

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE CIENCIAS ADMINISTRATIVAS Y CONTABLES**

**MEJORAMIENTO DE LOS PROCESOS DE DESARROLLO DE
SOFTWARE PARA LA EMPRESA HSB SOFTECUADOR C.A.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERA COMERCIAL**

CRISTINA ALEJANDRA CÁRDENAS IZQUIERDO

DIRECTOR: PAÚL MICHAEL IDROBO DÁVALOS, MBA

**LÍNEA DE INVESTIGACIÓN: SISTEMAS DE GESTIÓN DE
PRODUCCIÓN Y OPERACIONES**

QUITO, JUNIO DE 2018

DIRECTOR:

Ing., Paúl Idrobo, MBA

INFORMANTES:

Irina Verkovitch

Bolívar Córdor

ÍNDICE

INTRODUCCIÓN	1
1. CAPÍTULO 1.- ANÁLISIS SITUACIONAL.....	3
1.1. Análisis externo.....	4
1.1.1. Factores Socioculturales	4
1.1.2. Factores Políticos	7
1.1.3. Factores Legales	9
1.1.4. Factores Tecnológicos	12
1.1.5. Factores Económicos	14
1.2. Análisis interno	17
1.2.1. Generalidades de la empresa.....	17
1.2.2. Productos	18
1.2.3. Estructura Orgánica	19
1.2.4. Funciones de los cargos	20
1.2.5. Misión	22
1.2.6. Visión.....	22
1.2.7. Objetivos.....	23
1.2.8. Políticas.....	23
1.3. Análisis FODA.....	25
1.3.1. Fortalezas	25
1.3.2. Oportunidades.....	26
1.3.3. Debilidades	28
1.3.4. Amenazas.....	29
2. CAPÍTULO 2.- MARCO TEÓRICO	30
2.1. Desarrollo de Software.....	31
2.1.1. Metodologías de desarrollo de Software	31
2.1.2. Metodologías Ágiles vs. Tradicionales.....	32
2.1.3. Bases de medición	46
2.1.4. Casos de análisis	49
2.2. Dirección de Proyectos	55

2.2.1.	Comparación de PMBOK6 y las metodologías ágiles.....	59
3.	CAPÍTULO 3.- DIAGNÓSTICO.....	60
3.1.	Objetivos del diagnóstico.....	61
3.2.	Metodología del diagnóstico.....	61
3.3.	Resultados.....	62
3.3.1.	Diagnóstico de Procesos.....	62
3.3.2.	Análisis de Cumplimiento.....	72
3.3.3.	Rendimiento de proyectos pasados.....	83
3.4.	Oportunidades de mejora.....	89
4.	CAPÍTULO 4.- PROPUESTA DE IMPLEMENTACIÓN.....	91
4.1.	Propuesta de Mapa de Procesos.....	92
4.2.	Propuesta de Diseño de Procedimientos.....	95
4.2.1.	Análisis de requisitos.....	95
4.2.2.	Diseño de Proyectos.....	97
4.2.3.	Desarrollo de Prototipos.....	98
4.2.4.	Elaboración de Pruebas y Ensayos.....	100
4.2.5.	Gestión de Mejoras.....	101
4.2.6.	Operación y Mantenimiento.....	103
4.2.7.	Mejoras al proceso de Contratación e Inducción.....	103
5.	CAPÍTULO 5.- PLAN DE IMPLEMENTACIÓN.....	105
5.1.	Objetivos de la implementación.....	106
5.2.	Riesgos y estrategias de la implementación.....	106
5.3.	Plan de Implementación.....	108
5.4.	Cronograma de implementación.....	110
5.5.	Beneficios.....	114
5.5.1.	Beneficios Cuantificables.....	114
5.5.2.	Beneficios No Cuantificables.....	120
5.6.	Análisis Costo-Beneficio.....	120
6.	CAPÍTULO 6.- CONCLUSIONES Y RECOMENDACIONES.....	122
6.1.	Conclusiones.....	122
6.2.	Recomendaciones.....	125
	BIBLIOGRAFÍA.....	128

ÍNDICE DE TABLAS

Tabla 1.- Inventario de procesos de HSB SOFTECUADOR C.A.....	62
Tabla 2.- Priorización de procesos por impacto y frecuencia.....	63
Tabla 3.- Principales proyectos del año 2016 y 2017	83
Tabla 4.- Tiempo de finalización de producto final.....	85
Tabla 5.- Alcance de productos a la fecha	86
Tabla 6.- Desviación financiera respecto a la planificación inicial	87
Tabla 7.- Mejoras realizadas en base a las observaciones durante pruebas.....	88
Tabla 8.- Riesgos y Estrategias de Implementación	107
Tabla 9.- Costo de la Implementación	108
Tabla 10.- Cronograma de actividades	111
Tabla 11.- Beneficios en tiempo de la implementación.....	115
Tabla 12.- Costo por hora de cada trabajador	116
Tabla 13.- Reducción del costo de trabajadores por proyecto	117
Tabla 14.- Beneficios de calidad de la implementación	118
Tabla 15.- Beneficios en productividad de la implementación	119
Tabla 16.- Beneficios económicos anuales de implementación	119
Tabla 17.- Relación Costo-Beneficio por incremento de ventas	121
Tabla 18.- Relación Costo/Beneficio por incremento en ventas y reducción de costos.....	121

ÍNDICE DE FIGURAS

Figura 1 - Población total del Periodo 1950 – 2010	6
Figura 2 - Porcentaje de crecimiento poblacional	7
Figura 3 - Ventas Anuales de Software en Ecuador	15
Figura 4 - Porcentaje de Crecimiento del Sector de Software	16
Figura 5 - Estructura Orgánica HSB SOFTECUADOR C.A.	19
Figura 6.- Funcionamiento de escuadrones y tribus en Spotify.....	51
Figura 7.- Modelo de desarrollo de Belike Software.....	54
Figura 8.- Kanban de HSB SOFTECUADOR C.A.	67
Figura 9. - Gestión de proyectos en la iteración del proceso	74
Figura 10.- Brechas de cumplimiento de la etapa de Inicio de desarrollo de software ágil	76
Figura 11.- Brechas de cumplimiento de la etapa de Planificación de desarrollo de software ágil.....	77
Figura 12.- Brechas de cumplimiento de la etapa de Ejecución de desarrollo de software ágil	79
Figura 13.- Brechas de cumplimiento de la etapa de Mantenimiento y Control de desarrollo de software ágil	80
Figura 14.- Brechas de cumplimiento de la etapa de Cierre de desarrollo de software ágil	81
Figura 15.- Brechas de cumplimiento por etapas de desarrollo de software ágil	82
Figura 16.- Propuesta de mapa de procesos.....	93
Figura 17.- Secuencia de los procesos propuestos.....	94

RESUMEN EJECUTIVO

El presente estudio tiene como objetivo mejorar los procesos de desarrollo de software de la empresa HSB SOFTECUADOR C.A., con la finalidad de que la empresa puede generar sus productos de forma eficaz y eficiente sin generar malestar entre sus trabajadores ante la sobrecarga de trabajo en las últimas etapas apresuradas del desarrollo de los productos.

El estudio parte de un análisis situacional tanto externo como interno de la empresa, para determinar los aspectos positivos y negativos con los que cuenta la empresa y en base a eso determinar los factores de mejora que puedan servir de apoyo o impedimento para la ejecución de nuevas metodologías de desarrollo. Los factores que se analizaron en el análisis externo se determinaron por aspectos socioculturales, políticos, legales, tecnológicos y económicos. Los factores que se analizaron en el análisis interno se determinaron por generalidades de la empresa, su gama de productos, su estructura orgánica y funcional, objetivos y políticas.

Posteriormente, se consideran en el marco teórico los conceptos y fundamentos de desarrollo de software y las distintas metodologías que pueden utilizarse según el tamaño y las necesidades de la empresa. Estos conceptos se utilizaron de base para generar la propuesta de mejora de los procesos de la empresa enfocada en la utilización de metodologías ágiles de desarrollo de software tras analizar las grandes diferencias de las metodologías tradicionales de cascada y las metodologías ágiles modernas, así como los beneficios con los que estas cuentan.

A continuación se determinó la situación actual de la empresa, considerando el proceso de desarrollo que posee actualmente, así como también se consideraron las brechas con las que cuenta la metodología actual de desarrollo de software con la metodología ágil que se propone utilizar, para de esta forma, en base a los rendimientos de proyectos pasados de la empresa, pueda determinarse los aspectos que requieren cambios para generar la mayor productividad. Para lograr este análisis se utilizaron diagramas de flujo

para determinar los procesos actuales de la empresa y una lista de verificación ágil para determinar las brechas que posee la empresa entre su metodología actual y la metodología ágil propuesta.

En el cuarto capítulo se considera la mejora de los procesos de desarrollo de software de la empresa, representada por procedimientos más completos y detallados que permitan determinar los objetivos, responsables y políticas necesarias para cumplir de manera más objetiva los procesos de desarrollo y así evitar el trabajo acumulado al final del desarrollo.

Finalmente, se cuenta con el plan de implementación que determina todos los factores que se generarían con la aceptación de la propuesta, tanto como el tiempo y costo que esto representa y los beneficios que se darían, para así realizar un análisis de costo y beneficio para determinar si efectivamente, la implementación de la propuesta sería lo ideal para la empresa.

El resultado final de la investigación ha determinado que implementar la propuesta significaría un beneficio neto \$58 420 con un costo de \$18 000 en un periodo de 7 meses, concluyendo en que es beneficiosa para el futuro de la empresa.

INTRODUCCIÓN

El desarrollo de software en tiempos modernos posee varias posibilidades de metodologías eficientes que permitan a las empresas generar productos de forma más veloz y precisa a las necesidades del mercado. Al vivir en una época que la tecnología y el software representan gran parte de nuestra vida personal y profesional, es necesario adaptarse a los cambios que permitan a los nuevos programas a sobrevivir en un mundo de mayor conectividad.

Es por eso que se estudia la empresa HSB SOFTECUADOR C.A., una extensión ecuatoriana de la empresa HSBCAD nacida en Bélgica, que busca generar extensiones y herramientas adicionales de software de diseño de casas de madera en varias regiones de Europa y Japón.

HSBCAD se encuentra en 12 países en la actualidad, como son Bélgica, Reino Unido, Italia, Japón, Países Bajos, Polonia, República Checa, Suecia, Alemania, Canadá y Ecuador, siendo este último el generador de todos los productos, es decir el país donde se desarrolla el software que se distribuye al resto de países.

El problema con el que cuenta la empresa se detalla en que actualmente posee dificultades en los tiempos de entrega, además de que no cuenta con suficiente organización que promueven las sobrecargas de trabajo en las etapas finales del desarrollo de los productos. A su vez, los empleados de la empresa han comunicado su insatisfacción sobre la incorrecta estructuración y repartición del trabajo.

A pesar de que HSBCAD posee presencia internacional, el presente estudio únicamente se centrará en la empresa en Ecuador, considerando que HSB SOFTECUADOR C.A. no se dedica a la comercialización de los productos, simplemente se dedica a la generación de los mismos, generando posteriormente una propuesta que permita mejorar los procesos que formen parte del desarrollo de software general.

Se espera que el siguiente trabajo de investigación apoyé a mejoras en el proceso de desarrollo de software de la empresa HSBCAD SOFTECUADOR C.A., ya que el mejoramiento del proceso de desarrollo de software permitirá a la empresa a desarrollar mejor su actividad comercial, incrementar la productividad, reducir costos innecesarios y tiempos ociosos.

Por estas razones, se ha determinado el objetivo de realizar un diagnóstico del funcionamiento actual de la empresa para poder detectar las oportunidades de mejora que se cuentan y así generar una propuesta de trabajo flexible y pertinente al ritmo de trabajo que se requiere de los trabajadores de HSB SOFTECUADOR C.A.

El alcance con el que cuenta la investigación se limitará únicamente a la filial ecuatoriana HSB SOFTECUADOR C.A. y a aquellos procesos que tengan directa relación con el desarrollo de software, únicamente comprendiendo el análisis teórico de los resultados que genere, no su implementación.

1. CAPÍTULO 1.- ANÁLISIS SITUACIONAL

El objeto de estudio del presente capítulo se basa en conocer el entorno en el que se desarrolla la empresa HSB SOFTECUADOR C.A. mediante un análisis interno y externo de factores que la afectan de forma positiva o negativa.

El entorno externo en el que se desarrolla la empresa, se conocerá mediante el análisis de aspectos socioculturales, políticos, legales, tecnológicos y económicos, para poder determinar la relación que posee la empresa con el entorno y sector de la que forma parte.

En el entorno interno en el que se desarrolla la empresa, se conocerá los fundamentos bajo los que trabaja, desde su Visión y Misión, así como también los productos más importantes que posee, los cargos requeridos para cumplir con el proceso de estudio, los objetivos que persigue y las políticas que la rigen.

Posteriormente, con la ayuda de ambos análisis, se determinan las fortalezas y oportunidades, así como las debilidades y amenazas con las que se cuenta, permitiendo entender el panorama de desarrollo normal de la entidad.

En este capítulo se introduce a la empresa y como funciona, para permitir una mejor comprensión de las necesidades que requiere para generar la propuesta de mejora de los procesos de desarrollo de software que poseen.

1.1. Análisis externo

1.1.1. Factores Socioculturales

1.1.1.1. Perfil de clientes

HSB SOFTECAUDOR C.A. cuenta con dos segmentos de clientes muy específicos con diferentes perfiles, que se pueden especificar de la siguiente forma:

-Grandes constructoras en países europeos y asiáticos, donde buscan un programa confiable para elaborar diseños de casas de madera de calidad en grandes cantidades, donde se pueda ocupar los materiales de forma eficiente, mientras los diseños sean confiables y seguros para las personas que pudiesen habitar dichas viviendas.

-Profesionales independientes que desean diseñar casas de madera eventuales bajo pedidos específicos de personas que busquen construir su propio hogar.

1.1.1.2. Viviendas de materiales alternativos en Ecuador

Vivimos en un entorno en el que las viviendas generalmente son elaboradas de hormigón, ladrillos y otros materiales similares, donde la madera no es considerada como un material primordial para la construcción, sino como un material secundario para cualquier estructura, es por eso que, si HSB SOFTECUADOR C.A. desea incrementar sus

ventas en el territorio local, se deben generar estrategias para lograr contrarrestar la cultura de construcción de viviendas en la actualidad.

Según el Censo de Población y Vivienda del 2010, el 46,9% de la población cuenta con vivienda propia, porcentaje con tendencia creciente; es decir que las personas tienden a adquirir o construir su vivienda, donde la madera como material alternativo de construcción puede resultar en una solución habitacional familiar de bajo costo, donde la empresa garantice la resistencia de los diseños elaborados para dichas construcciones.

1.1.1.3. Demografía Ecuatoriana

Como se puede observar en la Figura 1, la población ecuatoriana se encuentra en constante crecimiento, sin embargo, al comparar con la Figura 2, se puede observar que este crecimiento tiende a decrecer con el paso del tiempo.

De acuerdo al Censo de Población y Vivienda realizado el año 2010, la provincia con mayor concentración poblacional es Guayas, con un 25,2%, donde Pichincha le sigue con el 17,8% de la población nacional. Si se desea conocer los mayores porcentajes de población en cuanto ciudades, en primer lugar se encuentra Guayaquil con 2 291 158 habitantes, seguida de Quito con 1 619 146, Cuenca con 331 888 y Santo Domingo con 305 632 habitantes.

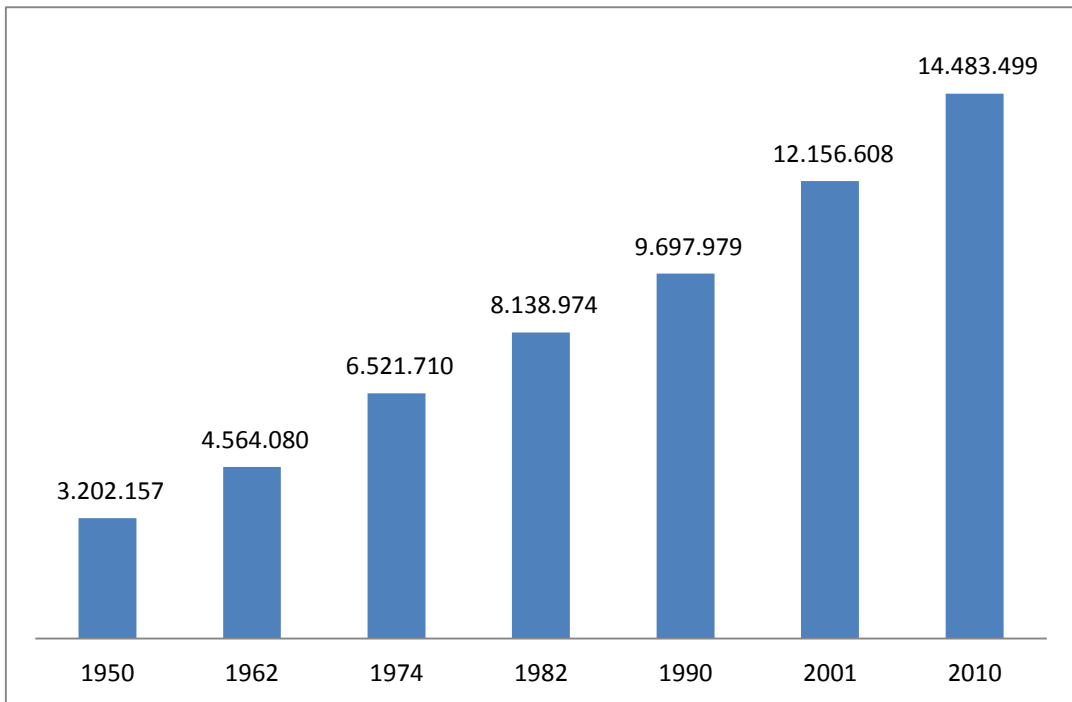


Figura 1 - Población total del Periodo 1950 – 2010

Fuente: INEC (2010)

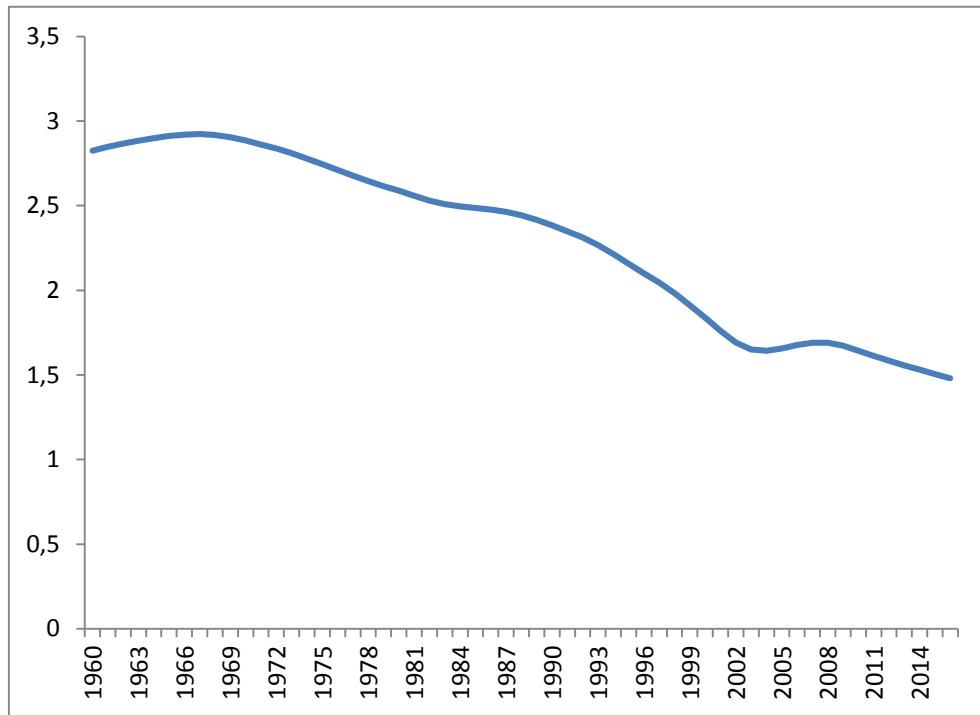


Figura 2 - Porcentaje de crecimiento poblacional
 Fuente: Banco Mundial (2014)

1.1.2. Factores Políticos

1.1.2.1. Crecimiento económico por el sector público

Como resultado de los últimos diez años de gobierno, el sector público se ha convertido en el mayor generador de fuentes de empleo del país en relación al sector privado, donde en diciembre del 2007, según el Instituto Nacional de Estadísticas y Censos (INEC), los empleos brindados por el sector público alcanzaban a 406 182 empleados, donde en septiembre de 2015, estos empleados se convirtieron en 717 125 personas, representando un incremento del 56%.

Es así como evidentemente el sector público ha incrementado su potencial económico en todos los sectores de trabajo, incluido el de la construcción, donde, es necesario el análisis de la posibilidad de

construcción con materiales alternativos para poder elaborar viviendas con menores costos y la misma resistencia.

1.1.2.2. Créditos para construcción de viviendas

El Banco del Instituto Ecuatoriano de Seguridad Social (BIESS) ofrece créditos de construcción de vivienda, entregando el 100% de financiamiento hasta 25 años plazo, donde dicho crédito puede ser utilizado para construir viviendas en terrenos, sobre o adyacente a construcciones existentes que no cuenten con gravamen y para terminar la construcción de viviendas cuyas condiciones actuales no permitan habitar a ninguna persona. Estos créditos que ofrece el BIESS cuentan con las menores tasas de interés que el mercado ofrece.

Estos créditos facilitan la construcción de casas, un factor de suma importancia para las constructoras y para las empresas de software que proveen los programas de diseño de casas.

1.1.2.3. Proyectos de Vivienda

Diario El Universo (2017) señala que el proyecto de gobierno “Casa para todos” del plan de gobierno “Toda una Vida” iniciará con su primera fase en la capital, donde se construirán en los sectores centro y sur 3 100 viviendas.

El proyecto “Casa para todos” busca concluir con la construcción de 325 000 viviendas en los próximos 4 años de gobierno del actual presidente Lenin Moreno, donde 191 000 de ellas serán gratuitas. Es por ello que se

ha propuesto el objetivo de construir entre 40 000 a 50000 viviendas hasta julio del 2018, con una inversión de \$500 millones, considerando que las provincias que cuentan con mayor déficit habitacional son Guayas, Manabí, Los Ríos y Pichincha.

Al contar con el objetivo de construir más de 300 000 viviendas en un periodo de 4 años, es necesario considerar los costos y el tiempo que esto conlleve al presupuesto ecuatoriano, donde la construcción en materiales alternativos como la madera, no solo disminuye el tiempo de producción de las viviendas, pero reduce costos en materiales de construcción.

1.1.3. Factores Legales

1.1.3.1. Incentivos para el sector de desarrollo de software

De acuerdo a PROECUADOR (2014), el sector de software en la economía ha presentado gran crecimiento en la última década, con una tasa anual del 22%. Considerando que el sector de las TICs, en la que se incluye al desarrollo de software es un sector priorizado por la ley, por motivo de promover aún más este crecimiento que se cuenta con ciertos incentivos para acelerar este desarrollo:

- “No se pagará el Impuesto a la Renta durante 5 años desde que se generen ingresos para: nuevas inversiones que se realicen en sectores priorizados ...” (PROECUADOR, 2015)

Este incentivo aplica para compañías nuevas creadas posteriormente al 29 de diciembre del 2010, que cuenten con nueva inversión, se

establezcan fuera del perímetro urbano de las ciudades de Quito y Guayaquil.

- “Exención del pago del (Impuesto a la Salida de Divisas) ISD para dividendos pagados por compañías establecidas en Ecuador a compañías en el exterior o personas no residentes en el Ecuador (excepto a paraísos fiscales)” (PROECUADOR, 2015)

Esta exención no es específica para el sector de software, sin embargo es importante para HSB SOFTECUADOR C.A. debido a que todas los dividendos que se produzcan (si se producen), son enviados a Bélgica, donde residen los dueños y accionistas de la empresa.

1.1.3.2. Salida de Divisas

De acuerdo al Servicio de Rentas Internas (SRI), la salida de divisas del país, ya sea en efectivo, giro de cheques, transferencias, pagos de cualquier naturaleza, entre otros; constituye un hecho generador de impuesto del 5%, llamado Impuesto a la Salida de Divisas (ISD).

Sin embargo, como se detalló con anterioridad, en el caso del pago de dividendos al exterior se encuentra exento del pago de este impuesto, un factor de gran importancia para HSB SOFTECUADOR C.A., ya que sus propietarios y accionistas residen en el extranjero.

1.1.3.3. Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación

El “Código Ingenios” tiene el objetivo de establecer un marco legal para estructurar la economía social del conocimiento, creatividad e innovación.

Sin embargo, en lo que respecta al desarrollo de software, es importante destacar ciertos artículos:

En primer lugar, el artículo 111 es importante debido a que hace referencia a la propiedad intelectual bajo relación de dependencia, en donde, a las obras que se realicen por encargo de un empleador, la titularidad le corresponderá a este pero de forma no exclusiva, ya que la persona autora de la propiedad guarda el derecho absoluto de explotarla de forma distinta a la acordada, solo si lo hace en buena fe y no perjudique al empleador. Sin importar la situación, el autor tiene derecho irrenunciable a recibir mínimo el 10% de los beneficios que provengan de la explotación de la propiedad.

Según Pérez, (2014), este cambio significa que las empresas de desarrollo de software deben establecer en sus contratos de licencia la prohibición de uso del programa a través de redes, donde los usuarios deben controlar por sí mismos si la prohibición existe antes de utilizar cualquier software y así evitar la violación de derechos de propiedad intelectual.

Para HSB SOFTECUADOR C.A. este artículo implica que se deben adaptar la contratación de los programadores de la empresa, donde, para conservar los derechos de propiedad sobre los trabajos realizados, se

debe estipular de forma expresa en el contrato laboral de cada empleado; si esta acción no se realiza, se entiende que los programadores son titulares de la propiedad realizada, así como las futuras obras por realizar e ideas planteadas en la empresa.

Finalmente, el artículo 136 hace referencia a la obligatoriedad del sector público e instituciones educativas a usar software libre y estándares abiertos, donde se prohíbe adquirir actualizaciones de software propietario adquirido antes de la existencia del Código, así como sistemas que no puedan ser reemplazados por software libre por razones técnicas o de naturaleza comercial. Esto obliga a estas entidades a contratar proveedores que posean una nube virtual o centro de datos ubicado en el Ecuador.

1.1.4. Factores Tecnológicos

1.1.4.1. Software Libre y propietario

Software libre hace referencia a aquel que puede distribuirse, modificarse, copiarse y usarse libremente, por lo que lo acompaña siempre el código fuente para efectivizar todas sus características.

El software no libre, propietario, privativo, privado o con propietario hace referencia a aquel en donde los usuarios tienen limitaciones para distribuirlo, usarlo o modificarlo, por lo que su código fuente es restringido, donde quien posee los derechos de autor provee la autorización para su utilización, estudio o adaptación.

Es importante conocer estas definiciones debido a que HSB SOFTECUADOR C.A. posee autorización de otras empresas de desarrollo de software para elaborar aplicaciones y extensiones a sus programas de diseño de casas de madera.

1.1.4.2. Desarrollo en la nube

La empresa BBVA, (2016), considera que el desarrollo de software en la actualidad se está trasladando a la nube, no solo para almacenar programas y aplicaciones, pero también para facilitar el proceso de desarrollo de las mismas.

Es por ello que en 2016, se estableció la meta global de que la mayoría de aplicaciones que no sean parte de la nube, sean compatibles con ella por cualquier necesidad de traslado de información y facilitación de edición del código.

Las ventajas con las que cuenta el uso de la nube para el desarrollo de software son las siguientes: los desarrolladores pueden programar desde cualquier parte del mundo, con trabajo colaborativo en tiempo real, donde el tiempo de implementación es menor y los costos de comunicación son menores.

1.1.5. Factores Económicos

1.1.5.1. Sector de Desarrollo de Software en el PIB

El sector de las tecnologías de la información y comunicación (TIC's), aportaba en 3,5% aproximadamente al PIB ecuatoriano, del cual, únicamente el 7% correspondía al desarrollo de software en el año 2014.

Sin embargo, las ventas y crecimiento del sector se han visto afectadas de forma negativa, en donde, para el año 2015, el sector decreció en 9%, opuesto a lo que se creía, pensando que el sector podría crecer hasta en un 20% según el ministro ecuatoriano de Telecomunicaciones en ese entonces, Augusto Espín.

El sector del software ecuatoriano comprende un aproximado de 600 empresas, donde mantiene el optimismo de incrementar sus ventas en los próximos 3 años de 400 millones de dólares a 1,500 millones de dólares, así como también se espera incrementar la oferta de trabajo de 8 000 plazas a 24 000 plazas de trabajo, que finalmente permitiría el incremento de exportaciones de software de 30 a 300 millones de dólares, especificado así por la Asociación Ecuatoriana de Software (Aesoft).

Como se puede observar en la Figura 3, las ventas de Software en el país se han mantenido en constante crecimiento, pero, al llegar al año 2015, han decaído, como se puede observar también en la Figura 4, donde no solo el nivel de crecimiento del sector de software ha crecido en menor nivel cada vez más, pero también, ha iniciado una etapa de decrecimiento.

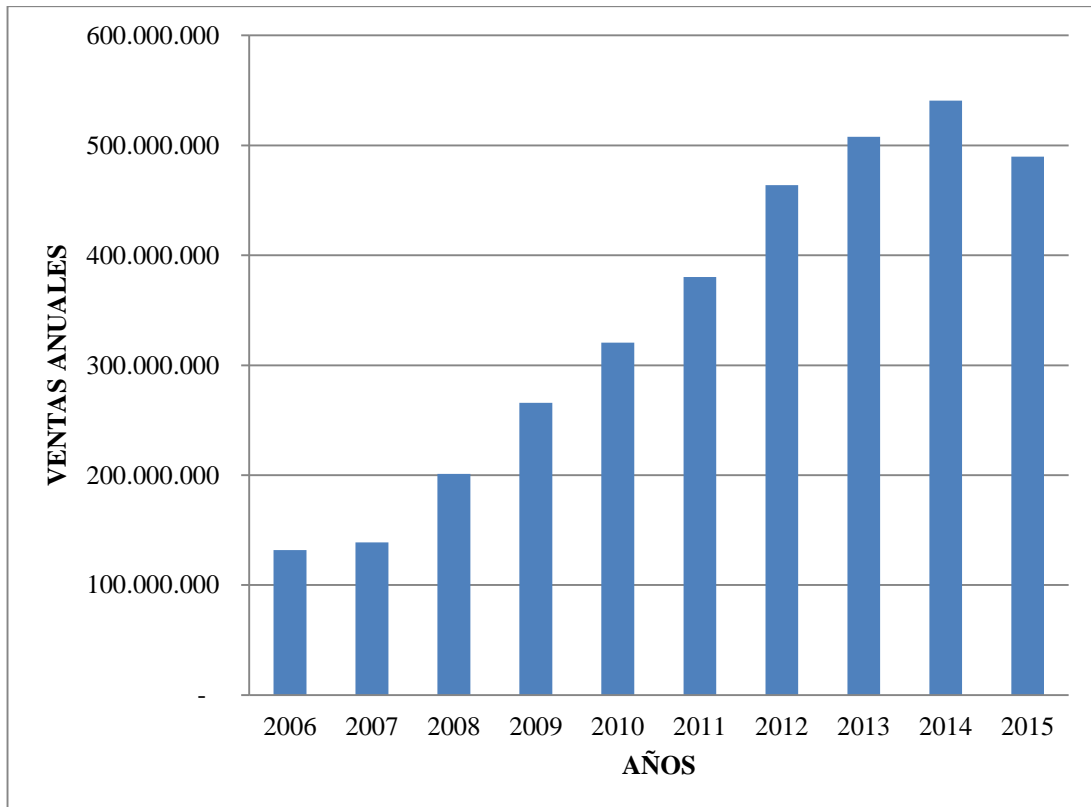


Figura 3 - Ventas Anuales de Software en Ecuador

Fuente: Asociación Ecuatoriana de Software (AESOFT) (2016)

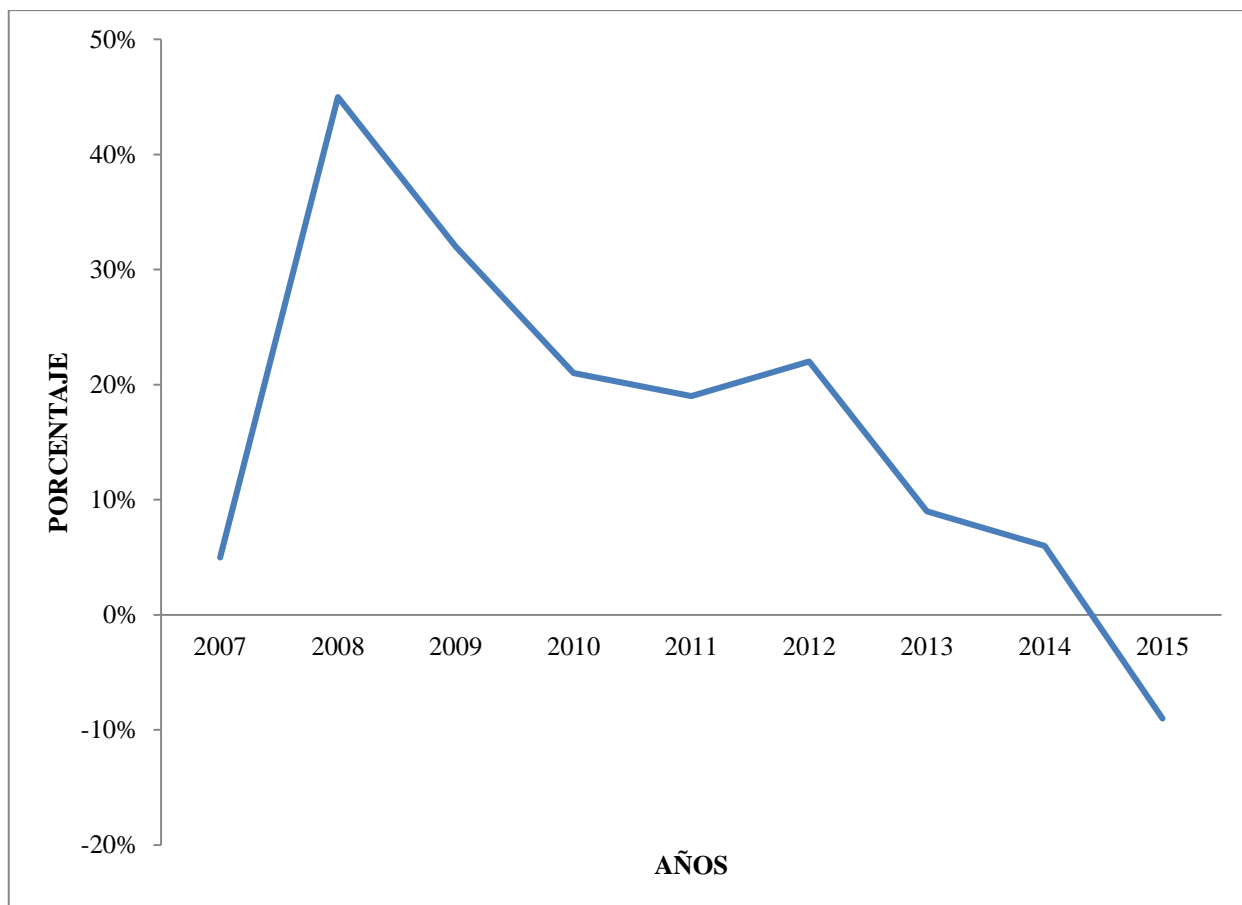


Figura 4 - Porcentaje de Crecimiento del Sector de Software

Fuente: Asociación Ecuatoriana de Software (AESOFT) (2016)

1.1.5.2. Sector de la Construcción en el PIB

Según el Banco Central del Ecuador, el sector de la construcción es el quinto sector que ha tenido gran crecimiento en la década del 2005 al 2015 en un 85,5%. (Mundo Constructor, 2016)

Por esta razón, el sector de la construcción ha aportado al PIB con un 10,1%, convirtiéndose en el cuarto sector de aportaciones, después de manufactura, comercio y petróleo y minas. (Mundo Constructor, 2016)

1.2. Análisis interno

1.2.1. Generalidades de la empresa

La empresa HSBCAD nació en Bélgica, con el propósito de desarrollar software que ayuda a “convertir diseños de arquitectura en un diseño de producción”.

HSBCAD cuenta con varias sucursales alrededor del mundo, como son Bélgica, Reino Unido, Italia, Japón, Países Bajos, Polonia, República Checa, Suecia, Alemania, Canadá y Ecuador.

HSBCAD en Ecuador se dedica al desarrollo del software de acuerdo al pedido de cada una de las sucursales, mas no a la comercialización directa con el consumidor final. Los diseños que realiza se basan en un prototipo establecido y estándar, que es adaptado a las necesidades señaladas por su cliente, es decir, otras sucursales; pero también cuentan con la posibilidad de iniciar un diseño completamente diferente sin basarse en el prototipo, de así ser requerido.

La manera en que HSBCAD en Ecuador trabaja se basa en proyectos, donde se crean equipos de trabajo, asignando tareas específicas de cada proyecto a cada equipo, así como sus responsables.

Los proyectos con los que cuenta la empresa pueden darse por corrección de un error o por creación de aplicaciones nuevas, en donde se tiene la lista de proyectos en un orden de prioridad a ser tomado en cuenta.

Los productos con los que cuenta la empresa en Ecuador se enfocan en 3: hsbcad (y componentes), hsb on Revit (y sus componentes) y hsbshare, donde los dos primeros son proyectos establecidos por la matriz en Bélgica y el último es un proyecto autónomo de los programadores de Ecuador.

1.2.2. Productos

Todos los productos que ofrece la empresa fueron elaborados para facilitar el uso de programas existentes de diseño de casas de madera, en donde se generan herramientas adicionales y complementos que permiten al usuario realizar planos más realistas, donde los cortes en la madera se puedan visualizar de mejor manera y con una mejor planificación al momento de cortar las piezas en las planchas de madera.

- **Hsbcad:** es un conjunto de extensiones, aplicaciones y complementos para el programa Autocad o Revit de diseño de casas de madera. Este es el producto de mayor distribución en la empresa, encontrándose en la cúspide de su vida como producto. Actualmente, la versión 21 es la disponible en el mercado, o versión 2017 para los clientes.
- **Hsbtimber on Revit o HsbRevit:** es un conjunto de extensiones, aplicaciones y complementos para el programa Revit de diseño de casas de madera. La empresa busca que este sea el nuevo producto estrella en el mercado, que reemplace a Hsbcad cuando su vida como producto decaiga.
- **Cross Laminated Timber:** Es la versión extendida de Hsbtimber, con mayores herramientas profesionales y de mayor alcance.

- **Hsbshare:** es un proyecto exclusivo de HSB SOFTECUADOR C.A., que se enfoca en el almacenamiento de los proyectos de los clientes en una nube para poder editarlos, movilizarlos y utilizarlos con mayor facilidad.
- **HsbNester:** es un aplicativo de Autocad para optimizar los paneles de plancha grande de madera, donde las piezas se plasman con el objetivo de ocupar la mayor superficie de la plancha y evitar desperdicios al momento de cortarlas.

1.2.3. Estructura Orgánica

La estructura orgánica de HSB SOFTECUADOR C.A. se ha visto reducida en los pasados meses debido a traslados de los programadores a oficinas de la empresa en otras partes del mundo, especialmente Bélgica e Inglaterra. Es así como en la

Figura 5, se puede observar la limitada estructura orgánica actual que cuenta la empresa.

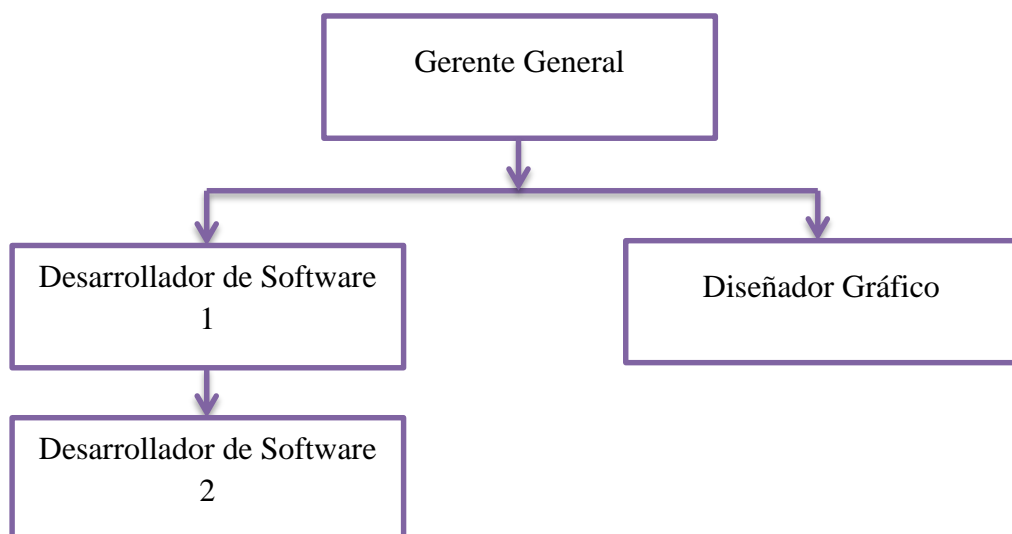


Figura 5 - Estructura Orgánica HSB SOFTECUADOR C.A.

1.2.4. Funciones de los cargos

A continuación se presentan los perfiles de los cargos establecidos anteriormente en la Figura 5, donde se establecen las responsabilidades y tareas que debe cumplir cada uno de ellos:

Gerente General

Sus funciones incluyen:

- Distribuir el software a los clientes.
- Administración de licencias a los clientes.
- Carga de actualizaciones en la web de la empresa para descarga de los clientes.
- Mejora de productos propios y adquiridos de terceros.
- Mantenimiento de aplicaciones existentes.
- Control de aplicación de registro de licencias.
- Entregar los permisos de producto en desarrollo a clientes aprobados.
- Administración de productos y módulos.
- Controlar el reinicio de los servidores cada semana para evitar futuras congestiones.

- Controlar las funciones de la contadora externa y los trámites legales que realiza.
- Contratar personal y capacitarlo.
- Pagos de sueldos, servicios externos, seguros e impuestos.
- Adquirir el equipo necesario para que el equipo trabaje de forma más ágil.

Desarrollador de Software 1

- Interpretar los diseños que elaboran los clientes para poder desarrollar herramientas que permitan facilitar cortes especiales en el modelo.
- Administrar el producto HsbNester.
- Carga de actualizaciones en la web de la empresa para descarga de los clientes.
- Mejora de productos propios y adquiridos de terceros.
- Mantenimiento de aplicaciones existentes.

Diseñador Gráfico

- Diseñar los iconos de las aplicaciones y los complementos de todos los productos.
- Diseñar la imagen de los productos.
- Mantener la imagen de la empresa.

Desarrollador de Software 2

- Aprender programación y desarrollo de software.
- Crear familias de códigos y extensiones de las mismas para facilitar el diseño de piezas de madera.
- Mejorar códigos existentes.
- Adaptar los códigos de hsbcad a códigos funcionales para el número programa hsbOnRevit.
- Probar las nuevas aplicaciones elaboradas desde un enfoque de arquitecto, para poder dar a conocer las fallas previas antes de presentarse el producto a Europa.

1.2.5. Misión

HSB SOFTECUADOR C.A. no cuenta con una misión establecida, pero en palabras de su Gerente General Luis Salazar, se puede considerar la siguiente:

“Satisfacer las necesidades del cliente, facilitando el uso de sus herramientas de diseño de casas de madera y la realización de sus planos.”

1.2.6. Visión

De igual manera que la misión, la empresa no cuenta con una visión establecida, pero en las palabras del Gerente General, se considera lo siguiente:

“Mantenerse a la vanguardia tecnológica de programas de diseño de casas de madera, en donde se encuentren cada vez más mejoras y aplicaciones de facilitación de diseño en los programas disponibles en el mercado de diseño de casas de madera.”

1.2.7. Objetivos

Los objetivos con los que cuenta la empresa HSBCAD SOFTECUADOR C.A. son:

- a. Para finales de 2017, crear un sistema para diseño de casas de madera llamado TimberFrame, con una versión estable para diseñar funciones de arquitectura.
- b. Migrar las aplicaciones de Windows Forms a Window Present Fund en el periodo de 3 meses, con el seguimiento correspondiente.
- c. Mejorar sustancialmente el sitio web de descarga de instalaciones en un periodo de 8 meses, con una mejora visible.

1.2.8. Políticas

Las políticas con las que cuenta la empresa HSBCAD SOFTECUADOR C.A. son:

- Manejar transparentemente la gestión administrativa para el mejor desarrollo.

- Evaluar cada 2 semanas el desarrollo de los proyectos en reuniones entre todo el equipo (internacionalmente).
- Empoderar a los empleados para adquirir los recursos necesarios para el desarrollo de su trabajo, con autorización final del Director Administrativo.
- Los desembolsos menores a \$250 no requieren autorización de la empresa en Bélgica para realizarse.
- Se promueve el diálogo y el intercambio de opiniones basado en la información.
- Se contrata a una contadora externa que se encarga de los trámites legales de la empresa, con el respaldo suficiente al Director Administrativo.
- El Director Administrativo revisa cada estado financiero elaborado por la contadora.
- Se dedica mayor tiempo a las actividades de mayor urgencia establecidas en las reuniones periódicas que se tiene cada 2 días.
- Contratar personas responsables e independientes.
- Se trabaja para cumplir objetivos, no horarios.
- Se permite el trabajo desde el hogar todos los martes y jueves.
- Cada vez que un miembro del equipo presente un problema, debe presentar mínimo una solución posible.

1.3. Análisis FODA

1.3.1. Fortalezas

En base al estudio previamente realizado, se pueden considerar las siguientes fortalezas para HSB SOFTECUADOR C.A.:

- Cada programador cuenta con una librería de códigos con métodos y procesos, que serán utilizados constantemente en el desarrollo de cada producto.
- Se cuentan con reuniones rotativas para que cada empleado puede exponer sus ideas y el desarrollo de sus proyectos, en donde el resto de empleados y el gerente general buscan todas las posibles fallas y problemas con las que cuentan los proyectos y así facilitar su desarrollo con varias soluciones.
- Se realizan periódicos ejercicios de programación para empleados nuevos para promover la solución creativa de los problemas que puede tener cada cliente de HSBCAD.
- Los programadores explican sus ideas a los administradores de Europa para poder ampliar el alcance y mantener sencillo el uso de cada herramienta.
- Se ahorra tiempo de corrección de errores al mantener exposición constante de ideas.
- Se trabaja sobre los mismos códigos que cuenta la empresa para poder mejorarlos constantemente.

- La empresa cuenta con una visión fuerte de continuar investigando y aprendiendo, en donde los programadores aprenden cómo diseñar casas de madera y así poder entender lo que realmente necesita y desea el cliente.
- Se cuenta con un empleado graduado de Arquitectura que se encuentra en proceso de aprendizaje de desarrollo de software, que es una ventaja debido a que el entiende de mejor manera las necesidades de cada cliente.
- Los objetivos de cada desarrollo de software que se elaboran no son finales, de hecho, son objetivos flexibles, considerados una fortaleza debido a que al desarrollar aplicaciones que no existen en ningún mercado, el cliente realmente no sabe lo que necesita hasta el momento exacto que lo necesita, de esta manera se puede comunicar a los programadores y adaptar de mejor manera los cambios requeridos por el cliente.
- Las diferentes zonas horarias de las oficinas proveen una ventaja a los programadores en cuanto a la corrección de problemas críticos y errores pequeños, ya que en la mañana (de Ecuador) se comunica el problema, contando con el resto del día para solucionarlo, en donde las oficinas europeas, al iniciar su jornada laboral el día siguiente, inician con el problema corregido para comunicar a los clientes.

1.3.2. Oportunidades

Considerando el entorno en el que se desarrolla la empresa HSB SOFTECUADOR C.A. en el país, se consideran como oportunidades las siguientes:

- El desarrollo de la programación en la nube permite a la empresa a ofrecer sus productos y su asistencia desde hacia cualquier parte del mundo en casi cualquier momento.
- El crecimiento y mayor seguridad de la nube permite a los usuarios de la empresa utilizar sus servidores para archivar sus proyectos y diseños con la mayor seguridad y confianza.
- Ligada a la amenaza de limitación de mercado de software de construcción de casas de madera en Ecuador y América latina, se considera una gran oportunidad el bajo nivel de competencia que existe en el país de empresas similares, que desarrollen extensiones y aplicaciones para software ya existente de diseño de casas de madera.
- Existencia de un mercado no satisfecho en la región latinoamericana de software que ayude a la construcción de casas de materiales alternativos y sostenibles.
- El creciente crecimiento de la población ecuatoriana determina la necesidad de generar casas diseñadas con programas que brinden seguridad.
- El crecimiento económico del sector de desarrollo de software en el país puede permitir a la empresa realizar ventas directas locales.
- El creciente incremento de créditos otorgados para la construcción de viviendas determina que un futuro a corto plazo se requerirá de apoyo tecnológico para el diseño de casas.
- No se requerirá a la empresa que desembolse en el Impuesto de Salida de Divisas en el caso que desee realizar ventas locales y necesite transferir los ingresos a otras filiales en el extranjero.
- El software libre y propietario permite a la empresa a realizar extensiones y herramientas de programas ya disponibles en el mercado.

- Al pertenecer a una empresa desarrollada en países de primer mundo, la empresa a nivel local puede aprovechar de este factor para demostrar una mejor imagen a sus posibles clientes en la región latinoamericana.

1.3.3. Debilidades

Las debilidades con las que cuenta HSB SOFTECUADOR C.A. se pueden resumir de la siguiente manera:

- Se dedica únicamente al desarrollo de software, mas no a la venta directa de sus aplicaciones.
- Los empleados consideran los códigos que crean como su propiedad.
- No tienen retroalimentación directa por parte del cliente para elaborar las correcciones y mejoras a cada producto, esta información viene de segunda mano de los directivos en Europa.
- Se tiene diferente zona horaria a los administrativos en Europa, por lo que las reuniones deben realizarse por medio electrónicos y a horas específicas, con un periodo de comunicación de 3 a 4 horas diarias, que provoca demoras y ruido en la comunicación de las necesidades y quejas de los clientes.
- La diferencia en zonas horarias también afecta a la solución de ciertos errores, donde si un programador cuenta con una duda no resuelta en el lapso de comunicación con el resto de oficinas, la duda no se resolverá sino hasta el día siguiente.
- Se cuentan con ideas muy creativas con corto alcance de uso en la práctica, en donde el tiempo invertido en su desarrollo se convierte en un desperdicio significativo y un factor desmotivante para el personal.

- La mentalidad de los trabajadores no se encuentra completamente abierta al cambio.
- La unión de las posiciones de Gerente General y un desarrollador de software más encarecen los costos de los productos y no permiten que la persona que ocupe la posición dedique su tiempo objetivamente a las tareas que se le otorguen.

1.3.4. Amenazas

Las amenazas examinadas con las que HSB SOFTECUADOR C.A. cuenta se reflejan a continuación:

- La limitación que existe en Ecuador y Latinoamérica del mercado de software de construcción de casas de madera y sus complementos adicionales.
- La aprobación del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, que arriesga la propiedad de la empresa sobre los productos elaborados en Ecuador.
- El artículo 136 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación obliga al sector público a optar por el uso de software libre sobre el software privado, donde, al ser uno de los sectores que más promueve la construcción, se reduce la participación de mercado de HSB SOFTECUADOR C.A. en el país.
- La disminución en los últimos tres años de ventas del sector de software provocaría que las ventas de la empresa puedan también reducirse.

2. CAPÍTULO 2.- MARCO TEÓRICO

En el presente capítulo se presentan de forma simplificada las bases y conceptos sobre el cual se desarrollará la propuesta de mejora de procesos de la empresa HSB SOFTECUADOR C.A.

Para poder entender las mejoras que pueden establecerse al proceso de desarrollo de software, es necesario comprender los tipos de metodologías que existen en el medio, por lo que se detallan las características principales de las metodologías tradicionales y ágiles, así como también los beneficios que traen cada una, para permitir que la propuesta se base en la que mejor convenga a la empresa.

También es necesario comprender las distintas bases de medición que se pueden utilizar en el desarrollo de software, para así poder medir el rendimiento que ha tenido la empresa a lo largo de su vida y generar las mejoras que faciliten una clara medición y evidente cambio.

Así mismo, en el capítulo se presentan algunos casos de análisis de las empresas de mayor reconocimiento internacional que han generado cambios o reestructuraciones a sus procesos de desarrollo de software, brindando una idea más completa de lo que puede requerir la empresa en base a las similitudes de los casos.

Sin embargo, también se requiere conocer los aspectos que conllevan la gestión y diseño de proyectos, ya que el desarrollo de software también genera proyectos que resultan en distintos productos, en donde la correcta gestión y diseño de estos proyectos puede facilitar de forma significativa el transcurso de los procesos de desarrollo de software que posee la empresa.

2.1. Desarrollo de Software

El proceso de desarrollar software tiene la finalidad de producir eficiente y eficazmente un producto de software que cumpla con las necesidades y expectativas del cliente. Este es un proceso de gran aporte intelectual, limitado por la creatividad y el raciocinio de quienes se encuentren diseñando y desarrollando el producto (Jacobson, Booch, & Rumbaugh, 2000).

Desarrollar software no es un proceso único, donde no exista un régimen universal efectivo para todo proyecto que se necesite llevar a cabo, donde se genera un producto que generalmente no existe, donde no se conoce el resultado final posible. Es por esto que no es factible automatizar o simplificar el proceso de desarrollar software (Jacobson, Booch, & Rumbaugh, 2000).

Según la Ingeniería del Software y Sistemas de Información de la Universidad Politécnica de Valencia (2003), el desarrollo de software 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.

2.1.1. Metodologías de desarrollo de Software

Debido a que el desarrollo de software posee infinitas posibilidades para cada proyecto y la forma en que se desarrollan, se han creado distintas metodologías y lineamientos que facilitan el proyecto de desarrollo de software, donde, estos métodos y lineamientos son sugerencias flexibles, más no reglas estrictas e irrompibles, ya que cada diseño y desarrollo es distinto y no cumple con las mismas características.

Clasificar cada metodología no ha sido un trabajo sencillo, ya que existen diversas propuestas con grandes diferencias en cada detalle, la información que brindan, sus alcances y diseño. Es así que existen distintas formas de clasificar dichas metodologías, sin embargo, se especificará únicamente la clasificación en consideración a la filosofía de desarrollo de cada proyecto (Díaz Polo, 2011).

Al considerar esta filosofía antes mencionada, se encuentran dos tipos: las Metodologías Tradicionales que se enfocan en la planificación y control del proyecto, los detalles, requisitos específicos y el diseño del producto; y las Metodologías Ágiles, que se enfocan en la generación de códigos, dirección de pequeños equipos de trabajo y el involucramiento del cliente en el proceso de diseño y desarrollo del producto (Díaz Polo, 2011).

Elegir la metodología que más se adecua a las necesidades del proyecto se refleja en el éxito o fracaso que pueda tener el producto resultado. El desarrollo de software es un proceso caótico a falta de guías, que se basa en la codificación, el control y el “arreglo” constante. Es por ello que elegir la metodología adecuada puede facilitar el proceso de desarrollo de software en medio de la confusión (Díaz Polo, 2011).

2.1.2. Metodologías Ágiles vs. Tradicionales

Existen variadas propuestas metodológicas que afectan en diferentes aspectos el proceso de desarrollo de software de un proyecto o producto, pero es necesario conocer y entender las diferencias de cada categoría de metodologías para poder seleccionar la más adecuada a las necesidades.

Como se mencionaba con anterioridad, se tienen en primer lugar aquellas Metodologías Tradicionales o Desarrollo en Cascada, centradas en el control del

proceso de desarrollo de software, estableciendo de forma muy específica y rigurosa cada una de las actividades que se realizarán en el proceso, los resultados que deben producirse y los recursos, herramientas y notaciones que serán utilizados para conseguir los resultados esperados. Estas metodologías se han probado a través del tiempo como herramientas efectivas y necesarias para innumerables productos a lo largo de los años, sin embargo, esto no justifica los problemas que se han presentado en varios de estos proyectos. Un camino para rectificar estas fallas se basa en la inclusión de más actividades, resultados y restricciones en cada proceso; pero esta adición a su vez tornará el resultado final a un proceso de desarrollo de software más complejo que conlleve a limitar las habilidades de los miembros del equipo para desarrollar el proyecto de la forma deseada o desarrollar el proyecto en absoluto (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

El modelo tradicional de cascada se basa en el diseño progresivo del desarrollo de un proyecto, donde se cuentan con etapas muy marcadas y separadas que son: (1) Concepción, (2) Iniciación, (3) Análisis, (4) Diseño, (5) Construcción, (6) Prueba, (7) Implementación y (8) Mantenimiento. Estas etapas se encuentran una por una en un flujo lento similar al de una cascada (de allí el nombre), donde para avanzar a una etapa se debe cumplir a cabalidad la anterior, progresando de etapa en etapa con las distintas metas que cuenta cada una. Este es un modelo de proceso estándar de flujo de trabajo, normalmente usado en la manufactura, pero ya no es tan común en el desarrollo de software de las organizaciones líderes en el terreno (SCRUMstudy, 2016).

Este proceso lento y de cierta inflexibilidad no se refiere a un método obsoleto e inservible, al contrario, aún se utiliza con buenos resultados si es bien utilizado, dependiendo del proyecto y al entorno al que se aplique, donde su mayor ventaja se centra en la división del proyecto en comportamientos separados, donde permite a los miembros del equipo del proyecto ser más independientes de sí, considerando que frente a la ausencia de uno de los miembros, la ejecución del

proyecto no se afectará en gran escala, debido a la documentación concreta que requiere el método, que sirve como un manual para que otro miembro pueda cumplir con las funciones de quien se ausente, reduciendo el impacto de los cambios (en cada etapa, mas no en el proyecto) (SCRUMstudy, 2016).

Sin embargo es importante considerar las desventajas con las que cuenta el método tradicional de cascada, donde la inflexibilidad de alterar el proyecto se puede llegar a convertir en tiempo, dinero y esfuerzo mal gastado, ya que al ser un proceso de cascada y considerando que el agua que lleva la cascada no regresa naturalmente, no se puede regresar a una etapa anterior sin afectar al diseño general del proyecto. Es por ello que es de vital importancia conocer y entender los requisitos y necesidades del cliente en la primera etapa del proyecto, ya que es un método que se basa en el concepto de que gran cantidad de tiempo invertido en la planificación y preparación del proyecto representa agilidad y ahorro de recursos en etapas posteriores (SCRUMstudy, 2016).

En segundo lugar se encuentran las metodologías enfocadas en otras dimensiones como son el talento humano y/o el producto final de software. Esta es la filosofía de las llamadas Metodologías Ágiles, que se centran en el valor que posee cada individuo, la colaboración que se tiene con el cliente y el desarrollo incremental de software con menores iteraciones. Estas metodologías se han visto efectivas en proyectos con requisitos muy fluctuantes y exigentes, así como se necesita reducir de forma drástica el tiempo invertido en el desarrollo de software manteniendo constante la calidad esperada (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

El Método Ágil es un acercamiento progresivo, a diferencia del método tradicional, donde el equipo trabaja en módulos pequeños del proyecto, donde las modificaciones de los requisitos por parte del cliente se realizan sin muchos cambios que afecten al proyecto general, con un diseño simple que permite

realizar correcciones a medida que el proyecto se vaya desarrollando (SCRUMstudy, 2016).

Al comparar estas metodologías con el método en “cascada”, se puede observar que en este tipo de proyectos se da prioridad a las interacciones entre las partes interesada por sobre los procesos y la burocracia (SCRUMstudy, 2016). Las Metodologías Ágiles aparecieron en la historia como resultado a las falencias y necesidades que existían con las Metodologías Tradicionales, sin embargo, ambas existen en la actualidad, donde no se descarta lo antiguo por lo nuevo, al contrario, se aplican aquellas que mejor se adapten a las necesidades y expectativas de todas las partes interesadas del proyecto (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

Mencionado lo anterior, es importante destacar que las Metodologías Ágiles han revolucionado la forma en que se ha desarrollado software con anterioridad, generando una amplia polémica entre quienes la prefieren y los desarrolladores escépticos o con necesidad de convencimiento que no las consideran como una alternativa viable que reemplace las Metodologías Tradicionales (Fitsilis, 2008).

Las metodologías Ágiles cuentan con la desventaja de requerir con desarrolladores calificados no solo en el desarrollo de software, pero también en la dirección de proyectos y el uso de estas metodologías, así como también cuentan con la desventaja de ser consideradas un acercamiento incompleto al desarrollo de ciertos proyectos, donde al no contar con todas las etapas del modelo en cascada y un proceso constante de retroalimentación, se puede llegar a modificar un proyecto en tal magnitud que se pierde la esencia de la meta que se buscaba lograr; mas es importante establecer que al enfocarse en áreas más pequeñas de trabajo, se pueden reducir gastos generales y el costo del proyecto en general, llegando a ser un método apropiado cuando se tiene gran nivel de

incertidumbre sobre las necesidades del cliente o el ambiente de negocio (SCRUMstudy, 2016).

Es necesario comprender que ambas metodologías cuentan con ventajas y desventajas, ninguna es completamente superior a la otra sin considerar el contexto y tipo de un proyecto, simplemente se entiende que las metodologías Ágiles nacieron para contrarrestar las falencias de las metodologías tradicionales en Cascada, que no deja de ser un método muy utilizado y apropiado en ambientes estables (SCRUMstudy, 2016).

2.1.2.1. Metodologías Tradicionales

Hasta hace poco, el proceso de desarrollo de software se enfatizaba en especial manera con el control de sí mismo, mediante un cuidadoso establecimiento de definiciones de roles, actividades, recursos, modelos y documentación completa y detallada. Este modelo de trabajo “tradicional” o “en cascada” para realizar el proceso de desarrollo de software se ha visto eficiente, eficaz y necesario en proyectos que cuenten con un tamaño considerablemente grande, en base al tiempo y recursos con los que se cuenta, donde la burocracia y el papeleo se halla en alto grado en el transcurso de todo el proceso (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

Aun así, esta clasificación de metodologías de desarrollo de software no es la óptima para proyectos contemporáneos y coyunturales, donde se vive un ambiente de constante cambio, con la exigencia absoluta de la reducción del tiempo invertido en el desarrollo del software, sin perder ante esta situación su alto grado de calidad. (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

Debido a estas dificultades del uso de las Metodologías Tradicionales ante las restricciones abruptas de tiempo y excesiva flexibilidad, varios equipos de desarrolladores de software se resignan a omitir el “buen hacer” de la ingeniería del software, asumiendo un gran riesgo ante esta omisión, un riesgo necesario ante las circunstancias extremas de desarrollo del producto (Fitsilis, 2008).

Es así que surgen ante la necesidad del vacío metodológico las Metodologías Ágiles como una posible solución de las exigencias actuales. Estas metodologías se dirigen principalmente a proyectos de pequeña escala, en las que aportan una importante simplificación del proceso de desarrollo de software, donde, aunque se reduzca el tiempo del proceso, no se pierde la calidad del producto, al contrario, se busca asegurarla (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

Una vez analizadas las principales características de ambas metodologías, la presente investigación se basará principalmente en el desarrollo de software en base a las metodologías ágiles como base para la propuesta que se desarrollará en los siguientes capítulos.

2.1.2.2. Movimiento Ágil

En febrero del 2001, se celebró una reunión en Utah, Estados Unidos, donde se crea y desarrolla el concepto de “Metodología Ágil” en el área de desarrollo de software. Esta reunión se generó entre 17 expertos y conocedores del proceso de desarrollo de software, entre ellos, varios creadores de algunas metodologías específicas. La principal razón por la que se llevó a cabo esta reunión fue el de delinear y proyectar los valores y principios que se pudieran permitir a los equipos de participar en el desarrollo de software de forma rápida, en donde se responda de adecuada manera a los cambios que se generen en la vida del

proyecto (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

Las organizaciones en la actualidad se enfrentan a la dificultad de mantenerse a la vanguardia de los múltiples y veloces cambios que enfrentan diariamente utilizando el método tradicional de Dirección de Proyectos. Estos cambios se ven reflejados en demandas periódicas, requisitos constantemente cambiantes, fallos en las actividades de soporte de la cadena de valor, entre otros. Es por esto que hoy en día los Directores de Proyecto y desarrolladores de sistemas se inclinan por el uso de las metodologías ágiles que existen en el entorno, incluyendo al Departamento de Defensa de los Estados Unidos, que en sus reglas de contratación de desarrolladores de software y directores de proyectos especifican que no requieren soluciones basadas en el modelo tradicional de “Cascada” mencionado anteriormente (SCRUMstudy, 2016).

Generar “nuevas” metodologías de desarrollo de software fue un plan necesario para poder poseer alternativas a los procesos de desarrollo tradicionales, que, como se mencionó con anterioridad, eran inflexibles, rígidos, burocráticos y llevados por una inmensa documentación que se genera en cada actividad realizada y desarrollada. Es por esto que tras la reunión del 2001, se crea The Agile Alliance, partiendo del Manifiesto Ágil, el documento en el que se engloba y conoce la filosofía de lo “ágil” (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

Las metodologías ágiles buscan dividir las tareas y procesos complejos en incrementos pequeños de planificación nominal. Estas divisiones son llamadas iteraciones que son ventanas de tiempo de corta duración, llegando a tomar de una a cuatro semanas. Cada iteración se ve envuelta por un equipo multidisciplinario, donde se cuenta con expertos en planificación, análisis de requerimientos, diseño, codificación, prueba de unidades y pruebas de

aceptación. Finalizada cada iteración se presenta un prototipo de producto que cumpla con los requisitos elementales del cliente, reduciendo el riesgo general que trae consigo la necesidad de adaptarse a los cambios bruscos de dichos requisitos o el entorno en general (SCRUMstudy, 2016).

2.1.2.3.El Manifiesto Ágil

El Manifiesto Ágil valora varios puntos del proceso de desarrollo de software, especificados a continuación:

- Al desarrollador o programador y las interacciones con el equipo de desarrollo sobre el proceso como tal y las herramientas que utilizan. Se reconoce que las personas son el factor principal de éxito o fracaso de un proyecto, donde es de mayor importancia generar un óptimo equipo de desarrollo que la construcción del entorno de desarrollo. Es así que se reconoce el frecuente error de priorizar la construcción del entorno con la esperanza que el equipo de desarrollo se adapte. Es recomendable crear el equipo en primer lugar, para que este decida, configure y adapte el entorno ante las necesidades que el equipo posea (Manifiesto Ágil, 2001).
- Se enfatiza la mayor importancia que tiene el desarrollar software funcional ante una amplia documentación. De este concepto parte la regla que no se deben producir documentos constantes, a menos que estos documentos sean necesarios inmediatamente para tomar decisiones vitales. Aun así, los documentos deben ser concisos y centrados en los aspectos fundamentales de su razón de ser (Manifiesto Ágil, 2001).
- Se prioriza el contacto y colaboración con el cliente ante la negociación de contratos. Es de esta forma que se busca que se genere una interacción constante entre el cliente y el equipo de desarrollo. Esta interacción deberá

ser de retro alimentación, donde se marque la velocidad y la marcha del proyecto, asegurando así su éxito o fracaso (Manifiesto Ágil, 2001).

- Se busca responder ágilmente a los cambios en lugar de seguir un plan de forma estricta con instrucciones precisas. Cada proceso de desarrollo y el equipo de trabajo poseen la habilidad de responder a cualquier cambio que pueda existir o surgir en la vida del proyecto, ya sea en base a requisitos, tecnología, el equipo mismo, entre otros; esta resiliencia determina el resultado del proyecto, por lo que toda planificación no debe ser rígida y cerrada, al contrario, procurar la flexibilidad y apertura al cambio (Manifiesto Ágil, 2001).

Estos valores son aquellos que inspiraron los doce principios del Manifiesto Ágil, diferenciando en alto grado un proceso de desarrollo de software Ágil ante uno Tradicional. Los doce principios fueron generados en base a las necesidades de adaptación que requieren los entes del nuevo milenio y la forma en la que lo logran. Estos principios son:

1. Tener como máxima prioridad la satisfacción del cliente a través de entregas a tiempo y continuas de software que posea valor.
2. Aceptar de que los requisitos cambien, incluyendo en las etapas más avanzadas de un proyecto de desarrollo, ya que los procesos ágiles aprovechan dicho cambio para generar una ventaja competitiva para el cliente.
3. Entregar software funcional de forma frecuente, entre dos semanas a dos meses, preferiblemente, el periodo más corto de tiempo posible para el equipo y el proyecto.

4. Trabajar conjuntamente, tanto los administradores del negocio o proyecto y los desarrolladores de software, donde se trabajará cotidianamente sin altercados.
5. Tener desarrolladores motivados en el proyecto, donde se les facilite el entorno y apoyo necesarios, así como la total confianza de su ejecución en su trabajo.
6. Hacer uso constante y frecuente del método más eficiente y eficaz de comunicación de la información al equipo de desarrollo y sus integrantes, el cual es la comunicación personal “cara a cara”.
7. Entender que el software que funciona es la medida principal para medir el progreso.
8. Promover el desarrollo sostenible a través de procesos de desarrollo de software ágiles, donde los administradores, desarrolladores y usuarios sean capaces de mantener la constancia indefinidamente.
9. Saber que para mejorar la agilidad, se debe prestar atención continua a la excelencia técnica y al diseño.
10. Conocer que es esencial la simplicidad, también expresada como “el arte de maximizar la cantidad de trabajo no realizado”
11. Generar las mejores arquitecturas, requisitos y diseño a través de equipos auto-organizados.
12. Ajustar y perfecciona el comportamiento del equipo de desarrollo tras la reflexión periódica de formas de lograr ser más efectivo, eficiente y eficaz.

2.1.2.4. Metodologías Ágiles

El movimiento de la Alianza Ágil, Movimiento Ágil o Agile Alliance (previamente mencionado), cuenta con un importante número de miembros y se ha convertido en un frecuente concepto en el desarrollo de software como el “Desarrollo Ágil de Software” o “Desarrollo Ágil” (Díaz Polo, 2011). Es importante establecer las cuatro características más importantes que tienen todas las metodologías ágiles en común, las cuales son:

1. **Desarrollo en ciclos cortos:** las metodologías ágiles cuentan con “Sprints” que no suelen ser mayores a 30 días, donde se buscan soluciones parciales que funcionen en el corto plazo, o a su vez se cuenta con ciclos iterativos muy cortos donde se busca entregar software funcional en cualquier etapa del proceso de desarrollo (SCRUMstudy, 2016).
2. **Valorar a las personas:** Los miembros del equipo son valorados, ya que representan el factor más importante para el desarrollo del proyecto, brindándoles libertad creativa para solucionar las necesidades del cliente y los problemas del proyecto, trabajando a un ritmo desafiante pero continuo, que es capaz de mantenerse indefinidamente y sin interrupciones (SCRUMstudy, 2016).
3. **Involucrar al cliente:** Al intentar conseguir resultados veloces, se debe considerar al cliente como parte fundamental en el equipo, donde incluso puede llegarse a asignarles un espacio físico junto al equipo para mantener cerca su retroalimentación y necesidades actualizadas (SCRUMstudy, 2016).

4. **Mantener la simplicidad:** Se mantiene el pensamiento de que, ante la elección de una solución sofisticada o una solución simple, siempre se debe optar por la opción simple, ya que se busca que el producto final cumpla a cabalidad con lo que se necesita, evitando trabajo arduo, extra e innecesario, donde incluso la codificación se mantiene en un formato simple (SCRUMstudy, 2016).

Es importante destacar que estas metodologías ágiles son una categoría, es decir, engloba un gran número de metodologías variadas enfocadas en la eficacia del proceso de desarrollo de software, donde se brinde agilidad sobre una base de reglas y prácticas audaces, donde las más conocidas, más famosas y más utilizadas son: eXtreme Programming y SCRUM, pero para efectos del estudio, se detallaran ellas y otras a continuación:

Programación Extrema, Extreme Programming o XP

La Programación Extrema es una metodología que busca centrarse en potenciar las relaciones interpersonales de los miembros del equipo, reconociendo que este factor es una clave del éxito del desarrollo de software, por lo que es vital para el proyecto facilitar y promover el trabajo en equipo, haciendo énfasis en el aprendizaje y desarrollo personal de cada uno de sus integrantes, y finalmente, brindar el más apto y sencillo clima de trabajo (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

La metodología XP vive de la retroalimentación continua entre el usuario/cliente y el equipo de desarrolladores, a través de una comunicación constante y fluida entre todos quienes intervienen en el proceso, generando soluciones simples y sencillas para poder ser implementadas ante las fallas del proceso y la capacidad de resiliencia necesaria para poder afrontar los cambios de las necesidades o el

entorno. Esta metodología se define como la más apta para proyectos cuyas necesidades y requisitos no sean precisos o a su vez sean de cambios volátiles, que generen un alto nivel de riesgo técnico. (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

Scrum

Esta metodología posee un marco para la gestión de proyectos que ha sido utilizado ya aproximadamente por 10 años con varios casos de éxito. Su base de trabajo se enfoca en proyectos que tengan cambios de requisitos constantes y frecuentes y se caracteriza por dos motivos principales. Primordialmente, el desarrollo de software se realiza mediante iteraciones o sprints que duran 30 días y obtienen como resultado de cada uno de ellos un incremento funcional del proyecto para ser utilizado por el cliente. En segundo lugar, la metodología Scrum consta de reuniones frecuentes durante la vida del proyecto, estas reuniones son de 15 minutos aproximadamente todos los días, para poder coordinar el trabajo e integrar al equipo de desarrollo (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

Esta metodología ha adquirido gran importancia en las organizaciones de desarrollo de software debido a que es un área de gran competencia empresarial, donde diariamente incrementan las compañías de esta índole, generando la necesidad de optimizar los recursos al máximo y generar más productos para poder mantener una posición aventajada en una carrera digital. Es por ello que los profesionales especializados en esta metodología son demandados con mayor frecuencia en los proyectos de desarrollo (Goikolea, 2014).

Dynamic Systems Development Method (DSDM)

Se caracteriza por promover un proceso iterativo e incremental, donde el equipo de desarrolladores y el usuario trabajan en conjunto. Desarrolló cinco fases para los proyectos, las cuales son: (1) estudio de viabilidad, (2) estudio de negocio, (3) modelado funcional, (4) diseño y construcción y (5) implementación. Las fases 3, 4 y 5 son iterativas y brindan retroalimentación a las fases 1 y 2 (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

Adaptative Software Development (ASD)

Es una metodología iterativa, orientada a los componentes del software desarrollado en lugar de a las tareas a realizar, tornándose tolerante al cambio. El ASD propone distintas fases como DSDM, con la diferencia que propone tres de ellas en lugar de cinco, las cuales son: (1) especulación, donde se inicia el proyecto y se planifican las características que el producto tendrá; (2) colaboración, en donde se desarrollan las características especificadas en la fase 1 y (3) aprendizaje, el cual controla la calidad del producto elaborado y lo entrega al cliente, siendo una fase de gran importancia, en donde la revisión de los componentes del software permiten a los desarrolladores aprender de sus errores y reiniciar el proceso en la etapa 2 de desarrollo (Grupo ISSI: Ingeniería del Software y Sistemas de Información, 2003).

Lean Development (LD)

“Los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente.” (Poppendieck.LLC, 2015)

2.1.3. Bases de medición

Se puede asumir que en la empresa todos saben cómo realizar su trabajo, sin embargo, es difícil para los programadores explicar de qué manera su trabajo ha mejorado, ya que el desarrollo de mejor software es un tema subjetivo al corto plazo.

Según Albaladejo (2008), existen cuatro principales formas para medir las mejoras de los proyectos ágiles que se presentan a continuación:

2.1.3.1.Métricas de productividad y efectividad de la entrega

- Velocidad del cumplimiento de objetivos en cada iteración: Permite extrapolar las fechas de entrega de productos.
- Lead Time: Tiempo de entrega de un requisito tras su petición.
- Prioridad de los requisitos completados: Se mide en relación a los objetivos de los proyectos y los requisitos que se han cumplido, procurando se mantengan alineados.

2.1.3.2.Métricas de resultados del proyecto

- Velocidad para aportar valor a la empresa: Es una medición realizada desde el punto de vista del cliente.

- Valor Entregado: Esta medición se basa en un único indicador económico, que puede estar ligado a las ventas directas de cierta aplicación o la inversión realizada según convenga a la empresa.
- Requisitos completados en la iteración.
- Próximos requisitos a desarrollar.
- Entrega de objetivos y velocidad: Esta medición relaciona el cumplimiento de los objetivos establecidos en cada sprint y el tiempo en que se toma cumplirlos, dividiendo cada objetivo en categorías según su nivel de complejidad.
- Defectos: Se mide el número de defectos y cambios necesarios para corregirlos en el paso del tiempo.
- Cambios incorporados: Se realiza en contraste con el alcance inicial de cada proyecto.
- Número de requisitos completados: Respecto al total de requisitos.
- Desviación de resultados: Análisis de los resultados del proyecto respecto a su planificación inicial.

2.1.3.3. Métricas de situación financiera

- Retorno de Inversión (ROI) pendiente, el valor pendiente respecto al coste pendiente, para saber cuándo finalizar el proyecto.
- Presupuesto disponible y/o presupuesto gastado.
- Desviación financiera respecto a la planificación inicial.

2.1.3.4. Métricas de calidad

- Expectativas:
 - Satisfacción del usuario, respecto a los resultados del proyecto.
 - Ambiente en el equipo.

- Calidad funcional:
 - Defectos detectados por el usuario.
 - Defectos detectados internamente.
 - Cobertura de las pruebas.
 - Trazabilidad.

- Mantenibilidad:
 - Cumplimiento de estándares, normativas, regulaciones, etc.
 - Comentarios en el código.
 - Tamaño de las operaciones.
 - Calidad de la documentación.

2.1.3.5. Métricas de riesgos, impedimentos, proceso y mejora continua

- Riesgos e impedimentos.

- Actividades de mejora a planificar: En relación a las comunicaciones, formaciones, soporte, herramientas, etc.

- Situaciones anómalas.

- Porcentaje de desarrolladores que no intervienen en las reuniones diarias.

2.1.4. Casos de análisis

Para poder seleccionar una metodología de desarrollo de software adecuada para la mejora de procesos de la empresa HSBCAD SOFTECUADOR C.A., o incluso si el cambio de metodología es necesario, es importante conocer algunos casos de implementación ágil de software como parte de la experiencia.

Healthcare.gov

Healthcare es un proyecto gubernamental de los Estados Unidos de América, donde se buscaba ofrecer información completa y transparente acerca del mercado de seguros de salud, de esta manera, los usuarios pudiesen escoger el de mayores beneficios y mejor valor agregado. Fue así que se decidió implementar la Metodología Ágil de desarrollo de Software SCRUM (Goikolea, 2014).

El proyecto consistió en un fracaso por variadas causas, ya que no existió adecuada coordinación entre el Front End y el Back End, el liderazgo fue insuficiente en un proyecto en el que colaboraban más de 20 consultoras y finalmente, el proyecto fue lanzado al público sin antes haber realizado una prueba que permita el aprendizaje y las mejoras, por ello, no se pudieron detectar errores importantes para el usuario (Goikolea, 2014).

Este proyecto fue desarrollado por varios años, sin embargo, solo se contó con seis días para poder probar su aplicación, ya que existía presión constante por el lanzamiento rápido del producto a los usuarios. Este periodo insuficiente de prueba es una de las causas claves para que el resultado del proyecto consistiese

en fracaso, sin importar que el Back End trabajó arduamente como la metodología ágil solicitaba, excepto en respecto al segundo principio del Manifiesto Ágil, referente a la aceptación del cambio. Es por ello que, aunque el equipo de Back End cumplió su trabajo en unos meses, el proyecto no funcionó adecuadamente (Goikolea, 2014).

Google

Google cuenta con 15 000 desarrolladores de software que trabajan bajo la metodología ágil de SCRUM, estos colaboradores trabajan en el desarrollo de la rama del código. Realizan cambios múltiples veces y a su vez, se elaboran 75 millones de pruebas automáticas, todo esto en un enfoque diario (Goikolea, 2014).

Spotify

Spotify se enfoca en la figura de Scrum Master. La compañía contrata frecuentemente un Agile Coach o Entrenador Ágil externo, con el requerimiento de que cuente con amplia experiencia para tener la capacidad de liderar todos sus proyectos. Es así como se puede observar la importancia de poder contar con roles especializados, que sean expertos de las metodologías que se buscan implementar, así poder garantizar casi en su totalidad el éxito del proyecto (Goikolea, 2014).

Spotify también contrata a un experto en otros roles de importancia como es el Product Owner, que es el responsable de conocer y entender al cliente/usuario y

sus necesidades, así poder transferir la información al equipo de desarrolladores de forma eficiente. La compañía decidió adoptar la metodología Scrum de forma sistemática, en donde, para poder mantenerse en el negocio ante competidores como Google o Apple, debían ser más veloces, económicos y mejores en todos los aspectos (Goikolea, 2014).

En la compañía, se organizan escuadrones de equipos pequeños de Scrum que tengan la competencia y habilidad de implementar el producto que desarrollen el finalizar cada uno de sus sprints, sin afectar a otro equipo. Spotify posee un funcionamiento poco convencional, en donde cada pequeño equipo de desarrolladores es poseedor de una porción del producto que le pertenece. Posteriormente, se generan tribus que son el conjunto de diferentes escuadrones. Esta dinámica se puede observar en la Figura 6 a continuación (Goikolea, 2014).

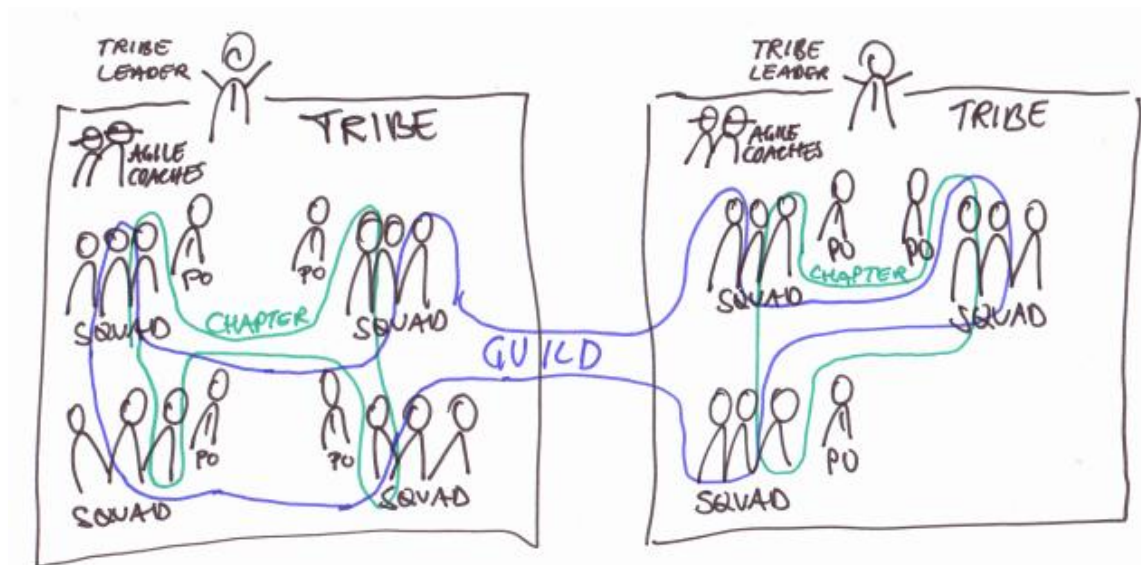


Figura 6.- Funcionamiento de escuadrones y tribus en Spotify

Fuente: (Goikolea, 2014)

Sin embargo, Spotify aún tiene la necesidad de implementar, cambiar y actualizar su código constantemente sin afectar a ningún área adicional. Es por ello que, para la compañía, una excelente coordinación central es primordial para su supervivencia (Goikolea, 2014).

Para poder ser el más veloz en una industria, se necesita deshacerse de aquello que pueda entorpecer el avance de un proceso o proyecto. Spotify contaba con un excelente equipo de operaciones encargado de implementar producto, excepto que lo hacían de una manera demasiado lenta. Por ello, se eliminó la fase de implementación y se buscó que cada desarrollador implemente sus códigos y productos al sistema de la empresa (Goikolea, 2014).

Belike Software

La empresa de software no considera a lo ágil como metodologías, en cambio lo considera un método de trabajo adaptable a cada proyecto, manteniendo en cuenta su tipo, su historial, el perfil de sus clientes y el conjunto de recursos que se hayan utilizado, se conozcan o se deseen probar. Es una empresa que indica que esta herramienta les ha permitido la eficiencia y velocidad en el desarrollo que planearon al inicio. Como tantas otras compañías, Belike decidió trabajar bajo la metodología Scrum, que, no es seguida al pie de la letra, de hecho, ha sido adaptada con la unión de añadidas reglas que reflejan los 4 valores del Manifiesto Ágil (Bienve, 2016).

Belike considera que su método de trabajo es regido a cabalidad tanto en el desarrollo de software como en la negociación y venta de un producto. La comunicación en su proyecto de construcción de un sistema (de más de 3000 horas de desarrollo) es crítica y vital para su éxito. A su vez, existen roles

imprescindibles en la empresa que no pueden fallar: Product Owner, Technical Leader y Executive Manager (Bienve, 2016).

El Product Owner es el que posee la responsabilidad operativa y ejecutiva más grande de un proyecto, completando un perfil de habilidades técnicas y de gestión. Esta persona es considerada la guía visible de todo proyecto, a quien se solicita mayor dedicación y horas de trabajo. El Technical Leader es el responsable operativo de la empresa, siendo quien coordina los trabajos a desarrollar y el canal principal de comunicación que existe en el proyecto. El Executive Manager es un rol que cumplen dos personas, el responsable ejecutivo del proyecto con el cliente y su semejante en Belike. Para el éxito de sus proyectos, Belike también ha desarrollado tres conceptos principales: (1) la comunicación directa con clientes y proveedores, (2) software funcional a diario y (3) maximización del tiempo de programación. Esta dinámica puede ser observada en la Figura 7 (Bienve, 2016).

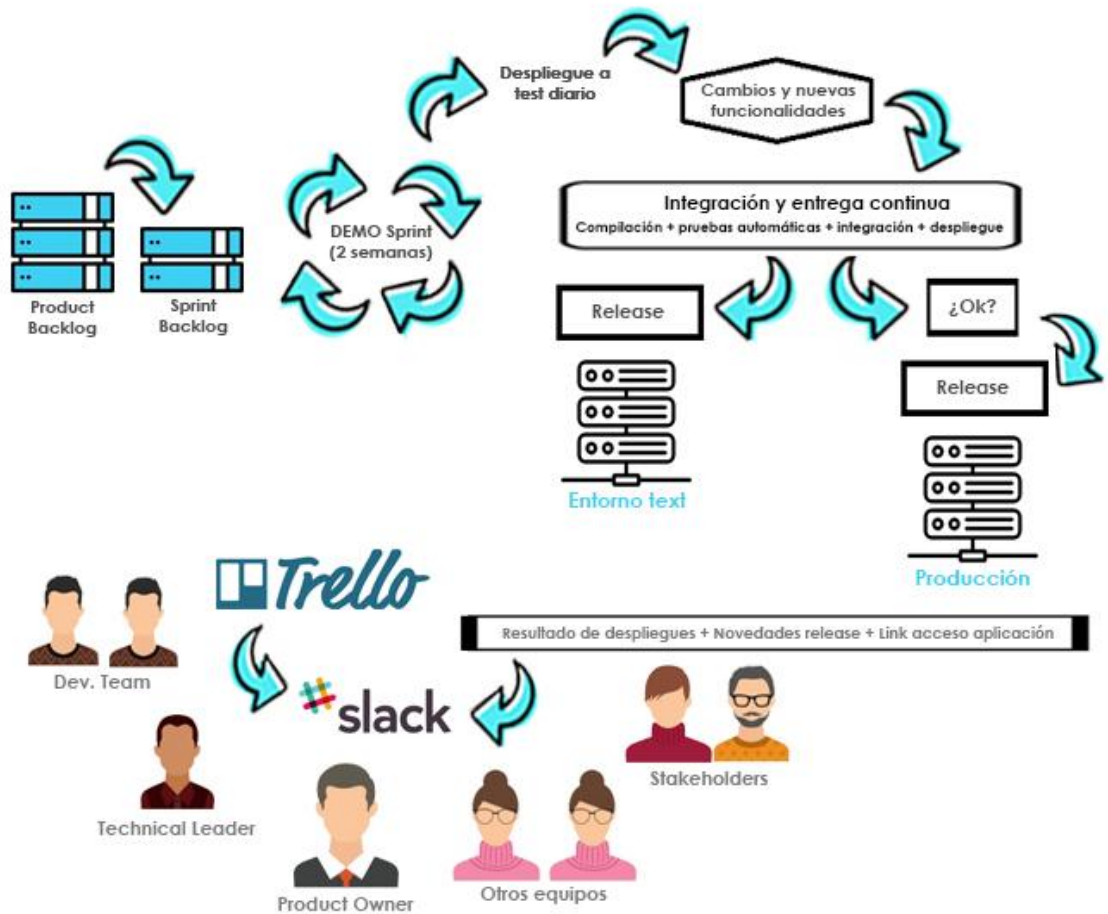


Figura 7.- Modelo de desarrollo de Belike Software

Fuente: (Bienve, 2016)

La comunicación directa se apoya en herramientas colaborativas como: Trello que facilita la organización del trabajo, Slack que conecta a los integrantes del equipo para conocer los cambios que ocurren constantemente a tiempo real, Google Drive como almacenamiento, Google Hangout y Skype como apoyo a las reuniones a distancia con clientes, proveedores e integrantes (Bienve, 2016).

El concepto de software funcional a diario se basa en que el usuario o cliente pueda percibir los cambios en el tiempo más real posible, para así poder tener la

retroalimentación necesaria para realizar mejoras o enmendar errores. Para poder realizar esta hazaña, se utilizan dos estrategias: la primera consiste en la Integración Continua, que se basan en integraciones automáticas continuas de ejecución de pruebas para detectar errores a tiempo y en segundo lugar se encuentra la Entrega Continua, donde se adaptan los cambios al código en pruebas. Este entorno de prueba para el cliente y el equipo facilita al equipo a mantener actualizados los cambios y brindarlos a todos los clientes una vez sean aceptados (Bienve, 2016).

La maximización del tiempo de programación hace referencia a dedicar el mayor tiempo posible de los desarrolladores a programar, en lugar de realizar reuniones largas. Las medidas que se utilizaron para lograr esto son: en caso de duda, se pregunta o comparte información sin necesidad de reuniones formales por asuntos que podrían solucionarse sencillamente, en las dos reuniones diarias de 10 minutos (la primera con el Product Owner y el Technical Leader y la segunda con el equipo de desarrollo) se tratan los temas realmente importantes, en presencia de quienes realmente deben estar en esa reunión, para no quitar valioso tiempo a los desarrolladores (Bienve, 2016).

2.2. Dirección de Proyectos

PMBOK (Project Management Body of Knowledge) es una útil herramienta para la dirección y desarrollo de un Proyecto, donde la versión número 6 del año 2017 hace referencia al desarrollo de proyectos en base al manifiesto ágil (Wordpress, 2017).

Las áreas de conocimiento del PMBOK6 son una guía básica para entender todas las partes que un proyecto de desarrollo de software con fundamentos ágiles debería contener, las cuales son:

1. Administración de la Integración del Proyecto: describe los procesos y actividades que integran los diferentes aspectos de la Dirección de Proyectos. Esta área se divide en:
 - a. Desarrollo del plan del proyecto
 - b. Desarrollar la declaración preliminar del alcance del proyecto
 - c. Desarrollar el plan de Dirección del Proyecto
 - d. Dirigir y Manejar la Ejecución del Proyecto
 - e. Monitorear y controlar el trabajo del Proyecto
 - f. Control del Cambio integrado
 - g. Cierre del Proyecto

2. Dirección del Alcance del Proyecto: Considera a los procesos que son responsables de controlar el alcance del proyecto, consistiendo de:
 - a. Planificación del Alcance
 - b. Definición del Alcance
 - c. Creación de la estructura de la división del trabajo (WBS)
 - d. Verificación del Alcance
 - e. Control del Alcance

3. Administración del Tiempo del Proyecto: describe a los procesos en relación a la finalización a tiempo del proyecto. Conforman a:
 - a. Definición de la actividad

- b. Secuencia de la actividad
 - c. Estimación de los recursos de la actividad
 - d. Estimación de la duración de la actividad
 - e. Desarrollo de cronograma
 - f. Control del cronograma
4. Administración del Costo del Proyecto: incluye a los procesos en relación al costo. Estos procesos son:
- a. Estimación del costo
 - b. Presupuesto del costo
 - c. Control del Costo
5. Administración de la Calidad del Proyecto: describe a los procesos involucrados en asegurar que el proyecto cumplirá los objetivos para los que se ha creado. Consiste en:
- a. Planificación de la calidad
 - b. Aseguramiento de la calidad
 - c. Control de calidad
6. Administración del Recurso Humano del Proyecto: incluye a todos los procesos necesarios para organizar y administrar a los integrantes del equipo. Se desarrolla por:
- a. Planificación del recurso humano
 - b. Creación del equipo de trabajo
 - c. Desarrollo del equipo de trabajo
 - d. Administración del equipo de trabajo

7. Administración de la Comunicación del Proyecto: Describe a los procesos que hacen referencia a todos los medios de comunicación del proyecto, relacionados con que sean oportunos y apropiados, además de que exista adecuados procesos de generación, recolección, análisis y almacenamiento de la información del proyecto. Sus procesos consisten en:
 - a. Planificación de la Comunicación
 - b. Distribución de la Información
 - c. Informes de rendimiento
 - d. Administración de los grupos de interés

8. Administración del riesgo del proyecto: describe a los procesos relacionados con la administración de las amenazas con las que cuenta un proyecto, los cuales son:
 - a. Planificación de la administración del riesgo
 - b. Identificación del riesgo
 - c. Análisis cualitativo del riesgo
 - d. Análisis cuantitativo del riesgo
 - e. Planificación de la respuesta al riesgo
 - f. Control y monitoreo del riesgo

9. Administración del Logro del Proyecto: incluye a los procesos que se relacionan con adquirir los productos y servicios necesarios para completar el proyecto, estos son:
 - a. Planificación de compras y adquisiciones
 - b. Plan de contratación
 - c. Cotizaciones
 - d. Selección de proveedores

- e. Administración de contratos
- f. Cierre de contratos

2.2.1. Comparación de PMBOK6 y las metodologías ágiles

Es importante entender que la guía PMBOK es una lista de prácticas que, en forma de procesos, pueden llegar a ser adaptadas a las diferentes necesidades de un proyecto, siendo una base reconocida y aceptada en el uso de las metodologías ágiles en la actualidad (Fitsilis, 2008).

3. CAPÍTULO 3.- DIAGNÓSTICO

El presente capítulo tratará del diagnóstico de la empresa HSB SOFTECUADOR C.A., para determinar la situación en la que se encuentra el proceso de desarrollo de software de la empresa en la actualidad, para así poder establecer las mejoras necesarias y a su vez generar una base de comparación de los beneficios que pudieran otorgar la propuesta.

El diagnóstico se inicia a través de la determinación del flujo de todos los procesos que se considere, intervengan directamente con el desarrollo de software, permitiendo conocer las falencias y fortalezas que pueden ser utilizadas para generar una mejora futura.

El diagnóstico también es conformado por el estudio de las brechas que posee el funcionamiento y diseño de los procesos de desarrollo de software de la empresa actualmente en comparación con la metodología ágil que se pretende utilizar en su totalidad y los aspectos que se deben llevar a cabo para cumplir los objetivos que se poseen.

Finalmente, se analizan los proyectos finalizados más recientes que posee la empresa, su rendimiento en medidas de tiempo, costos, calidad y productividad, permitiendo conocer si la empresa debería mantenerse en su estado actual o adoptar la propuesta que se genera en posteriores capítulos.

3.1. Objetivos del diagnóstico

Los objetivos del diagnóstico de la empresa HSB SOFTECUADR C.A. se enfocan en la situación actual de la empresa y son los siguientes:

1. Conocer el proceso de desarrollo de software actual de la empresa.
2. Conocer las brechas que existen entre la metodología de trabajo de la empresa y la metodología de trabajo ágil de desarrollo de software.
3. Conocer el rendimiento de la empresa en proyectos pasados.

3.2. Metodología del diagnóstico

Para cumplir el objetivo 1 del punto anterior se procederá a levantar los procesos actuales a través de diagramas de flujo con la ayuda de información provista por el personal a través de entrevistas, así como documentación que permita evidenciar los resultados del diagnóstico. Se realizará un inventario de procesos que determinará los procesos de desarrollo de software más importantes para HSB SOFTECUADOR C.A. que deben ser analizados.

Relacionando al objetivo 2, se desarrollará un análisis de cumplimiento de la metodología ágil de desarrollo de software mediante la recopilación de información a través de cuestionarios, análisis de documentación, observación del trabajo del equipo y entrevistas tanto personales como grupales

Finalmente, para cumplir el objetivo 3 se elaborará un análisis de los proyectos más importantes de la empresa, con la colaboración de la información aportada por la empresa partiendo de sus registros e indicadores de desempeño.

3.3. Resultados

3.3.1. Diagnóstico de Procesos

Para poder iniciar con el detalle de los diagramas de Flujo de los procesos principales actuales de la empresa HSB SOFTECUADOR C.A., se iniciará con un inventario de procesos detallado en la Tabla 1.- Inventario de procesos de HSB SOFTECUADOR C.A. Tabla 1; **Error! No se encuentra el origen de la referencia.**, donde se analizarán los más relevantes para poder cumplir con los proyectos establecidos por la alta dirección.

Tabla 1.- Inventario de procesos de HSB SOFTECUADOR C.A.

N°	PROCESOS DE HSB SOFTECUADOR C.A.
1	Contratación e Inducción
2	Desarrollo de Software
3	Distribución de productos
4	Comunicación Interna
5	Mantenimiento de Equipos
6	Adquisición de nuevos Equipos
7	Control de proveedores

Para poder determinar los procesos que serán analizados, se priorizaron aquellos con mayor impacto y frecuencia como se puede ver en la Tabla 2, en donde con una escala del 1 al 5, siendo uno de menor impacto o frecuencia y 5 de mayor impacto a frecuencia, se calificaron a todos los procesos con los que cuenta la empresa, eligiendo finalmente a aquellos cuyo impacto por frecuencia sean mayores a 10.

Tabla 2.- Priorización de procesos por impacto y frecuencia

Procesos	Impacto	Frecuencia	Impacto por frecuencia
Contratación e Inducción	5	3	15
Desarrollo de Software	5	5	25
Distribución de productos	4	3	12
Comunicación Interna	3	3	9
Mantenimiento de Equipos	2	2	4
Adquisición de nuevos Equipos	3	1	3
Control de proveedores	2	2	4

Una vez consultado el impacto y frecuencia de los procesos, se detallan a continuación los procesos seleccionados por HSB SOFTECUADOR C.A. para ser analizados en el presente diagnóstico: Contratación e Inducción, Desarrollo de Software y Distribución de Productos. Los diagramas de flujo correspondientes a los procesos seleccionados serán elaborados por el método ANSI y se encuentran en el Anexo A.

3.3.1.1. Contratación e Inducción

Uno de los procesos más importantes que determinan el éxito de todos los proyectos es el proceso de Contratación e Inducción de los nuevos empleados, ya que tener el equipo apropiado es de vital importancia para el cumplimiento a tiempo de todos los objetivos. Sin embargo, cabe recalcar que el proceso de Contratación e Inducción no se lo realiza por cada proyecto nuevo que se desarrolla, al contrario, se lo lleva a cabo cuando la Alta Dirección Internacional lo demanda en base a los planes futuros con los que se cuenta.

El proceso de Contratación e Inducción inicia en el momento en el que la Alta Dirección Internacional solicita la contratación de nuevo personal como se

mencionaba con anterioridad. Este proceso se lleva a cabo únicamente por el Gerente General y el nuevo personal contratado, que generalmente son estudiantes universitarios de semestres iniciales en carreras afines a las actividades de la empresa.

El nuevo personal se mantiene en un periodo de aprendizaje de aproximadamente un ciclo de desarrollo de software, es decir 6 meses en el caso de HSB SOFTECUADOR C.A., en este periodo únicamente el personal se dedica a la investigación y aprendizaje de los métodos de trabajo de la empresa. Este periodo de aprendizaje se base en una guía didáctica que se entrega a los nuevos empleados con temas necesarios para el desarrollo de software en la empresa, estos temas deben ser consultados y posteriormente reportados los puntos de interés para demostrar el aprendizaje.

Una regla importante en el periodo de aprendizaje se resume en que si el nuevo empleado se toma demasiado tiempo en consultar e investigar sobre un tema, debe solicitar ayuda a los demás miembros del equipo, únicamente si es absolutamente necesario y nadie ha asistido a su retroalimentación. Esta regla se basa en patrones subjetivos, donde generalmente el nuevo empleado considera que se ha tomado demasiado tiempo en la investigación cuando empieza a sentirse “ansioso” ante la falta de resultados. Así mismo, la empresa considera el tiempo estimado que debe tomar el nuevo personal en sus investigaciones y para su labor si el tiempo se extiende 15 minutos más allá de lo que él pretendía.

Una vez terminado el primer ciclo de desarrollo de software en el que el empleado ha sido contratado, se considera si se la persona se encuentra preparada para aplicar los conocimientos adquiridos en el primer ciclo de desarrollo, en donde se les entrega tareas que cumplir y un tiempo sugerido en el que deben acabar cada una de ellas, sin embargo, el tiempo se estima en base a

experiencias pasadas de forma subjetiva, ya que no cuentan con un registro útil de las tareas realizadas en el pasado.

El nuevo personal trabaja cercanamente con el Gerente General para minimizar los errores iniciales de adaptación a un nuevo empleo, sin embargo, al inicio, sin importar la posición para la que han sido contratados, sus tareas se basan en cumplir con las tareas de un asistente personal y/o recepcionista, provocando malestar inicial con los nuevos miembros del equipo.

Entre las fortalezas del proceso se considera que la empresa pone énfasis especial en contratar personas que busquen aportar a un proyecto que ayude a la comunidad, en lugar de contratar personas que buscan desarrollar software bajo órdenes y sin criterio. Tal como lo indica, el equipo necesita personas comprometidas que disfruten trabajar en los proyectos en lugar de buscar cumplir con un horario de trabajo.

Sin embargo, como debilidades, la empresa no cuenta con métodos establecidos de contratación del personal, en donde la mayoría de pruebas y cualidades de selección son de carácter subjetivo. La mayoría de las pruebas que realiza son de temas de conceptos básicos de desarrollo de software y escenarios en los que los postulantes deben proponer soluciones.

En el periodo de aprendizaje, los nuevos empleados no conocen mucho sobre los productos que la empresa ofrece, así como no pueden abrir libremente el repositorio de códigos para entender mejor el funcionamiento de dichos productos. Así mismo, de forma incongruente, durante todo el proceso de Contratación e Inducción, la empresa busca motivar a los nuevos empleados promoviendo el concepto de realizar algo que ayude y facilite la vida de muchas

personas, un concepto que se puede evidenciar con fuerza en todos los empleados de HSB SOFTECUADOR C.A.

En el proceso de Contratación e Inducción de los nuevos empleados, es de gran importancia dar a conocer todo el organigrama funcional de la empresa a nivel internacional, para que los nuevos miembros del equipo tengan conocimiento de con quienes trabajan, cómo aportan a la empresa y lo que hacen todos y cada uno de sus componentes. Este proceso sin embargo es bastante subjetivo, basado en los criterios del Gerente General.

3.3.1.2. Proceso de Desarrollo de Software

Este proceso inicia cuando la Alta Dirección Internacional detecta las necesidades de nuevas aplicaciones en los programas de diseño de casas de madera en Europa y envían así un plan de desarrollo para los próximos 6 o 12 meses generalmente.

Uno de los conceptos que la empresa ha logrado implementar de la metodología SCRUM enfocada al desarrollo de software es el concepto del cuadro Kanban, aunque lo aplican con el término de SCRUM internacional, observado en la Figura 8, que permite la transparencia del proceso y pretende agilizar la comunicación interna con las empresas de HSB a nivel internacional.

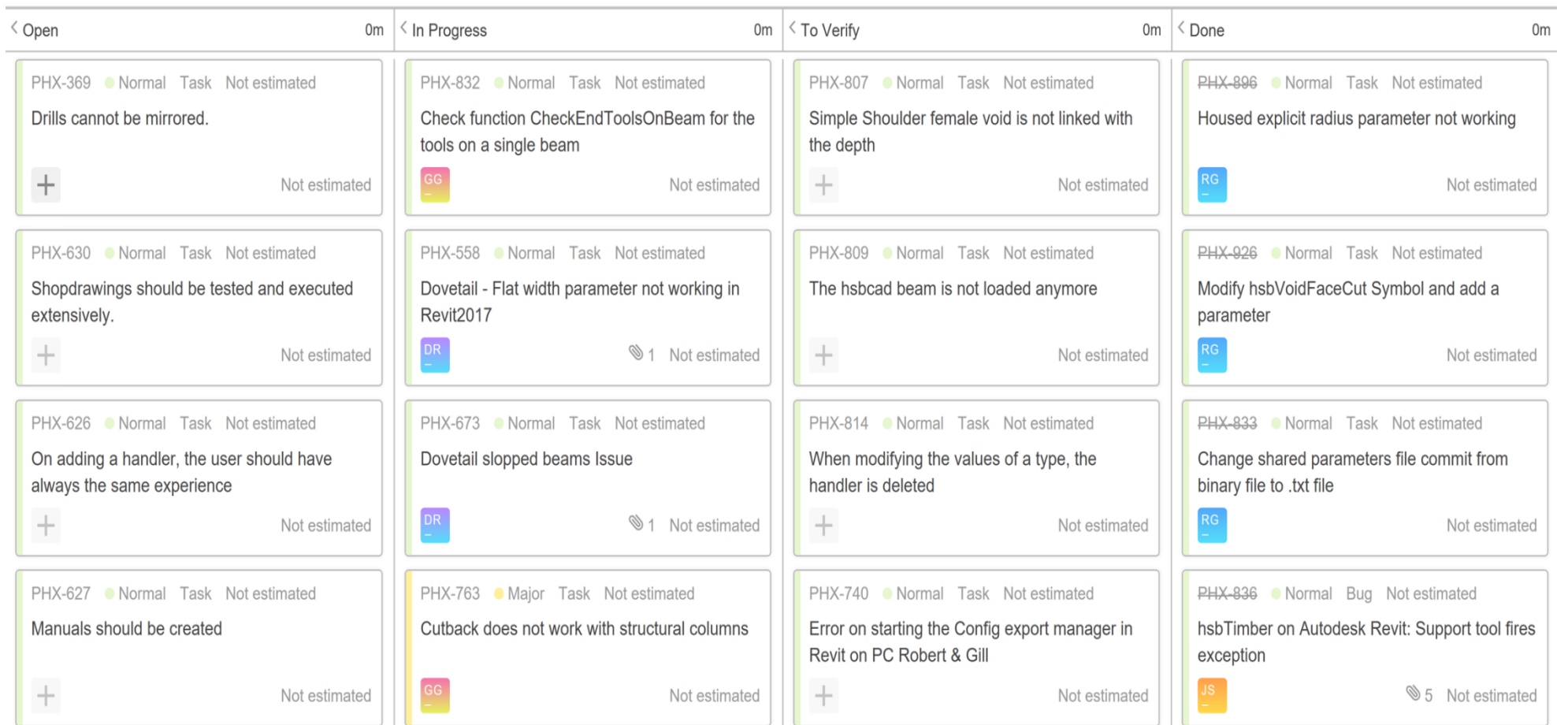


Figura 8.- Kanban de HSB SOFTECUADOR C.A.

El cuadro Kanban, utilizado internamente como un tablero de control de tareas, se actualiza cada vez que la Alta Dirección Internacional envía la planificación de desarrollo a HSB SOFTECUADOR C.A.

Esta herramienta consta de cuatro columnas: la primera hace referencia a las tareas nuevas o correcciones que se deben realizar, la segunda detalla las actividades que se encuentran en proceso y la persona que las lleva a cabo, la tercera columna especifica el proceso de verificación por el que cada actividad ha atravesado, para finalmente en la cuarta columna detalla las actividades completadas y verificadas en su totalidad.

Los consumidores finales detallan a la Alta Dirección Internacional el producto ideal que desean y los directivos transmiten los requisitos a los desarrolladores en Ecuador.

En el proceso de Desarrollo de Software, todas las tareas nuevas o las correcciones parten del uso de códigos anteriores del repositorio digital de la empresa, donde reutilizan todos los posibles y los adaptan, ya que crear nuevos códigos es una tarea más compleja y dominante.

Para asignar cada tarea a los miembros del equipo se considera en primer lugar si la tarea es una mejora o corrección de un código ya existente, en ese caso, la persona encargada de realizar la tarea será quién trabajó en el código anterior. En caso de que la tarea sea nueva, se designan a los desarrolladores según su campo de experticia o según quien posea la menor cantidad de tareas pendientes.

El proceso consta de 3 etapas de prueba para verificar diferentes aspectos del producto. La primera etapa de prueba es la de funcionalidad, donde todo el

equipo revisa y verifica que el producto que se encuentra en desarrollo cumple con las necesidades del cliente. La segunda etapa de prueba es llamada la etapa Beta, donde se verifica la simplicidad del producto, analizando la forma de facilitar el uso del producto para los usuarios. La última etapa de prueba es llamada la etapa de prueba Release o de pre distribución, donde se entrega el producto a los altos directivos de la empresa en Europa, así como a clientes estratégicos que prueban el producto y detectan fallas que el equipo no ha podido destacar y solucionar.

Posteriormente a las etapas de prueba con las que cuenta el producto, se confirma la finalización del mismo cuando no se detectan más errores o se detallan sugerencias. Sin embargo, esto no ocurre en la realidad, la decisión de dar un proyecto o producto como finalizado recae en el Gerente General y la Alta Dirección Internacional de la empresa de forma subjetiva. Es necesario comprender que en el desarrollo de software se busca crear productos nuevos, por lo que es realmente complicado saber cuándo el producto ha sido finalizado, esto reflejado en el hecho de que los consumidores finales al solicitar el producto tienen una vaga y ligera idea de lo que necesitan, mas no poseen un concepto tangible de lo que desean y necesitan en realidad.

Una vez finalizado el producto, la empresa solicita a la alta dirección internacional que adquieran las licencias de los sitios web que servirán de herramienta para crear los instaladores finales del producto y así poder iniciar el proceso de Distribución.

El proceso de Desarrollo de Software es bastante generalizado y no cuenta con procesos específicos en los enfocarse, contando con varios aspectos que requieren mejoras detallados a continuación.

La empresa HSB SOFTECUADR C.A. busca trabajar bajo la metodología SCRUM, aunque actualmente no cumplen todavía las condiciones específicas para considerar que desarrollan software bajo este modelo.

La comunicación interna a nivel internacional ha visto mejoras al utilizar esta herramienta de control de las tareas de forma significativa. Sin embargo, esto no se ha reflejado a su vez en la comunicación entre la empresa HSB SOFTECUADOR C.A. y los consumidores finales, debido a que los consumidores finales residen en Europa principalmente y es complicado para los desarrolladores de Ecuador contactarse con ellos ante la diferencia de horario y la diferencia de realidad socio-cultural. Por estos motivos, los desarrolladores en Ecuador necesitan que la Alta Dirección en Europa se convierta en el canal que transmita las necesidades de los clientes.

El proceso de comunicación se entorpece considerablemente al comprender que toda la información llega al receptor con un día aproximado de retraso debido a las zonas horarias separadas de las empresas en Europa y Ecuador, ya sea en caso de comunicar nuevas tareas o correcciones al equipo de desarrollo de software o se trate de realizar consultas y confirmaciones hacia la Alta Dirección.

La empresa adapta la metodología SCRUM según mejor le convenga, tal es el caso de las reuniones diarias de 15 minutos que la metodología establece, en donde el equipo se reúne casi todos los días por periodos mayores a los 15 minutos, donde el lapso de menor duración fue de 30 minutos y el de mayor duración fue de 2 horas.

Al iniciar el trabajo con la metodología SCRUM, se generó mayor tensión laboral entre los miembros del equipo, sin embargo, recalcan que se han visto

beneficiados en siempre trabajar en los más importante cada vez, ya que antes de las reuniones diarias, el equipo trabajaba en lo que cada uno personalmente consideraba lo más importante, pero ahora, en cada reunión, se llega al conceso de lo que realmente es lo que más importa, avanzando a mejores resultados, a costa de mayores tiempos de desarrollo cada vez.

Así mismo, la empresa considera que mientras la metodología ágil les permite avanzar a mayor velocidad individual (no total) en sus proyectos, también consideran que se ha perdido gradualmente la capacidad interna para investigar nuevos métodos y capacitarse en códigos nuevos, se establece que el equipo debe lograr esta capacitación fuera de su horario de trabajo.

En las etapas de prueba se han encontrado fallos importantes para el desarrollo del proyecto, en donde los clientes que prueban el producto no tienen el interés ni el tiempo de probar todos los aspectos del mismo, por lo que no prueban el producto en casos extremos, únicamente en casos simples, donde varios errores importantes no fueron detectados a tiempo, provocando que se extendiera la fecha de culminación del proyecto.

Finalmente, se ha detectado que los dueños de la empresa, residentes en Europa, son quienes toman las decisiones finales sin considerar la opinión del equipo de trabajo, indicando los aspectos en los que el equipo debe enfocarse en cada sprint, sin embargo, mientras esto genera malestar inicial, finalmente promueve buenos resultados ya que los dueños generalmente toman buenas decisiones.

3.3.1.3. Proceso de Distribución

El proceso de Distribución de Productos parte inmediatamente después de la finalización del proceso de Desarrollo de Software, en donde el Directorio

Internacional adquiere y entrega las licencias de los sitios web que se utilizarán para crear los instaladores de los productos de HSB SOFTECUADOR C.A.

Tras crear los instaladores de los productos, la empresa puede compartirlos en su centro de descargas de su página web donde cada cliente solicita licencias de uso y descarga de los productos. La persona encargada de otorgar y actualizar los permisos de cada cliente es el Gerente General, que informa al cliente que ha sido habilitado para manejar los productos de la empresa.

Así mismo, los consumidores finales pueden descargar los productos gratuitamente a través de los portales de Autocad, pero en formato de prueba, que tiene un periodo de 1 mes y limitado acceso a las herramientas del programa.

Entre los aspectos negativos que se cuentan, la empresa no posee sitios web predeterminados para poder crear los instaladores de los productos de cada proyecto, utilizan los sitios seleccionados de forma subjetiva por el dueño de la empresa global.

Así mismo es importante detallar a favor del proceso que las versiones de prueba del programa son lo suficientemente atractivas para generar nuevos clientes de los productos de cada proyecto.

3.3.2. Análisis de Cumplimiento

El análisis de cumplimiento de la metodología ágil de desarrollo de software se elaborará a través una lista de verificación elaborada a partir del manifiesto ágil y las diez áreas de conocimiento de la administración de proyectos.

Es así necesario detallar la relación de las etapas de cada iteración del proceso de desarrollo de software con las áreas de conocimiento de la administración de proyectos como se puede observar en la Figura 9 para poder contar con un análisis más extensivo de cómo se controla el desarrollo de software a través de un método ágil.

Las etapas que cuenta cada iteración del desarrollo de software son: inicio, planificación, ejecución, monitoreo y control y cierre. (Larrañaga, 2015) Así mismo, las diez áreas de administración de un proyecto son: (Project Managment Institute, 2017)

- Gestión de los interesados, entendiendo quienes se benefician o afectan del proyecto, no solo los desarrolladores y usuarios.
- Gestión de las adquisiciones, manejando adecuadamente la adquisición de insumos y herramientas.
- Gestión de los riesgos, anticipándose y adaptándose a ellos.
- Gestión de las comunicaciones, para conectar a todo el equipo.
- Gestión de los recursos, asignándolos eficientemente.
- Gestión de la calidad, manteniendo un control constante y periódico.
- Gestión de los costos, con los desembolsos necesarios.
- Gestión del tiempo, optimizándolo para poder realizar más mejoras.
- Gestión del alcance, entendiendo lo que se desea lograr.
- Gestión de la integración, donde las cinco etapas mencionadas con anterioridad conforman un todo para brindar un producto final al usuario o cliente.

ETAPAS DEL PROCESO	ÁREAS DE CONOCIMIENTO DE LA ADMINISTRACIÓN DE PROYECTOS									
INICIO										
PLANIFICACIÓN	Gestión de los interesados	Gestión de las adquisiciones	Gestión de los riesgos	Gestión de las comunicaciones	Gestión de los recursos	Gestión de la calidad	Gestión de los costos	Gestión del tiempo	Gestión del alcance	Gestión de la integración
EJECUCIÓN										
MONITOREO Y CONTROL							Gestión de los costos	Gestión del tiempo	Gestión del alcance	
CIERRE										

Figura 9. - Gestión de proyectos en la iteración del proceso

Fuente: (Project Managment Institute, 2017)

En la Figura 9 se puede observar que existen 5 etapas fundamentales en cada iteración de software, sin embargo, a diferencia del método tradicional de cascada, no es necesario cumplir con el orden observado de estas etapas, al contrario, se estimula regresar o prescindir etapas por el beneficio de la mejora inmediata de un producto si este así lo requiere a través de soluciones veloces y precisas.

De esta forma y con ayuda al marco teórico presentado en el capítulo anterior, se puede desarrollar la lista de verificación que se puede encontrar el Anexo B, elaborada con entendimiento del funcionamiento de los métodos ágiles de desarrollo de software, la cual consiste en 42 preguntas cerradas clasificadas según la etapa del desarrollo al que hacen referencia en un proyecto. Al ser un cuestionario elaborado únicamente con preguntas cerradas, las respuestas aceptadas son “sí”, “no” o “no aplica” según sea el caso, permitiendo la ampliación de la respuesta en una explicación a determinarse en las observaciones.

A través de entrevistas con el equipo de trabajo, se pudo observar ciertos aspectos que contestaban con facilidad las preguntas de la lista de verificación detallada en la Anexo B. Las preguntas que no pudieron ser contestadas a través de la simple observación, se las realizaron al gerente general para conocer las respuestas necesarias para completar el análisis.

Para poder analizar el cumplimiento que tiene el equipo de desarrollo de software en base a las metodologías ágiles de diseño de software, se analizan las brechas por cada una de las etapas de desarrollo. Este análisis se realiza ponderando el cumplimiento Total con un valor de 2, el cumplimiento Parcial con un valor de 1 y el cumplimiento Nulo con un valor de 0.

3.3.2.1. Etapa de Inicio

En la etapa de Inicio se analizaron 8 aspectos de desarrollo ágil de software, brindando los siguientes resultados:

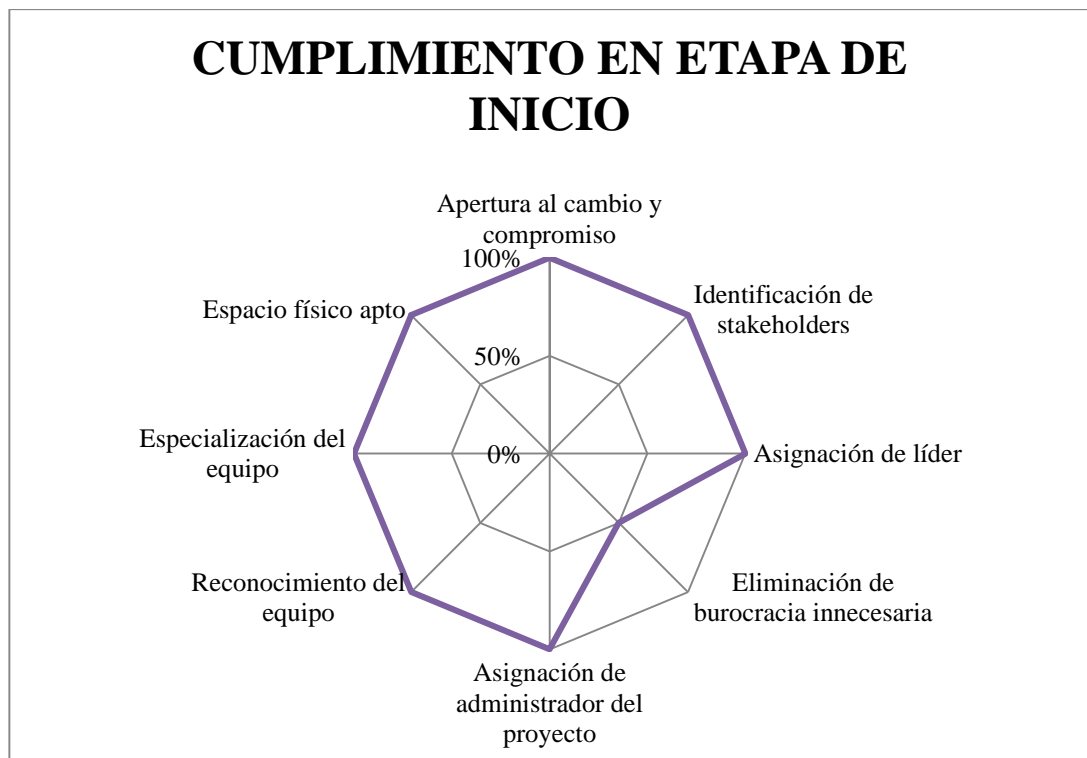


Figura 10.- Brechas de cumplimiento de la etapa de Inicio de desarrollo de software ágil

La máxima calificación posible de cumplimiento que se podía obtener es de 16 con un cumplimiento total. El puntaje obtenido por la empresa fue de 15/16, es decir un cumplimiento en la etapa de inicio de 93,75%, como se puede observar en el Anexo C del estudio. Las brechas de cumplimiento se pueden observar en la Figura 10. La empresa cuenta generalmente con una etapa de inicio ágil y eficiente, ya que se cumplen con la mayoría de los aspectos del Anexo C en su totalidad, donde el único aspecto que realmente genera inconvenientes a futuro del desarrollo se basa en que el líder del proyecto no identifica ni elimina completamente los impedimentos burocráticos que afectan al desarrollo e impiden su agilidad.

La empresa no considera necesario eliminar los impedimentos burocráticos desde en esta etapa, trasladando esta labor a etapas posteriores que dificultan la facilidad del desarrollo en cuanto se deben realizar pausas para poder solucionar dichos impedimentos.

3.3.2.2. Etapa de Planificación

En la etapa de Planificación se analizaron 17 aspectos que reflejan los siguientes resultados:



Figura 11.- Brechas de cumplimiento de la etapa de Planificación de desarrollo de software ágil

La máxima calificación posible de esta etapa de cumplimiento que se podía obtener es de 34 con un cumplimiento total. El puntaje obtenido por la empresa fue de 17/34, es decir un cumplimiento en la etapa de planificación con 50%.

Las brechas de cumplimiento se pueden observar en la Figura 11. Esta etapa es la más débil en cumplimiento de agilidad y eficiencia de software, debido a que no se elabora ninguno de los planes necesarios para dirigir un proyecto, como son los planes de: seguridad de nuevos componentes de software, administración de costos, administración de recursos, administración de la comunicación, de gestión de riesgos y de respuesta al riesgo; así como no se elabora análisis cuantitativo de riesgos. Esta falta de planificación de estos componentes significa problemas en el proceso de desarrollo de software en cuanto surgen mayores eventualidades e imprevistos que no se consideraron ni previnieron en el inicio por consecuencia, obligando a adaptar el proceso de forma innecesaria para solucionar cualquier problema.

3.3.2.3. Etapa de Ejecución

En la etapa de Ejecución se analizaron 10 preguntas, basadas en los siguientes aspectos:

CUMPLIMIENTO EN ETAPA DE EJECUCIÓN

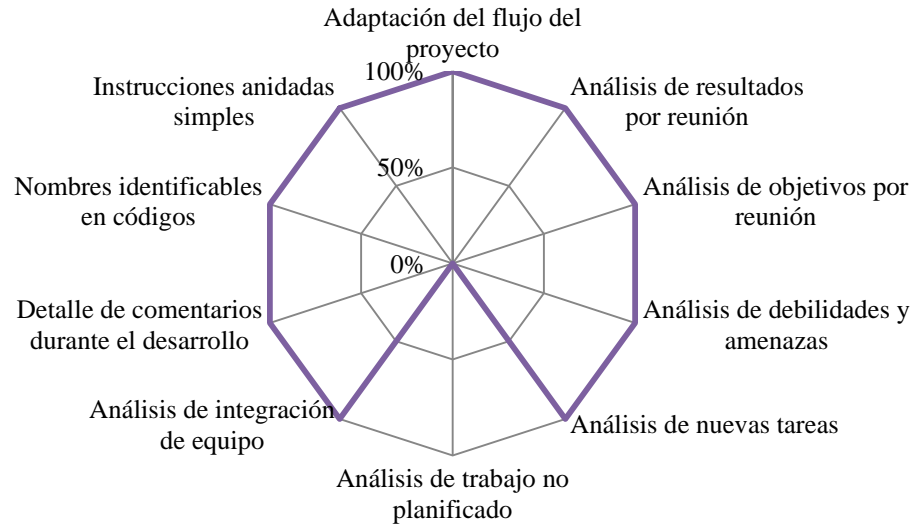


Figura 12.- Brechas de cumplimiento de la etapa de Ejecución de desarrollo de software ágil

En esta etapa, la máxima calificación posible de cumplimiento que se podía obtener es de 20 con un cumplimiento total. El puntaje obtenido por la empresa fue de 18/20, es decir un cumplimiento en la etapa de planificación con 90%.

Las brechas de cumplimiento se pueden observar en la Figura 12. Esta etapa, al igual que la etapa de inicio, cuenta con un considerablemente buen cumplimiento de agilismo y eficiencia en el proceso de desarrollo de software, donde únicamente no se realiza el análisis del trabajo que se realiza y no aporta al proyecto; el Gerente General indica que ocasionalmente este fenómeno ocurre, pero cuando lo hace, implica pérdida de tiempo que pudo ser utilizado en desarrollo del proyecto planificado.

3.3.2.4. Etapa de Mantenimiento y Control

En la etapa de Mantenimiento y Control se cuentan con 13 preguntas que reflejaron los siguientes resultados:



Figura 13.- Brechas de cumplimiento de la etapa de Mantenimiento y Control de desarrollo de software ágil

La máxima calificación posible de la etapa de Mantenimiento y Control de cumplimiento que se podía obtener es de 26 con un cumplimiento total. El puntaje obtenido por la empresa fue de 19/26, es decir un cumplimiento en la etapa de planificación con 73,1%.

Esta es la etapa con el segundo más bajo cumplimiento de una metodología ágil y eficiente de desarrollo de software, como se puede ver en la Figura 13, donde los mayores problemas se dan al no contar con criterios objetivos para analizar el cumplimiento de los resultados, así como realmente no se cuenta con registros apropiados de los resultados de cada proyecto, donde no se miden cuantitativamente ni cualitativamente, es decir que los criterios de culminación de un proyecto se dan subjetivamente por parte del Gerente General.

3.3.2.5. Etapa de Cierre

En la etapa de Cierre se cuentan con 2 preguntas con los siguientes resultados:

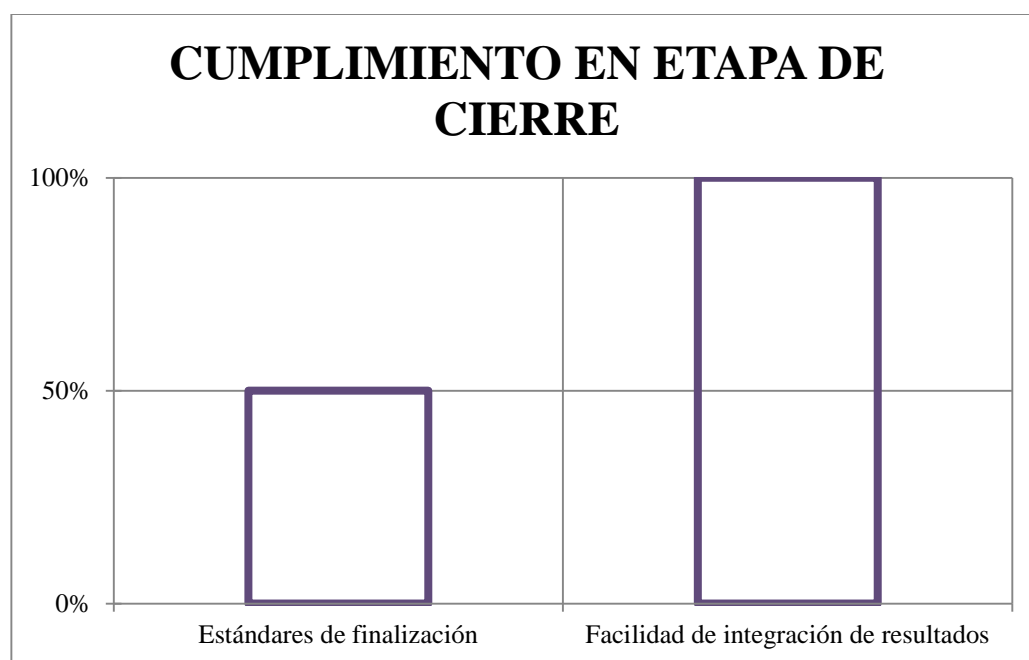


Figura 14.- Brechas de cumplimiento de la etapa de Cierre de desarrollo de software ágil

Esta etapa posee una máxima calificación posible de cumplimiento que se podía obtener es de 4 con un cumplimiento total. El puntaje obtenido por la empresa fue de 3/4, es decir un cumplimiento en la etapa de planificación con 75%.

Las brechas de cumplimiento se pueden observar en la Figura 14. Como se mencionaba en la etapa de Mantenimiento y Control, al no tener criterios para considerar el cumplimiento de los resultados, en la etapa de Cierre del proyecto se genera moderada dificultad en considerar un proyecto finalizado y listo para ser distribuido. Es una dificultad moderada debido a que el equipo de desarrollo busca acercarse lo más posible a las necesidades del cliente, pero considerando que desarrollar software es crear algo que no existe todavía, los límites para continuar o finalizar un proyecto se tornan muy ambiguos y difíciles de discernir.

3.3.2.6. Análisis general

Finalmente, analizando la Figura 15, se puede considerar un análisis por etapa del cumplimiento que se identificó al contestar la lista de verificación del Anexo B.

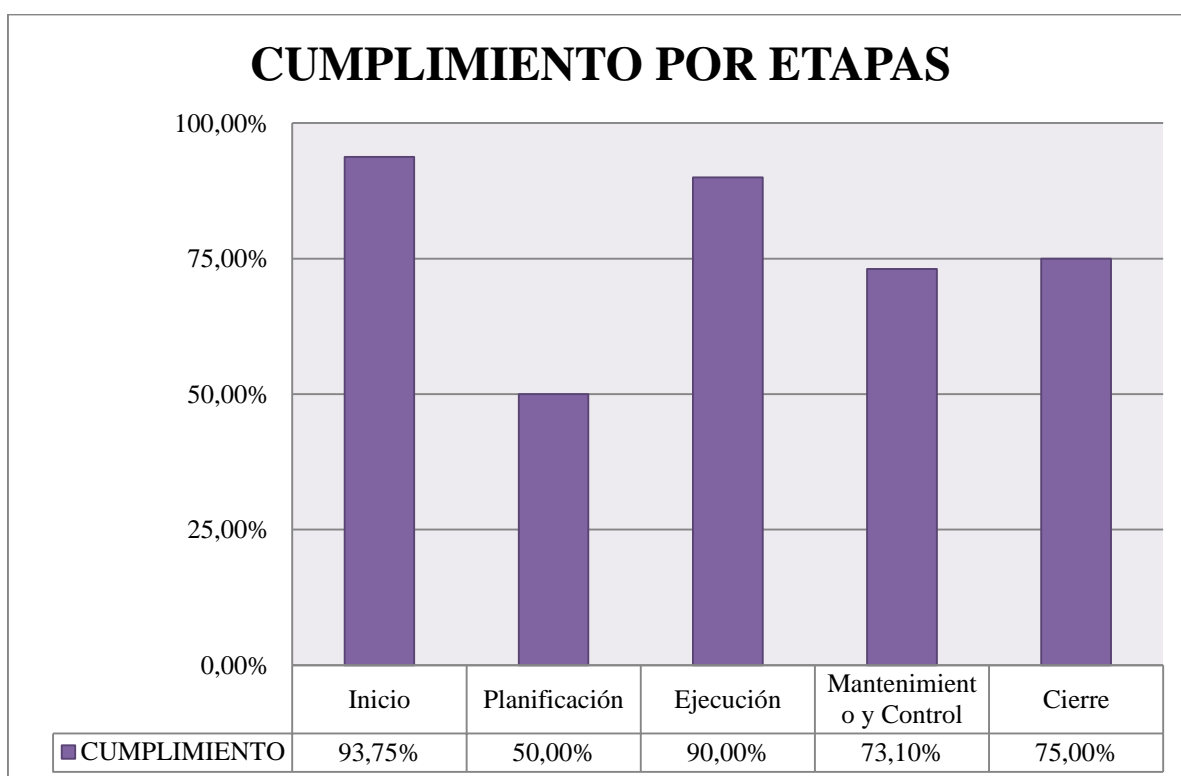


Figura 15.- Brechas de cumplimiento por etapas de desarrollo de software ágil

Se puede considerar que la etapa con menor eficiencia según el desarrollo de software ágil es la etapa de Planificación con un cumplimiento del 50% y la etapa de Mantenimiento y Control con un cumplimiento de 73,1%.

Cabe destacar que la etapa de Inicio y de Ejecución demuestran un alto nivel de eficiencia en el desarrollo de software ágil, en donde se entiende la importancia de un equipo bien preparado y equipado con los recursos necesarios para cumplir un proyecto exitosamente, así como también el trabajo técnico es llevado a cabo con el mejor detalle posible.

3.3.3. Rendimiento de proyectos pasados

Los principales proyectos de la empresa HSB SOFTECUADOR C.A. son: Cross Laminated Timber (CLT), HSB Timber for Autodesk Revit Versión II y su Versión I, detallados en resumen a continuación en la Tabla 3.

Tabla 3.- Principales proyectos del año 2016 y 2017

PROYECTO	DESCRIPCIÓN	ALCANCE ESTIMADO	TIEMPO PLANIFICADO	PRESUPUESTO ASIGNADO
Cross Laminated Timber CLT (2017)	Facilita la unión de planchas de madera en el diseño de arquitectura	Proyecto experimental distribuido a 5 clientes	6 meses	\$110 000
Hsb Timber For Autodesk Revit Versión II (2017)	Facilita la conexión entre vigas y columnas de madera	2100 clientes aproximadamente	6 meses	\$105 000

PROYECTO	DESCRIPCIÓN	ALCANCE ESTIMADO	TIEMPO PLANIFICADO	PRESUPUESTO ASIGNADO
Hsb Timber For Autodesk Revit Versión I (2016)	Facilita la conexión entre vigas y columnas de madera	1800 clientes aproximadamente	12 meses	\$200 000

Considerando que la empresa busca trabajar bajo la metodología ágil de SCRUM, no la llevan a cabo a totalidad como se pudo comprobar en el punto anterior, uno de los factores más importantes se debe a que la empresa no posee registros que permitan realizar las mediciones necesarias para conocer el rendimiento de proyectos pasados o presentes, ya sea de costos incurridos para realizar cada proyecto, el tiempo promedio en el que se ha llevado a cabo cada desarrollo, número de descargas de cada producto o el número de defectos detectados y correcciones realizadas de ninguno de los productos mencionados con anterioridad.

La falta de registro de algún tipo de métrica para constar el desempeño del producto se debe a que el Gerente General no las considera necesario, ya que su equipo mide su rendimiento en base al logro del producto final en el tiempo solicitado por la Alta Dirección a nivel internacional, donde indican 6 o 12 meses de tiempo para el desarrollo. Es por esto que las métricas especificadas a continuación son rendimientos estimados en base a cálculos aproximados de la realidad detallados en cada caso.

3.3.3.1. Métricas de productividad y efectividad de la entrega

De las métricas de esta categoría detalladas en el capítulo anterior, para medir la efectividad en tiempo de los proyectos de HSC SOFTECUADOR C.A. se analizará el Lead Time de cada uno de ellos a continuación.

Tabla 4.- Tiempo de finalización de producto final

PROYECTO	INICIO	FINALIZACIÓN	LEAD TIME	CUMPLIMIENTO SOBRE PLAN
Cross Laminated Timber CLT (2017)	Abril 2017	Septiembre 2017	6 meses	100%
Hsb Timber For Autodesk Revit Versión II (2017)	Octubre 2017	Abril 2018	6 meses	100%
Hsb Timber For Autodesk Revit Versión I (2016)	Julio 2016	Abril 2017	10 meses	120%

Se puede observar en la Tabla 4 que en la primera versión de Hsb Timber for Autodesk Revit, es el único que culminó el tiempo de entrega desde la petición con un lapso menor al planificado en la Tabla 3, los demás proyectos cumplieron el lapso establecido en la planificación inicial.

3.3.3.2. Métricas de resultados del proyecto

La métrica seleccionada para medir los resultados del proyecto es el alcance, medida a través de los usuarios que utilizan cada uno de los productos resultantes de los proyectos analizados.

Tabla 5.- Alcance de productos a la fecha

PROYECTO	DESCARGAS ESTIMADAS	DESCARGAS REALES	CUMPLIMIENTO DEL PLAN
Cross Laminated Timber CLT (2017)	5	4	80%
Hsb Timber For Autodesk Revit Versión II (2017)	2100	-	-
Hsb Timber For Autodesk Revit Versión I (2016)	1800	2000	111,11%

Como se puede observar en la Tabla 5, el único proyecto que no puede ser medido por esta métrica es la versión actualizada de Hsb Timber, ya que acaba de ser finalizada. En cuanto al proyecto Cross Laminated Timber, que con 7 meses de vida, no completa las descargas esperadas en la planificación, mientras que la primera versión de Hsb Timber, que posee 12 meses de vida, ha superado la estimación con 200 descargas adicionales.

3.3.3.3. Métricas de situación financiera

Para analizar los proyectos en medida financiera, se contará la desviación financiera respecto a la planificación inicial. Esta medida ha sido seleccionada debido a que la falta de registro de información detallada impide conocer con facilidad otras medidas de rendimiento y permite únicamente analizar esta medida de presupuesto aproximada a la realidad de la empresa.

La información presentada a continuación ha sido calculada a través de valores aproximados de los registros disponibles de la empresa en cuanto a los costos anuales que posee, para así generar un valor proporcional al tiempo de desarrollo del producto.

Tabla 6.- Desviación financiera respecto a la planificación inicial

PROYECTO	PRESUPUESTO	COSTO	DIFERENCIA
Cross Laminated Timber CLT (2017)	\$110 000	\$113 000	+2,72%
Hsb Timber For Autodesk Revit Versión II (2017)	\$105 000	\$110 000	+4,76%
Hsb Timber For Autodesk Revit Versión I (2016)	\$200 000	\$205 000	+2,5%

Como se puede observar en la Tabla 6, todos los proyectos tuvieron excedentes en el presupuesto menores al 5%, sin embargo al considerar que tanto los valores del presupuesto y del costo son valores estimados, es difícil conocer con seguridad la brecha financiera exacta de cada proyecto.

Asumiendo la exactitud de los valores de la Tabla 6, el proyecto con mayor desviación financiera es la versión actualizada de Hsb Timber, debido a que se esperaba tener un costo menor a su primera versión casi al 50% menor, considerando que se trata únicamente de la mejora de un proyecto mas no el inicio de uno nuevo de diferentes características.

La empresa considera que la causa más probable para todas estas diferencias se debe a los costos adicionales que se generaron al encontrar obstáculos no

planificados. Esta teoría se podría alinear al análisis de cumplimiento elaborado en el punto anterior, en donde en la etapa de inicio no se elimina la burocracia innecesaria y en la etapa de planificación no se realiza una adecuada planificación.

3.3.3.4. Métricas de calidad

Para analizar la calidad de los proyectos, se detallará el número de observaciones brindadas por todos los usuarios de prueba (entre errores y oportunidades de mejora), respecto a las mejoras realizadas finalmente. Al igual que las anteriores métricas, los resultados mostrados a continuación fueron adquiridos por estimaciones debido a la falta de registros.

Tabla 7.- Mejoras realizadas en base a las observaciones durante pruebas

PROYECTO	OBSERVACIONES	MEJORAS REALIZADAS	DIFERENCIA
Cross Laminated Timber CLT (2017)	1500	1000	-33,33%
Hsb Timber For Autodesk Revit Versión II (2017)	800	600	-25%
Hsb Timber For Autodesk Revit Versión I (2016)	1400	950	-31,14%

Es importante detallar que la empresa busca que las mejoras que finalmente se realizan sean menores a las observaciones realizadas por los usuarios de prueba, debido a que no todas las observaciones son aspectos que competen a la empresa como desarrollador, pero competen al usuario en el manejo de las herramientas.

Como se puede observar en la Tabla 7, todos los proyectos cumplen la política interna de la empresa de realizar menores mejoras a las observaciones detalladas.

Sin embargo, la información brindada no es suficiente para determinar completamente medidas de calidad de la empresa, debido a que no se conoce el número exacto de observaciones que corresponden a competencia de la empresa y cuales corresponden a competencia del usuario.

3.4. Oportunidades de mejora

Considerando todos los aspectos detallados en todo el capítulo, existen varias oportunidades de mejora con las que la empresa HSB SOFTECUADOR C.A. cuenta, resumidas en los siguientes aspectos:

- La falta de registros impiden el análisis de rendimiento de los proyectos, dificultando el análisis de eficiencia del proceso de desarrollo de software ágil.
- El Gerente General procura asumir roles de líder, administrador, supervisor y desarrollador de software al mismo tiempo.
- El Gerente General es un administrador del equipo de desarrollo efectivo, mas no es el líder que requiere generar proyectos ágiles.
- El equipo de desarrollo de software comprende bien sus funciones, sin embargo no pueden explicar fácilmente los procesos con los que cuenta la empresa, así como no logra identificar los más importantes fácilmente.

- La planificación de la empresa es tomada a la ligera y muy poco considerada, en cuanto la empresa no cuenta con una persona especializada en el ámbito.
- Al inicio del proceso de desarrollo de software, no se buscan eliminar las barreras e impedimentos burocráticos con facilidad, logrando que se generen barreras a lo largo de todo el proceso.
- No se analiza el trabajo que realiza el equipo de trabajo que realmente no aporta a la culminación de los proyectos.
- No se cuentan con criterios objetivos para poder determinar que un proyecto cumple en su mayoría las necesidades y expectativas de los clientes.
- No se cuentan con sitios web establecidos para generar los instaladores de los productos de la empresa.
- Las pruebas de los productos de cada proyecto no son tomadas con seriedad por parte de los clientes estratégicos que las realizan.
- No se determina el origen de todas las observaciones, así como tampoco se determina si se generan antes o después del lanzamiento de un producto.

4. CAPÍTULO 4.- PROPUESTA DE IMPLEMENTACIÓN

El presente capítulo detalla los cambios necesarios para lograr la mejora de todos los procesos que conforman el desarrollo de software de la empresa HSB SOFTECUADOR C.A., en donde se podrá visualizar la metodología ágil de eficiencia para lograr dichos resultados.

Así como se puede estudiar en el Anexo D, la propuesta de implementación de la mejora de los procesos de software de la empresa está representada por procedimientos exactos determinados objetivamente por sus objetivos, alcance, responsables y políticas necesarias para cumplir de manera más objetiva los procesos de desarrollo y así evitar el trabajo acumulado al final del desarrollo.

4.1. Propuesta de Mapa de Procesos

Para iniciar la propuesta de mejora de los procesos de desarrollo de software de la empresa HSB SOFTECUADOR C.A., es necesario establecer un mapa de procesos adecuado que permita a todos los procesos de la empresa a trabajar de forma integral y eficiente.

Los primordiales cambios realizados en comparación a la lista de procesos detallada en el capítulo anterior, se debe a que ahora, los procesos de desarrollo de software se han dividido en 6, en lugar del proceso general de Desarrollo de Software que existía, en donde se asimilaron las etapas más importantes de un proceso de software ágil, que puedan ejemplificar de manera simple los propósitos de cada uno de ellos.

A su vez, todos los procesos de la propuesta de mapeo han sido codificados para facilidad de identificación en cada uno de los procedimientos que se pueden observar en el Anexo D.

Los 6 procesos principales de desarrollo de software han sido visualmente situados en cada una de las etapas de la gestión de proyectos de software mencionadas en el capítulo anterior, en donde es necesario recalcar que al ser un proceso ágil, cada una de las etapas se puede dar en cualquier momento de la vida del proyecto, no necesariamente en cadena como se puede observar en la Figura 16.

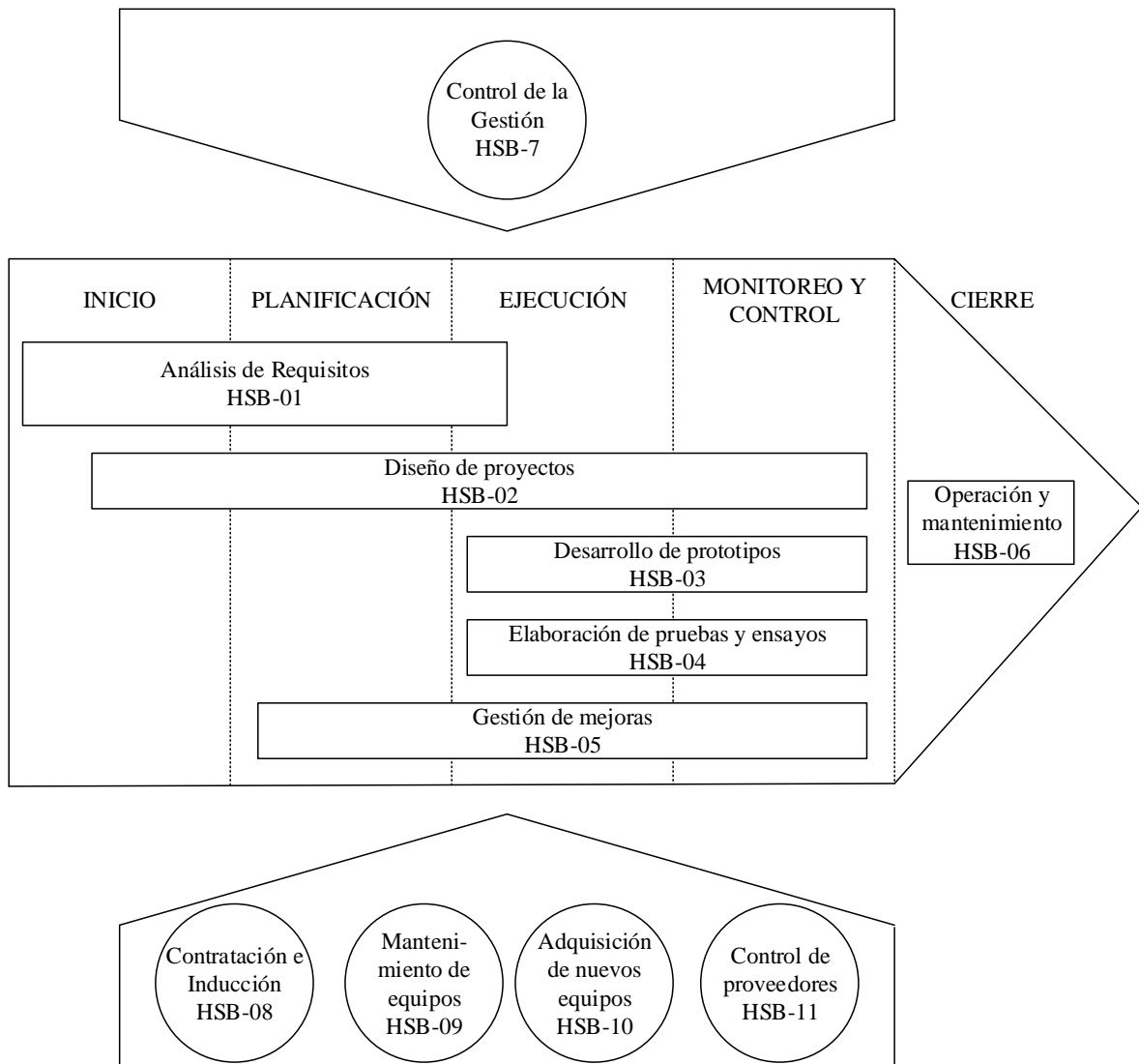


Figura 16.- Propuesta de mapa de procesos

Como se puede observar en la Figura 16, es necesario establecer que los procesos no necesariamente siguen el orden visual en el que se encuentran, así como cada etapa no cae en secuencia de cascada. Cada etapa o proceso pueden darse a cualquier momento de la vida del proyecto o a su vez, omitirse si el caso lo amerita.

En la Figura 17 se puede observar que los procesos no siguen una secuencia directa, sin embargo, poseen una secuencia lógica, que permite comprender que todos los procesos se encuentran en un alto nivel de cooperación que culmina en el cierre.

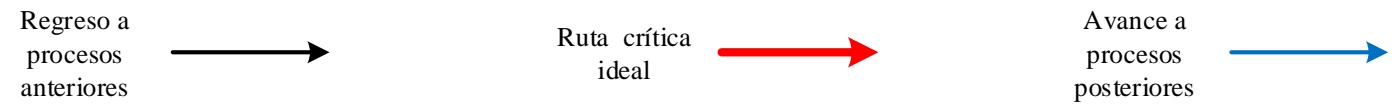
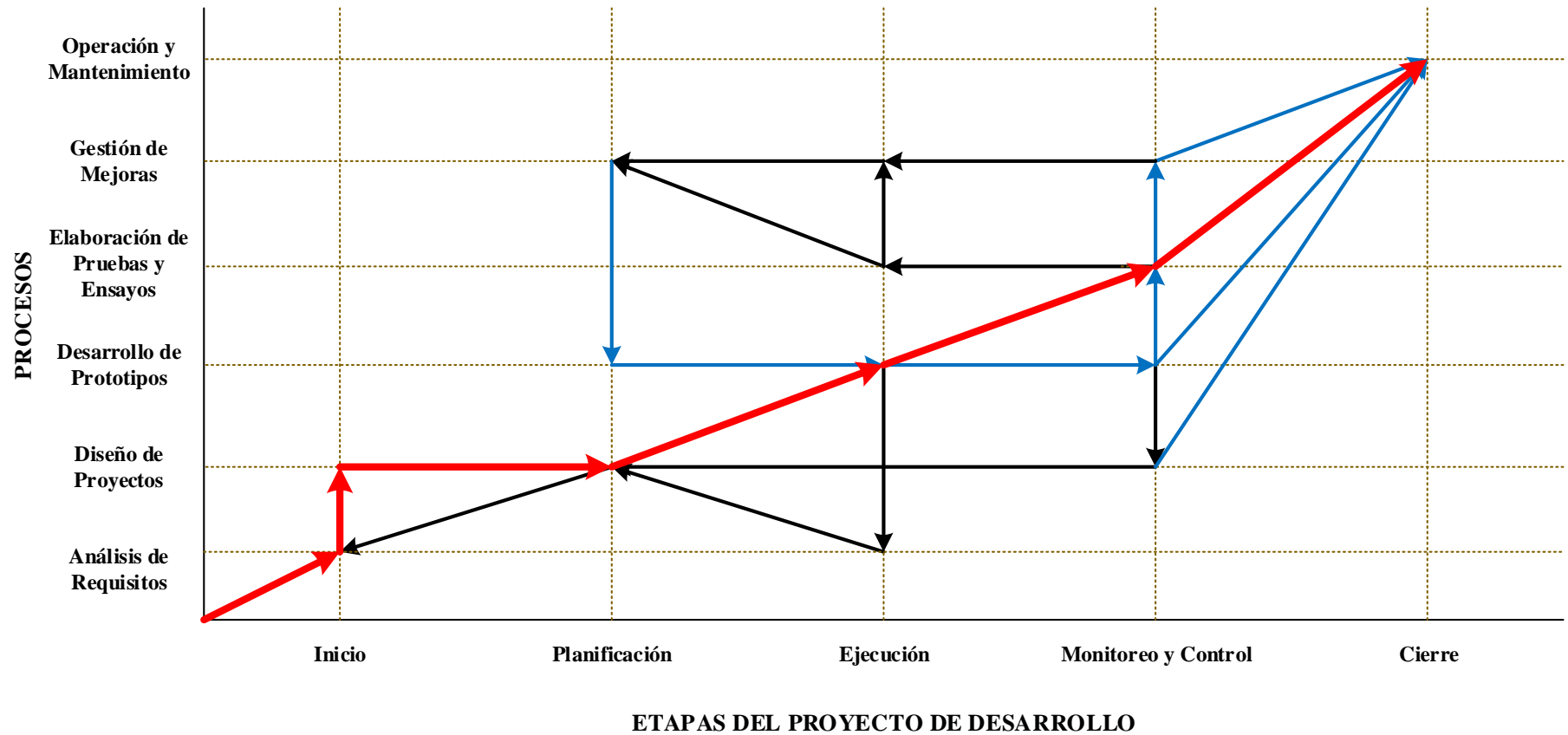


Figura 17.- Secuencia de los procesos propuestos

4.2. Propuesta de Diseño de Procedimientos

Los procedimientos que serán diseñados a continuación solo serán aquellos misionales que se llevan a cabo en cada proyecto nuevo realizado, a diferencia de los procesos de gestión y de soporte que se realizan eventualmente indiferentemente al desarrollo de cada proyecto de desarrollo de software.

Así mismo, es importante recordar que el proceso de Contratación e Inducción fue diagnosticado en el capítulo anterior como uno de los procesos más importantes de desarrollo de software, sin embargo, considerando lo mencionado con anterioridad, este proceso no corresponde al desarrollo de software propiamente dicho, de hecho, solo forma parte del apoyo para generar el personal competente que lleve a cabo el desarrollo de software necesitado por la empresa. Es por este motivo que el proceso de Contratación e Inducción no contará con un procedimiento para el caso de su estudio, pero contará con aclaraciones pertinentes para la mejora de su proceso al final del presente capítulo.

Todos los procedimientos elaborados de los procesos mencionados a continuación se encuentran en el Anexo D, que explican con mayor detalle las definiciones y políticas necesarias para aclarar las explicaciones que se realizarán a continuación, así como todos los elementos que no se pueden graficar en el diagrama de flujo de cada uno de los procesos.

4.2.1. Análisis de requisitos

El proceso de Análisis de requisitos será aquel del que parten los proyectos, en donde para cada uno de los proyectos de desarrollo de software que se tengan, el Gerente General será el responsable de velar por el cumplimiento del mismo. Es

el único proceso que inicia la generación de un proyecto, por lo que es sumamente importante en el posterior desarrollo.

Las mejoras propuestas para este proceso en específico parten del establecimiento de límites de cada proyecto, en donde se determinarán objetivos, alcance y partes interesadas de cada proyecto para poder tener los lineamientos de creación de productos necesarios. Es importante recalcar que el proceso actualmente no establece estos parámetros o requisitos con anticipación, pero los generan a medida que el proyecto se desarrolla.

En este proceso todo el equipo de desarrollo será de vital importancia para determinar el nicho del mercado que no ha sido cubierto por los productos existentes, así como clientes y potenciales clientes colaborarán con su punto de vista de las herramientas que los nuevos productos deben contener.

Entendiendo que no todas las necesidades pueden ser cubiertas por un solo producto a la vez, es importante limitarse al alcance del proyecto y determinar todas las características innecesarias o irrealizables en el próximo proyecto a iniciar; es por esto que de las necesidades detectadas por el equipo de desarrollo, así como aquellas detectadas por el cliente, serán filtradas en categorías.

Las categorías propuestas para filtrar las necesidades detectadas se darán por naturaleza de las herramientas, analizando si pueden elaborarse dentro del proyecto a iniciar con las capacidades de desarrollo que el equipo posea en dicho momento.

Así mismo, nuevas necesidades se pueden dar más adelante en el proceso de Desarrollo de Prototipos, en donde al mantener la premisa ágil, el listado de

requisitos podrá ser fácilmente adaptado y elaborado a medida que amerite la situación futura.

Este proceso se medirá en relación a todas las necesidades y requisitos detectados que puedan ser realizables, dando el porcentaje de la capacidad actual que posee la empresa de desarrollar el producto necesitado.

4.2.2. Diseño de Proyectos

El proceso de Diseño de Proyectos es desencadenado por el proceso de Análisis de Requisitos, pero también puede darse por consecuencia del proceso de Desarrollo de Prototipos (como parte de mejoras del diseño realizado) que se detallará más adelante. Este es un proceso que no se considera actualmente en la empresa, así como se pudo evidenciar en el Capítulo 3, en el análisis de cumplimiento de la etapa de Planificación.

Este proceso tiene la finalidad de generar la planificación y borradores de prototipos necesarios para poder realizar un proyecto ágil con el mayor control de cambio y riesgo posibles; esto se logra a través de la determinación propicia de todos los riesgos que tiene el proyecto de necesitar cambio alguno, además de controlar y eliminar todos los impedimentos burocráticos que representa crear cualquier proyecto como se pudo observar en el Capítulo 1.

Al ser un proceso ágil, los nuevos proyectos deben considerar los proyectos pasados, así como todos los percances y cambios que se realizaron, para tener una mayor perspectiva de los nuevos riesgos que pueden enfrentar los nuevos desarrollos y así adaptar la función de desarrollo en cada nuevo producto elaborado. Todos los entregables de gestión elaborados en este proceso son necesarios para mantener apertura y preparación ante cualquier cambio que se

pueda dar. Estos entregables en sí son sumamente flexibles ante los cambios a lo largo del proyecto.

Finalmente, los entregables técnicos, que son considerados los más importantes de todos los entregables en el ámbito de la empresa, son un borrador tentativo que seguirá la programación de cada producto y un cronograma del proyecto. Ambos de estos entregables son extremadamente sensibles a cualquier cambio que se pueda dar a lo largo de todo el proceso de Desarrollo de Software, por lo que los entregables de gestión deben elaborarse con el mayor detalle para minimizar todos estos cambios.

Este proceso será medido en relación a todos los cambios que se realizan al borrador inicial, debido a que, si bien es cierto que debe ser un documento flexible y abierto al cambio, debe poseer la menor cantidad de cambios que permitan un desarrollo ligero y libre de inconvenientes o barreras.

4.2.3. Desarrollo de Prototipos

Este proceso adopta varias premisas de la metodología SCRUM que la empresa actualmente busca utilizar en su totalidad aunque aún no usa todas sus herramientas y guías para ser un proceso ágil de esta metodología. Los conceptos principales que se han proponen añadir al proceso son: el Kanban Board y los sprints de desarrollo que poseen sencilla aplicación a los procesos actuales para permitir una transición simple y efectiva a la propuesta.

El Kanban Board, que la empresa actualmente llama SCRUM Internacional, es una herramienta visual y de orden que permite a todos los miembros del equipo conocer a tiempo real el estado de cada actividad de desarrollo y los responsables a cargo de desarrollarla. Es una herramienta que beneficia a la

comunicación interna para facilitar la integración de los códigos de todos los prototipos. Esta herramienta debe estar en constante control sobre todos los cambios que se efectúen en el proceso de Gestión de Mejoras que se detallará más adelante.

Los sprints de desarrollo inician una vez que todas las actividades se han distribuido satisfactoriamente, con una opinión casi total del tiempo que debe llevar el cumplimiento de las actividades más importantes. Esto ayuda a cada desarrollador a generar los códigos y herramientas solicitados de forma rápida y precisa, en donde todos los miembros del equipo participan en reuniones ágiles diarias de pocos minutos, para generar soluciones a todos los inconvenientes que se generen en las actividades realizadas por cualquier persona.

Este proceso de Desarrollo de Prototipos no puede llevarse a cabo sin un borrador tentativo del proyecto proveniente del proceso de Diseño de Proyectos, ya que este borrador permite distribuir las actividades anteriormente mencionadas. Durante este proceso, es importante asegurarse que no se generen nuevas necesidades pertinentes al alcance del proyecto, así como no sea necesario re asignar el rol de una o más personas en el equipo; también se debe controlar que no existan nuevos riesgos o a su vez, que algún riesgo planificado con anterioridad no se haya tornado efectivo, provocando la necesidad de cambios de los entregables de procesos anteriores.

Es importante también entender que a medida que se generen los prototipos, si se evidencia la posibilidad de mejorar el borrador del proyecto, por motivos no relacionados con riesgos adquiridos, al contrario, por oportunidades de generar códigos más sencillos o más efectivos, el borrador deberá mejorarse y adaptarse, sin que se vea afectado el trabajo ya realizado del equipo de trabajo.

Este proceso se mide en relación de los requisitos que se logran cumplir en cada prototipo, así como el cumplimiento del cronograma establecido. No se considera la medición del cumplimiento del borrador inicial debido a que este se encuentra sujeto a cambios tanto de fondo como de forma.

4.2.4. Elaboración de Pruebas y Ensayos

Cada prototipo generado o mejorado debe obligatoriamente atravesar el proceso de Elaboración de Pruebas y Ensayos, es en este proceso que el prototipo será rigurosamente analizado para confirmar que no existan fallas o faltas en los productos resultantes, en donde el fenómeno más grave será la existencia de fallas.

La propuesta de mejora de este proceso añade una fase adicional de prueba, en la cual los clientes Beta Testers formarán una mejora parte del control de los prototipos y la detección de irregularidades.

A su vez, también se propone iniciar el proceso con un la elaboración de un plan de pruebas que permita a cada usuario, interno o externo, realizar las pruebas que permitan detectar la mayoría o todas las fallas y faltas existentes en los productos resultantes del Desarrollo de Prototipos.

La primera fase de pruebas se basa en la funcionalidad, en la que todos los miembros de equipo verificarán que todas las herramientas del producto cumplan con el propósito de su creación. En esta etapa, cada persona comprobará las herramientas de los otros integrantes del equipo, mas no las herramientas generadas por sí mismos.

La segunda fase de pruebas se basa en la simplicidad, en donde el equipo volverá a utilizar las herramientas mejoradas y corregidas, pero esta vez para comprobar que el uso de cada una sea el más simple y entendible posible. Al igual que la primera etapa de pruebas, cada persona probará las herramientas que no hayan sido generadas por sí mismos.

La tercera fase de pruebas será realizada por usuarios externos a la empresa, es decir los Beta Testers, que serán clientes interesados en probar los nuevos productos para añadir o corregir herramientas a su beneficio. Los clientes recibirán el prototipo y listado de todas las herramientas que pueden y se sugieren que prueben. La lista de herramientas es importante para que los Beta Testers conozcan las pruebas que deben realizar y así minimizar el número de fallas y faltas no detectadas.

La última fase de pruebas se realizará nuevamente por el equipo de desarrollo de la empresa, esperando que con todas las correcciones y mejoras existentes a lo largo del proceso, el prototipo no contenga más fallas o faltas. Esa una prueba final de seguridad que permite al equipo de desarrollo distribuir posteriormente el producto con seguridad.

Este proceso será controlado por la medida de observaciones que se generen a lo largo de las pruebas realizadas, considerando que en cada etapa, los límites de tolerancia ante las observaciones se reducirán significativamente.

4.2.5. Gestión de Mejoras

Este proceso va estrechamente ligado al proceso de Elaboración de Pruebas y Ensayos y al proceso de Desarrollo de Prototipos, ya que en este proceso se gestionan las acciones correctivas y mejoras detalladas anteriormente. Se

propone categorizar cada observación que se genera a lo largo de todo el proceso de desarrollo de software, en donde en la actualidad, no se conoce el origen de cada observación, tornando inefectivo cualquier método de medición utilizado para mejorar la calidad.

La propuesta para este proceso parte de la categorización ya mencionada, en las que las observaciones se analizarán para determinar si se deben por defecto del producto o por mal manejo de las herramientas por defecto del usuario, de igual forma, se analizará si las observaciones recibidas no se deben por defecto del producto o del usuario, cuando se encuentran fuera de los límites del proyecto establecidos en el proceso de Análisis de Requisitos.

Todas las observaciones que inmediatamente no sean categorizadas como Observaciones por Defecto del Producto, provocarán que el Gerente General contacte a los Beta Testers para obtener información adicional que permita clasificar las restantes y así buscar soluciones.

Para poder manejar eficientemente el tiempo de desarrollo del software, se genera un plan de acción que permita realizar todas las acciones correctivas y mejoras necesarias al proceso, tras priorizar todas las no conformidades con la ayuda de un diagrama de Pareto que permita identificar las “pocas vitales” que necesitan ser atendidas efectivamente.

Finalmente, al conocer las no conformidades de mayor peso, se puede determinar las causas que necesitan ser corregidas con la ayuda de un diagrama de Ishikawa, permitiendo al equipo de desarrollo actualizar el Kanban Board del proceso de Desarrollo de Prototipos para corregir las fallas y mejorar las faltas desde la raíz, permitiendo que la empresa salvaguarde su tiempo en futuras correcciones o futuros proyectos, donde una vez se haya verificado la corrección de las causas, se puede dar por concluido el proceso.

4.2.6. Operación y Mantenimiento

Este proceso se basa en la distribución de los productos y el mantenimiento que se brinda a los usuarios después de su adquisición, en contraste al proceso actual de Distribución con el que cuenta la empresa, en donde el enfoque se dirigía en la operación de los productos y no a su correcta operación y completa satisfacción de los usuarios.

La propuesta de mejora de este proceso se basa primordialmente en la generación de asignación de categorías para cada cliente, que según la lista de precios en la que se encuentre y el paquete de productos que adquiera, permitirá a la empresa otorgar los permisos que merezcan cada uno de los usuarios. La segunda propuesta mayor que genera este proceso es el control de la instalación y uso de los productos oficiales distribuidos, asegurando la satisfacción del cliente.

El proceso también asume la probabilidad de que aún con los filtros de pruebas para distribuir productos que aseguren la satisfacción total de los usuarios, se pueden generar nuevas observaciones que ya no pueden ser corregidas o mejoradas con productos ya disponibles al público, por lo que estas nuevas observaciones conformarán parte de una base de datos que permita evitar esas fallas y faltas en proyectos futuros.

4.2.7. Mejoras al proceso de Contratación e Inducción

Como se mencionaba al inicio de este punto del capítulo, el proceso de Contratación e Inducción es considerado por la empresa actualmente como parte del desarrollo de software general, sin embargo, se ha determinado en esta propuesta que no forma parte de los procesos misionales de desarrollo, ya que es un proceso que se lleva a cabo eventualmente y no se genera en la creación de

cada proyecto. Aun así, existen ciertas pautas que se pueden considerar para mejorar el proceso y promover su efectividad como un proceso de apoyo.

En primer lugar, es necesario determinar el nuevo puesto del personal que se busca contratar e incluir al equipo de desarrollo o simplemente al equipo empresarial, planificando el perfil de la persona que se necesita para cumplir con el cargo, así como analizar las funciones que cumplirá y los beneficios que recibirá.

También es importante que las pruebas que cada candidato deba realizar sean elaboradas bajo parámetros medibles, en donde todos los postulantes puedan ser comparados en términos técnicos y objetivos, para así evitar la dificultad en el cumplimiento de tareas futuras que puedan tener los nuevos empleados.

Así mismo, el programa de estudios que cada nuevo empleado reciba, debe ser planificado con anticipación según las actividades que se están desarrollando en los proyectos presentes o aquellas que se desarrollarán en los proyectos futuros, para así poder alinear con facilidad los nuevos conocimientos con las habilidades que se requieren en el equipo.

Cada nuevo empleado de la empresa que se encuentre en el periodo de aprendizaje no debe ser únicamente monitoreado por el Gerente General, pero cada etapa puede dividirse por cada miembro del equipo ya existente, para así minimizar el tiempo de cada uno de las personas que no cumple con las actividades del Kanban Board.

5. CAPÍTULO 5.- PLAN DE IMPLEMENTACIÓN

El presente capítulo detalla las características que conlleva implementar la propuesta presentada en el capítulo anterior, en donde se analizan importantes factores de costo y beneficio que se presentarían para HSB SOFTECUADOR C.A. en caso de tomar la decisión de aplicarla en el futuro.

Es pertinente determinar los riesgos que sufre el plan de implementación y las estrategias necesarias para poder mitigarlos, considerando la propuesta utilización de metodologías ágiles, el plan de implementación debe poseer aptitudes flexibles y resilientes que permitan lograr mejores resultados a la empresa en cualquier circunstancia.

Así mismo también es importante determinar todas las actividades necesarias que significaría la implementación de la propuesta más allá de este estudio e investigación, que permita identificar tanto el costo en tiempo y dinero requeridos para llevar en marcha el plan.

Una vez determinado el costo necesario para permitir implementar la propuesta, resulta primordial identificar todos los beneficios que se generan a la par y así poder comparar y ponderar el éxito o fracaso del plan de implementación a través de un análisis de costo-beneficio.

5.1. Objetivos de la implementación

Los objetivos para la implementación de la propuesta se enfocan en los resultados que se buscan obtener de esta operación, especificando los siguientes:

1. Reducir el tiempo de entrega promedio de los productos en un 15%.
2. Incrementar el número de descargas anuales realizadas en un 10% en productos de mayor alcance y 20% en productos de menor alcance.
3. Incrementar el nivel de mejoras de los productos desarrollados en un 30%.
4. Reducir los costos de desarrollo de los productos generados de los proyectos en un 5%.

Es importante determinar que el objetivo número 1 se puede generar por dos factores: en primer lugar al reducir el tiempo de los desarrollos actuales y en segundo lugar al incrementar el alcance de los proyectos para generar más herramientas por productos en el mismo tiempo actual de desarrollo.

5.2. Riesgos y estrategias de la implementación

Considerando los beneficios y mejoras que la propuesta plantea generar, existen también riesgos en la implementación que puedan afectar a los resultados de la empresa HSB SOFTECUADOR C.A., explicados a continuación en la Tabla 8, junto a estrategias que se pueden utilizar para mitigar o eliminarlos.

Los riesgos considerados no deben confundirse con las amenazas con las que cuenta la empresa analizadas en el Capítulo 1. Los riesgos son considerados

únicamente por los factores que puedan tornarse impedimentos a la implementación de la propuesta.

Tabla 8.- Riesgos y Estrategias de Implementación

RIESGO		ESTRATEGIA
1	Falta de cooperación del equipo de desarrollo para laborar bajo una nueva metodología.	Realizar capacitaciones sobre metodologías ágiles de trabajo y los beneficios que estas conllevan.
2	Oposición por parte del Directorio Internacional a generar cambios que no consideran necesarios para la región de la empresa.	Comunicar que los cambios están enfocados en metodologías internacionales de aplicación que permiten la fácil adaptación de los procedimientos a la región en la que se encuentren.
3	Falta de tiempo del Gerente General a gestionar la implementación al dedicar su tiempo en la mayoría al desarrollo de software.	Separar el cargo de Gerente General y de Desarrollador de Software, asignando ambos roles a personas distintas según sus capacidades.
4	Falta de recursos que permitan implementar la propuesta completamente.	Implementar paulatinamente los procedimientos, iniciando por el nivel más básico de cambios. No se recomienda implementar un procedimiento a la vez debido a que todos se encuentran estrechamente ligados.
5	Falta de integración de los procedimientos de forma simultánea, considerando su cumplimiento en forma de cascada.	Capacitar a los empleados en el funcionamiento integrado de forma ágil de los procesos.

5.3. Plan de Implementación

El presente plan de Implementación presentado en la Tabla 9 permite visualizar las actividades que se deben llevar a cabo para poder implementar la propuesta con éxito, así como los responsables, la duración y recursos financieros y no financieros necesarios para su implementación. Así mismo, permite fácilmente identificar los responsables de cada una de las actividades, así como el tiempo y los recursos tanto físicos como económicos que estos conllevan.

Todas las actividades han sido divididas según la etapa del proyecto a las que se realizan los cambios y se benefician de estos. Los costos se han considerado con una base anual de cálculo.

Tabla 9.- Costo de la Implementación

ACTIVIDADES	RESPONSABLE	DURACIÓN	RECURSOS	COSTO TOTAL
INICIO				
Aprobar la propuesta de mejora de procesos de desarrollo de software y los procedimientos adjuntos.	Directorio Internacional	1 semana	El Directorio Internacional estudiará la presente propuesta.	\$0
Capacitar al personal sobre los procedimientos propuestos, el impacto y los cambios que estos pueden generar a largo plazo.	Gerente General	2 semanas	Tiempo de desarrollo del equipo	\$1 600

ACTIVIDADES	RESPONSABLE	DURACIÓN	RECURSOS	COSTO TOTAL
PLANIFICACIÓN				
Capacitar al personal en la gestión de proyectos ágiles.	Directorio Internacional	1 mes	Curso para cada desarrollador	\$1 000
EJECUCIÓN				
Contratar a un Desarrollador de Software adicional.	Gerente General	1 mes	Reclutamiento, pruebas de selección y contrato	\$9 600
Promover al Desarrollador más capacitado y antiguo a Desarrollador Líder.	Directorio Internacional	2 semanas	Pruebas de selección y contrato	\$3 600
Implementar programas de medición del trabajo para determinar el trabajo no relacionado al desarrollo de productos.	Gerente General	2 semanas	Suscripción mensual por desarrollador	\$1 400
MANTENIMIENTO Y CONTROL				
Capacitar al personal en temas de planificación y control de pruebas realizadas a los productos.	Directorio Internacional	1 mes	Curso para cada desarrollador	\$800
Generar bases de datos de proyectos que permitan identificar las lecciones aprendidas.	Equipo de Desarrollo	1 mes	Integración de resultados de proyectos pasados	\$0

ACTIVIDADES	RESPONSABLE	DURACIÓN	RECURSOS	COSTO TOTAL
CIERRE				
Generar bases de almacenamiento de la comunicación escrita con los clientes y usuarios.	Equipo de Desarrollo	2 meses	Integración de las líneas de comunicación con los clientes	\$0
Gestionar un proyecto piloto bajo los lineamientos de la propuesta para corrección de errores.	Gerente General	4 meses	Generación de un nuevo proyecto	\$0
			TOTAL	\$ 18 000

5.4. Cronograma de implementación

Es necesario determinar la distribución del tiempo requerido para cumplir con la implementación de la propuesta. En la Tabla 10 se puede apreciar dicha gestión del tiempo, en donde las actividades se asignan a un periodo específico de realización asignado semanalmente. Como se puede observar, no todas las actividades se realizan de forma consecutiva y linear, se realizan a medida que sean requeridas en uno o más periodos de implementación. El periodo de implementación consta de 7 meses, asignado a 10 actividades principales de la siguiente manera:

Tabla 10.- Cronograma de actividades

ACTIVIDADES		Mes 1				Mes 2				Mes 3				Mes 4				Mes 5				Mes 6				Mes 7			
		Semana				Semana				Semana				Semana				Semana				Semana							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	Aprobar la propuesta de mejora de los procesos de desarrollo de software y los procedimientos adjuntos.																												
2	Capacitar al personal sobre los procedimientos propuestos, el impacto y los cambios que estos pueden generar a largo plazo.																												
3	Capacitar al personal en la gestión de proyectos ágiles.																												
4	Contratar a un Desarrollador de Software adicional.																												

ACTIVIDADES		Mes 1				Mes 2				Mes 3				Mes 4				Mes 5				Mes 6				Mes 7			
		Semana				Semana				Semana				Semana				Semana				Semana							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
5	Promover al Desarrollador más capacitado y antiguo a Desarrollador Líder.																												
6	Implementar programas de medición del trabajo para determinar el trabajo no relacionado al desarrollo de productos.																												
7	Capacitar al personal en temas de planificación y control de pruebas realizadas a los productos.																												
8	Generar bases de datos de proyectos que permitan identificar las lecciones aprendidas.																												
9	Generar bases de almacenamiento de la comunicación escrita con los clientes y usuarios.																												

ACTIVIDADES		Mes 1				Mes 2				Mes 3				Mes 4				Mes 5				Mes 6				Mes 7			
		Semana				Semana				Semana				Semana				Semana				Semana							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
10	Gestionar un proyecto piloto bajo los lineamientos de la propuesta para corrección de errores.																												

5.5. Beneficios

La propuesta presentada trae consigo beneficios reflejados en tiempo, dinero y cooperación interna, siendo este uno de los puntos más importantes que conlleva el trabajo de investigación, para probar que la propuesta de mejoramiento realmente trae consigo la eficiencia y eficacia detallada.

Sin embargo, es necesario considerar que los beneficios que trae consigo la propuesta son cuantificables y no cuantificables por igual. Es importante considerar ambos tipos de beneficios para poder asimilar los correctos resultados de una implementación posterior. Todos los beneficios cuantificables y no cuantificables presentados a continuación han sido determinados a través de los registros históricos disponibles de la empresa.

5.5.1. Beneficios Cuantificables

Los beneficios cuantificables presentados son valores basados en estimaciones aproximadas calculadas a partir de escasos registros históricos de la empresa, es por esto que se debe considerar que los beneficios detallados a continuación pueden variar significativamente de forma positiva o negativa si se llegase a realizar la implementación respectiva. Estos beneficios representan la base de cálculo de los beneficios económicos de la propuesta.

Considerando que uno de los objetivos de la implementación es reducir el tiempo de entrega de los productos, se ha considerado periodos por proyecto actuales con los que cuenta la empresa de 6 y 10 meses, en el que se conseguiría un beneficio en tiempo anual de aproximadamente 2 meses para los proyectos más sencillos y un beneficio anual aproximado de 1 mes y medio para los proyectos de mayor complejidad como se puede observar en la Tabla 11.

Estos beneficios se han logrado al determinar los límites de cada proyecto en su inicio, así como se la eliminación de burocracia innecesaria y la limitación de las acciones correctivas necesarias para cada producto.

Estos beneficios en tiempo estimados, a su vez representan beneficios económicos que se consideran para detallar el beneficio total que representará la implementación de la propuesta, como se detalla más adelante.

Tabla 11.- Beneficios en tiempo de la implementación

Tiempo por proyecto	6 meses	10 meses
Capacidad de proyectos por año	2	1
Tiempo por proyecto esperado	5,1 meses	8,5 meses
Beneficio por proyecto	0,9 meses	1,5 meses
Beneficio Total	1,8 meses	1,5 meses

Considerando las actividades detalladas anteriormente sobre las nuevas posiciones requeridas, se facilita el cálculo del valor que cuesta cada trabajador directo en los proyectos reflejados en la Tabla 12 en un análisis anterior a la implementación y posterior a la misma, donde se evidencia la eliminación del valor por hora del Gerente General, el incremento del costo del Desarrollador Jefe y la generación del costo de un nuevo desarrollador.

Analizando la suma del valor por hora de todos los trabajadores sea \$37,52 en el escenario actual y la suma del valor por hora propuesto de \$31,26, se ha logrado una reducción por hora de 16,68%.

Tabla 12.- Costo por hora de cada trabajador

ANTES			
POSICIÓN	Sueldo mensual	horas/mes	Valor hora
Gerente General	2100	160	13,13
Desarrollador 1	1500	160	9,38
Desarrollador 2	1500	160	9,38
Diseñador Gráfico	900	160	5,63
DESPUÉS			
POSICIÓN	Sueldo mensual	horas/mes	Valor hora
Desarrollador Jefe	1800	160	11,25
Desarrollador 1	1500	160	9,38
Desarrollador 2	800	160	5,00
Diseñador Gráfico	900	160	5,63

Una vez determinado el costo por hora de cada uno de los trabajadores, se ha calculado el costo por proyecto que se generan. Considerando la capacidad actual de la empresa de generar un proyecto de 6 meses simultáneamente a uno de 10 meses, se puede observar en la Tabla 13 que se han reducido los costos en 27%, así como se ha incrementado el número de horas disponibles para nuevos proyectos en 39,92%.

Si bien es cierto, la reducción de costos no representa un ingreso adicional, representa un valor no desembolsado que permite generar una nueva inversión para un nuevo proyecto, que junto con el tiempo adicional no utilizado en un año de desarrollo de software, genera aún mayor capacidad de generar más productos que a su vez incurren en un mayor número de ventas.

Tabla 13.- Reducción del costo de trabajadores por proyecto

ANTES					
POSICIÓN	Proyecto de 6 meses		Proyecto de 10 meses		Horas Disponibles
	horas/trabajador	\$	horas/trabajador	\$	
Gerente General	324	4.252,50	900	11.812,50	504
Desarrollador 1	360	3.375,00	1000	9.375,00	560
Desarrollador 2	360	3.375,00	1000	9.375,00	560
Diseñador Gráfico	360	2.025,00	1000	5.625,00	560
TOTAL	13.027,50		36.187,50		2184
DESPUÉS					
POSICIÓN	Proyecto de 5,1 meses		Proyecto de 8,5 meses		Horas Disponibles
	horas/trabajador	\$	horas/trabajador	\$	
Desarrollador Jefe	306	3.442,50	850	9.562,50	764
Desarrollador 1	306	2.868,75	850	7.968,75	764
Desarrollador 2	306	1.530,00	850	4.250,00	764
Diseñador Gráfico	306	1.721,25	850	4.781,25	764
TOTAL	9.562,50		26.562,50		3056

Reducir los costos de los proyectos e incrementar las horas disponibles para generar nuevos proyectos representa una mejora en la productividad de la empresa que permite que se generen un flujo de entrega de productos más veloz, así como se permite un mayor tiempo de ventas que pueda generar ganancias adicionales.

En el ámbito de la calidad, se espera resolver gestionar de forma más efectiva las mejoras necesarias para los proyectos, considerando que con un incremento promedio del 30% de mejoras realizadas a los productos, generaría que las observaciones que se tienen, serían manejadas aproximadamente al 90% en contraste al 70% aproximado con el que se cuenta actualmente y se puede apreciar en la Tabla 14.

Estos beneficios se pueden lograr mediante la determinación inicial del alcance de cada proyecto, así como los objetivos que se buscan cumplir, ante la correcta aplicación de los procesos de Elaboración de Pruebas y Ensayos y Gestión de Mejoras.

Tabla 14.- Beneficios de calidad de la implementación

Observaciones actuales promedio	1240
Mejoras realizadas actuales	850
Mejoras realizadas esperadas	1105
Porcentaje de mejoras actuales sobre el nivel de observaciones	70%
Porcentaje de mejoras esperadas sobre el nivel de observaciones	90%
Beneficio porcentual	+20%

Considerando el objetivo de la implementación relacionado al incremento de las descargas, se pudo estimar en base a promedios de capacidad de la empresa, que los productos disponibles a los usuarios pueden incrementarse en un 10% para los productos de mayor distribución y alcance al público general, mientras que se espera un incremento del 20% en productos de menor distribución y alcance al público general. Esto representaría un incremento de ventas para la empresa en su ámbito internacional, pero representa un incremento de la productividad de la empresa en el campo nacional como se puede observar en la Tabla 15.

El incremento de descargas representa a su vez un incremento financiero consecuente directamente de este factor, que al considerar los registros de la empresa y su capacidad actual de generar simultáneamente dos proyectos de diferentes alcances, genera el beneficio de \$45 330.

El incremento del nivel de descargas se puede lograr porque al determinar los requisitos de cada producto de forma analítica y objetiva, se puede conocer las necesidades que poseen los posibles clientes, en contraste al método actual de determinar en etapas finales del desarrollo.

Tabla 15.- Beneficios en productividad de la implementación

PRODUCTO	Promedio de descargas actuales anuales	Porcentaje de incremento	Promedio de descargas estimadas anuales	Incremento de descargas	Precio	Ingreso Adicional
Menor alcance	4	20%	4,8	1	\$330	\$330
Mayor alcance	2000	10%	2200	200	\$225	\$45 000
TOTAL						\$45 330

Finalmente, considerando todos los factores detallados separadamente, se puede determinar el beneficio estimado total que representa la reducción de tiempos de programación, observaciones por parte de los usuarios y costos de programación, así como el incremento del número de descargas de los productos, determinados en la Tabla 16, con un beneficio total de \$58 420

Tabla 16.- Beneficios económicos anuales de implementación

Reducción de costos de trabajadores al año	\$13 090
Incremento en el número de descargas anuales	\$45 330
Beneficio anual estimado	\$58 420

5.5.2. Beneficios No Cuantificables

Los beneficios no cuantificables que resultarían de la implementación de la propuesta se basan en la reducción de trabajo innecesario para los desarrolladores, así como se genera un mejor canal de comunicación entre ellos para poder integrar de forma más eficiente los resultados de los desarrollos realizados.

Así mismo, la propuesta provee mayor seguridad por parte del equipo en el lanzamiento de cualquier producto al mercado, ya que se busca la reducción de fallas y faltas en los productos que generen poca aceptación de cualquiera de los productos disponibles.

A su vez, la propuesta pretende generar mayor proactividad ante los riesgos que genera un proyecto ágil que no sigue una línea en cascada de creación, ya que no se requiere que termine una etapa para poder continuar con la corrección de los errores generados en el proceso de desarrollo.

5.6. Análisis Costo-Beneficio

Una vez determinado los valores tanto del costo como del beneficio que trae la implementación consigo, se puede realizar un análisis para determinar si la propuesta realmente es favorable para la empresa HSB SORTECUADOR C.A.

El índice se mide de dos formas distintas, en donde en la Tabla 17 únicamente se considera el beneficio del incremento en ventas en relación a los costos de implementación, otorgando un valor de **\$2,52**, mientras que en la Tabla 18 se considera el beneficio total del incremento de ventas añadido el valor de la reducción del costo de desarrollo con un valor de **\$3,25**.

Considerando que ambos índices son mayores a 1, se puede asegurar que la implementación de la propuesta significaría un impacto positivo para la empresa en el caso de implementar la propuesta.

Tabla 17.- Relación Costo-Beneficio por incremento de ventas

Beneficio por incremento en ventas	\$45.330,00
Costo de implementación	\$18.000,00
Beneficio/Costo	\$ 2,52

Tabla 18.- Relación Costo/Beneficio por incremento en ventas y reducción de costos

Beneficio por incremento en ventas y reducción de costos	\$58.420,00
Costo de implementación	\$18.000,00
Beneficio/Costo	\$ 3,25

6. CAPÍTULO 6.- CONCLUSIONES Y RECOMENDACIONES

Al considerar todos los aspectos detallados en los anteriores puntos, es importante determinar las conclusiones que la investigación ha generado, así como también las recomendaciones que se sugieren para la empresa HSB SOFTECUADOR C.A.

6.1. Conclusiones

Las conclusiones que se listan a continuación se han determinado al considerar no solo la situación actual de la empresa, sino también sus antecedentes y la propuesta generada:

- Las empresas pequeñas que se dedican al desarrollo de software, se benefician en mayor ámbito al poseer un modelo ágil de trabajo, en lugar del modelo en cascada que es más rígido a los cambios que se puedan necesitar en el futuro de un desarrollo.
- La comunicación tanto interna como externa que posee una empresa es de vital importancia para generar productos que cumplan con las expectativas y requisitos del mercado en general.
- La empresa actualmente no considera importante el mantenimiento de registros que permitan identificar rendimiento a futuro.
- A nivel internacional, se considera únicamente como mercado meta a distintas regiones de Europa y Japón, generando que el enfoque de desarrolladores de América Latina deba adaptarse al pensamiento de esas localidades.

- Se cuenta con la posibilidad de expandir su mercado meta a distintas regiones ecuatorianas al considerar los Proyectos de Vivienda que el actual mandato lleva a cabo.
- El sector de desarrollo de software en Ecuador ha mantenido un crecimiento constante en la última década.
- El Código INGENIOS (Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación) promueve precauciones adicionales a la empresa al momento de contratar a sus desarrolladores para que ellos no sean los titulares de los productos realizados.
- El Gerente General cumple con la función de Desarrollador, elevando los costos directos de cada producto al considerar que su hora de trabajo vale más que la del resto de desarrolladores.
- La empresa no cuenta con una visión y misión establecidas que permitan a los empleados a identificarse con los objetivos de la empresa. Así mismo, no considera de importancia planificar la seguridad, la gestión del riesgo, de los recursos y la comunicación.
- Los empleados de la empresa no pueden determinar o limitar con facilidad los procesos con los que cuenta la empresa en la actualidad.
- Las etapas de desarrollo de software de las metodologías en cascada pueden ser utilizadas en las metodologías ágiles siempre y cuando se tenga en cuenta que no siguen un cumplimiento lineal, al contrario, se promueve que las etapas se cumplan de forma cíclica y no mantengan un orden específico si la situación lo requiere.
- Las metodologías ágiles de desarrollo de software son útiles en empresas con similares características a HSB SOFTECUADOR C.A., debido a que promueven el control continuo sobre la ejecución de los planes, generando

acciones correctivas inmediatas y precisas en cortos periodos de tiempo, es por esto que la empresa busca emplear la metodología ágil SCRUM para el desarrollo de sus productos, sin embargo únicamente aplica pocos conceptos de la metodología actualmente.

- No se determina el alcance de cada producto al inicio de cada proyecto, generando incertidumbre al momento de finalizar e integrar los productos.
- Al momento de desarrollar los productos la empresa actualmente no cuenta con un diseño del proyecto o borrador del prototipo que le permita desarrollar con facilidad el código de los productos.
- La empresa posee ambiguos métodos de medición para las pruebas de sus productos, así como el rendimiento que proveen los mismos.
- Se ha identificado 6 procesos que permiten mejoras en el proceso de desarrollo de software general, los cuales son: Análisis de Requisitos, Diseño de Proyectos, Desarrollo de Prototipos, Elaboración de Pruebas y Ensayos, Gestión de Mejoras y Operación y Mantenimiento.
- La implementación de la propuesta se puede realizar en un periodo de 7 meses con un costo de \$18 000.
- La propuesta generaría un índice de Costo-Beneficio Total de \$3,25, provocando un impacto positivo al rendimiento de la empresa.

6.2. Recomendaciones

Las recomendaciones que se listan a continuación se han determinado al considerar los beneficios que la propuesta puede traer a la empresa:

- Determinar una Visión y Misión que permitan a los empleados entender lo que se espera de ellos en la empresa y así ellos puedan alinear sus objetivos con los de la organización.
- Mantener registros más detallados que evidencien cada desarrollo que han realizado para medir de forma más efectiva los resultados de los proyectos.
- La posición de Gerente General no debe cumplir con las funciones de la posición de Desarrollador de Software, debido a que el trabajo de la persona exige que divida su tiempo hacia ambas tareas y no dediquen todo su potencial a una u otra.
- Es necesario delimitar sus procesos para que facilite el desarrollo de software al entender lo que se espera de sus desarrolladores.
- Delimitar con anticipación el alcance de cada proyecto para poder conocer el momento en el que cada producto se ha finalizado y debe ser integrado.
- Entender que al desarrollar software para países con cultura diferente, pueden existir varios cambios a lo largo del desarrollo de cada producto, por lo que necesita planificar la gestión a los riesgos que puede tener y así poseer productos flexibles ante cualquier necesidad de cambio.
- Tener reuniones diarias de menor duración para poder utilizar su tiempo de forma más eficiente en el cumplimiento de las actividades necesarias para cumplir con los requisitos de un producto.

- Planificar la forma en la que se realizarán las pruebas de todos sus prototipos para que de esta forma no se obtengan observaciones por defecto del producto posterior a su distribución a los usuarios.
- Para mejorar su productividad, la empresa debe adaptar su metodología de cascada de desarrollo de software a metodologías ágiles.
- Delimitar los procesos de desarrollo de software de la empresa en base a las 5 etapas de gestión de proyectos.
- Se necesita que la empresa delimite e identifique los procesos que realmente forman parte del desarrollo de software y aquellos que únicamente brindan soporte.
- La empresa no debe enfocarse únicamente a la corrección de las fallas o la mejora de las faltas de los productos, debe enfocarse en eliminar las causas de las fallas, que pueden ser la razón de ser de varias observaciones.
- Manejar sus procesos de gestión y apoyo de forma ágil para alinearse a su modelo de gestión de proyectos.
- Mantener registros de la vida de todos los proyectos, así como registros administrativos y contabilidad detallada que permitan conocer el funcionamiento con el que se cuenta así como una base de lecciones aprendidas de proyectos pasados, ya que de esta forma se conoce los problemas y errores que tuvo en el pasado para evitarlos en el futuro.
- Dejar de considerar al proceso de Contratación e Inducción como parte de su proceso de Desarrollo de Software.
- Seleccionar un Administrador para cada nuevo proyecto que se genere, en base a las capacidades de cada uno de los desarrolladores.

- Separar las posiciones de Gerente General y Desarrollador de Software para que puedan utilizar su tiempo de trabajo imparcialmente hacia las actividades que corresponde a cada uno de ellos.
- Las empresas de desarrollo de software con similares características a HSB SOFTECUADOR C.A. se beneficiarían de la implementación de metodologías ágiles para generar sus productos.
- Un estudio similar de mayor alcance a nivel internacional se debería realizar para poder mejorar la productividad de la empresa en todos sus aspectos y funciones.
- Es importante que personas de profesión desarrollada en el ámbito administrativo seas resilientes a cualquier tipo de entorno e industria, sin embargo, se debe mantener un nivel básico de conocimiento y gusto por el entorno en el que se desarrollará.
- La gestión de proyectos es un tema de gran utilidad para los entornos cambiantes contemporáneos, por lo que tanto profesionales técnicos como administrativos deberían capacitarse en ambos ámbitos para entender el soporte de su trabajo a un resultado final.

BIBLIOGRAFÍA

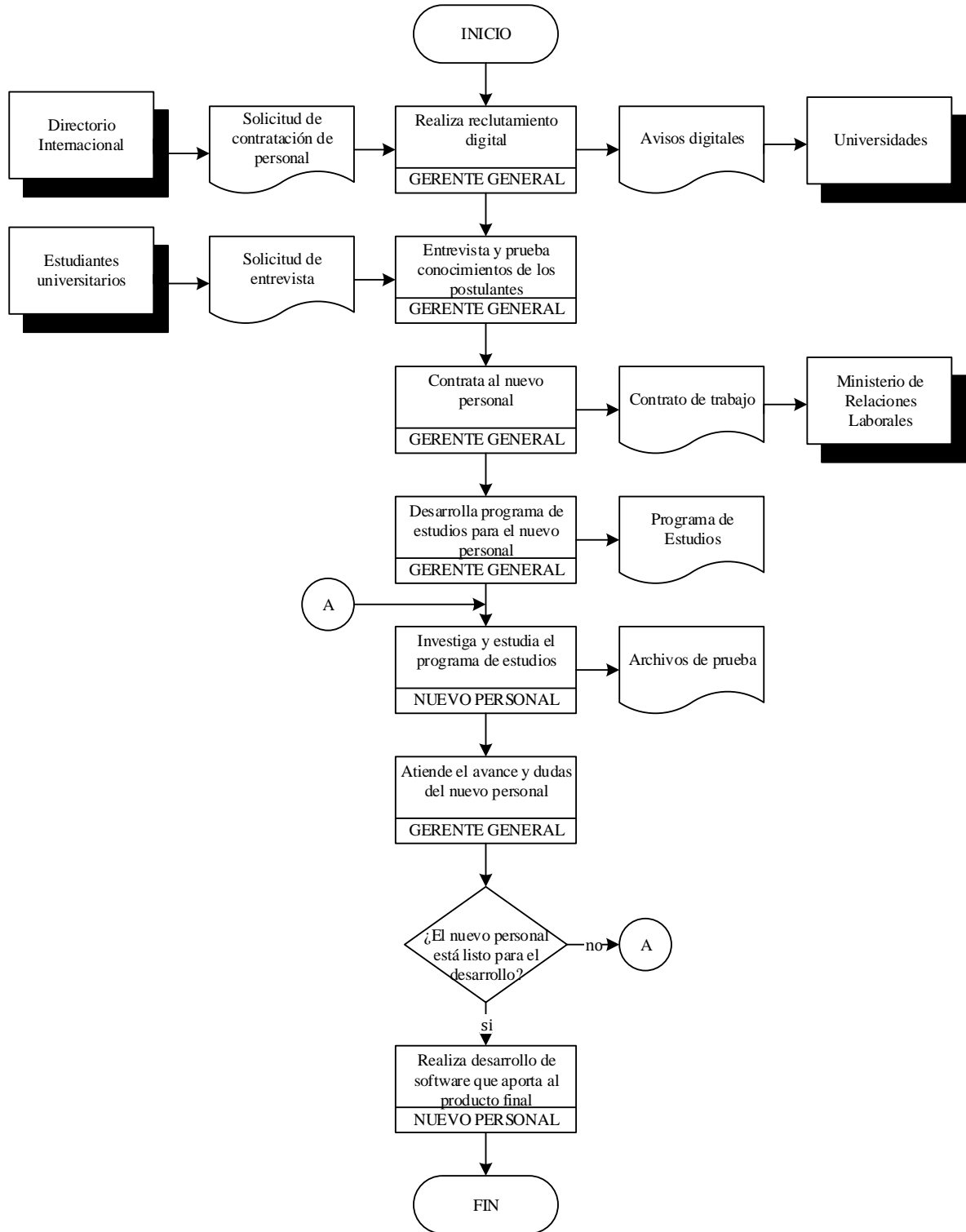
- Bienve, S. (16 de septiembre de 2016). *Belike Software*. Recuperado el septiembre de 2017, de <http://www.belikesoftware.com/desarrollo-agil-metodo-trabajo/>
- Díaz Polo, D. (2011). *Definición de un proceso de desarrollo de software en un entorno universitario*. La Habana, Cuba: Instituto Superior Politécnico José Antonio Echeverría. CUJAE.
- Fitsilis, P. (2008). Comparing PMBOK and Agile Project Management Software Development Processes. En S. Media, *Advances in Computer and Information Sciences and Engineering* (pág. 378). Springer.
- Goikolea, M. (17 de julio de 2014). *Innovation & Entrepreneurship Business School*. Recuperado el septiembre de 2017, de <https://www.google.com/url?hl=es&q=http://www.iebschool.com/blog/exitos-y-fracasos-en-proyectos-scrum-spotify-vs-healthcare-agile-scrum/&source=gmail&ust=1505833926317000&usg=AFQjCNHfrGeBXH-fvV9TTHq7zRvdGDuYRg>
- Grupo ISSI: Ingeniería del Software y Sistemas de Información. (2003). *Metodologías Ágiles en el Desarrollo de Software*. Alicante: Grupo ISSI.
- Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El Proceso Unificado de Desarrollo de Software*. Addison Wesley.
- Larrañaga, F. (2015). *Plan de Verificación y Validación*.
- Manifiesto Ágil*. (2001). (W. Cunningham, Productor) Recuperado el septiembre de 2017, de <http://agilemanifesto.org>
- Manifiesto Ágil*. (2001). (W. Cunningham, Productor) Recuperado el septiembre de 2017, de <http://agilemanifesto.org/iso/es/principles.html>

- Mundo Constructor. (20 de Abril de 2016). *Mundo Constructor*. Obtenido de <http://www.mundoconstructor.com.ec/construccion/comercial/544-el-sector-de-la-construccion-en-un-dia-cil-2016.html>
- Poppendieck.LLC. (2015). *The Lean Mindset*. Obtenido de <http://www.poppendieck.com/>
- PROEcuador. (2015). *Perfil Sectorial de Tecnologías de la Información y las Comunicaciones (TIC's) para el inversionista*. Ministerio de Comercio Exterior.
- Project Management Institute. (2017). *A Guide to the Project Management Body of Knowledge* (Sexta ed.). Chicago: Project Management Institute.
- SCRUMstudy. (2016). *Tenstep*. (E. Ledesma, Editor) Recuperado el septiembre de 2017, de Scrum y Ágil: ¿Qué tan diferentes son?: https://www.tenstep.ec/portal/articulos-boletin-tenstep/311-scrum-y-agil-que-tan-diferentes-son?utm_source=Suscriptores+TenStep&utm_campaign=3ccd779e48-Boletin_79_qna&utm_medium=email&utm_term=0_f4fd9feec7-3ccd779e48-364105713
- Wordpress. (2017). *WHAT IS THE PURPOSE OF THE NEW PMI AGILE PRACTICE GUIDE?* Recuperado el septiembre de 2017, de <http://managedagile.com/purpose-new-pmi-agile-practice-guide/>

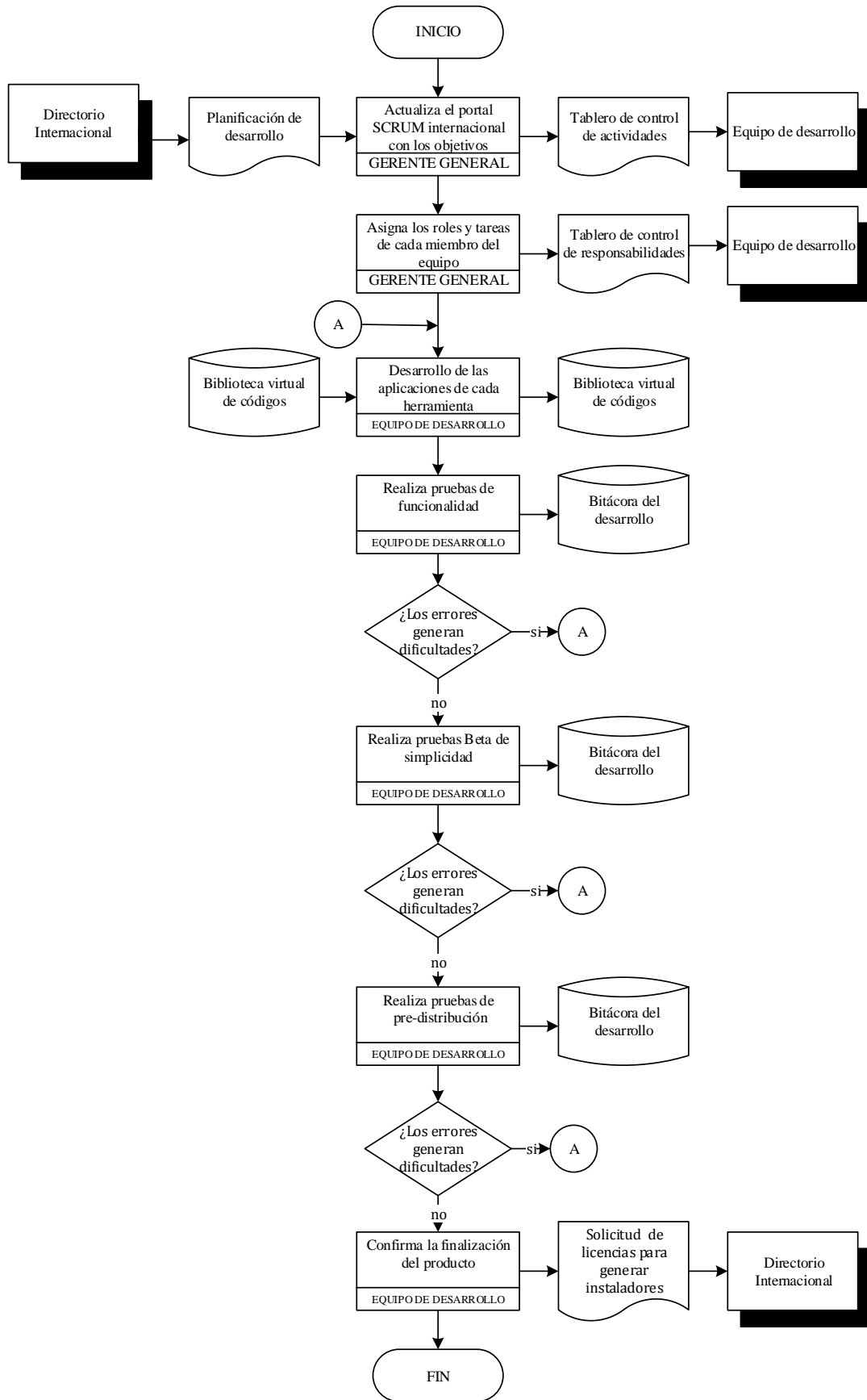
ANEXOS

ANEXO A: Diagramas de flujo de los actuales procesos de desarrollo de Software

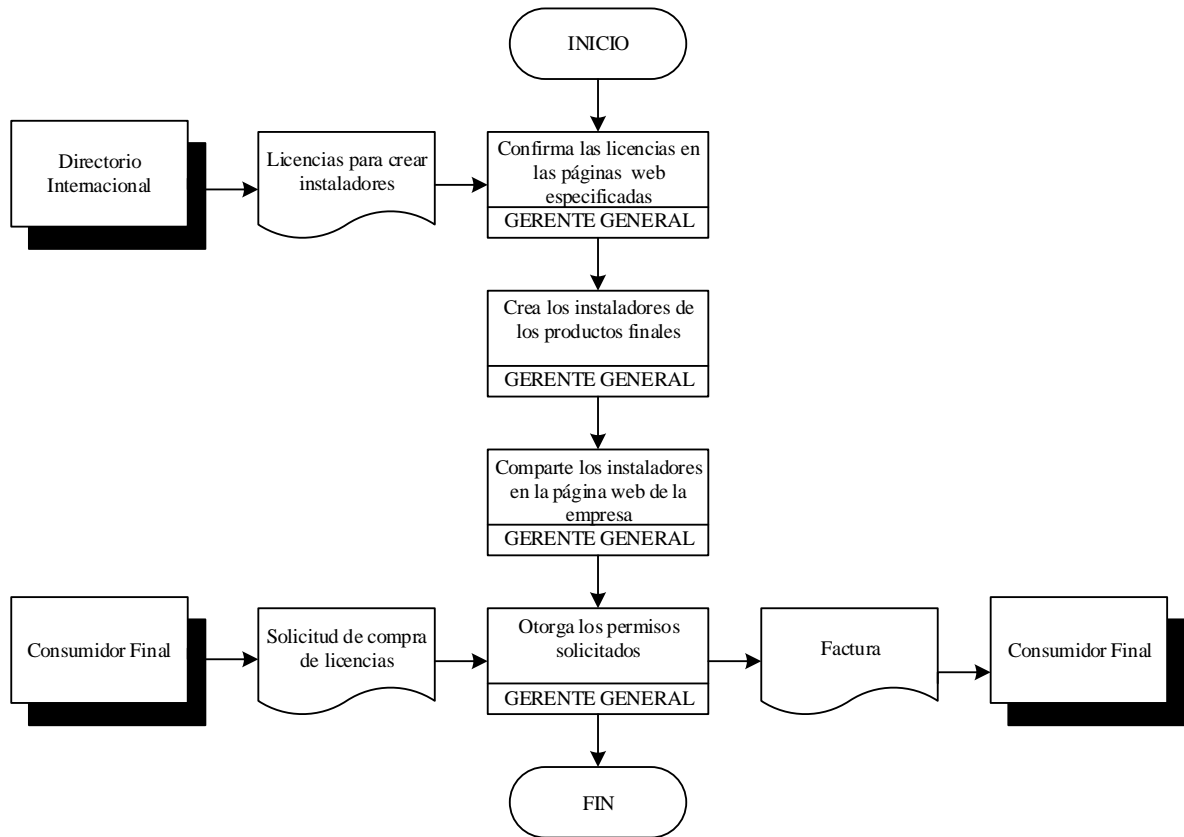
CONTRATACIÓN E INDUCCIÓN



DESARROLLO DE SOFTWARE



DISTRIBUCIÓN DE PRODUCTOS



ANEXO B: Lista de verificación de Desarrollo de Software ágil

LISTA DE VERIFICACIÓN "ÁGIL "					
CRITERIO	SUBCRITERIO	CUMPLIMIENTO			OBSERVACIONES
		TOTAL	PARCIAL	NULO	
<i>INICIO</i>	¿Tienen los integrantes del equipo apertura al cambio y compromiso al trabajo integrado?	x			Poseen apertura en el desarrollo de productos, no en su administración
	¿Se identifican las partes interesadas del proyecto?	x			Le dan mucha importancia
	¿Se asignan los roles del líder?	x			
	¿El líder identifica y elimina los impedimentos burocráticos que entorpecen el proceso?		X		No lo consideran de importancia
	¿Se asigna al proyecto un administrador del mismo?	x			Siempre es el gerente y el líder
	¿El equipo es considerado como parte integral del proyecto, en lugar de mano de obra?	x			Por ello consideran la contratación e inducción de gran impacto
	¿Se asegura que los miembros del equipo sean especialistas en el campo requerido en el proceso?	x			
	¿El equipo cuenta con el espacio físico apto para desarrollar sus funciones?	x			Se permite el trabajo en casa justificado
<i>PLANIFICACIÓN</i>	Los miembros del equipo pueden contestar fácilmente las siguientes preguntas:				
	- ¿Por qué se realiza este proyecto?		X		Algunos no conocen el propósito de los proyectos que realizan
	- ¿Quiénes se benefician de este proyecto? ¿Cómo lo hacen?	x			
	- ¿Cómo se puede saber que el proyecto ha finalizado?		X		Se tienen métodos ambiguos de medición
	- ¿Cómo trabajarán juntos como equipo?	x			Siempre se apoyan en el desarrollo y pruebas posteriores
	¿Se realiza un análisis retrospectivo de proyectos pasados?	x			Se lo realiza pero no le dan importancia
	¿Se analiza la seguridad de los programas a utilizar?	x			
	¿Se especifican las necesidades del software a desarrollar?	x			Sin embargo no determinan un alcance
	¿Se elabora un plan de seguridad para los nuevos componentes de software?			x	

CRITERIO	SUBCRITERIO	CUMPLIMIENTO			OBSERVACIONES
		TOTAL	PARCIAL	NULO	
<u>PLANIFICACIÓN</u>	¿Se elabora un cronograma?		x		Lo elaboran pero no lo siguen como lo planifican
	¿Se definen actividades?	x			
	¿Se elabora un plan de administración de costos?			x	No se considera importante
	¿Se elabora un plan de administración de recursos?			x	No se considera importante
	¿Se elabora un plan de administración de la comunicación?			x	A pesar de conocer las fallas de la comunicación, no resulta molestia
	¿Se elabora un plan de gestión de riesgo?			x	No se considera importante
	¿Se elabora un análisis cuantitativo de riesgos?			x	No se considera importante
	¿Se prepara un plan de respuesta al riesgo?			x	No se considera importante
	¿Se genera un borrador del flujo tentativo que puede llevar el proyecto?	x			Lo elaboran pero se considera que no es necesario
<u>EJECUCIÓN</u>	¿Se refina el flujo del proyecto a medida que aparecen los cambios?	x			
	¿Se realizan reuniones diarias para comprender los aspectos listados a continuación?				
	- ¿Qué se ha completado desde la última reunión?	x			
	- ¿Qué se pretende completar hasta la próxima reunión?	x			
	- ¿Cuáles son los impedimentos, riesgos, cuellos de botella y problemas?	x			Generalmente varios al no eliminar la impedimentos iniciales
	- ¿Qué se debe hacer para avanzar en el proyecto?	x			
	- ¿Algún miembro trabajo en algo que no se contenga en el flujo del proyecto? ¿Aporta al proyecto?			x	Nunca han considerado el aspecto
	- ¿Qué se debe completar como equipo?	x			
	¿Se detallan comentarios durante el desarrollo de productos?	x			
	¿Se utilizan nombres fácilmente identificables en los códigos de los productos?	x			
¿Se mantiene al mínimo las instrucciones anidadas en los productos?	x				

CRITERIO	SUBCRITERIO	CUMPLIMIENTO			OBSERVACIONES
		TOTAL	PARCIAL	NULO	
<u>MANTENIMIENTO Y CONTROL</u>	¿Se realizan pruebas y demostraciones de lo avanzado en el proyecto?	x			
	¿Se brinda retroalimentación de las pruebas realizadas por todos los miembros del equipo?	x			
	¿Se adapta el proyecto en base a la retroalimentación brindada?	x			
	¿Se integra continuamente el trabajo de todos los miembros del equipo?	x			
	¿Se realizan pruebas en todos los niveles de desarrollo del proyecto?	x			
	¿El equipo en conjunto establece los criterios de cumplimiento de los resultados del proyecto?	x			
	¿El equipo genera pruebas automatizadas para testear los criterios establecidos?			x	Las pruebas son manuales por cada herramienta
	¿Se miden los resultados del proyecto en forma cualitativa?		x		Se los miden pero no se cuentan con bases establecidas de medición
	¿Se miden los resultados del proyecto en forma cuantitativa?			x	No se poseen suficientes registros para poder medir el rendimiento
	¿Se verifica la estructura del software?	x			
	¿Se controla el cronograma?	x			
	¿Se controlan los costos, recursos y comunicación?		x		No se considera importante
¿Se controlan los posibles riesgos?		x		No se piensa frecuentemente en los riesgos	
<u>CIERRE</u>	¿Poseen estándares para determinar la finalización de un proyecto?		x		La finalización es muy subjetiva en base a conversaciones para llegar a acuerdos
	¿Se integran los resultados de todos los miembros con dificultad?	x			

ANEXO C: Puntuación del cumplimiento ágil por cada etapa de desarrollo

ETAPA DE INICIO

SUBCRITERIO	CUMPLIMIENTO
Apertura al cambio y compromiso	100%
Identificación de stakeholders	100%
Asignación de líder	100%
Eliminación de burocracia innecesaria	50%
Asignación de administrador del proyecto	100%
Reconocimiento del equipo	100%
Especialización del equipo	100%
Espacio físico apto	100%

ETAPA DE PLANIFICACIÓN

SUBCRITERIO	CUMPLIMIENTO
Entendimiento de objetivos	50%
Entendimiento de beneficiarios	100%
Entendimiento de finalización	50%
Expectativas de desempeño del equipo	100%
Análisis retrospectivo de proyectos pasados	100%
Análisis de seguridad	100%
Especificación de necesidades	100%
Plan de seguridad	0%
Elaboración de cronograma	50%
Definición de actividades	100%
Plan de administración de costos	0%
Plan de administración de recursos	0%
Plan de administración de la comunicación	0%
Plan de gestión de riesgo	0%
Análisis cuantitativo de riesgos	0%
Plan de respuesta al riesgo	0%
Borrador de flujo de proyecto	100%

ETAPA DE EJECUCIÓN

SUBCRITERIO	CUMPLIMIENTO
Adaptación del flujo del proyecto	100%
Análisis de resultados por reunión	100%
Análisis de objetivos por reunión	100%
Análisis de debilidades y amenazas	100%
Análisis de nuevas tareas	100%
Análisis de trabajo no planificado	0%
Análisis de integración de equipo	100%
Detalle de comentarios durante el desarrollo	100%
Nombres identificables en códigos	100%
Instrucciones anidadas simples	100%

ETAPA DE MANTENIMIENTO Y CONTROL

SUBCRITERIO	CUMPLIMIENTO
Realización de pruebas	100%
Retroalimentación de pruebas	100%
Adaptación en base a pruebas	100%
Integración del trabajo individual	100%
Pruebas continuas	100%
Establecimiento de criterios de finalización	100%
Pruebas de criterios de finalización	0%
Medición cualitativas de resultados	50%
Medición cuantitativas de resultados	0%
Verificación de estructura de software	100%
Control de cronograma	100%
Control de recursos y costos	50%
Control de riesgos	50%

ETAPA DE CIERRE

SUBCRITERIO	CUMPLIMIENTO
Estándares de finalización	50%
Facilidad de integración de resultados	100%

**ANEXO D: PROCEDIMIENTOS PROPUESTOS DE DESARROLLO DE
SOFTWARE PARA HSB SOFTECUADOR C.A.**

1. PROPÓSITO

Generar los lineamientos y bases del proyecto, realizables, dentro sus límites del proyecto, mediante estudio de necesidades y entrevistas a clientes.

2. ALCANCE

Este proceso aplica para todos los proyectos de desarrollo.

3. RESPONSABLE DEL PROCESO

Gerente general

4. RECURSOS

- Equipo de programación
- Clientes aleatorios
- Software del que se generarán aplicaciones adicionales
- Software del que se generarán actualizaciones
- Observaciones de productos pasados
- Equipos de cómputo personales
- Oficina compartida

5. DEFINICIONES

- **Actualizaciones:** Nuevas versiones de productos de la empresa con mejoras y herramientas adicionales.
- **Aplicaciones adicionales:** Extensiones de programas ya existentes en el mercado.

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

**CODIGO:
HSB-01**
ANÁLISIS DE REQUISITOS
Edición No. 01
Pág. 2 de 4

- **Observaciones de productos pasados:** Observaciones que los clientes realizaron una vez que un producto fue comercializado, impidiendo la corrección o mejora de la versión disponible.
- **Requisitos realizables:** Son requisitos del producto final que se mantienen dentro de los límites del proyecto.

6. POLÍTICAS

- Todos los empleados deben participar en el estudio del software que se plantea mejorar.
- Todos los empleados participarán en el debate de selección de requisitos, donde se filtrarán los requisitos por categorías realizables y no realizables.
- Los clientes a los que se entrevisten serán contactados en persona en lo posible, caso contrario se los contactarán a través de entrevistas electrónicas.
- Todos los requisitos seleccionados se encontrarán dentro del alcance del proyecto.

7. INDICADORES

Código	IND-01_HSB-01					
Nombre	Requisitos realizables					
Tipo de medida	Eficacia					
Tipo de relación	Porcentaje					
Descripción	Mide la relación de porcentaje de la cantidad de requisitos detectados que son realizables dentro del alcance del proyecto.					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
(Requisitos realizables/ Requisitos detectados)*100	Por proyecto	Positivo	80%	95%	Gerente General	Alta Dirección

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

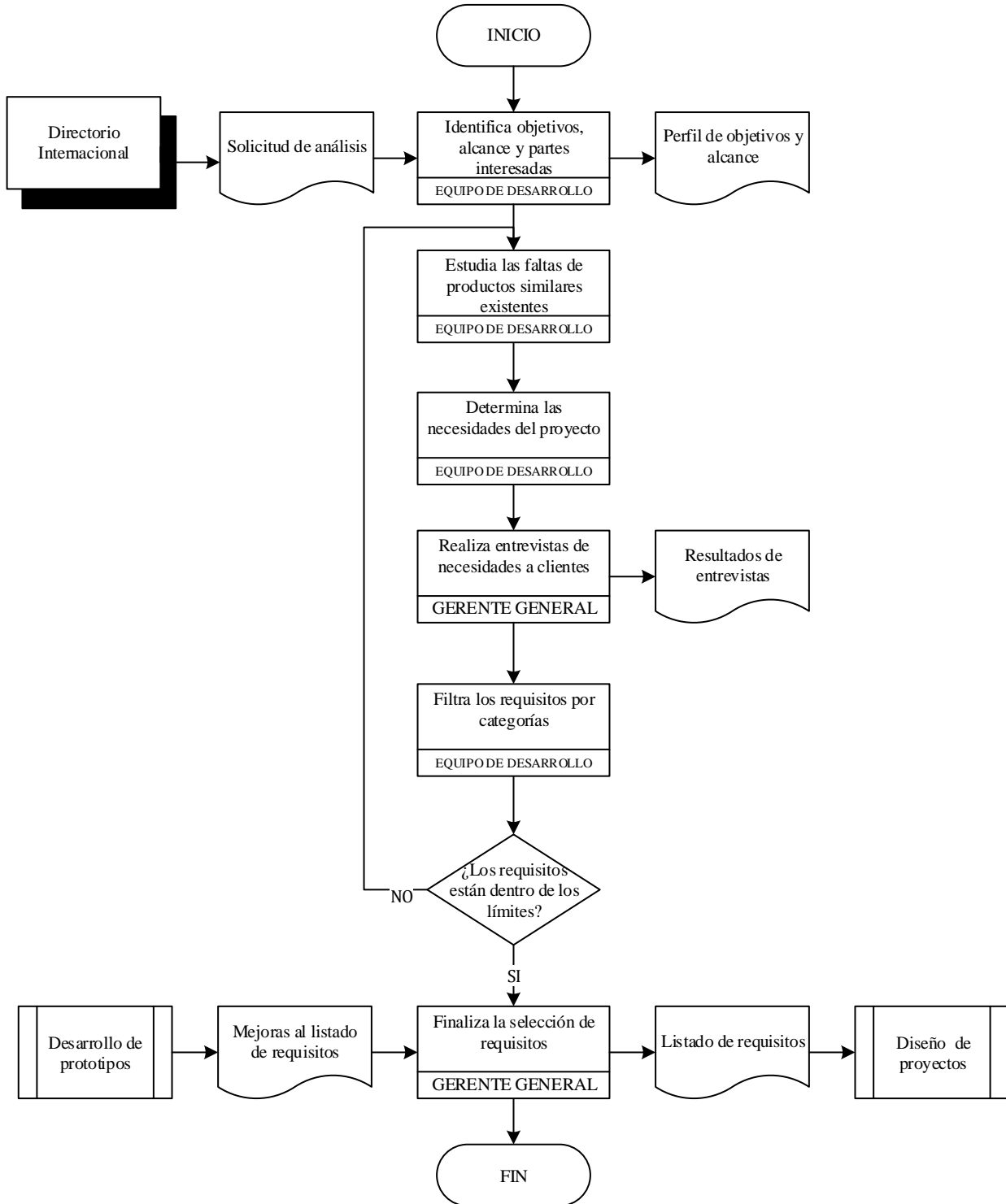
**CODIGO:
HSB-01**

ANÁLISIS DE REQUISITOS

Edición No. 01**Pág. 3 de 4****8. INFORMACIÓN DOCUMENTADA**

Código	Origen	Nombre	Soporte	Conservación	Disposición
R01_HS B-01	Externo	Solicitud de análisis	Digital	3 años	Sacar respaldo
R02_HS B-01	Interno	Perfil de objetivos y alcance	Impreso	3 años	Sacar respaldo
R03_HS B-01	Interno	Resultados de entrevistas	Impresos	3 años	Enviar a archivo pasivo
R03_HS B-03	Interno	Mejoras al listado de requisitos	Digital	5 años	Sacar respaldo
R04_HS B-01	Interno	Listado de requisitos	Digital	5 años	Sacar respaldo

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

9. DIAGRAMA DE FLUJO


Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

1. PROPÓSITO

Determinar los entregables técnicos y de gestión necesarios para desarrollar cada proyecto, mediante planes de seguridad, recursos, gestión de la comunicación y de la gestión del riesgo.

2. ALCANCE

Este proceso aplica para todos los proyectos de desarrollo.

3. RESPONSABLE DEL PROCESO

Administrador del proyecto

4. RECURSOS

- Equipo de programación
- Equipos de cómputo personales
- Oficina compartida

5. DEFINICIONES

- **Administrador del proyecto:** Miembro del equipo de desarrollo encargado de gestionar el desarrollo del mismo.
- **Borrador tentativo:** Es un diseño provisional del proyecto, ya que al ser ágil es completamente adaptable a lo largo del proceso de desarrollo de software.
- **Impedimentos burocráticos:** cualquier barrera política, legal o social que impida el desarrollo normal del proyecto.

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

CODIGO:
HSB-02

DISEÑO DE PROYECTOS

Edición No. 01

Pág. 2 de 5

6. POLÍTICAS

- La asignación de roles puede ser cambiada a futuro si el proyecto no cumple con más del 75% el cronograma y planificaciones resultados de este proceso.
- Para determinar los impedimentos burocráticos, es necesario que todos los miembros del equipo de desarrollo opinen sobre los aspectos que puedan afectar los proyectos.
- Siempre se debe considerar todos los proyectos pasados similares que puedan evitar iniciar cada proyecto sin códigos básicos.

7. INDICADORES

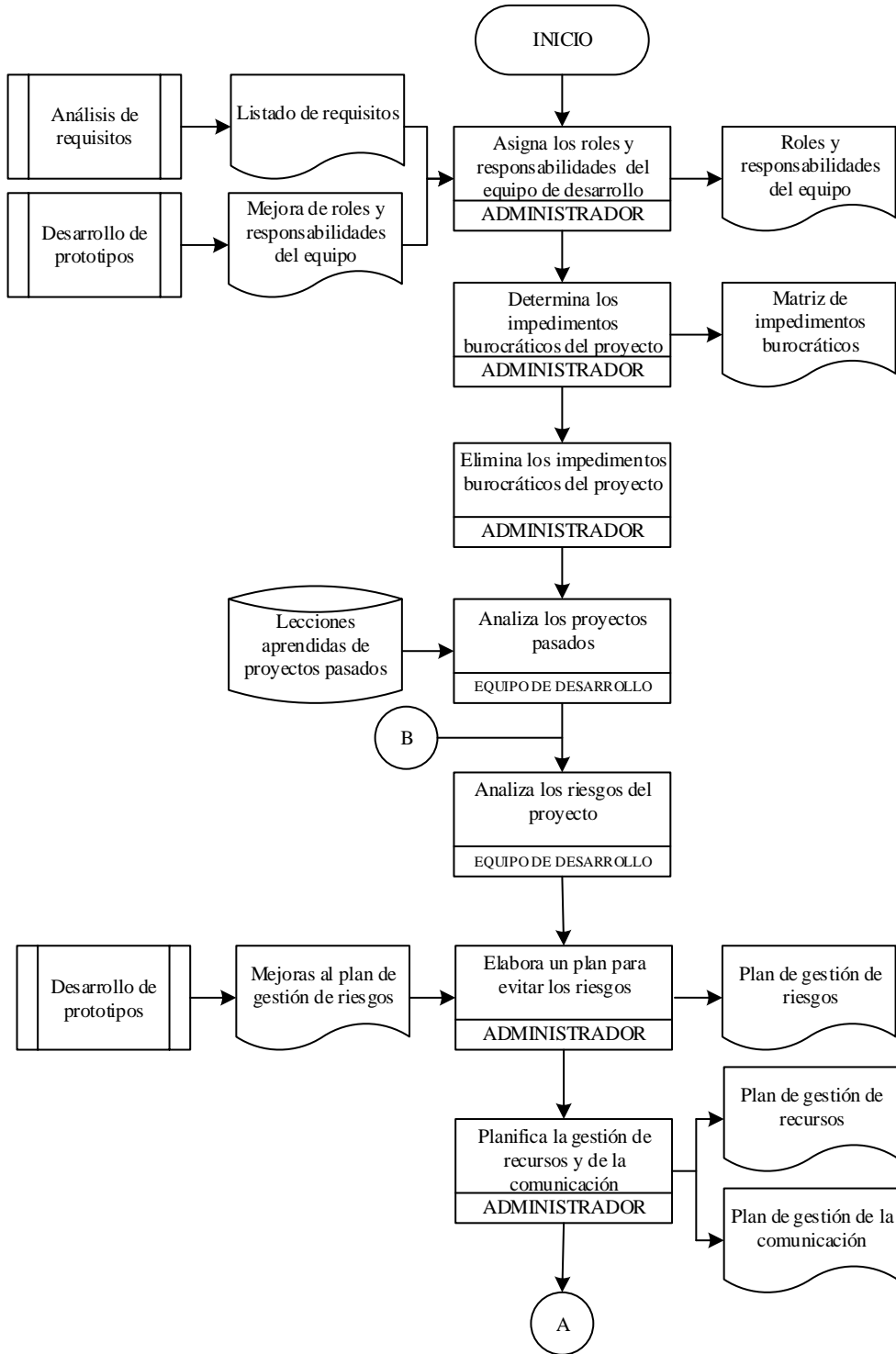
Código	IND-01_HSB-02					
Nombre	Cambios no planificados del borrador inicial					
Tipo de medida	Eficacia					
Tipo de relación	Porcentaje					
Descripción	Mide en términos de porcentaje el nivel de cambios que ha sufrido el borrador inicial del proyecto					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
(Cambios no planificados/ Cambios totales)*100	Por proyecto	Negativo	5%	10%	Administrador del proyecto	Gerente General

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

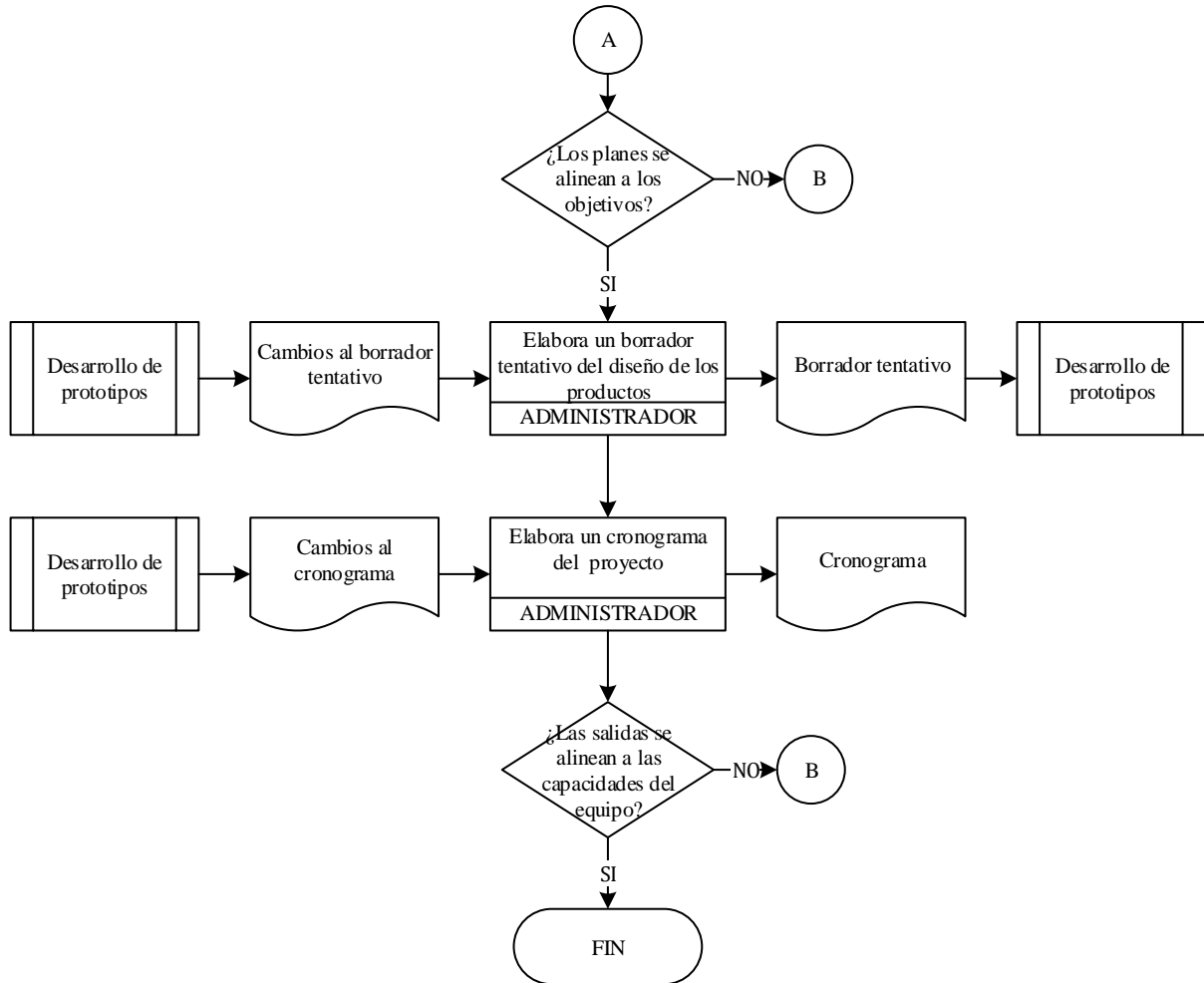
**CODIGO:
HSB-02****DISEÑO DE PROYECTOS****Edición No. 01****Pág. 3 de 5****8. INFORMACIÓN DOCUMENTADA**

Código	Origen	Nombre	Soporte	Conservación	Disposición
R04_HS B-01	Interno	Listado de requisitos	Digital	5 años	Sacar respaldo
R04_HS B-03	Interno	Mejora de roles y responsabilidades del equipo	Digital	Vigencia del proyecto	Sacar respaldo
R01_HS B-02	Interno	Roles y responsabilidades del equipo	Digital	Vigencia del proyecto	Sacar respaldo
R02_HS B-02	Interno	Matriz de impedimentos burocráticos	Digital	3 años	Sacar respaldo
R03_HS B-02	Interno	Base Lecciones Aprendidas de Proyectos Pasados	Digital	10 años	Sacar respaldo
R05_HS B-03	Interno	Mejoras al plan de gestión de riesgos	Digital	5 años	Sacar respaldo
P01_HS B-02	Interno	Plan de gestión de riesgos	Digital	5 años	Sacar respaldo
P02_HS B-02	Interno	Plan de gestión de recursos	Digital	5 años	Sacar respaldo
P03_HS B-02	Interno	Plan de Administración de la comunicación	Digital	5 años	Sacar respaldo
R06_HS B-03	Interno	Cambios al cronograma	Digital	3 años	Sacar respaldo
R04_HS B-02	Interno	Cronograma	Digital	3 años	Sacar respaldo
R07_HS B-03	Interno	Cambios al borrador tentativo	Digital	3 años	Sacar respaldo
R05_HS B-02	Interno	Borrador tentativo	Digital	3 años	Sacar respaldo

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

9. DIAGRAMA DE FLUJO


Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:



Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

1. PROPÓSITO

Asegurar el cumplimiento de los requisitos establecido mediante la generación de versiones preliminares de los productos.

2. ALCANCE

Este proceso aplica para todos los productos finales de la organización.

3. RESPONSABLE DEL PROCESO

Administrador del proyecto

4. RECURSOS

- Equipo de programación
- Kanban del proyecto
- Equipos de cómputo personales
- Oficina compartida

5. DEFINICIONES

- **Análisis de culminación:** Una reunión de todo el equipo del desarrollo para analizar si todos los miembros del equipo cumplieron sus actividades de cada sprint.
- **Kanban board:** Herramienta visual que permite optimizar el flujo del proyecto al asignar actividades y responsables.
- **Reunión ágil:** Reunión diaria de minutos donde se comparten inquietudes, observaciones y soluciones sencillas al trabajo de los demás.

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

**CODIGO:
HSB-03**
DESARROLLO DE PROTOTIPOS
Edición No. 01
Pág. 2 de 6

- **Reunión de sprint:** Reunión que se realiza al finalizar cada sprint de desarrollo para analizar el cumplimiento de las actividades y formular nuevas actividades de desarrollo.
- **Sprints:** Periodo de codificación de varios días en el que el equipo de desarrollo cumple con sus actividades asignadas en el Kanban de la empresa.

6. POLÍTICAS

- Los sprints de desarrollo serán de 15 días cada uno.
- El análisis de culminación no puede ser menor a una hora.
- Cada reunión ágil no durará más de 20 minutos al día.
- El Kanban board solo puede ser actualizado por el administrador del proyecto.
- Las actividades de cada miembro del equipo de desarrollo pueden ser intercambiadas si existe una persona que se encuentre más familiarizada con dichas actividades.
- Todos los documentos generados en la adaptación del proyecto serán solo actualizaciones de los documentos existentes en otros procesos.

7. INDICADORES

Código	IND-1_HSB-03					
Nombre	Requisitos cumplidos					
Tipo de medida	Eficacia					
Tipo de relación	Porcentaje					
Descripción	Mide en términos de porcentaje la relación entre los requisitos seleccionados y los requisitos cumplidos dentro del proyecto.					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
(Requisitos cumplidos/ Requisitos seleccionados)*100	Por prototipo	Positivo	85%	100%	Administrador del proyecto	Gerente General

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

CODIGO:
HSB-03 DESARROLLO DE PROTOTIPOS

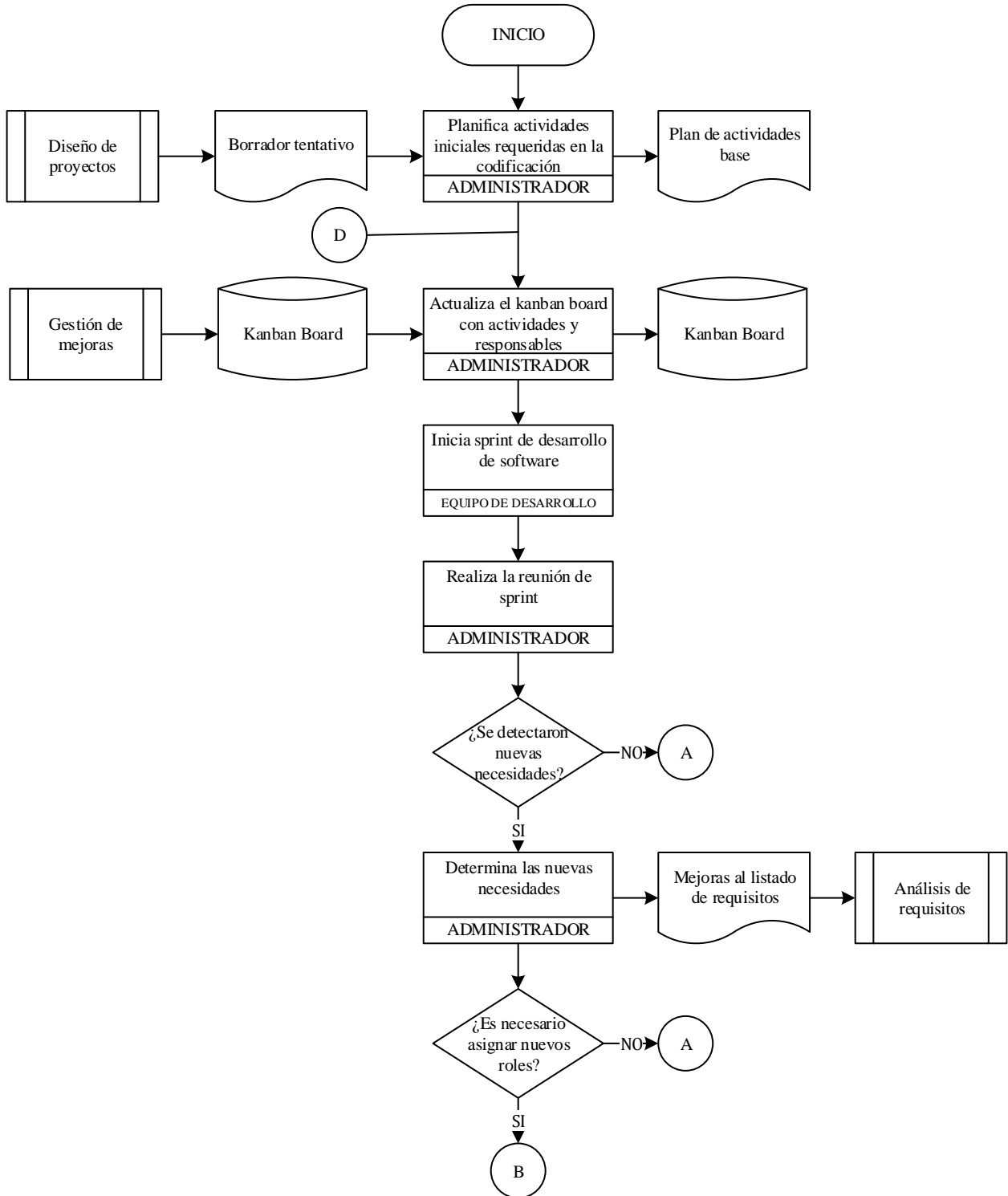
Edición No. 01
Pág. 3 de 6

Código	IND-2_HSB-03					
Nombre	Cumplimiento de cronograma					
Tipo de medida	Índice					
Tipo de relación	Porcentaje					
Descripción	Indica en términos de porcentaje el tiempo adicional que se ha ocupado para desarrollar los prototipos.					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
Tiempo utilizado/ Tiempo en cronograma	Por prototipo	Negativo	1	1,15	Administrador del proyecto	Gerente General

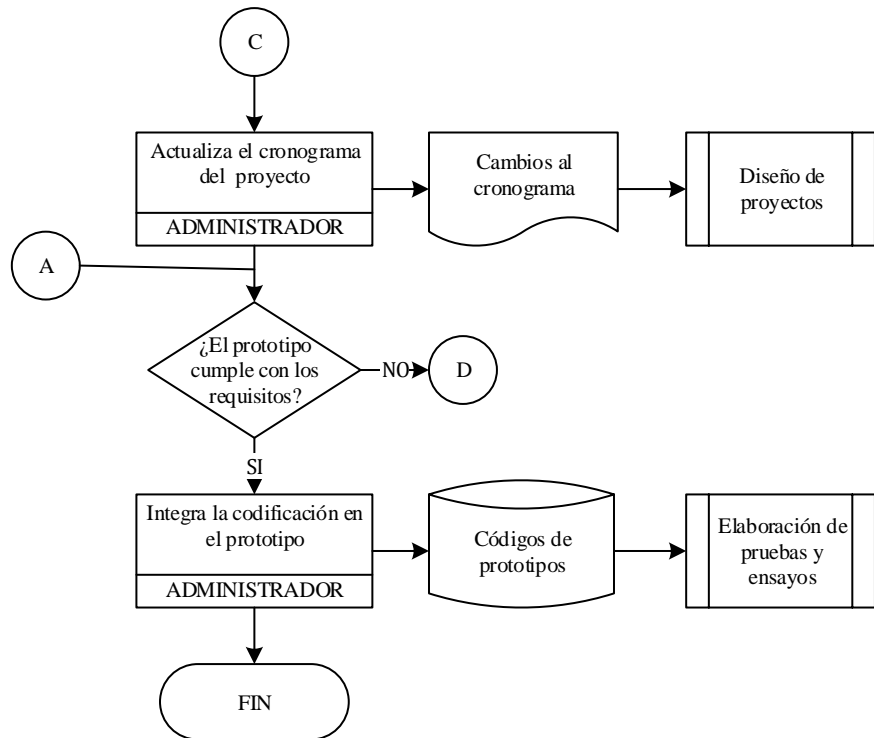
8. INFORMACIÓN DOCUMENTADA

Código	Origen	Nombre	Soporte	Conservación	Disposición
M01_HS B-03	Interno	Manual de codificación	Impreso	3 años	Actualizar y sacar respaldo de versión anterior
R05_HS B-02	Interno	Borrador tentativo	Digital	3 años	Sacar respaldo
R01_HS B-03	Interno	Plan de actividades base	Digital	3 años	Sacar respaldo
R02_HS B-03	Interno	Kanban Board	Digital	Vigencia del proyecto	Sacar respaldo
R03_HS B-03	Interno	Mejoras al listado de requisitos	Digital	5 años	Sacar respaldo
R04_HS B-03	Interno	Mejora de roles y responsabilidades	Digital	Vigencia del proyecto	Sacar respaldo
R05_HS B-03	Interno	Mejoras al plan de gestión de riesgos	Digital	5 años	Sacar respaldo
R06_HS B-03	Interno	Cambios al borrador tentativo	Digital	3 años	Sacar respaldo
R07_HS B-03	Interno	Cambios al cronograma	Digital	3 años	Sacar respaldo

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

9. DIAGRAMA DE FLUJO


Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:



Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

CODIGO:
HSB-04

ELABORACIÓN DE PRUEBAS Y ENSAYOS

Edición No. 01

Pág. 1 de 7

1. PROPÓSITO

Controlar los productos que se generan, identificando las faltas o fallas que existan, mediante 4 etapas de pruebas.

2. ALCANCE

Este proceso aplica para todos los productos finales de la organización.

3. RESPONSABLE DEL PROCESO

Administrador del proyecto

4. RECURSOS

- Equipo de programación
- Clientes de prueba
- Equipos de cómputo personales
- Oficina compartida

5. DEFINICIONES

- **Beta testers:** Clientes de la organización que han accedido a ser parte de una de las etapas de prueba de los productos.
- **Fallas:** Errores en las herramientas disponibles al usuario que necesitan ser corregidas.
- **Faltas:** Herramientas solicitadas por el cliente que no han sido realizadas en el desarrollo.
- **Observación por defecto del producto:** Observaciones provenientes de errores o faltas en el diseño en sí.

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

CODIGO:
HSB-04

ELABORACIÓN DE PRUEBAS Y ENSAYOS

Edición No. 01

Pág. 2 de 7

- **Observaciones no competentes al producto o al usuario:** Observaciones que solicitan mejoras o correcciones que o fueron predeterminadas al alcance del producto.
- **Observaciones por fallo del usuario:** Observaciones provenientes del uso incorrecto del prototipo por parte del usuario.

6. POLÍTICAS

- Todas las solicitudes de prueba a clientes potenciales o clientes de prueba deberán adjuntar un listado de todas las herramientas adicionales de los productos para que ser probadas.
- Todos los empleados deberán probar las herramientas de los productos en casos no convencionales.
- Cualquier observación realizada por el cliente externo deberá clasificarse como: observación por defecto del producto, observación por fallo del usuario u observación no competente al producto ni al usuario.

7. INDICADORES

Código	IND-01_HSB-04					
Nombre	Observaciones en pruebas de funcionalidad					
Tipo de medida	Eficacia					
Tipo de relación	Porcentaje					
Descripción	Mide la cantidad de observaciones totales realizadas que son de naturaleza de defecto del producto y solucionables por el equipo de desarrollo detectadas en las pruebas de funcionalidad.					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
(Observaciones por defecto del producto/ Observaciones Totales) * 100	Por proyecto	Negativo	15%	30%	Administrador del proyecto	Alta Dirección

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

**CODIGO:
HSB-04**

ELABORACIÓN DE PRUEBAS Y ENSAYOS

Edición No. 01
Pág. 3 de 7

Código	IND-02_HSB-04					
Nombre	Observaciones en pruebas de simplicidad					
Tipo de medida	Eficacia					
Tipo de relación	Porcentaje					
Descripción	Mide la cantidad de observaciones en las pruebas de simplicidad en relación a las observaciones totales por defecto del producto.					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
(Observaciones de simplicidad/ Observaciones por defecto del producto) * 100	Por proyecto	Negativo	12%	20%	Administrador del proyecto	Alta Dirección

Código	IND-03_HSB-04					
Nombre	Observaciones en pruebas Beta externas					
Tipo de medida	Eficacia					
Tipo de relación	Porcentaje					
Descripción	Mide la cantidad de observaciones en las pruebas Beta externas en relación a las observaciones totales por defecto del producto.					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
(Observaciones externas Beta/ Observaciones por defecto del producto) * 100	Por proyecto	Negativo	10%	15%	Administrador del proyecto	Alta Dirección

Código	IND-04_HSB-04					
Nombre	Observaciones en pruebas Beta de pre-distribución					
Tipo de medida	Eficacia					
Tipo de relación	Porcentaje					
Descripción	Mide las observaciones en las pruebas Beta en relación a los defectos.					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
(Observaciones pre-distribución/ Observaciones por defecto del producto) * 100	Por proyecto	Negativo	5%	8%	Administrador del proyecto	Alta Dirección

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

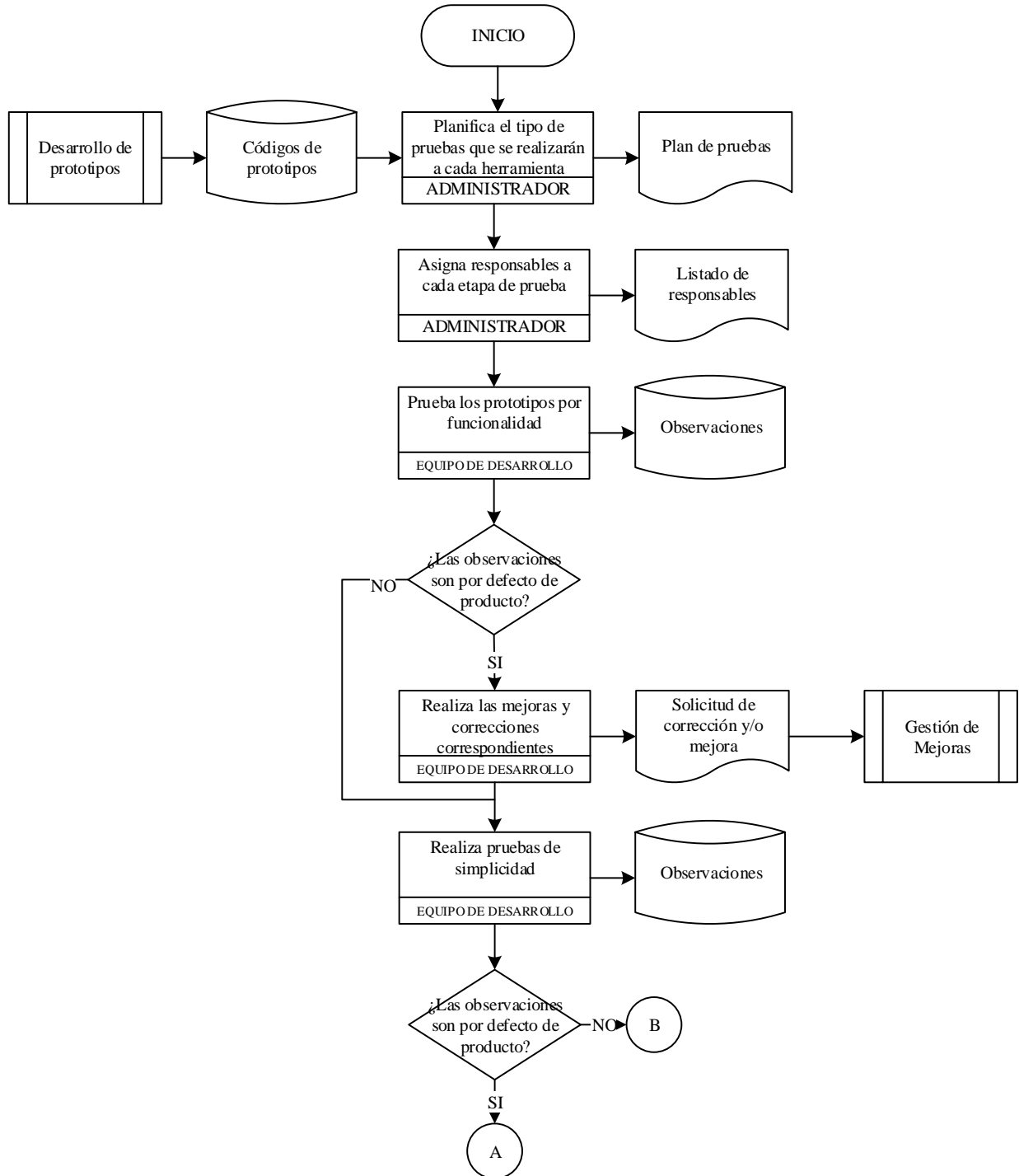
**CODIGO:
HSB-04**

ELABORACIÓN DE PRUEBAS Y ENSAYOS

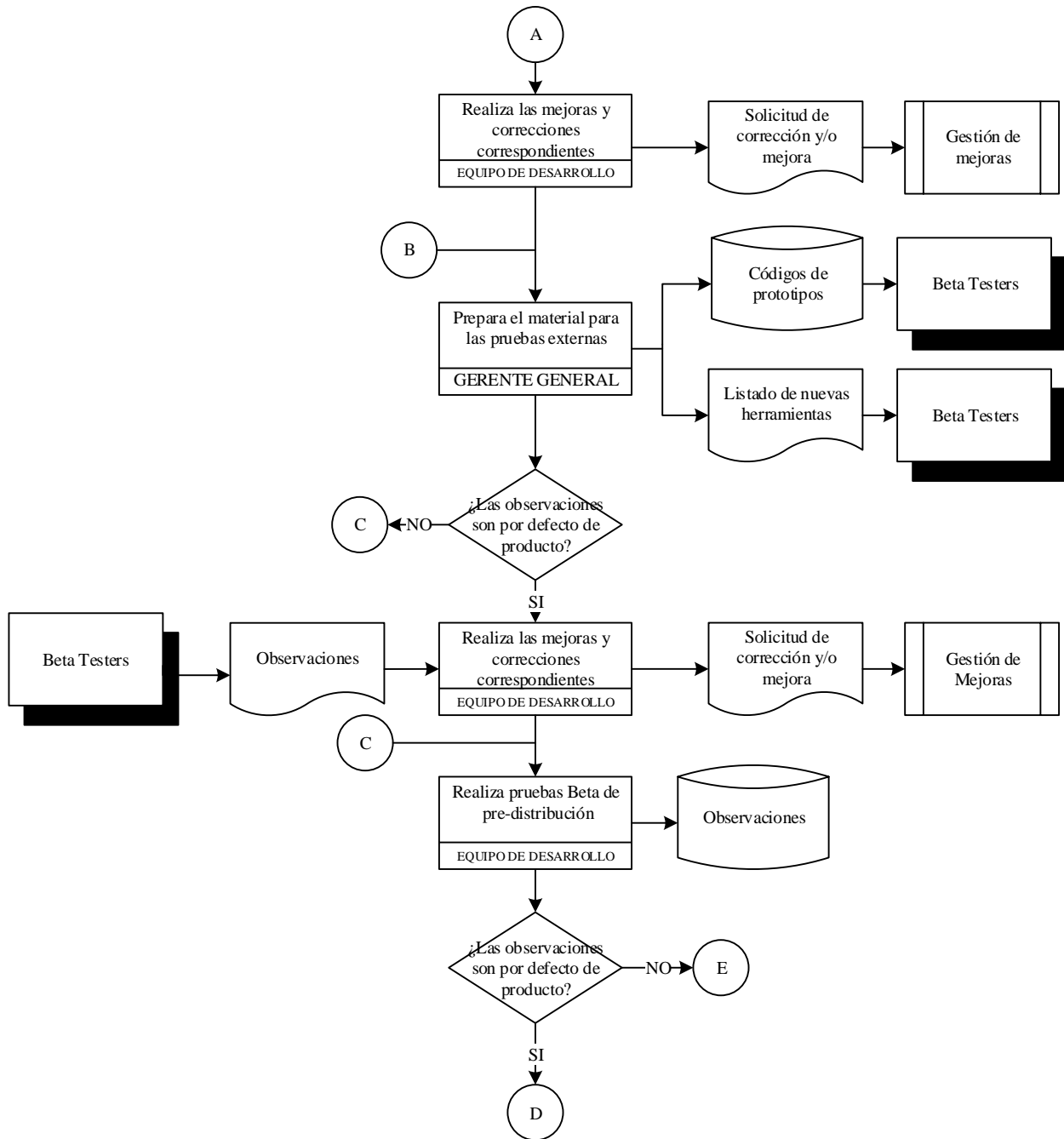
Edición No. 01**Pág. 4 de 7****8. INFORMACIÓN DOCUMENTADA**

Código	Origen	Nombre	Soporte	Conservación	Disposición
R01_HS B-04	Interno	Plan de pruebas	Digital	5 años	Sacar respaldo
R02_HS B-04	Interno	Listado de responsables	Impreso	Duración del proyecto	Enviar a archivo pasivo
R03_HS B-04	Interno	Base de observaciones	Digital	5 años	Sacar respaldo
R04_HS B-04	Interno	Listado de nuevas herramientas	Digital	3 años	Sacar respaldo
R05_HS B-04	Interno	Solicitud de licencias para instaladores	Digital	3 años	Sacar respaldo
R06_HS B-04	Interno	Solicitud de corrección y/o mejora	Digital	3 años	Sacar respaldo

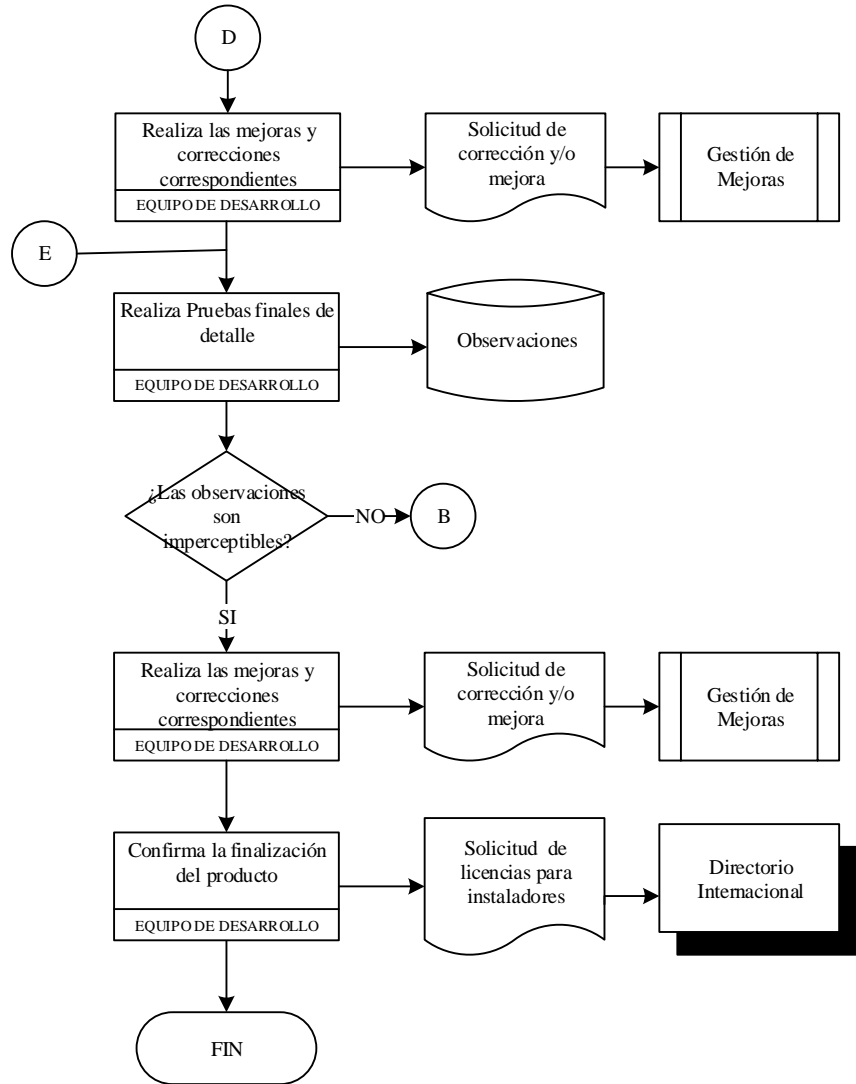
Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

9. DIAGRAMA DE FLUJO


Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:



Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:



Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

CODIGO:
HSB-05

GESTIÓN DE MEJORAS

Edición No. 01

Pág. 1 de 6

1. PROPÓSITO

Asegurar la conformidad de proyectos actuales y anteriores, mediante la categorización de observaciones y el análisis de sus causas.

2. ALCANCE

Este proceso aplica para todos los productos finales de la organización.

3. RESPONSABLE DEL PROCESO

Administrador del proyecto

4. RECURSOS

- Equipo de programación
- Observaciones de productos pasados y presentes
- Equipos de cómputo personales
- Oficina compartida

5. DEFINICIONES

- **Acciones Correctivas:** Acción tomada para eliminar las causas de no conformidades detectadas.
- **Agrupación de observaciones por causas probables:** Segmentación de las observaciones por defecto del producto en las que se agrupan distintas de ellas para una sola causa común.
- **Corrección:** Acción que permite reformar una falla o eliminar una no conformidad.
- **Fallas:** Errores en las herramientas disponibles al usuario que necesitan ser corregidas.

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

CODIGO:
HSB-05

GESTIÓN DE MEJORAS

Edición No. 01

Pág. 2 de 6

- **Faltas:** Herramientas solicitadas por el cliente que no han sido realizadas en el desarrollo.
- **Mejora:** Acción que permite eliminar una falta y aumentar la capacidad de cumplir requisitos.
- **Observación por defecto del producto:** Observaciones provenientes de errores o faltas en el diseño en sí.
- **Observaciones de productos pasados:** Observaciones que los clientes realizaron una vez que un producto fue comercializado, impidiendo la corrección o mejora de la versión disponible.
- **Observaciones de productos presentes:** Observaciones que los clientes realizan a lo largo del desarrollo del producto, antes de finalizarlo y comercializarlo.
- **Observaciones no competentes al producto o al usuario:** Observaciones que solicitan mejoras o correcciones que o fueron predeterminadas al alcance del producto.
- **Observaciones por fallo del usuario:** Observaciones provenientes del uso incorrecto del prototipo por parte del usuario.

6. POLÍTICAS

- Deben generarse una base de soluciones a las observaciones de proyectos pasados.
- Las observaciones de proyectos pasados deben ser analizadas cada semana si la empresa no tiene un proyecto nuevo en el que trabajar en el presente.
- Las observaciones no competentes al producto o al usuario deben considerarse para mejorar el alcance de proyectos futuros.
- Las observaciones por fallo del usuario deben ser comunicadas en un periodo de 36 horas.
- Todo el equipo de desarrollo debe participar la reducción de causas probables.

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

CODIGO:
HSB-05 **GESTIÓN DE MEJORAS**
Edición No. 01
Pág. 3 de 6
7. INDICADORES

Código	IND-01_HSB-05					
Nombre	Correcciones realizadas					
Tipo de medida	Eficacia					
Tipo de relación	Porcentaje					
Descripción	Mide la cantidad de correcciones partiendo de las fallas detectadas.					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
(Correcciones realizadas/ Total de fallas detectadas)*100	Por proyecto	Positivo	95%	100%	Administrador del proyecto	Gerente General

Código	IND-02_HSB-05					
Nombre	Mejoras realizadas					
Tipo de medida	Eficacia					
Tipo de relación	Porcentaje					
Descripción	Mide la cantidad de mejoras partiendo del número de faltas detectadas.					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
(Mejoras realizadas/ Total de faltas detectadas) *100	Por proyecto	Positivo	90%	100%	Administrador del proyecto	Alta Dirección

Código	IND-03_HSB-05					
Nombre	Cumplimiento de acciones correctivas					
Tipo de medida	Eficacia					
Tipo de relación	Porcentaje					
Descripción	Mide la cantidad de acciones correctivas realizadas partiendo del número de actividades planificadas.					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
(Actividades realizadas/ Actividades planificadas) *100	Por proyecto	Positivo	85%	95%	Administrador del proyecto	Alta Dirección

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

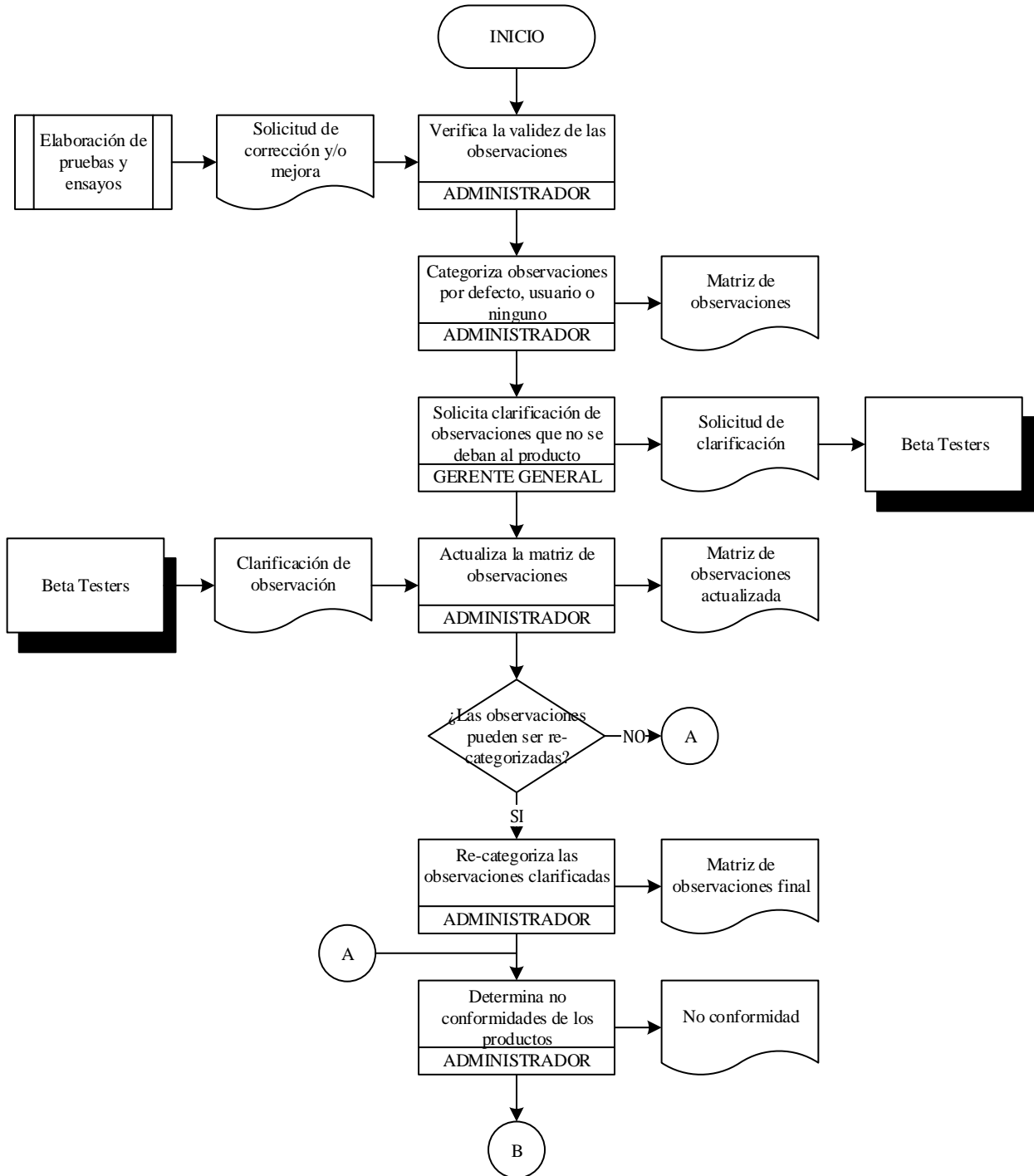
**CODIGO:
HSB-05**

GESTIÓN DE MEJORAS

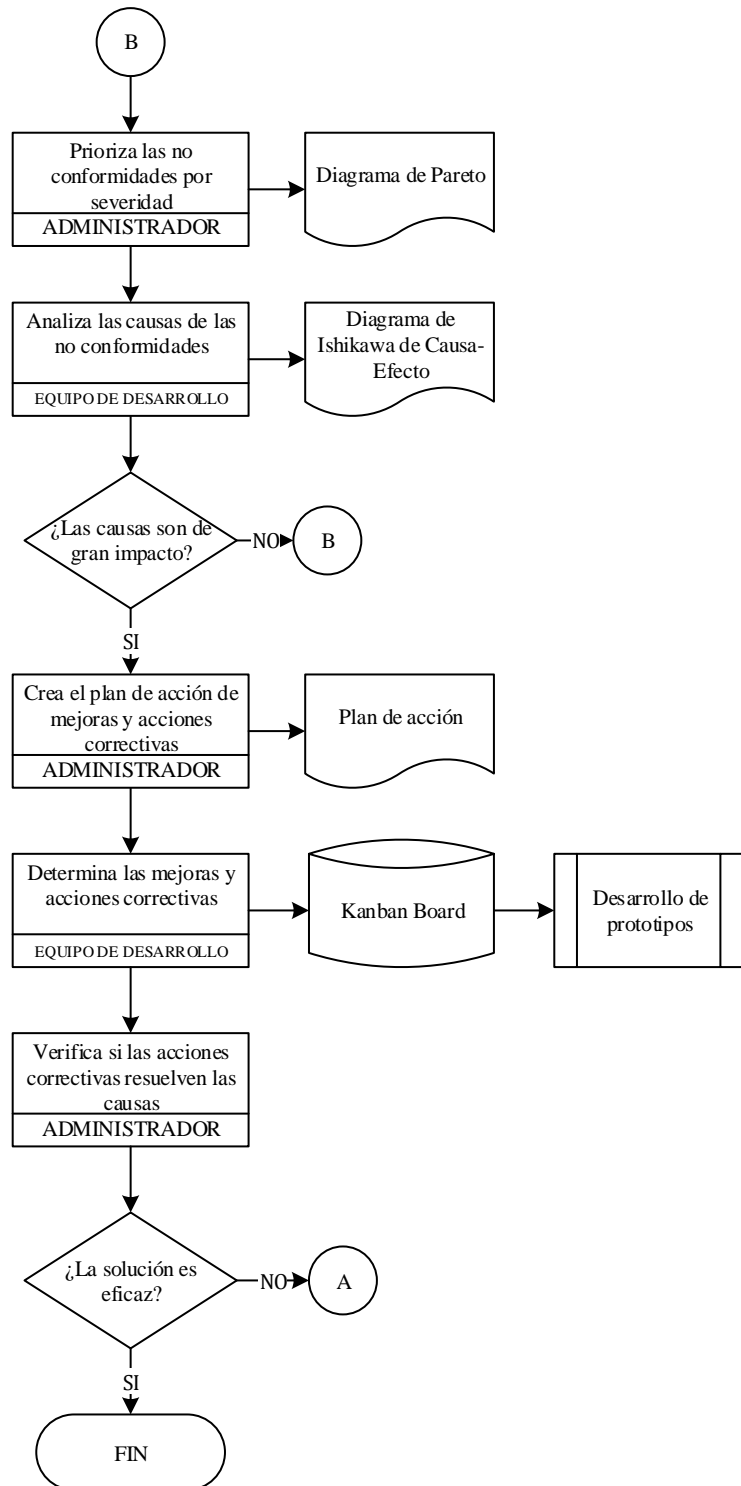
Edición No. 01**Pág. 4 de 6****8. INFORMACIÓN DOCUMENTADA**

Código	Origen	Nombre	Soporte	Conservación	Disposición
R06_HS B-04	Interno	Solicitud de corrección y/o mejora	Digital	3 años	Sacar Respaldo
R01_HS B-05	Interno	Matriz de observaciones	Digital	5 años	Sacar Respaldo
R02_HS B-05	Interno	Solicitud de clarificación	Digital	2 años	Sacar Respaldo
R03_HS B-05	Externo	Clarificación de observación	Digital	2 años	Sacar Respaldo
R04_HS B-05	Interno	No conformidad	Digital	5 años	Sacar Respaldo
R05_HS B-05	Interno	Diagrama de Pareto	Digital	5 años	Sacar Respaldo
R06_HS B-05	Interno	Diagrama de Ishikawa de Causa-Efecto	Digital	5 años	Sacar Respaldo
R07_HS B-05	Interno	Plan de acción	Digital	5 años	Sacar Respaldo
R02_HS B-03	Interno	Kanban Board	Digital	Vigencia del Proyecto	Sacar Respaldo

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

9. DIAGRAMA DE FLUJO


Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:



Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

CODIGO:
HSB-06

OPERACIÓN Y MANTENIMIENTO

Edición No. 01

Pág. 1 de 4

1. PROPÓSITO

Distribuir los productos finales y brindar soporte a los usuarios con estándares de calidad, en base a las observaciones de los mismos, mediante permisos planteados acorde a los precios de cada producto y el alcance que desea el usuario.

2. ALCANCE

Este procedimiento aplica para todos los productos finales de la organización.

3. RESPONSABLE DEL PROCESO

Gerente general

4. RECURSOS

- Equipo de programación
- Licencias de creación de instaladores
- Listas de precios por categoría de cliente
- Lista de los paquetes de los productos por categoría de cliente
- Equipos de cómputo personales
- Oficina compartida

5. DEFINICIONES

- **Categoría de cliente:** Categoría del cliente dentro de la empresa que le permite mayor o menor acceso a las herramientas de cada producto.
- **Licencias de creación de instaladores:** Licencias de distintos programas que permiten crear instaladores de productos.

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

CODIGO:
HSB-06

OPERACIÓN Y MANTENIMIENTO

Edición No. 01

Pág. 2 de 4

- **Listas de precios:** Precios asignados a cada categoría de cliente en base al precio que pagan por cada paquete de productos.
- **Paquetes de productos:** Integraciones de productos en base a la categoría de cada cliente.

6. POLÍTICAS

- Las listas de precios y paquetes de productos serán establecidos únicamente por la dirección internacional de la empresa.
- Las listas de precios y paquetes de productos deberán ser compartidos en el sitio web de la empresa.
- Los permisos de los clientes deben ser controlados cada mes.
- Todas las observaciones provenientes posteriormente al lanzamiento del producto serán almacenadas para ser usadas en proyectos futuros.

7. INDICADORES

Código	IND-01_HSB-06					
Nombre	Soporte post-venta					
Tipo de medida	Eficacia					
Tipo de relación	Tasa					
Descripción	Indica el número de veces que se otorga soporte a los usuarios tras el lanzamiento del producto					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
Nº solicitudes de soporte del usuario/ Nº productos del proyecto	Por proyecto	Negativo	50 solicitudes/proyecto	150 solicitudes/proyecto	Gerente General	Alta Dirección

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

CODIGO:
HSB-06

OPERACIÓN Y MANTENIMIENTO

Edición No. 01

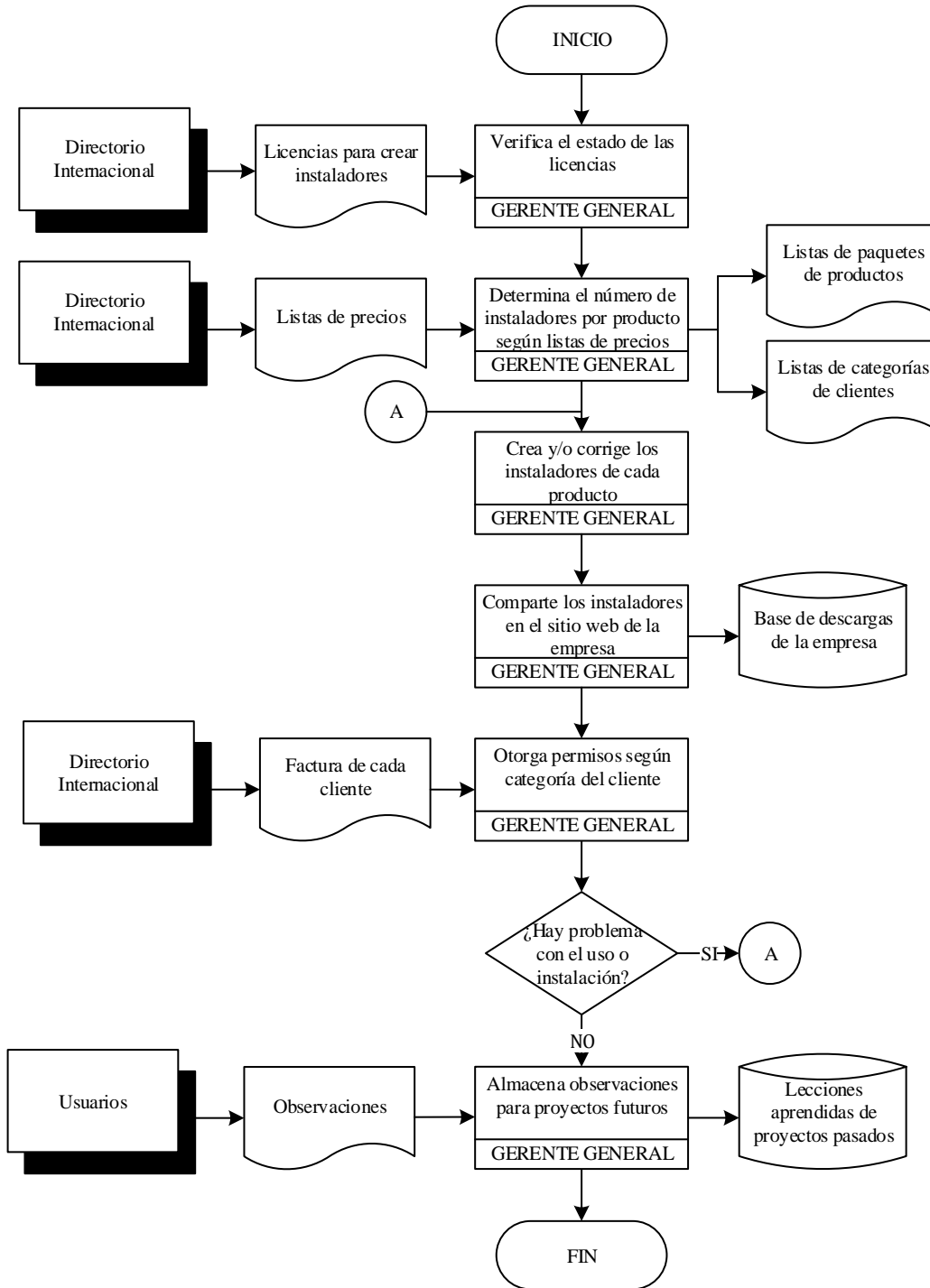
Pág. 3 de 4

Código	IND-02_HSB-06					
Nombre	Observaciones post-venta					
Tipo de medida	Eficacia					
Tipo de relación	Tasa					
Descripción	Mide en relación por cada producto, el número de observaciones que se generan tras el lanzamiento del mismo					
Fórmula	Frecuencia	Sentido	LI	LS	Responsable de Medición	Responsable de Análisis
N° observaciones/ N° de productos del proyecto	Por proyecto	Negativo	30 observaciones/ proyecto	80 observaciones/ proyecto	Gerente General	Alta Dirección

8. INFORMACIÓN DOCUMENTADA

Código	Origen	Nombre	Soporte	Conservación	Disposición
R01_HS B-06	Externo	Licencias para instaladores	Digital	5 años	Sacar respaldo
R02_HS B-06	Externo	Listas de precios	Digital	5 años	Sacar respaldo
R03_HS B-06	Interno	Listas de paquetes de productos	Digital	5 años	Sacar respaldo
R04_HS B-06	Interno	Listas de categorías de clientes	Digital	5 años	Sacar respaldo
R05_HS B-06	Interno	Base de descargas de la empresa	Digital	10 años	Sacar respaldo
R06_HS B-06	Externo	Factura de cada cliente	Digital e impreso	5 años	Sacar respaldo/ Enviar a archivo pasivo
R07_HS B-06	Externo	Observaciones	Digital o impreso	5 años	Sacar respaldo
R08_HS B-06	Interno	Base de observaciones	Digital	10 años	Sacar respaldo

Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha:

9. DIAGRAMA DE FLUJO


Responsable del Proceso	Jefe de Unidad
Fecha:	Fecha: