-type: none">➤ Limitaciones de tiempo.
--	---

[Tabla N° 13] FODA (Elaboración propia.)

Capítulo III. Propuesta de solución para la mejora del proceso de software.

En este capítulo se realiza un análisis de alternativas y un estudio económico de la solución propuesta; y de los insumos del diagnóstico proponer una guía para el control de calidad donde la UCCC podrá aplicar una evaluación a todos los sistemas desarrollados por la UST con el fin de mejorar los controles y la calidad de su servicio reflejado en la eficiencia de los sistemas.

3.1 Análisis de alternativas

Se definió cuatro alternativas de solución, luego de un proceso de evaluación se tomará una decisión sobre la solución a implementar. Estas alternativas se describen a continuación.

3.1.2.1 Alternativa 1. Utilizar una metodología de desarrollo de software existente.

3.1.2.2 Alternativa 2. Utilizar una metodología de desarrollo de software a medida, de acuerdo a las necesidades y expectativas de los proyectos de la organización.

3.1.2.4 Alternativa 3. Contratar asesor experto en metodologías o/y aumentar los recursos humanos de programación y capacitarlos en metodologías.

3.1.2.5 Alternativa 4. Aplicar controles mínimos que verifiquen la calidad del producto tomando como base estándares y metodologías y aplicarlos al ciclo de vida actual.

Alternativa	Descripción
1	Existen metodologías de desarrollo de software existentes que utilizan algunas organizaciones y se encuentran ya documentada.
2	Crear una metodología de desarrollo de software especial, híbrida o a medida para los proyectos de la DIS, según las necesidades, requerimientos, expectativas, enfoque, tipo de proyecto, prioridades y estrategia de la organización.
3	Adquirir más personal o contratar personal experto y capacitado para aumentar la eficiencia en la realización de procesos.

Alternativa	Descripción
4	<p>Aplicar evaluación del software, en la etapa de entrega del producto a la UST donde control de calidad utilizando métricas, mejores prácticas de estándares de calidad, software y seguridad, verifique los cumplimientos.</p> <p>Los puntos a tomar en cuenta pueden mejorarse.</p>

[Tabla N° 14]. Descripción de las alternativas de solución (Elaboración propia).

Criterios de selección de una metodología:

Tomando en cuenta el estudio del estado del arte y la encuesta realizada, uno de los factores de la no aplicación de metodologías durante el ciclo de vida del software es la disponibilidad de tiempo de los programadores por lo que la elección de una metodología debe considerar, antes de seleccionarse los siguientes aspectos:

- Presupuesto de la institución
- Metodología de fácil comprensión
- Tamaño del proyecto
- Tipo de proyecto
- Tamaño del equipo de trabajo
- Tiempo limitado de entrega
- Necesidades de documentación
- Personal necesario
- Adaptabilidad y respuesta al cambio
- Capacidad individual de los programadores
- Nivel tecnológico
- Metas adecuadas
- Debe incluir formas de evaluación o auditorías del proceso.

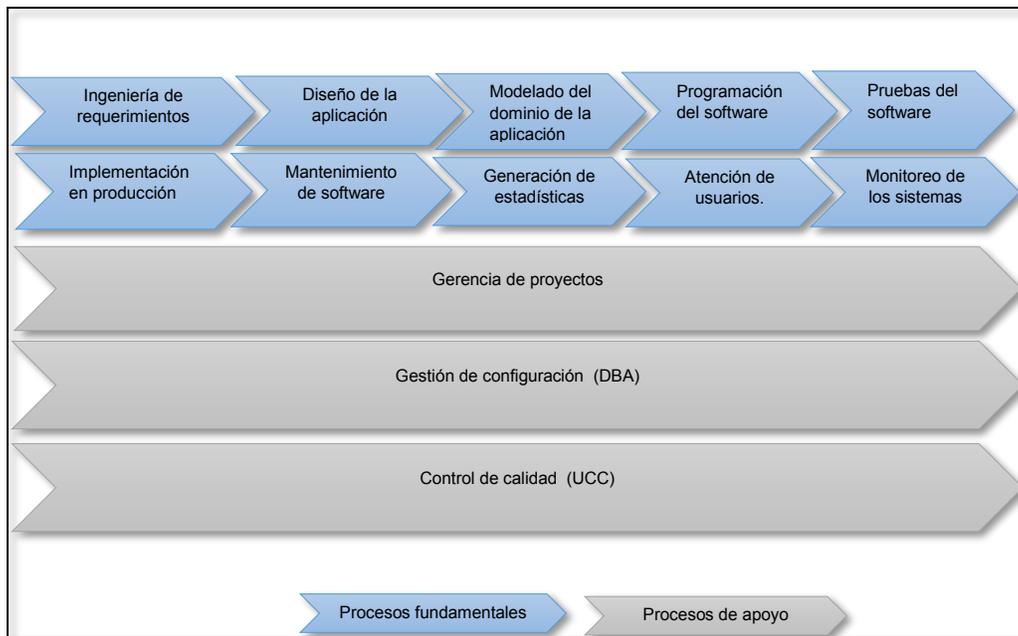
Selección de alternativa

Los programadores y analistas de control de calidad están dispuestos a implementar una metodología sin embargo no es un asunto sencillo de la noche a la mañana, requiere de presupuesto, desarrollo de una cultura, compromiso desde la alta gerencia hasta las áreas involucradas y otros factores, no se trata de solo capacitar al personal u obtener una certificación por un curso recibido sino de aplicarlo y eso conlleva a una serie de procesos a seguir.

Lo ideal es implementar una metodología y el personal sea capacitado tanto en el uso como en la revisión del cumplimiento de ésta, pero por el momento la alternativa que se propone implementar es la numero 4 “Aplicar controles mínimos que verifiquen la calidad del producto tomando como base estándares y metodologías y aplicarlos al ciclo de vida actual” por ser flexible a cambios, aplicable a corto plazo y mejorable tras la implementación de un proyecto, requiere poco presupuesto, lo único que es exigido es el compromiso de realización para todo proyecto por muy pequeño que sea, esto se puede utilizar como la ante sala para la aplicación de la metodología que mejor se ajuste a la DIS, ya sea una ágil o una hibrida y así llevar un mejor control de los proyectos.

Cadena de valor.

A continuación se muestra cadena de valor propuesta, donde se identifican los procesos de importantes del área:



[Figura. N° 25] Cadena de valor del proceso de desarrollo propuesto. (Elaboración propia)

3.2 Diagramas del proceso propuesto

A continuación se muestra el flujo propuesto por cada etapa del ciclo de vida del software.

Descripción del proceso de software

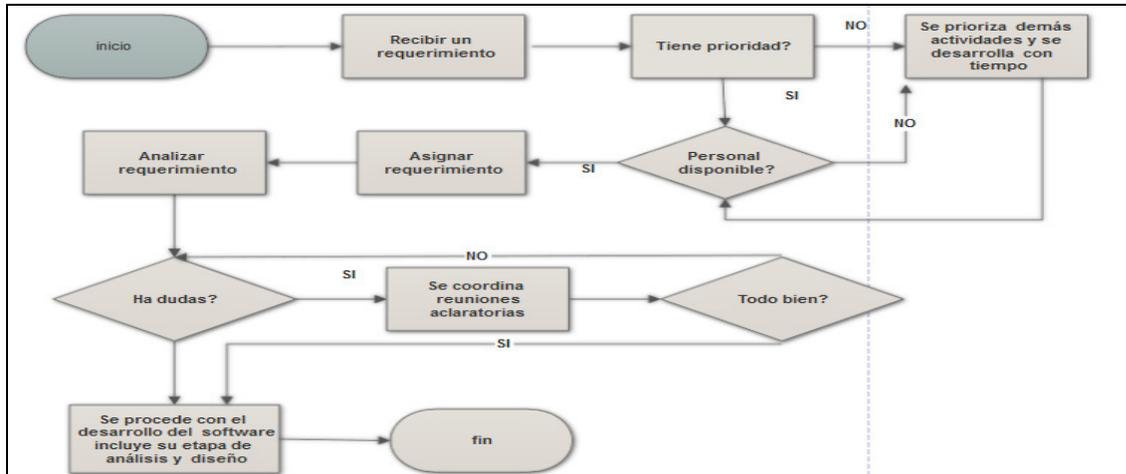
A continuación se realiza una breve descripción de cada uno de los procesos propuestos para el desarrollo de software:

Subproceso: Recepción de requerimientos

Comprende desde la recepción, realizando las actividades necesarias de revisión del cumplimiento de la calidad del requerimiento para su atención oportuna, pasando por las etapas de investigación, reuniones con el cliente solicitante y aceptación del requerimiento.

El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Dentro del proceso de análisis es fundamental que a través de una colección de requerimientos funcionales y no funcionales, el desarrollador o desarrolladores del software comprendan completamente la naturaleza de los programas que deben

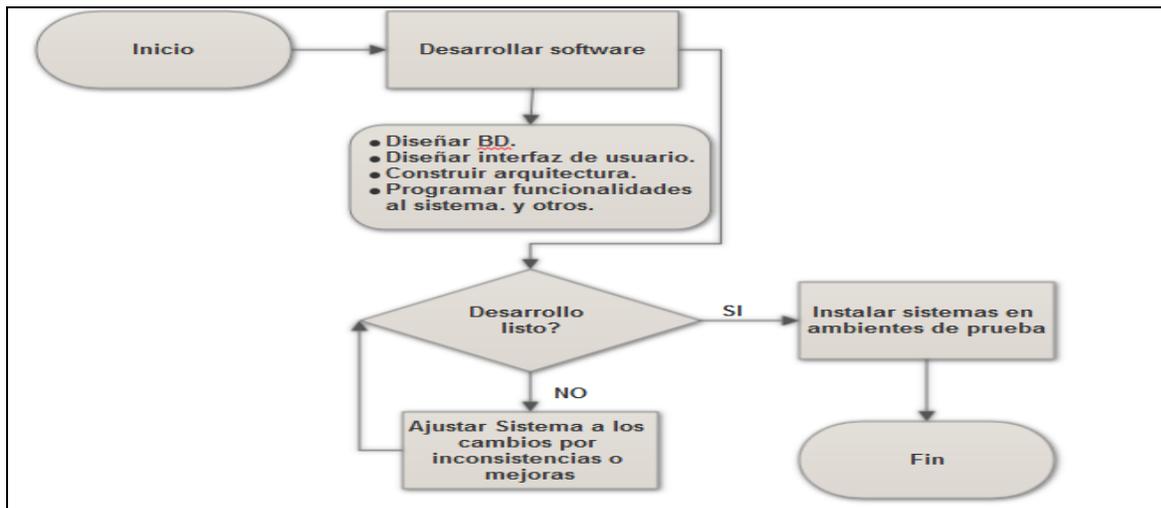
construirse para desarrollar la aplicación, la función requerida, comportamiento, rendimiento e interconexión.



[Figura. N° 26] Flujograma Planificación del desarrollo. (Elaboración propia).

Subproceso: Análisis, diseño y desarrollo de software.

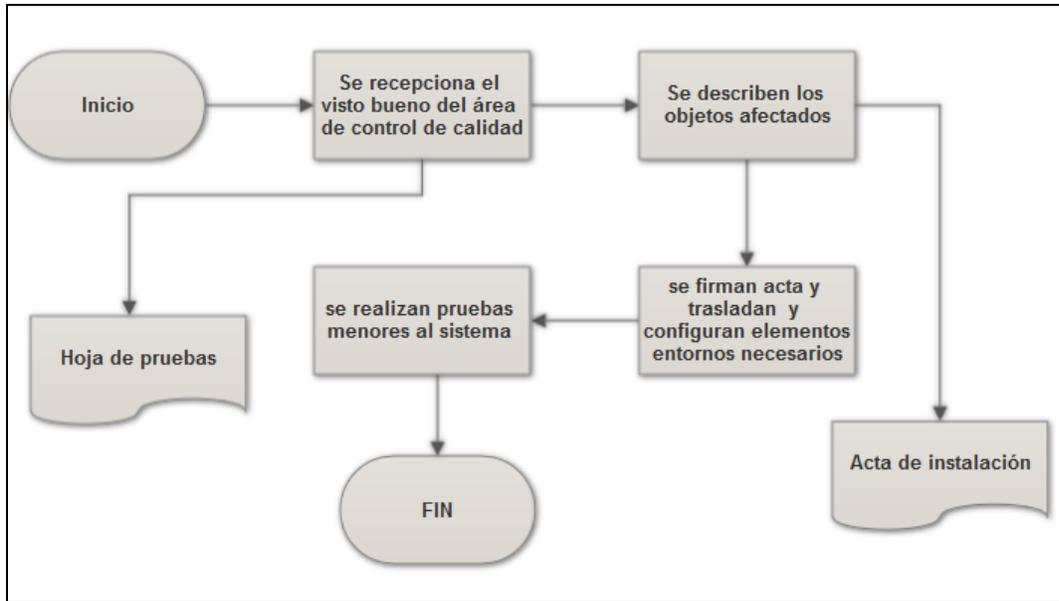
En esta parte se describe brevemente los elementos del diseño, análisis y desarrollo, es decir traducir los requisitos expuestos por al área solicitante a un prototipo o sistema.



[Figura. N°27] Flujograma Diseño, análisis y desarrollo. (Elaboración propia)

Subproceso: Implementación en producción.

A continuación se muestra el flujo propuesto para el subproceso de implementación, donde luego de realizar las pruebas pertinentes y obtener los artefactos que evidencian el proceso y la aceptación por control de calidad se procede a actualizar ambiente productivo.



[Figura. N°28] Flujograma implementación de software. (Elaboración propia).

Conclusiones.

En los últimos tres años sistemas como la VET en el cual se realiza el 99% de los trámites permiten que se recaude el 94% de los ingresos, el objetivo de los sistemas de cara al contribuyente es de facilitar el cumplimiento de las obligaciones tributarias y elevar la cultura tributaria, y los demás sistemas facilitar a la Administración Tributaria el manejo y control de dichos trámites, así como auditar y ser auditable con facilidad, por lo que es importante la calidad de sus sistemas y la administración de estos desde su concepción hasta su implementación en producción.

Primero.

La DIS actualmente utiliza Metodologías técnicas de desarrollo pero no cuenta con estándares definido para el proceso de desarrollo de software y control de calidad, por lo que el desarrollador y analista de calidad decide sus propios controles y formas de trabajo conforme a su conocimiento y tiempo dado que no está normado; la mayoría de sus desarrollos son justo a la medida sin embargo hay desarrollos por terceros para los cuales también es un aspecto opcional; todos están conscientes de lo necesario que es implementar una metodología, lograr esto es un trabajo continuo y adquirir un compromiso que requiere un presupuesto.

Segundo.

De forma general, utilizando un híbrido de estándares y metodologías, se concluye que el proceso de desarrollo a pesar de ciertas limitaciones abordadas en el árbol de problemas y diagnóstico del proceso, es mejorable tanto para el producto como el proceso por lo que se cumple con la implantación de los procesos de forma eficaz, lo cual significa que funcionalmente los sistemas en su mayoría cumplen el propósito por el cual fueron creados, algunos sistemas presentan incidencias y fallas relacionado a omisión de algún proceso ya sea en el requerimiento, en el desarrollo o prueba previo a su instalación en producción por lo que son puntos negativos, sin embargo el hecho de aplicar una metodología no significa que nunca haya un fallo en el proceso o en el producto pero sí reduce la brecha y mejora el control del proceso y su calidad, a su vez

se trabaja de una forma estructurada y estándar que permite la mantenibilidad del software y la inclusión de nuevos integrantes al equipo que administra el sistema. En cuanto a su seguridad hay un refuerzo de configuraciones como protocolos y monitoreos sobre los sistemas y se aplican criterios de seguridad a nivel de aplicación pero no son verificables por el equipo de control de calidad, más que por el analista de desarrollo lo cual no es tan recomendable.

Tercero. Los controles de calidad se basan en su mayoría en el resultado de las pruebas de caja blanca, funcionales, integrales y del control de incidencia; en cuanto a calidad empezando por que no se utiliza una metodología o estándar para su evaluación según los estándares existentes no hay evidencias que soporte el cumplimiento de algunos aspectos, sin embargo se procedió a evaluar la calidad según los aspectos abordados en entrevistas con jefes.

Resulta importante utilizar un modelo o estándar de calidad como guía para el proceso de evaluación de la calidad de los productos de software, para un mejor control y evaluación de la calidad.

Recomendaciones

De acuerdo con el estudio realizado, y los resultados obtenidos se determina importante que la organización adopte las siguientes recomendaciones en su proceso. Estas permitirán fortalecer el proceso de desarrollo de software:

- Es importante que exista una base de conocimiento de los procedimientos que ejecutan en el área, con el fin de evitar re-procesos por el desconocimiento.
- Definir una estructura básica de lo que debe contener un requerimiento y al ser remitido a la DIS esté con el aprobado de las áreas involucradas con el contenido.
- Estandarización gradual de los procesos de desarrollo y gestión de calidad.
- Se recomienda implementar las buenas prácticas que ofrece la guía de OWASP, esta guía incluye: pruebas de seguridad, pruebas funcionales, pruebas unitarias, pruebas de caja de negra, entre otras.
- Capacitar a los usuarios normativos en herramientas y métodos para mejorar la forma en que actualmente se elaboran los requerimientos.
- Introducir al equipo de control de calidad y bases de datos como parte del proyecto desde un principio para que vaya estructurando su plan de pruebas, diseño de base de datos y luego se le permita revisar ciertos aspectos del software relacionados a la calidad.
- Involucrar al área de bases de datos en el proceso de diseño de bases de datos aprovechando su especialidad; esto en la etapa de diseño del sistema; para ello proporcionar requerimiento o especializar un área encargada de la parte de diseño de interfaz, base de datos, metodología y revisión del cumplimiento.

Glosario de términos

- **Analista de sistemas:** Persona que analiza el requerimiento, diseña bases de datos e interfaz de los sistemas de información; se encarga de la documentación e implementación de los mismos en los ambientes pre producción.
- **CMMI:** Integración de modelos de madurez de capacidades o Capability Maturity Model Integration (CMMI) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.
- **Calidad funcional.** Refleja en qué medida el *software* cumple con o se ajusta a un determinado diseño, basado en requerimientos funcionales. Éstos abarcan las actividades del *software* que involucran procesamiento de datos de entrada.
- **Calidad estructural.** Refleja en qué medida el software cumple con los requerimientos no funcionales, como rendimiento, capacidad de mantenimiento o escalabilidad.
- **Eficacia:** Medida del grado de cumplimiento de los objetivos. Una actividad es eficaz si consigue los resultados que pretendía conseguir.
- **Eficiencia:** Medida de la utilización de los recursos necesarios para conseguir un objetivo. Una actividad es eficiente si reduce al mínimo la utilización de los recursos, o produce mejores resultados con los mismos recursos.
- **Evaluación,** según la es la “Estimación de la eficacia, eficiencia, utilidad y relevancia de un servicio o instalación”.
- **Gobierno Electrónico:** Es la aplicación de las tecnologías de la información y la comunicación (TIC) al funcionamiento del sector público, con el objetivo de brindar mejores servicios al ciudadano e incrementar la eficiencia, la transparencia y la participación ciudadana.

- **IEEE** (Instituto de Ingeniería Eléctrica y Electrónica): es una asociación mundial de ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas.
- **ISO/IEC**: (International Organization for Standardization) e IEC (International Electrotechnical Commission) estándar reconocido internacionalmente en gestión de servicios de TI (Tecnologías de la Información).
- **ISO 27001**: es una norma internacional que permite el aseguramiento, confidencialidad e integridad de los datos y de la información así como de los sistemas que la procesa.
- **ISO 19011**: es una norma internacional desarrollada por la Organización Internacional de Normalización (ISO), que establece las directrices para la auditoría de los sistemas de gestión de la calidad.
- **Impacto** según (Bazó, 2005) son los cambios o efectos positivos y negativos producidos directa o indirectamente en una situación inicial, como resultado de la ejecución de un programa o proyecto.
- **Implementación**: Es la realización de una especificación técnica o algoritmos como un programa, componente software, u otro sistema de cómputo. Muchas implementaciones son dadas según a una especificación o un estándar.
- **Instalación**: Es el proceso fundamental por el cual los nuevos programas son transferidos a un computador con el fin de ser configurados, y preparados para ser ejecutados en el sistema informático, para cumplir la función para la cual fueron desarrollados.
- **Indicador**: Magnitud utilizada para medir o comparar los resultados efectivamente obtenidos, en la ejecución de un proyecto, programa o actividad. Resultado cuantitativo de comparar dos variables.

- **La trazabilidad:** Serie de procedimientos que permiten seguir el proceso de evolución de un producto en cada una de sus etapas.
- **MDS (Metodologías de desarrollo de software):** Término que se refiere a las diversas metodologías de desarrollo de software existentes de forma general.
- **NAGIOS:** es un sistema de monitorización de redes ampliamente utilizado, de código abierto, que vigila los equipos y servicios que se especifiquen, alertando cuando el comportamiento de los mismos no sea el deseado.
- **NEXUS:** Sistema de registro de requerimientos y solicitudes.
- **Disponibilidad:** Capacidad de un sistema de estar operativo y accesible para su uso cuando se necesite.
- **Offshoring:** Subcontratación internacional o deslocalización (del inglés offshoring) es una subcontratación de procesos de negocios de un país a otro, usualmente en busca de costos más bajos o mano de obra. Incluye procesos como producción, manufactura, servicios e incluso innovación o investigación y desarrollo (I+D)
- **Proceso:** Es un conjunto de actividades inter relacionadas que transforman entradas en salidas. Un proceso define quién, qué, cuándo, y cómo, para alcanzar un determinado objetivo.
- **Test de aceptación:** el cliente prueba el software y verifica que cumpla con sus expectativas. Como no tiene acceso a la implementación interna de la aplicación sólo se evalúan las entradas que se realizan sobre el software y las salidas que produce, sin preocuparse por su comportamiento interno.

- **TIC (Tecnologías de la información y la comunicación):** El término tecnologías de información y comunicación tiene dos acepciones. Por un lado, se utiliza con bastante frecuencia el término 'tecnologías de la información'. Este lo hace para referirse a cualquier forma de hacer cómputo.
- **TI (La tecnología de la información)** (TI, más conocida como IT por su significado en inglés, information technology) es la aplicación de ordenadores y equipos de telecomunicación para almacenar, recuperar, transmitir y manipular datos.
- **Verificación:** Método empleado para garantizar que el producto resultante de una actividad o fase del desarrollo cumple con los requisitos de esa actividad o fase en lo relativo a corrección, calidad, rendimiento, cumplimiento de agendas y usabilidad.
- **Validación:** Método que garantiza que el producto es conforme para el uso que tiene previsto.

Bibliografía

1. Acosta, N. J., Espinel, L. A., & García, J. L. (2017). Estándares para la calidad de software. *Tecnol.Investig.Academia Tia*, 5(1). Obtenido de <http://revistas.udistrital.edu.co/ojs/index.php/tia/article/view/8388/pdf>
2. agesic. (16 de 02 de 2019). Modelo de Calidad de Software (MCS). Obtenido de <https://centroderecursos.agesic.gub.uy/web/optimizacion-de-ti/modelo-de-calidad-de-software>
3. Alonso Amo, F. L. (2005). Introducción a la ingeniería del software. En E. J. B. Rubio (Ed.). Zaragoza, España: DELTA publicaciones. Obtenido de https://books.google.com.ni/books?id=rXU-WS4UatYC&pg=PR5&lpg=PR5&dq=Introducci%C3%B3n+a+la+ingenier%C3%A9a+del+software.+Alonso+Amo,+F.,+Lo%C3%AFc,+M.&source=bl&ots=wtKxdZiV-&sig=bH7eKs-5DXozzfzqGr_a9dTBriQ&hl=es&sa=X&ved=2ahUKEwjQ06r2k-XfAhWjl-AKHXE0CnA
4. Arias, F. (2006). El proyecto de investigación: Introducción a la metodología científica. (5ta ed.). Caracas - Venezuela: Episteme.
5. Bayona, J. C., Pineda, O. L., & Pardo, O. D. (14 de 01 de 2015). El papel de la Ingeniería de Software en el desarrollo de aplicaciones. *TIA*, 4(1).
6. Bazó, R. M. (2005). Una propuesta para la evaluación de programas sociales: Lineamientos generales. *Red de Revistas Científicas de América Latina y el Caribe, España y Portugal*, 11(1). Obtenido de <https://www.redalyc.org/articulo.oa?id=33611105>
7. Bórdon, N. (12 de 03 de 2013). NORMA DE EVALUACIÓN ISO/IEC 9126. Obtenido de <http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html>
8. Cantone, D. (2006). Implementación y Debugging. Mp Ediciones Corp.
9. Castro, M. (2003). El proyecto de investigación y su esquema de elaboración. Caracas: Uyapal.
10. Cristina Lopez, J. L. (2008/2009). Obtenido de FUNDAMENTOS DE SISTEMAS DE INFORMACIÓN:

http://phpwebquest.org/cursocep/webquest/soporte_tabbed_w.php?id_actividad=18518&id_pagina=1

11. CUARTAS, D. H. (2008). Principios de la administracion. En &. L. L. Valencia (Ed.). Medellin, Colombia: INSTITUTO TECNOLOGICO METROPOLITANO. Obtenido de <https://books.google.com.ni/books?id=1Fp55-1oXv8C&pg=PA132&lpg=PA132&dq=Un+Si+es+un+conjunto+formal+de+procesos+que,+operando+sobre+una+colecci%C3%B3n+de+datos+estructurada+seg%C3%BA+las+necesidades+de+la+empresa,+recopilan,+elabora+y+distribuyen+la+inf>
12. DATADEC. (25 de 04 de 2018). Obtenido de CÓMO IMPLANTAR UN SOFTWARE DE GESTIÓN EMPRESARIAL CON ÉXITO: <https://www.datadec.es/blog/como-implantar-un-software-de-gestion-empresarial-con-exito>
13. Elkan, M. (18 de 09 de 2017). Ingeniería d software. Obtenido de <http://tic01-ingenieria-de-software-lau.blogspot.com/2017/09/ventajas-y-desventajas-de-metodologias.html>
14. Fuente, F. G. (2004). Los sistemas de información en la sociedad del conocimiento. MADRID, ESPAÑA: ESIC Editoria. Obtenido de <https://books.google.com.ni/books?id=94sv48wCJAMC&pg=PA47&dq=Los+elementos+que+forman+un+Sistema+de+informacion&hl=es&sa=X&ved=0ahUKEwi42a67gr3bAhVB0IMKHSflBnQQ6AEIKjAB#v=onepage&q&f=false>
15. Garciatorres, J. V. (2008). Guía De Los Fundamentos De La Dirección De Proyectos. Project Management Institute (PMI). Obtenido de https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjP6u_JvrrbAhXQ2VMKHaE6AGIQFggIMAA&url=http%3A%2F%2Femem.uniovi.es%2F%2Fdocument_library%2Fget_file%3Fuuid%3D2abb8e4b-78c3-4b02-879c-27541dcc1a7d%26groupid%3D517164
16. Gloria P. Gasca-Hurtado1, J. A. (2017). Evaluación de mejora de procesos software para entornos multi-modelo. International Journal of Information Systems and Software Engineering for Big Companies (IJSEBC).

17. Herrera, J. A. (2012). Modelos de Madurez en los Datos de una Organización. revistasbolivianas. Obtenido de http://www.revistasbolivianas.org.bo/scielo.php?pid=S1683-07892012000200004&script=sci_arttext
18. IEE. (1990). Computer Dictionary. Estándar. IEE. doi:10.1109/IEEESTD.1991.106963
19. IEEE. (2008). Especificación de Requisitos según el estándar de IEEE 830. Recuperado el 07 de 02 de 2019, de <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
20. ingertec. (13 de 02 de 2018). ISO/IEC 12207:2008. Obtenido de ingertec: <http://ingertec.com/iso-15504/iso-iec-12207/>
21. INTECO. (2009). INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA. España. Obtenido de https://kejochama.files.wordpress.com/2013/03/guia_ingenieria_software.pdf
22. IRAM-ISO/IEC9126-1. (s.f.). Tecnología de la Información. Ingeniería de Software. Calidad del Producto. Parte 1-Modelo de Calidad.
23. Iso 9000-2000. (01 de 01 de 2019). ISO 9000-2000. Obtenido de ISO 9000-2000: <https://www.iso.org/home.html>
24. ISO/IEC/IEEE 12207:2008(E). (2008). ISO/IEC/IEEE 12207:2008(E). Vernier, Geneva, Switzerland: Institute of Electrical and Electronics Engineers, Inc. Obtenido de ISO/IEC 12207:2008: ingertec
25. Jimmy R. Molina, M. P. (2017). ESTADO DEL ARTE: METODOLOGÍAS DE DESARROLLO EN APLICACIONES WEB. 3C Tecnología, 6(3). doi:<http://dx.doi.org/10.17993/3ctecno.2016.v6n3e23.54-71>
26. Kendall, K. E. (2005). Análisis y diseño de sistemas (Vol. 6). Mexico: Pearson Educación.
27. Imendoza, m. a. (2005). Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software. Computación y sistemas. Obtenido de <https://docplayer.es/16998030-Prototipo-de-modelo-sistemico-de-calidad-mosca-del-software-prototype-of-software-quality-systemic-model-sqsm.html>

28. Lynch, Q. w. (04 de 10 de 2015). Obtenido de Chaos Report: <https://www.infoq.com/articles/standish-chaos-2015>
29. McCall, R. &. (1977). Factores en la calidad del software. Roma: Development Center.
30. microsoft. (01 de 11 de 2018). Obtenido de Valores de las métricas de código: <https://docs.microsoft.com/es-es/visualstudio/code-quality/code-metrics-values?view=vs-2017>
31. Molina Ríos, J. Z. (2018). COMPARACIÓN DE METODOLOGÍAS EN APLICACIONES WEB. 3ciencias. doi:<http://dx.doi.org/10.17993/3ctecno.2018.v7n1e25.1-19>
32. Morales, E. (23 de 05 de 2010). ¿Por qué es importante implementar Modelos y Estándares de Calidad de Software? Obtenido de <https://estandarsw.wordpress.com/category/estandares-del-ciclo-de-vida-del-software/>
33. Muñoz, M., Mejía, J., Calvo, J., Cuevas, G., & San Felio, T. (2012). Evaluación de los procesos software de una organización enfocado en la disminución de la resistencia al cambio. Computer Science and Engineering, 1-4.
34. Navarro Cadavid, A., Fernández Martínez, J. D., & Morales Vélez, J. (20 de 09 de 2013). Revisión de metodologías ágiles para el desarrollo de software. 11(2). Recuperado el 10 de 11 de 2018, de <https://www.redalyc.org/pdf/4962/496250736004.pdf>
35. OIT. (18 de 02 de 2019). Obtenido de CINTENFOR: <http://guia.oitcinterfor.org/como-evaluar/como-se-construyen-indicadores>
36. Paul J. Gertler, S. M. (2011). La evaluación de impacto en la práctica. Washington DC 20433: BANCO MUNDIAL. doi:10.1596/978-0-8213-8541-8
37. Perse, J. R. (2007). Project Management Success with CMMI, Seven CMMI Process Areas. Prentice Hall.
38. Pressman, R. (2002). Ingeniería del Software. Un enfoque práctico. Mexico: McGraw-Hill Interamericana.
39. Ramos, M. D. (05 de 2013). LA INGENIERÍA DE SOFTWARE APLICADA A LA MEDICIÓN DE LA CALIDAD DE LOS SISTEMAS HIPERMEDIA. Obtenido de

- https://www.researchgate.net/figure/Figura-4-Mapeo-del-modelo-de-McCall-con-el-ISO-IEC-9126-Adaptado-de-Castaneda-et-al_fig1_271843274
40. Redrován Castillo, F. L. (2017). Estado del arte: Métricas de calidad para el desarrollo de aplicaciones web. 3C Tecnología, 24(6). doi:<http://dx.doi.org/10.17993/3ctecno.2017.v6n4e24.1-12>
 41. Rivas, C. C. (DICIEMBRE de 2015). Metodologías actuales de desarrollo de software. Revista Tecnología e Innovación. Revista Tecnologica e innovación, 2(5). Obtenido de http://www.ecorfan.org/bolivia/researchjournals/Tecnologia_e_innovacion/vol2num5/Tecnologia_e_Innovacion_Vol2_Num5_6.pdf
 42. Rodríguez, C. T. (2017). Impacto de los requerimientos en. 5(2). Obtenido de <https://revistas.udistrital.edu.co/ojs/index.php/tia/article/viewFile/7607/pdf>
 43. Rodríguez, G. J. (15 de 01 de 2012). Técnicas efectivas para la toma de requerimientos. Obtenido de https://www.northware.mx/tecnicas_efectivas_toma_requerimientos/
 44. Ruvalcaba, M. (05 de 03 de 2019). Procesos de Software. Obtenido de <https://sg.com.mx/revista/1/procesos-software>
 45. Santander. (2008). Ingeniería de programación: Procesos de desarrollo de software.
 46. SOLÍS, S. (2015). COMPENDIO DE INGENIERÍA DE SOFTWARE. En & C. I.. Obtenido de https://books.google.com.ni/books?id=3dZ5CwAAQBAJ&printsec=frontcover&dq=COMPENDIO+DE+INGENIER%C3%8DA+DE+SOFTWARE&hl=es&sa=X&ved=0ahUKEwj20_KZluXfAhWsT98KHUK5Bm8Q6AEIJzAA#v=onepage&q=COMPENDIO%20DE%20INGENIER%C3%8DA%20DE%20SOFTWARE&f=false
 47. Sommerville, I. (2005). Ingeniería del software (Vol. 7). Madrid: Pearson Educación.
 48. sparxsystems. (07 de 02 de 2019). Obtenido de sparxsystems: http://www.sparxsystems.com.ar/products/ea_editions.html
 49. Tamayo, M. (1997). El Proceso de la Investigación científica. México: Limusa S.A.

50. The Standish Group International, Inc. (2015). CHAOS REPORT. INFOQ.
Obtenido de
[https://www.standishgroup.com/sample_research_files/CHAOSReport2015-
Final.pdf](https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf)

Apéndice

A: Evaluaciones

A.1 Encuesta UST

El objetivo de esta encuesta es evaluar el impacto de la aplicación de metodologías de software en el proceso de implementación de los Sistemas Tributarios de la División de Informática y Sistemas de la Dirección General de Ingresos.

Muestra o Población: Personal del área de desarrollo UST de la DIS.

A.1.1 Administración de requerimientos:

- ¿Quién define los requerimientos?
 DIS Otros _____
 Usuarios Normativos/Admón.
- ¿Existe una estructura establecida para la elaboración de un requerimiento?
 SI NO
▪ Si, su respuesta es sí: ¿Se cumple con la estructura establecida?
 SI NO
- ¿Se entiende con facilidad los requerimientos recibidos?
 SI NO
- ¿Se gestionan los cambios en los requerimientos?
 SI NO
- ¿Existe alguna práctica definida para manejar la trazabilidad de los requisitos?
 SI NO

A.1.2 Prácticas de diseño, análisis, desarrollo e implementación de un proyecto.

- ¿Utiliza alguna metodología de desarrollo de software para el diseño y programación de sistemas informáticos?
a. SI
o ¿Qué metodología de desarrollo de software está usando actualmente?
 Ágil Ninguna
 Tradicional Otros _____
o ¿Se siente cómodo usando la mencionada metodología de desarrollo de Software?
 SI
 NO

[H1] Hoja de encuesta UST (Elaboracion propia)

¿Le resulta complicado seguir los procesos de la mencionada metodología de desarrollo de software?
 SI NO

¿Siente que el uso de la mencionada metodología de desarrollo de software ha contribuido a mejorar la calidad de los sistemas informáticos resultantes?
 SI NO Lo desconozco

b.No

¿Por cuáles de las siguientes razones no utiliza una metodología de software?
 No me gusta Complejidad del proyecto Plazos de entrega cortos
 Lo considero innecesario, lo que importa es el funcionamiento.
 Complejidad de la metodología Tengo procedimientos propios a seguir
 Desconozco de metodologías Es opcional, nadie la solicita.
 Todas las anteriores Otros _____

¿Cuál de los siguientes procesos considera que ha influido en el proceso de desarrollo de software?
 La gestión de requerimientos La gestión de cambios
 La gestión de riesgos Cambios de prioridades
 Compresión del requerimiento Estimación de los tiempos.
 Uso de metodologías y estándares Formas de programación y diseño
 Compromiso Gestión del conocimiento
 Planificación de proyectos Gestión de documentación
 Experiencia del equipo en programación y diseño. Tecnología aplicada
 Disponibilidad Recursos físicos, hardware o software
 Sobre carga de asignaciones Todas las anteriores

¿Estaría dispuesto a seguir los lineamientos de una metodología de desarrollo de software con la única finalidad de mejorar la calidad los sistemas informáticos resultantes?
 SI NO ¿Por qué? _____

6 ¿Se realiza un análisis de los requerimientos versus los recursos disponibles y procesos existentes relacionados a lo que se solicita?
 SI NO

7 ¿Se realiza modelado o diagramas de los proceso del sistema y de la base de datos relacionada?
 SI NO

8 ¿Se utilizan estándares de calidad, seguridad, de mejores prácticas en el durante el proceso de desarrollo?
 SI NO Lo desconozco Muy poco

[H2] Hoja de encuesta UST (Elaboracion propia)

9 ¿Cuál de las siguientes herramientas utiliza para diseñar diagramas o modelos?
 Rational Rose Enterprise Architect Toad Data Modeler Ninguno
 Visual Studio Microsoft Visio Otros _____

10 ¿Tiene costumbre de realizar comentarios dentro del código del programa?
 SI NO Algunas veces

11 ¿Existen políticas o estándares establecidos que deben seguirse al desarrollar un software?
SI NO Lo desconozco

12 ¿Realiza horas extras de desarrollo para cumplir con lo planificado?
 SI NO Algunas veces

13 ¿Cuál de las siguientes técnicas ampliamente difundidas utiliza?:
 Análisis coste beneficio Diagramas UML Diagramas de Flujo de Datos
 Diagrama de Interacción Diagramas de paquetes Diagramas de estados
 Modelado de procesos Modelo entidad/relación Normalización y optimización
 Técnicas matriciales Todas Ninguna

A.1.3 Gestión de documentación.

14 De los siguientes procedimientos documentales ¿Cuáles se realizan?
 Manuales técnicos. Plan Migración de Datos Manuales de uso
 Diagramas UML. Modelo de Datos Actas de Entrega
 Actas de instalación Documentos de Diseño Plan de Pruebas
 Control de Cambio Plan de Instalación y Puesta en Marcha Ninguno
 Documento de Seguridad e infraestructura. Estándares o metodologías utilizados
 Análisis y Levantamiento de Requerimientos Otros _____

15 ¿Se actualiza documentación al realizarse un cambio al sistema?
 SI NO: ¿por qué razón? tiempo y carga trabajo

16 ¿La documentación está respaldada en algún repositorio y está disponible al equipo de desarrollo y responsables del proyecto?
 SI NO: ¿por qué razón? _____

A.1.4 Evaluar monitorización y control.

17 ¿Se utiliza un control de versiones del código fuente?
 SI: mencione Team foundation NO

18 ¿Se revisa periódicamente los avances de los proyecto?
 SI NO

A.2 Encuesta Control de calidad

El objetivo de esta encuesta es evaluar el impacto de la aplicación de metodologías de software en el proceso de implementación de los Sistemas Tributarios de la División de Informática y Sistemas de la Dirección General de Ingresos.

Muestra o Población: Personal del área de desarrollo UCCC de la DIS.

A.2.1 Aseguramiento de la calidad del software

1. ¿Su área realiza un plan de prueba basado en los requerimientos?
 SI NO
2. ¿Existen parámetros establecidos para medir la calidad del producto?
 SI NO
3. ¿Qué tipo de prueba se realizan a los sistemas previos a su implementación en ambiente productivo?
 Caja blanca Caja negra Funcionales Otros _____
4. ¿Se lleva un control de los defectos que se va encontrando en el software?
 SI NO A veces
5. ¿Se aplica algún estándar de calidad o forma de valorar si un sistema se puede considerar con calidad mínima permitida?
 SI NO
6. ¿Se documentan las pruebas y procedimientos utilizados?
 SI NO Muy poco En su mayoría
7. ¿Se realiza alguna evaluación posterior a la implementación de proyectos o mantenimientos?
 SI NO

[H4] Conocimiento general de la forma de evaluar calidad en UCCC (Elaboración propia)

A.3 Entrevista Jefes

El objetivo de esta encuesta es evaluar el impacto de la aplicación de metodologías de software en el proceso de implementación de los Sistemas Tributarios de la División de Informática y Sistemas de la Dirección General de Ingresos.

Preguntas:

1- ¿Existen estándares y metodología establecidos para el proceso de desarrollo de software?

SI NO N/A N/S

1.1 ¿Si su respuesta fue si: mencione al menos dos relacionadas al proceso de desarrollo?

1.2 ¿Se verifica el cumplimiento de dicho estándares y metodología por el equipo desarrollador o de control de calidad?

SI NO N/A N/S

2- ¿Tiene la DIS una metodología de desarrollo de software definida, aplicada a todos los proyectos?

SI NO N/A N/S

3- ¿La mayoría de los desarrollos son justo a la medida, pero es cierto que existen desarrollo o adquisiciones de software tributarios por terceros?

SI NO N/A N/S

3.1 ¿Si su respuesta fue si: los terceros no siempre se encargan del mantenimiento del sistema y el equipo de desarrollo interno hereda dicho sistema?

SI NO N/A N/S

3.1.1 ¿Importa la metodología que se utilizó en dicho desarrollo?

SI NO N/A N/S

4 – ¿De maneja general y de acuerdo a su experiencia cree que la aplicación o no aplicación de una metodología ha influido en la calidad del software?

SI NO N/A N/S

5 – ¿Se estima el grado de avance de los proyectos con facilidad, dando una estimación de entrega de software más exacta?

SI NO N/A N/S

Questionario 1: Aspectos generales (Elaboración propia)

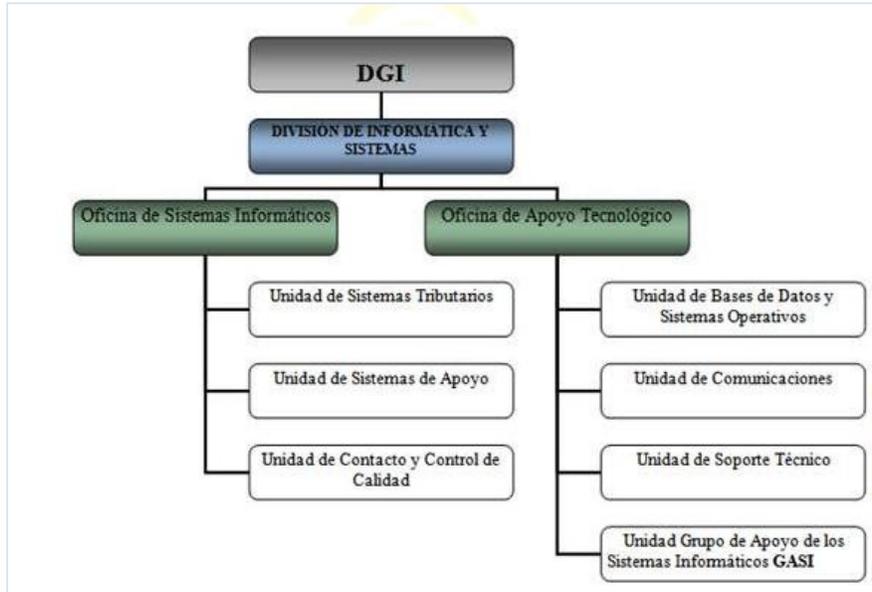
B: Estructura organizacional

B.1 Descripción de la DIS

División de informática y sistemas	
Unidad Administrativa	División de informática y sistemas
Dependencia Jerárquica	Dirección superior
Unidades subordinadas	Oficina de sistemas informáticos
	Oficina de apoyo tecnológico
Objetivo:	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 la 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.
Funciones	<ul style="list-style-type: none"> • Organizar, dirigir y controlar los servicios y sistemas automatizados de acuerdo a las leyes, reglamentos y disposiciones de la institución. • Diseñar, definir y elaborar los programas y normas que regulen el ordenamiento informático en la institución. • Recomendar, proponer e implantar nuevas aplicaciones tecnológicas en software y hardware, para mejorar los sistemas de información, equipos, normas y procedimientos de desarrollo, instalación, mantenimiento, operación y producción. • Ser miembro de la comisión en los procesos de licitación de los equipos y soportes técnicos y la elaboración de los presupuestos de inversión de tecnología a adquirirse, así como definir los requisitos que deben reunir la plataforma informática y el personal profesional que intervendrá en su desarrollo. • Administrar el mantenimiento preventivo y correctivo de toda la infraestructura informática, a fin de garantizar el funcionamiento de los sistemas automáticos y administrativos de la institución. • Apoyar los programas de análisis de informaciones económicas fiscales, para la administración tributaria, generando base de datos en herramientas y aplicaciones para obtener información estadística interna y externa a la institución. • Informar mensualmente a Planificación Estratégica del avance en las acciones del Plan Operativo Anual (POA) del Área, así como los insumos para la Evaluación del POA y otras tareas propias de esta Unidad Administrativa. • Informar mensual y periódicamente de las actividades y cambios en los sistemas de información gerencial, de control de gestión, administrativos y tributarios de la institución. • Apoyar plenamente la reforma de la DGI y el logro de los objetivos para el cambio tal como se definiera en el Plan Estratégico aprobado.

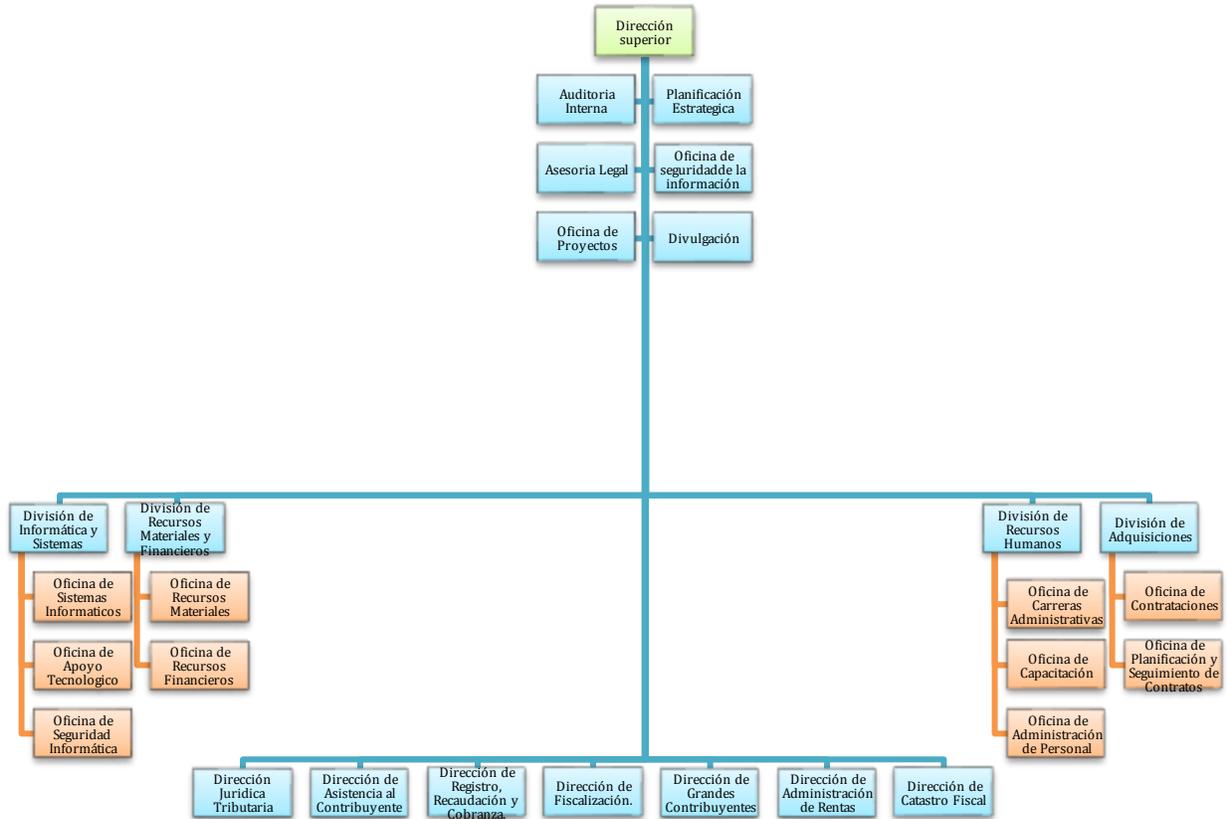
[Ficha Técnica N° 1]: Información extraída del Manual de Organización y Funciones Generales, Elaborado por Planificación Estratégica en Mayo de 2017.

B.2 Diagrama organizacional DIS



[Figura N°16] Organigrama de la DIS. (DGI-DIS, 2018)

B.3 Diagrama organizacional DGI



[Figura N°17]. Organigrama de la Dirección General de Ingresos. (DGI, 2018)

C: Otros

C.1.1 Preguntas de Investigación

Las preguntas que se planteó resolver con esta investigación fueron:

- ¿Qué procedimientos y/o metodologías utilizaron los analistas de desarrollo en la fase de análisis, diseño e implementación de software en la DIS?
- ¿Cuántos sistemas fueron implementados sin metodología y cuántos de estos fueron de resultado exitoso, medio o fracaso?
- ¿Cuántos sistemas fueron implementados con metodología y cuántos de estos fueron de resultado exitoso, medio o fracaso?
- ¿Cuántos sistemas fueron implementados utilizando parte de una metodología y cuántos de estos fueron de resultado exitoso, medio o fracaso?
- ¿En que se basan los desarrolladores para la estimación del tamaño del producto?
- ¿A qué involucrados afecta o beneficia la implementación de una metodología?
- ¿La disponibilidad del analista-desarrollador perjudica a la calidad del producto?
- ¿Existen incidencias de inconsistencias que afectaron la calidad del software que se consideran resultado de la metodología aplicada en el sistema?
- ¿Se ha cumplido con el alcance de los proyectos sin utilizar metodología?
- ¿Los sistemas existentes cumplen con alguna metodología, norma, y estándar que sirva para evaluar técnicas, procedimientos y recursos del área de desarrollo de sistemas tributarios y de apoyo?
- ¿Se evalúa los resultados obtenidos de los proyectos tras su implementación?

C.1.2 Criterios de evaluación

A continuación se propone la siguiente guía que permite evaluar el Ciclo de Vida del Desarrollo de Software. Teniendo como base el híbrido de metodologías, debido a que cada una plantea diferentes estrategias para alcanzar altos niveles de calidad y seguridad.

Identificación del programa	
Nombre: Creador: Responsables: Versión y fecha de última actualización: Dispone de prototipo: SI <input type="checkbox"/> NO <input type="checkbox"/>	
Base de datos	Flexibilidad de cambios a nivel de base de datos
	No de usuarios que pueden trabajar al mismo tiempo
	Cantidad y número de relaciones con bases de datos.
Formulación de documentos	¿El sistema permite Trabajar con documentos? ¿Qué tipo de formatos acepta?
	Los formatos y estructura de los documentos del sistema están conforme a las especificaciones de los requerimientos.
	Tamaño de fuente, justificación y formatos estándares así como la ortografía ¿están correctos?

<p>Control de documentos</p>	<p>¿Se registran las entradas de documentos?</p> <p>¿Se registra la salida de los documentos?</p> <p>¿Qué tipo de manuales tiene el sistema?</p> <p>¿Qué Tipo de documentos tiene el sistema?</p> <p><input type="checkbox"/> Documentos técnico <input type="checkbox"/> Requerimiento</p> <p><input type="checkbox"/> Documento de análisis <input type="checkbox"/> ¿Otros? _____</p>
<p>Seguridad</p>	<p>¿Accesos al sistema, tiene roles definidos y niveles?</p> <p>¿La contraseña esta encriptada?</p> <p>¿Qué tipo de encriptamiento utiliza?</p>

[Tabla N° 15] verificación propuesto para el control de calidad (Elaboración propia)

C.1.3 Método de verificación

A continuación el orden de importancia de las medidas de calidad con el orden de importancia de la calidad.

Medida	IMPORTANCIA	MCCALL	DROMEY	ISO
Facilidad de uso	5	X	X	X
Integridad	3	X		
Corrección	3	X		
Confiabilidad	4	X	X	X
Eficiencia	4	X	X	X
Facilidad de mantenimiento	5	X	X	X
Facilidad de pruebas	4	X		
Flexibilidad	3	X		
Reutilización	3	X		
Interoperabilidad	3	X		
Portabilidad	4	X	X	X
Productividad del recurso humano	3			
Facilidad de entendimiento	4			
Mantenibilidad	5			
Calidad de requisitos	5			
Estructura/Arquitectura definida	5			
Complejidad de la documentación	3			
Conformidad	3			
Calidad de los datos	3			
Efectividad de las pruebas	4			
Efectividad de la implementación	4			
Funcionalidad	5		X	X
Migrabilidad	3			
Seguridad	5			X

[Tabla N° 16]. Matriz de actividades (Elaboración propia)

Medida	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17
1-SC	R	R	R	MB	MB	B	B	B	B	B	B	B	B	B	R	R	B
1-CA	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
1-PD	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
2-SC	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
2-CA	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
2-PD	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
3-SC	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
3-CA	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
3-PD	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
4-SC	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
4-CA	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
4-PD	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
5-SC	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
5-CA	R																
5-PD	R																
SC: SUBCARACTERÍSTICA																	
CA: CARACTERÍSTICAS																	
PD: PRODUCTO																	

B:Bueno , MB=muy bueno
R: Regular ,D=Deficiente

[Tabla N° 17]. Evaluación del producto (Elaboración propia)

Métricas de mantenibilidad.

Visual Studio define ciertas métricas para medir la mantenibilidad del software, A continuación las medidas adecuadas (microsoft, 2018):

Índice de mantenimiento	Complejidad ciclomática	Profundidad de herencia	Acoplamiento de clases	Líneas de código
>50	<10	<4	<7	<25

[Tabla N° 18]. Valoración de la mantenibilidad producto (microsoft, 2018)

Se recolecto las metricas de mantenibilidad para todos lo proyectos evaluados pero para efectos de este trabajo se muestra captura de un proyecto simple y uno de crecimiento mayor.

Jerarquía	Índice de ma...	Complejidad ciclomát...	Profundidad de heren...	Acoplamiento de clases	Líneas de código
Capa.Datos (Debug)	76	135	2	87	537
Capa.Entidades (Debug)	92	832	1	21	845
Capa.Negocio (Debug)	70	157	1	32	259
Otros (Debug)	55	6	1	10	31
WebFisc (Debug)	82	109	5	70	292

[Figura N° 26]: Métrica generada por visual Studio para el proyecto 17. (Elaboración propia)

Este proyecto es del tipo simple, su índice de mantenibilidad es alto, pero al menos uno de ellos tiene una profundidad de herencia mayor a 4, tiene una alta complejidad ciclomatica por lo que se requerira mas pruebas para lograr la cobertura del codigo y sera menos facil de mantener. Por el momento para este caso es Tiene Buena mantenibilidad.

Jerarquía	Índice de ma...	Complejidad ciclomát...	Profundidad de heren...	Acoplamiento de clases	Líneas de código
1 - Presentacion (Debug)	72	7,512	4	845	14,657
1 - Presentacion Test (Debug)	95	2	1	3	2
2 - Aplicacion (Debug)	64	4,103	1	215	8,951
3 - Dominio (Debug)	100	329	0	304	0
3 - Dominio (Debug)	92	4,221	2	194	4,597
3 - Dominio (Debug)	93	386	1	6	387
3 - Dominio (Debug)	63	3,358	1	705	13,257
3 - Dominio (Debug)	69	172	1	55	541
4 - Infraestructura (Debug)	87	1,368	3	633	3,466
4 - Infraestructura (Debug)	81	39	1	60	106
4 - Infraestructura (Debug)	74	58	1	51	203
4 - Infraestructura PU (Debug)	79	13	1	11	41
4 - Infraestructura (Debug)	76	280	3	195	477
4 - Infraestructura (Debug)	76	141	2	139	483
4 - Infraestructura (Debug)	87	3	1	2	7
ManejadorTests2 (Debug)	83	2	1	4	4
ests (Debug)	96	7	1	4	8
ests (Debug)	100	2	1	3	2

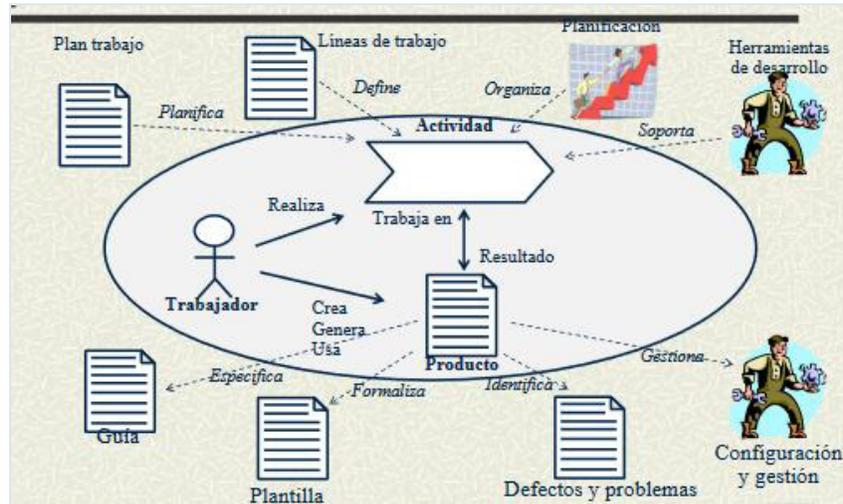
[Figura N° 27]: Métrica generada por visual Studio para el proyecto 4. (Elaboración propia)

En cambio este proyecto está catalogado por la cantidad de líneas de código como difícil, sus índices muestra que es altamente difícil de mantener.

Anexos

A: Ingeniería de software

A.1 Elementos básicos de un proyecto



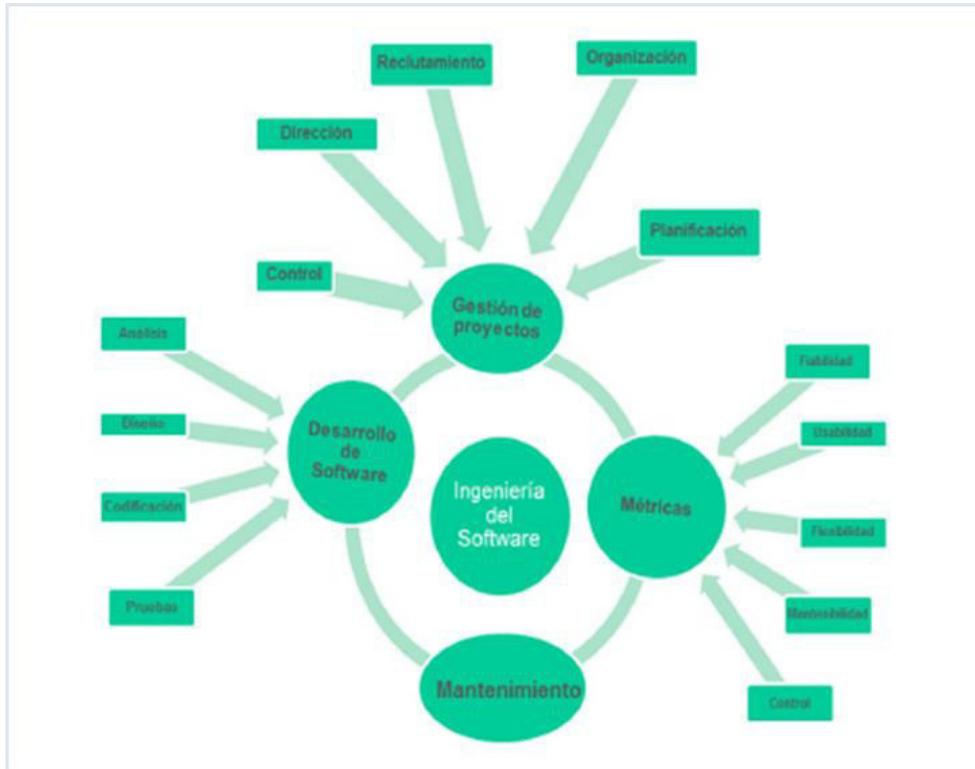
[Figura. N°1]. Elementos básicos de un proyecto. (Santander, 2008)

A.2 Factores de éxito de proyecto

CHAOS REPORT '15. Factores de éxito		
Factores éxito	P.	Evaluaron...
Apoyo de la dirección	15	Respaldo financiero y emocional.
Madurez emocional	15	Cómo trabajan juntas las personas: Comunicación, delegación, asertividad...
Implicación del usuario	15	Usuarios involucrados en toma de decisiones y proceso de recolección de información.
Optimización	15	Procesos, Valor relativo de Negocio.
Personal calificado	15	Tanto en el negocio como en la tecnología.
Arquitecturas y marcos estándar.	8	Prácticas, servicios y productos consistentes.
Procesos ágiles	7	Equipo y propietario del producto son expertos en metodologías ágiles.
Ejecución moderada y limitada.	6	Automatización en ejecuciones + Moderación en el uso de herramientas de PM.
Experiencia en P. Management.	5	Experiencia previa.
Objetivos de Negocio claros.	4	Visión, estrategia: Para todas las partes implicadas.

[Figura. N° 2] Factores de éxito (The Standish Group International, Inc, 2015)

A.3 Proceso ingeniería de software



[Figura. N° 5] Ingeniería de software (Bayona, Pineda, & Pardo, 2015).

A.4 Comparación de ciclos de vida.



[Figura. N°6] Ciclo de vida de metodologías. (Elkan, 2017)

A.5 Factores de la metodología.

FACTOR	TRADICIONAL	ÁGIL		
Prioridad de negocio	Cumplimiento	Valor		
Tiempo del proyecto	Largo	Corto		
Tamaño del equipo	Grande	Pequeño		
Interacción con el cliente	Mínima	Máxima		
Ambiente del desarrollo	Controlado	Incertidumbre		
Dirección	Organizativa	Colaborativa		
Rigidez del producto	Cerrado	Ampliable		
Requisitos	Claros	Ambiguos		
Riesgo de fallos	Bajo	Alto		
Enfoque de desarrollo	Procesos	Personas		
Necesidad de documentación	Alta	Baja		
Probabilidad de cambios en el equipo	Alta	Baja		
Roles intercambiables	No flexible	Flexible		
VALORES	1	2	3	4

[Tabla N° 19] Factores más importantes que intervienen en el desarrollo del proyecto.

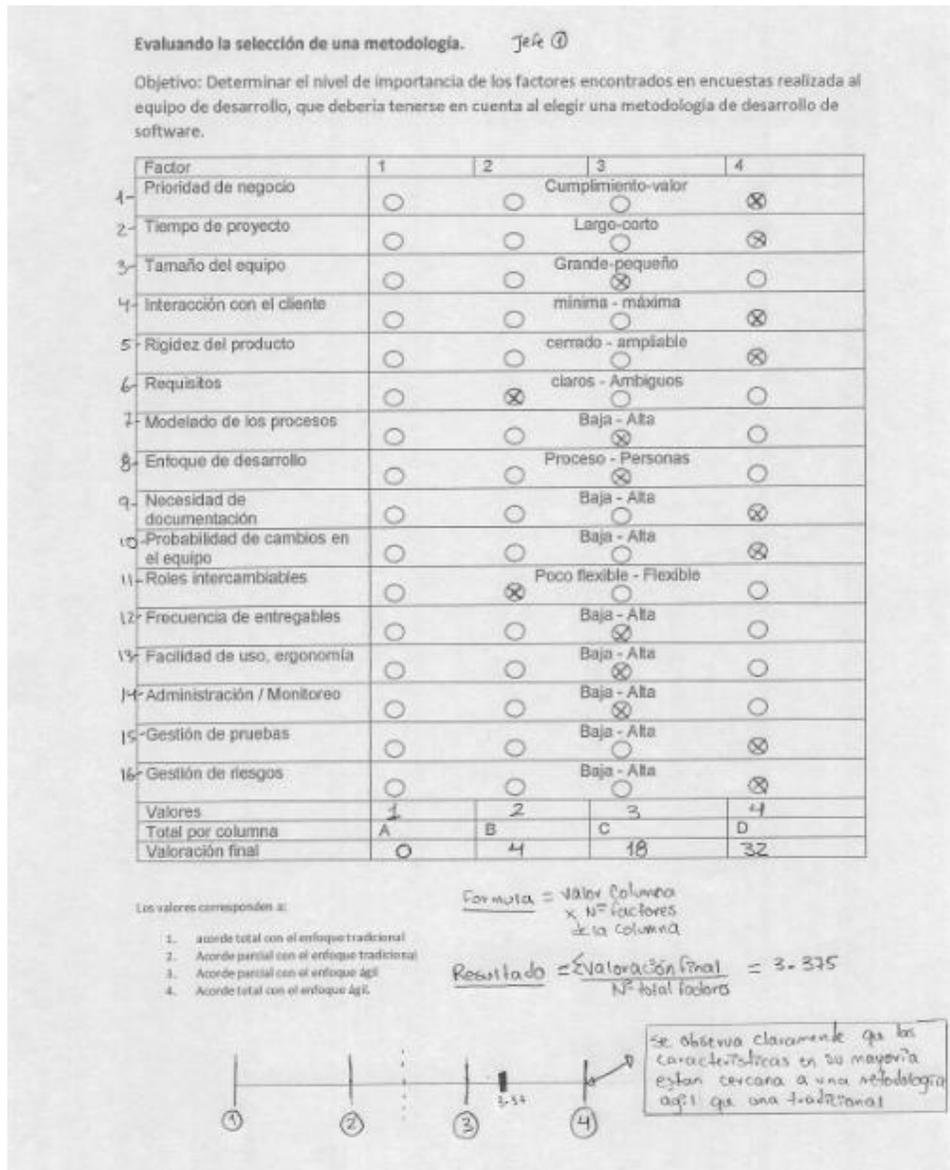
A.6 Comparación de metodología

TRADICIONAL	ÁGIL
<ul style="list-style-type: none"> Orientada a proyectos de cualquier tamaño 	<ul style="list-style-type: none"> Orientada a proyectos pequeños
<ul style="list-style-type: none"> Equipos grandes y dispersos 	<ul style="list-style-type: none"> Equipos pequeños, sobre 10 personas
<ul style="list-style-type: none"> Proyectos de media / larga duración 	<ul style="list-style-type: none"> Proyectos de corta duración
<ul style="list-style-type: none"> Proyecto cerrado 	<ul style="list-style-type: none"> Proyecto abierto a cambios
<ul style="list-style-type: none"> El cliente mantiene reuniones con la dirección 	<ul style="list-style-type: none"> El cliente está integrado en el equipo
<ul style="list-style-type: none"> Arquitectura prefijada 	<ul style="list-style-type: none"> Arquitectura se va mejorando
<ul style="list-style-type: none"> Documentación rigurosa 	<ul style="list-style-type: none"> Poca documentación
<ul style="list-style-type: none"> Roles específicos 	<ul style="list-style-type: none"> Roles genéricos
<ul style="list-style-type: none"> Roles no intercambiables 	<ul style="list-style-type: none"> Roles flexibles
<ul style="list-style-type: none"> Centrada en los procesos 	<ul style="list-style-type: none"> Centrada en las personas
<ul style="list-style-type: none"> Gestión dirigida 	<ul style="list-style-type: none"> Gestión colaborativa
<ul style="list-style-type: none"> Alto coste de prototipado 	<ul style="list-style-type: none"> Bajo coste de prototipado
<ul style="list-style-type: none"> Planificación inicial alta 	<ul style="list-style-type: none"> Planificación inicial baja
<ul style="list-style-type: none"> Basada en estándares de desarrollo 	<ul style="list-style-type: none"> Basadas en heurísticas
<ul style="list-style-type: none"> Poco <i>feedback</i> 	<ul style="list-style-type: none"> Continuo <i>feedback</i>
<ul style="list-style-type: none"> Proceso lineal 	<ul style="list-style-type: none"> Proceso iterativo
<ul style="list-style-type: none"> El coste se acerca a lo estimado 	<ul style="list-style-type: none"> El coste puede dispararse

[Tabla N° 20]: comparación de metodología ágiles vs tradicional

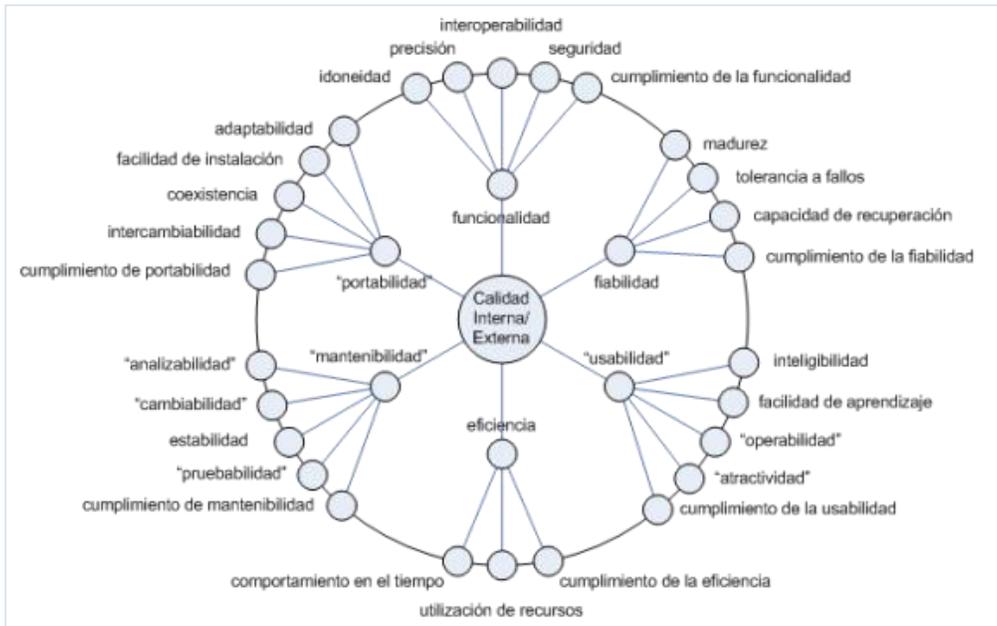
A.7 Evaluando selección de metodología

Si el resultado está más cerca del 1, sería más óptima una metodología tradicional, y si es más cercano a 4, sería más recomendable una metodología ágil.



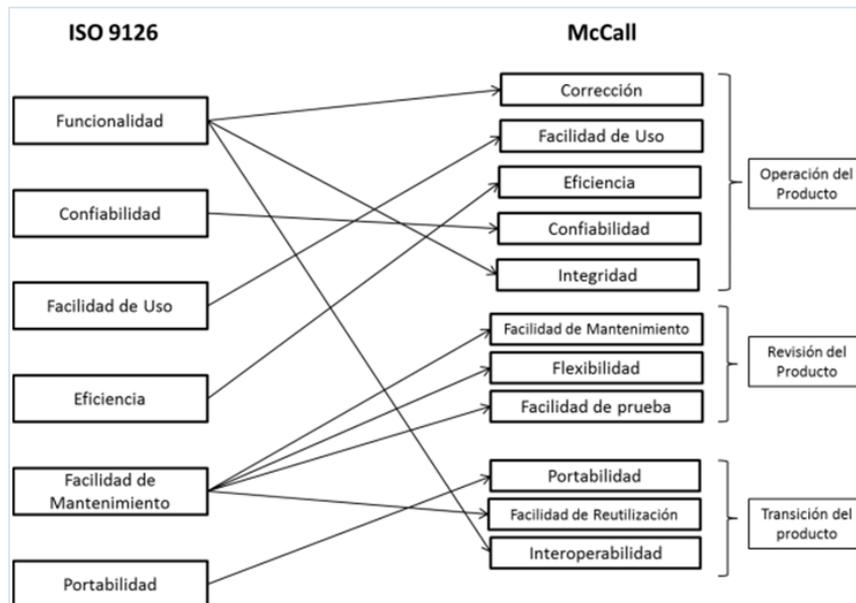
CK1. Valores de importancia de los factores al seleccionar una metodología de desarrollo (Elaboración propia)

A.8. Características de Calidad

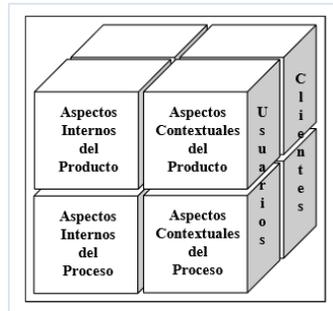


[Figura N° 10]. Características de la Calidad según la ISO/IEC 9126-1 (Fuente: Portal ISO 25000).

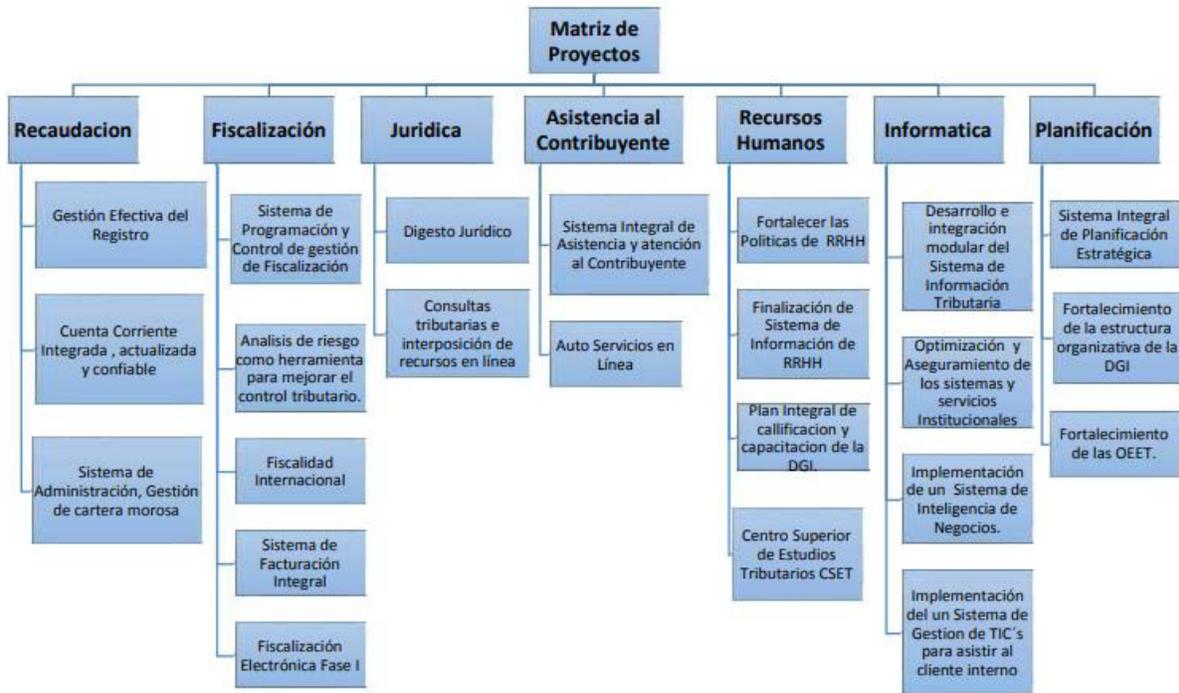
A.9 Calidad de software



[Figura. N° 11] Elementos de calidad. (Ramos, 2013)



[Figura N°. 12] Matriz de calidad global. (Imendoza, 2005)



[Figura N°. 29] Matriz de proyectos (DGI, 2016)