Universidad de las Ciencias Informáticas
Facultad 10

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Título: Propuesta de una Línea de Producción de Software para el proyecto Nova Linux.

Autores: Susel Peña Cruz
Anaisa Núñez Rizo

Tutor: Ing. Yoandy Rodríguez Martínez

Ciudad de la Habana
Junio/ 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Susel Peña Cruz

_____

Firma del Autor

Anaisa Núñez Rizo

_____

Firma del Autor

Ing. Yoandy Rodríguez Martínez

_____

Firma del Tutor

DATOS DE CONTACTO:

El tutor de la presente investigación: Yoandy Rodríguez Martínez, es graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas durante el año 2006. Incorporándose posteriormente a su servicio social en el propio centro. Ha cursado el diplomado de Profesor en la Educación Superior en dicho centro, y presentado los ejercicios que le otorgaron la categoría docente de Profesor Instructor.

Agradecimientos:

A la Revolución Cubana y al compañero Fidel por brindarnos la valiosa posibilidad de forjarnos como profesionales de bien.

A nuestros padres por su apoyo incondicional y la confianza depositada.

A nuestra familia, tíos, primos, abuelos, a los presentes y a los que ya no nos acompañan, por ser parte indiscutible de nuestro soporte.

Al tutor de nuestra investigación, el Ingeniero Yoandy Rodríguez Martínez por ser un guía incondicional.

A nuestros compañeros de tantas y tantas cosas.

A todos aquellos que nos han acompañado y han influido en nuestra preparación en estos 5 años.

Dedicatoria:

Susel:

A mi madre por darme la vida y confiar en mí, a mi padre por inculcarme la disciplina que ha me hecho una profesional revolucionaria, a ambos por brindarme el apoyo y el cariño que permitieron hacer este sueño realidad.

A Hilda, Evelyn y Dioscórides por estimularme a seguir siempre el buen camino y por su apoyo incondicional, gracias por todo.

A mi abuela que aunque ya no está con su cariño supo ser pilar fundamental en mi educación.

A mi futuro esposo Raidel por soportarme y hacer mi vida más fácil todos estos años.

Anaisa:

A mi mamá, por su constancia, su amor y sacrificio. Por estar siempre a mi lado, apoyándome en todas mis decisiones. Por ser como es.

Resumen:

En el presente trabajo investigativo se abordan algunos problemas que presenta la producción de software en la Universidad de las Ciencias Informáticas y una posible vía de solución para dichos inconvenientes; de forma específica se toma como ejemplo piloto el proyecto Nova Linux.

La falta de calidad en el producto que elabora dicho Grupo de Desarrollo está dada por la poca calidad del proceso de construcción empleado; una solución eficiente para este tipo de contratiempo es la adopción de la filosofía de una Línea de producción, por lo que el objetivo principal del trabajo es realizar una propuesta de este tipo para el proyecto que compete.

Para dar solución al problema que se presentase plantean un grupo de preguntas científicas relacionadas con la estrategia a seguir para la gestión y optimización de los diversos procesos que se ejecutan durante la construcción de las diversas personalizaciones de Nova.

Durante la investigación se emplea lo que se conoce como metodología de gestión por procesos, en pos de identificar los mismos durante el desarrollo de las personalizaciones, además de los diversos problemas que se presentan y sus oportunidades de mejora. De esta manera queda estructurado un nuevo proceso de liberación que permite agilizar la producción y que a su vez organiza de forma estricta el desarrollo.

El resultado del presente trabajo proporciona un grupo de mejoras al proceso de desarrollo del proyecto Nova, debido fundamentalmente a la agilización del proceso de construcción del mismo, así como la pertinente organización de su estructura productiva.

Palabras Clave: Línea de Producción, software, Ingeniería de Liberaciones, Nova Linux.

Índice:

Introducción:

El *desarrollo de software* ha devenido como respuesta a las necesidades de los usuarios o como una meta de mercado transformando al software en un producto. De forma general, el desarrollo de software no es más que el proceso de crear software. Hoy en día, para denominar este proceso se emplea el término *ingeniería de software*, que es conceptualizado como el conjunto de técnicas específicas destinadas a la producción de un programa de computadora, más allá de la mera actividad de la programación. De esta manera y aunque ambos conceptos están estrechamente relacionados, la ingeniería implica niveles de rigor y prueba de procesos que no son apropiados para todo tipo de desarrollo de software.[1]

Según estudios realizados uno de los más grandes problemas que se presenta durante el desarrollo de software es que se le presta muy poca atención a la discusión del problema a resolver. En general los desarrolladores se centran más en las posibles soluciones, dejando inexplorados los diversos matices del problema, por ende, quedan en el olvido vías de solución que por sí solas son más óptimas; dificultando a la larga el proceso de desarrollo.[2]

La introducción de la filosofía de una ***línea de producción*** en la construcción de software puede ser una solución eficiente en pos de evitar algunos de estos problemas, sin menospreciar por ello los métodos que la ingeniería de software brinda de forma efectiva en estos casos.

Sentando las bases en el estudio de un grupo de conceptos se puede entender como *línea de producción* (LP), a un grupo de operaciones o instrucciones ordenadas, establecidas con anterioridad, a través de las cuales se someten uno o varios materiales para lograr de forma eficiente un producto final con calidad. A través del uso de las LP se garantiza la consecutividad de los procesos implicados en la tarea a realizar, por lo que el gasto de recursos durante la transición es mínimo o casi nulo, constituyendo esto una de sus principales ventajas. Es necesario aclarar que las mismas son aplicadas a cualquier tipo de producto, en el caso específico de esta investigación la propuesta está orientada al software; de ahí el término *línea de producción de software* (LPS).

En la Universidad de las Ciencias Informáticas (UCI), surgida al fragor de la Batalla de Ideas se desarrollan programas de computadora en pos de incluir a Cuba en el mercado internacional del software. Después de diversos estudios realizados respecto a la influencia de los problemas que se presentan en esta rama, tanto en el mundo como en el resto del país; se determinó que es necesario que el centro comience a generar productos y no soluciones de software, lo cual constituye la

tendencia general en la producción.

En el polo productivo de Software Libre de dicha universidad se desarrolla un sistema operativo con el objetivo de facilitar la migración para algunas de las ramas de la sociedad. El grupo de desarrollo Nova se encarga de desarrollar Nova Linux, que es una personalización de Linux, basada en Gentoo, sistema operativo que permite compilar todo su software para obtener un sistema exactamente ajustado a la medida que se desea. Debido fundamentalmente a la necesidad de convertir las soluciones en productos de software se hace indispensable la realización de una transformación en el proceso de desarrollo que se ejecuta dentro del proyecto. Una de las opciones existentes para llevar a cabo esta transformación dentro del grupo de desarrollo Nova es el establecimiento de una LPS, para ello se hace necesario encontrar los parámetros que permitan su implantación para cualquiera de las personalizaciones que se desarrollan; teniendo en cuenta que el proceso de desarrollo de dicho proyecto carece de una descripción y optimización adecuada para lograr los nuevos objetivos propuestos por el centro.

Las diversas personalizaciones del Sistema Operativo Nova Linux sientan las bases de su desarrollo en la *Ingeniería de Liberaciones (Release Engineering)*. Este término hace referencia a una de las subdisciplinas de la IS, la cuál se encarga de los procesos de compilación, ensamblamiento y entrega del código fuente, tanto en productos terminados, como en los diversos componentes del software. Además incluye el proceso de control de versiones, entre otros elementos.[3]

Los procesos que se realizan para la confección de Nova Linux son realizados de forma manual y absorben gran cantidad de recursos humanos, lo que provoca demasiada carga laboral en los desarrolladores, afectando a la larga la calidad del producto final.

A partir de esta situación se plantea el siguiente **problema**:

- El proceso de desarrollo y gestión de software en el proyecto Nova Linux carece de un orden establecido para las operaciones que se realizan, así como de una descripción y optimización adecuada para las nuevas pautas establecidas en la Universidad de las Ciencias Informáticas respecto a la creación de productos de software.

El **objeto de estudio** de esta investigación es fundamentalmente:

- El proceso de desarrollo y gestión de software en Nova Linux.

Dicho objeto deriva en un **campo de acción** que se puede denominar como:

- La ingeniería de liberaciones (release engineering) en Nova.

El principal **objetivo** planteado para el logro de resultados en la presente investigación es:

- Reestructurar el proceso de desarrollo actual de Nova en una línea de producción de software.

Al problema planteado se le dará solución a través de las siguientes **preguntas científicas**:

- ¿Qué operaciones se deben tener en cuenta para transformar la actual metodología de desarrollo de Nova en una línea de producción de software?
- ¿Qué orden debe establecerse para las operaciones a realizar durante la ejecución de la línea de producción?
- ¿Qué procesos de los que se realizan actualmente en Nova pueden ser optimizados?

Para lograr el objetivo propuesto se han trazado una serie de **tareas** a realizar durante la investigación:

- Sistematizar los procesos que se están desarrollando de forma manual en el proyecto Nova Linux.
- Sistematizar las vías de optimización para la carga de procesos que se están realizando en el proyecto Nova.
- Proponer una línea de producción para Nova Linux.

En la presente investigación el uso de **métodos científicos** tales como la **inducción-deducción** posibilita a partir del estudio y la búsqueda de conceptos, arribar a definiciones y proposiciones generales de la materia a abordar tales como la metodología y el modelo de calidad que son empleados en este estudio; de la misma forma permite inferir a partir de conocimientos y conceptos generales casos mas específicos y particulares como es el caso de las líneas de producción. Un complemento fundamental dentro de estos métodos es el **histórico-lógico,** dicho método permite el análisis de la trayectoria completa del fenómeno, en este caso el proyecto Nova, además de analizar su condicionamiento a los diferentes períodos de la historia, revelando las etapas principales de su desenvolvimiento y las conexiones históricas fundamentales. También pone de manifiesto la lógica interna del desarrollo del proyecto, además de expresar en forma teórica su esencia. De manera general permite unir el estudio de la estructura del proceso de desarrollo y gestión de software en Nova Linux con su concepción histórica. La necesidad de buscar las contradicciones existentes dentro de

dicho proyecto, justifica el empleo del **método dialéctico**; y de esta forma poder explicar los cambios cualitativos que se producen en el sistema y revelar las relaciones contradictorias esenciales existentes. Este tipo de método permite estudiar a fondo las características del proceso de desarrollo y gestión de software en Nova Linux, apoyándose en el proceso de análisis y síntesis.

Para lograr buenos resultados se hace imprescindible el uso de técnicas que apoyan a los métodos anteriormente mencionados como la **entrevista** y la **observación**. En el primer caso, la misma proporciona el conocimiento de especificaciones y detalles tanto de la Ingeniería de Liberaciones como del proceso de desarrollo y gestión de software del proyecto especificado; a través de conversaciones planificadas con diferentes personas involucradas en ambos procesos y con amplios conocimientos de los mismos. En el caso de la observación y al ser uno de los autores de este trabajo integrante del proyecto productivo Nova, la misma facilita la apreciación de procesos presentes en los diversos grupos de desarrollo y la detección de presuntos problemas a resolver.

Este trabajo cuenta con 3 capítulos: el **Capítulo 1** hace referencia a la Fundamentación teórica de la investigación, o sea, contiene el estado actual de las líneas de producción de software en el mundo, haciendo alusión a las diversas herramientas que se emplean para un buen funcionamiento de este concepto en la producción de software. Además se abordan los problemas existentes en este mercado en Cuba y específicamente en la UCI, tomando como punto de referencia al proyecto Nova; y como pueden las LPS solucionar las carencias existentes. Un apéndice importante dentro de este capítulo lo constituye el análisis en profundidad de un grupo de metodologías de desarrollo de software, principalmente ágiles, así como diversos modelos de calidad que contribuyen con la aplicación de este concepto.

En el transcurso del **Capítulo 2** se realiza la descripción y el análisis del proceso de desarrollo del proyecto Nova Linux, empleando la metodología de Gestión por procesos, aunque no en su totalidad. Se realiza la identificación de los procesos fundamentales que se llevan a cabo en el desarrollo de las diversas personalizaciones, se describen los mismos y se diagraman, facilitando una posterior identificación de los problemas que se presentan en el proceso y las oportunidades de mejora. Además de establecer una propuesta de la metodología de desarrollo que se va a implantar en pos de transformarla en una línea de producción, así como una propuesta del modelo de calidad seleccionado con este mismo fin.

Durante el **Capítulo 3** se efectúan las propuestas fundamentales de los procesos que durante el estudio reportan problemas de funcionamiento, o sea, se proponen las vías de optimización de los

mismos, siguiendo en todo momento la filosofía de gestión por procesos. Además de realizar la propuesta que constituye el principal objetivo de la presente investigación, la Línea de Producción de Software.

1

Capítulo 1. Fundamentación Teórica

1.1 Introducción.

Hoy en día el desarrollo de software ha alcanzado niveles insospechados; la causa es simple, las Tecnologías de la Informática y las Comunicaciones (TIC´s) están en pleno auge, se vive en una era guiada por las computadoras; por ello la industria del software se expande día a día, mejorando sus vías de solución a los problemas que presenta la sociedad.

En la década de los noventa se empieza a reflexionar sobre la necesidad de un cambio en las formas y medios de producir software. Los ingenieros quieren evolucionar de una producción artesanal a una producción industrial. En estos momentos es cuando empiezan a aparecer en el mercado las herramientas y metodologías para el desarrollo de software; desde entonces, las empresas de servicios han ido avanzando en sus métodos y sistemas de producción. En esta línea se consolida el enfoque y la estructura a la prestación de los servicios que ofrecen como vía principal de solución los modelos de *factoría de software.*

Según estudios realizados de varios conceptos, los autores de la presente investigación llegan al concenso de que la *factoría de software* no es más que una institución u organización estructurada cuyo fin es el desarrollo de software, en la cuál los requisitos del producto son procesados hasta lograr un producto final; es necesario tener en cuenta que este concepto supone la provisión de soluciones basadas en las TIC, producidas con esquemas y métodos industriales, sin entrar en conflicto con las necesidades específicas de cada cliente, ya que aporta respuestas a medida mediante un proceso industrial.

El desarrollo a través de una factoría aporta al cliente beneficios tales como:

- Mayor garantía respecto al resultado final.
- Reducción del plazo y coste del producto.
- La fabricación a través de componentes ya probados permite una alta reutilización, que redunda directamente en el plazo y en el coste de los sistemas, aplicaciones o productos a obtener.
- El cliente dispone de una visión total del proceso, ya que existen sistemas e indicadores que le garantizan el seguimiento de la provisión del servicio con una sensación de control y en un ambiente de transparencia, desde la generación del pedido hasta la entrega del servicio.[4]

Las factorías de software han llegado como una eficiente solución a algunos de los problemas existentes en el desarrollo de software en el mundo. En la actualidad existen un grupo de importantes

empresas que aplican este concepto; la multinacional IBM es una de ellas arrojando excelentes resultados. Otra de las empresas exitosas en este campo es Ibermática, la cuál sigue al pie de la letra los aspectos que propone este concepto, el cual le reporta beneficios invaluables, al igual que INDRA, empresa española de software. Además de las anteriormente mencionadas se podrían tener en cuenta otras muchas empresas que aplican ventajosamente esta idea. Para lograr estos factores de éxito la producción industrial ha de sintonizar con la personalización de las soluciones. Este concepto incorpora el valor añadido de los sistemas industriales, manteniendo las ventajas que suponen las soluciones a la medida de las necesidades del cliente.

1.2 Líneas de Producción.

Las factorías de software han sido una alternativa eficiente a los problemas existentes en el desarrollo de este tipo de producto; aún así se ha abogado por una vía que permita además la informatización de los procesos de desarrollo, en pos de agilizarlos manteniendo la calidad que el software requiere, de ahí el surgimiento de las Líneas de Producción de Software como nuevo concepto.

Una de las soluciones más eficientes existentes en el empleo de las LPS es GeneXus, una herramienta inteligente, desarrollada por Artech, cuyo objetivo es asistir al analista y a los usuarios en todo el ciclo de vida de las aplicaciones.

La idea básica de GeneXus es automatizar todo aquello que es automatizable: normalización de los datos y diseño, generación y mantenimiento de la base de datos y de los programas de aplicación. De esta manera se evita que el analista deba dedicarse a tareas rutinarias y tediosas, permitiéndole poner toda su atención en aquello que nunca un programa podrá hacer: entender los problemas del usuario. Como un subproducto, GeneXus ofrece una documentación rigurosa, autosuficiente y permanentemente actualizada.

Artech es una empresa líder en herramientas de desarrollo de software basadas en gestión automática del conocimiento. La buena reputación de dicha empresa ha provocado que más de 5,000 clientes en todo el mundo utilicen más de 50.000 licencias GeneXus para crear e integrar aplicaciones de misión crítica que se adaptan fácilmente a los cambios y exigencias del negocio. GeneXus hace posible que las empresas usen su conocimiento (know-how) exclusivo en las plataformas tecnológicas líderes del mercado.

Este software desarrollado por Artech se ha transformado en un Referente de Desarrollo en

Latinoamérica.[5] La primera versión de este producto surge por idea de Nicolás Jodal en 1989, ya como versión de mercado. Su evolución tecnológica ha permitido a los clientes de la herramienta migrar y expandir aplicaciones críticas desde entornos centralizados como AS/400 a entornos cliente servidor primero y ahora entornos distribuidos basados en Java o .NET, y también a dispositivos móviles.

En los últimos 20 años la complejidad de las aplicaciones corporativas ha crecido enormemente, digamos un 2000%, y sin embargo la productividad de los lenguajes de programación ha aumentado sólo un 200 o 300% (por decir una cifra). El hecho es que el proceso de creación de aplicaciones complejas ya no se adapta a la realidad de los clientes corporativos, de ahí que Genexus constituya una solución a este problema, ya que su objetivo principal es el desarrollo basado en el conocimiento, y no en la programación.

Otro de los innumerables ejemplos de empresas que han adoptado la estrategia de las LPS en pos de mejorar sus servicios y a la vez sus ganancias es la Fábrica de Software HEINSOHN, la cual aplica efectivamente un modelo formal de construcción de Software basado en líneas de producción. La estrategia técnica, soportada en los modelos probados y estandarizados CMMI (Capability Maturity Model Integration) e ISO 9000, garantiza un alto nivel de calidad al servicio del desarrollo de proyectos. Otro de los ejemplos que se pueden citar a este respecto es la multinacional Capgemini; apenas dos años después de su llegada a Asturias, la multinacional ha logrado situar a la región como un referente mundial en el desarrollo de software. Su factoría langreana se acaba de consolidar como una de las más eficientes de Europa tras obtener la máxima puntuación en el modelo de mejora continua CMMI (Capability Maturity Model Integration). En la actualidad, la factoría langreana proporciona servicios de desarrollo y mantenimiento de software mediante líneas de producción totalmente industrializadas y cuenta con un equipo de alrededor 270 profesionales. Esta empresa apuesta por Asturias para crear un empleo de calidad y especializado que repercuta positivamente en el servicio prestado a los clientes; el grupo prevé un desembolso global para la planta langreana superior a los 20 millones de euros.[6] Debido al evidente auge de esta tendencia se puede mencionar otro ejemplo: La empresa tecnológica Sadiel instalará "de manera inminente" en la Bahía de Cádiz una filial de construcción de software, Software Factory, que prevé alcanzar en el horizonte de 2010 más de 400 trabajadores. Cada una de las líneas de negocio cuenta con un conjunto de actuaciones específicas para cumplir sus ambiciosos objetivos, entre las que se encuentra un fuerte incremento de la inversión en nuevos productos y soluciones (I+D+i). Igualmente está prevista la creación de líneas de producción especializadas para incrementar la excelencia en los procesos de creación de software.[7]

Los ejemplos anteriormente mencionados ponen de manifiesto como las líneas de producción contribuyen al mejoramiento de los procesos de desarrollo de software, además de reducir los costos; dejando al descubierto las innumerables razones que indican lo ventajoso que puede resultar aplicar esta nueva tendencia en Cuba.

1.3 Situación de la producción de software en Cuba. Necesidades.

Cuba intenta introducirse en el mercado mundial del software con productos que posean gran calidad y estén adaptados a las necesidades de los clientes. En el país se han llevado a cabo variadas estrategias con el fin de elevar los indicadores de producción de software aprovechando el alto capital humano disponible, pero las deficiencias en los modelos de producción afectan el proceso de desarrollo y entorpecen el avance en pos de las metas trazadas. Una forma eficaz de lograr estas metas es simplemente consolidar las empresas de este tipo en el país.

La consolidación en Cuba de las empresas de software necesita como primer paso una mejora de los aspectos que influyen en el producto:

- El proceso.
- La tecnología (que soporta el proceso).
- Las personas (comprenden y aplican el proceso de manera óptima).

Estos cambios necesitan tener en cuenta la optimización de los costos, los plazos de entrega y la calidad del producto final. Es posible afirmar que estos factores que afectan la inserción del país en tan importante mercado son producto de la inmadurez de las empresas que se encuentran al frente de este proceso, así como la producción artesanal de software, que causa lentitud en las entregas.

La inmadurez de una empresa es posible apreciarla según algunas características que definen este factor:

- Apaga fuegos: Esta característica se refiere a la rapidez con que se ejecutan los procesos, con el fin de entregar los productos.
- Pocos recursos propios.
- Hay altibajos en la producción por la rotación de recursos.

- Las planificaciones son pocos reales.
- Mucho esfuerzo dedicado a mantenimiento.
- Los plazos de entrega son impredecibles.
- Los empleados están descontentos.

Para realizar cambios significativos en cuanto a este aspecto es necesario definir un proceso de producción de software que cumpla con las metas que se establecen, referidas a la calidad, al tiempo, al costo, entre otros aspectos.

Las principales afectaciones que se evidencian obedecen a pequeños errores en la aplicación de las *metodologías de desarrollo de software* o simplemente a la ausencia de las mismas, por lo menos de forma categórica. Una *metodología de desarrollo de software* no es más que una rama dentro de la Ingeniería de software; en fin, es una estrategia para desarrollar este tipo de producto, y que además promueve prácticas adoptativas en vez de predictivas, centradas en las personas o en los equipos, orientadas hacia la funcionalidad y entrega del producto, de comunicación intensiva y que requieren implicación directa del cliente.

Las afectaciones mencionadas con anterioridad pueden resultar dañinas para el producto final; algunos ejemplos de estos problemas que existen en el desarrollo de software en Cuba son mencionados en la Revista Ciencia Holguín, en su segunda edición. La no elección de una metodología de desarrollo o modelo es uno de los elementos que se abordan en el artículo de dicha revista, esto provoca una falta de comunicación creciente entre clientes y desarrolladores afectando el levantamiento de requisitos, parte indispensable del desarrollo de software. Otro de los problemas a los que se hace referencia es que en muchas ocasiones se procede a implementar los requisitos de software directamente aún antes de haber realizado un análisis y un diseño profundo de estos. Por consiguiente, no se hace una buena elección de los lenguajes de programación o de la plataforma de explotación. Lo más usual radica en que no se determinan inicialmente los vínculos existentes entre los requerimientos de software. No se puede por tanto definir correctamente cuáles son las prioridades reales desde el punto de vista de los clientes y usuarios. Otro factor es que la estimación del tiempo de desarrollo o costos se ven afectadas si no se comprende el real alcance de lo que se realiza; aumentan, por tanto, no se satisfacen los convenios realizados, conllevando a desilusiones e informalidades. Además se adolece de falta de documentación en cada una de las etapas o de documentación realizada en distintos formatos, cada uno de ellos elegido por los desarrolladores a su conveniencia o entendimiento. El paso de una etapa a

la siguiente puede verse afectado por esto, pues los desarrolladores no comprenden la documentación entregada por otros.[8]

1.3.1 Situación de la producción de software en la Universidad de las Ciencias Informáticas.

La Universidad de las Ciencias Informáticas forma parte inequívoca de la producción de software en Cuba, y está encaminada entre otros factores a guiar el proceso de producción de software nacional; aún así, dentro de este centro subsisten algunos de los problemas anteriormente mencionados.
En la universidad se han realizado estudios que permitan brindar soluciones a estos problemas que se presentan, se ha abogado por introducir dentro de la universidad el concepto de factoría de software, aún así no se ha llegado a un acuerdo que permita poner en ejecución las investigaciones realizadas al respecto.
La Facultad 10, enclavada dentro de la universidad, tiene entre sus tareas el desarrollo de aplicaciones siguiendo los paradigmas del software libre, movimiento tecnológico que proporciona innumerables ventajas para el país en concepto de flexibilidad para la producción de software y sobre todo de índole económica. El polo productivo de software libre del centro incorpora a dicha facultad como su base , dentro del mismo se encuentra el grupo de desarrollo Nova, el cual se encarga de la producción de un Sistema Operativo personalizado basado en Gentoo (Distribución del Sistema Operativo GNU/Linux), colaborando en el proceso de migración hacia el software libre; dicho grupo presenta algunos problemas en el desarrollo del producto que lleva adelante, en su mayor parte los inconvenientes están relacionados con la metodología de desarrollo de software seleccionada. Además la inmensa mayoría de los procesos que se llevan a cabo en dicho proyecto (Nova Linux) son realizados de forma artesanal, lo que limita el proceso de desarrollo, así como la calidad del producto final. Entre otros aspectos que afectan también la calidad, es necesario tener en cuenta que el personal dedicado al desarrollo del mismo es mínimo, además se presentan problemas técnicos a las cuales urge darles solución, e incluso se han presentado situaciones que comprometen el software relacionadas con los recursos de los que se disponen para su producción.

1.4 Metodologías Ágiles.

El uso en la actualidad de las metodologías ágiles para el desarrollo de software es muy común debido a las ventajas que las mismas proporcionan con respecto a las metodologías de desarrollo tradicionales. En los años 90 muchos desarrolladores de software se dieron cuenta de la necesidad de

crear metodologías de desarrollo livianas y maniobrables, por el ambiente cambiante y turbulento dentro de los procesos de desarrollo. Las metodologías ágiles comparten algunos aspectos comunes, a pesar de que los detalles de desarrollo varían en dependencia de la metodología que se emplea.

Las metodologías ágiles forman parte del movimiento de desarrollo ágil de software, que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto. Se le denomina ágil como la habilidad de responder de forma versátil al cambio para maximizar los beneficios.

Las metodologías ágiles varían en su forma de “responder al cambio”, pero en general comparten características como las siguientes:

- Usan procesos de construcción iterativos.
- Entregan software funcional lo más pronto posible.
- Privilegian el valor de la gente sobre el valor del proceso.
- Fortalecen la comunicación y la colaboración.

De manera general este tipo de metodología favorece el desarrollo de software de manera colaborativa, debido principalmente a que se le presta más atención a los recursos humanos que a las herramientas de desarrollo, además de que son adaptables al cambio sin demasiada burocracia.

El principal objetivo de estas metodologías es permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto.

Las propuestas de metodologías más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán pueden incluso limitar la capacidad de los desarrolladores para llevar a cabo el proyecto.

En la tabla 1 se muestran las principales diferencias entre las metodologías ágiles y las tradicionales.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo de desarrollo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 1. Diferencias entre metodologías ágiles y no ágiles

Es importante llegar a un concenso de las metodologías ágiles a estudiar, de ahí que se realice una comparación con respecto a algunos parámetros que ofrecen una idea de cuán ágil es la metodología que se representa. La tabla 2 compara las distintas aproximaciones ágiles en base a tres parámetros: vista del sistema como algo cambiante, tener en cuenta la colaboración entre los miembros del equipo y características más específicas de la propia metodología como son simplicidad, excelencia técnica, resultados, adaptabilidad, etc. También incorpora como referencia no ágil el Capability Maturity Model10 (CMM).

	CMM	ASD	Crystal	DSDM	FDD	LD	Scrum	XP
Sistema como algo cambiante	1	5	4	3	3	4	5	5
Colaboración	2	5	5	4	4	4	5	5
Características Metodología (CM)								
- Resultados	2	5	5	4	4	4	5	5
- Simplicidad	1	4	4	3	5	3	5	5
- Adaptabilidad	2	5	5	3	3	4	4	3
- Excelencia técnica	4	3	3	4	4	4	3	4
- Prácticas de colaboración	2	5	5	4	3	3	4	5
Media CM	2.2	4.4	4.4	3.6	3.8	3.6	4.2	4.4
Media Total	1.7	4.8	4.5	3.6	3.6	3.9	4.7	4.8

Tabla 2. Ranking de "agilidad" (Los valores más altos representan una mayor agilidad)

El proyecto Nova Linux es desarrollado a través del empleo de una metodología ágil. Uno de los aspectos a tener en cuenta para la implantación de una línea de producción es el establecimiento de una metodología de desarrollo que permita el seguimiento de los pasos establecidos en la LPS. De ahí que sea necesario verificar de las metodologías ágiles existentes cuál se adapta al desarrollo de Nova Linux aplicando la filosofía de una LP.

Las metodologías a analizar son aquellas que después de realizada la comparación ofrecen resultados favorecedores, incluyendo a Lean Software Development debido a sus características principales, que la convierten en candidata para la aplicación de una línea de producción.

1.4.1 eXtreme Programming.

EXtreme Programming (XP) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos

imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí su nombre.

XP es un enfoque de la ingeniería de software y es la más destacada de los procesos ágiles de desarrollo de software en la actualidad. Al igual que estos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. La mayoría de los cambios de requisitos que se realizan en XP se hacen sobre la marcha, aspecto inevitable e incluso deseable dentro del desarrollo de proyectos. Es capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto como una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del mismo e invertir esfuerzos después en controlar los cambios en los requisitos. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo, y aplicarlo de manera dinámica durante el ciclo de vida del software.

Las características fundamentales de la metodología son:

- *Desarrollo iterativo e incremental*: pequeñas mejoras, unas tras otras.
- *Pruebas unitarias continuas*, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- *Programación en parejas*: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera, el código es revisado y discutido mientras se escribe.
- Frecuente *interacción del equipo de programación con el cliente* o *usuario*.
- *Corrección de todos los errores* antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- *Refactorización del código*, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenimiento pero sin modificar su comportamiento.
- *Propiedad del código compartida*: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto.
- *Simplicidad en el código*: La programación extrema apuesta que es más sencillo hacer algo

simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

Las características esenciales de XP están organizadas en 3 apartados: historias de usuario, roles, proceso y prácticas. Los cuales permiten de manera eficiente desarrollar cualquier software de manera ágil.

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento estas pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada una de ellas es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

Respecto a los roles XP existen diversas fuentes que abordan al respecto, aquí se hará referencia a los roles según el creador de la metodología. Los principales roles descritos por Beck son:

- *Programador*: Escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.
- *Cliente*: Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.
- *Encargado de Pruebas (Tester)*: Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- *Encargado de seguimiento (Tracker)*: Proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.
- *Entrenador (Coach)*: Es responsable del proceso global.
- *Consultor*: Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto.
- *Gestor (Big boss)*: Es el vínculo entre clientes y programadores. Su labor esencial es de

coordinación.

Dentro de los aspectos fundamentales en el proceso y las prácticas XP se encuentran las fases de la metodología: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimientos y Muerte del Proyecto.

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. XP apuesta por un crecimiento lento del costo del cambio y con un comportamiento asintótico. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas que forman parte del proceso:

- El juego de la planificación.
- Entregas pequeñas.
- Metáfora.
- Diseño simple.
- Pruebas.
- Refactorización (Refactoring).
- Programación en parejas.
- Propiedad colectiva del código.
- Integración continúa.
- 40 horas por semana.
- Cliente in-situ.
- Estándares de programación.
- Comentarios respecto a las prácticas.

Hay que tener presente una serie de inconvenientes y restricciones para su aplicación, tales como: está dirigida a equipos pequeños o medianos (Beck sugiere que el tamaño de los equipos se limite de

3 a 20 como máximo, otros dicen no más de 10 participantes), el entorno físico debe ser un ambiente que permita la comunicación y colaboración entre todos los miembros del equipo durante todo el tiempo, cualquier resistencia del cliente o del equipo de desarrollo hacia las prácticas y principios puede llevar al proceso al fracaso (el clima de trabajo, la colaboración y la relación contractual son claves), el uso de tecnologías que no tengan un ciclo rápido de realimentación o que no soporten fácilmente el cambio, etc.

1.4.2 Scrum.

Scrum es una metodología para el desarrollo ágil de productos, expuesta por Hirotaka Takeuchi e Ikujiro Nonaka, los cuales exponen según sus ideas que:

- El mercado competitivo de los productos tecnológicos, además de los conceptos básicos de calidad, coste y diferenciación, exige también rapidez y flexibilidad.
- Los nuevos productos representan cada vez un porcentaje más importante en el volumen de negocio de las empresas.
- El mercado exige ciclos de desarrollo más cortos.

Está metodología debe su nombre a un estudio realizado por sus creadores sobre nuevas prácticas de producción de software, comparando las mismas con un juego de rugby.

Nonaka y Takeuchi extraen las bases de Scrum de las prácticas que observan en las empresas con buenos resultados de rapidez y flexibilidad en la producción: Xerox, Canon, Honda, NEC, Epson, Brother, 3M o Hewlett-Packard. La principal premisa de esta metodología fue establecida por sus creadores y expresa “El conocimiento está en continua evolución a través del patrón dialéctico de tesis, antítesis y síntesis.”

Aunque esta metodología surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software.

Jeff Sutherland aplicó el modelo Scrum al desarrollo de software en 1993 en Easel Corporation (Empresa que en los macro-juegos de compras y fusiones se integraría en VMARK, luego en Informix y finalmente en Ascential Software Corporation). En 1996 lo presentó junto con Ken Schwaber como proceso formal, también para gestión del desarrollo de software en la 11na Conferencia anual realizada

por ACM (Association for Computing Machinery, siglas en inglés) sobre los Sistemas, lenguajes y aplicaciones de programación Orientada a Objetos (OOPSLA 96, siglas en inglés). Es importante acotar que en el desarrollo de software Scrum está considerado como modelo ágil por la Alianza Ágil.

Los principales elementos del proceso de desarrollo con Scrum son:

- *Roles*: Propietario del producto, Gestor o Manager del Scrum, Equipo e Interesados.
- *Componentes del proceso*: Pila del producto (*Product Backlog*), Pila del sprint (*Sprint Backlog*), Incremento.
- *Reuniones*: Planificación del sprint, Revisión diaria, Revisión del sprint.
- *Sprint* (Iteración).

El propietario del producto posee la responsabilidad del mismo, de la misma manera el Gestor o Management del Scrum tiene la responsabilidad del buen funcionamiento de la metodología; por último se puede decir que el equipo de trabajo posee la responsabilidad del desarrollo de manera general.

En Scrum los requisitos se expresan como elementos del *Product Backlog*. Esto no es más que una lista viva de requisitos funcionales y no funcionales priorizados por su valor para el cliente. Al decir que se trata de una lista viva, se deja claro que los requisitos que en ella aparecen y el orden de los mismos es cambiante a lo largo de la vida del proyecto. En Scrum, los requisitos se van abordando en Sprint en el orden en que aparecen en el Product Backlog. Esta forma de gestionar los requisitos brinda una vía para atajar los problemas relacionados con la misma.

Un elemento importante a tener en cuenta dentro de las ventajas que ofrece dicha metodología es el hecho de que el equipo de desarrollo Scrum se auto-gestiona y auto-organiza, o sea, dispone de atribuciones suficientes en la organización para tomar decisiones sobre como realizar su trabajo.

De manera general el Líder Scrum se encarga en su mayor parte de proporcionar a los desarrolladores un ambiente de trabajo estable y satisfactorio, garantizando un alto nivel de productividad, ya que el factor humano es el más importante según esta filosofía.

Scrum emplea dos herramientas fundamentales relacionadas con la gestión de proyectos. La primera se denomina Gráfico Burn-Up, la cual no es más que una herramienta de gestión y seguimiento para el propietario del producto; en fin, este gráfico presenta las versiones del producto previstas, las

funcionalidades de cada una, la velocidad estimada, las fechas probables para cada versión, el margen de error previsto en las estimaciones y el avance real del producto. La segunda herramienta es el Gráfico Burn-Down, que es la herramienta que emplea el equipo de desarrollo para gestionar y seguir el trabajo de cada Sprint, o sea, es una representación gráfica del avance del Sprint.

Scrum no se queda en el “cómo” de las prácticas, sino que trabaja desde el “porqué” para descartar, modificar o incorporar elementos según las características de la empresa y los proyectos.

Este modelo, al igual que el resto de las metodologías ágiles trabaja sobre la premisa de que la calidad y la eficiencia de la producción se deben sobre todo a los procesos empleados.

Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro, porque la gestión no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

1.4.3 Lean Software Development.

La metodología Lean Software Development (LSD) fue creada por Robert Charette y es importante aclarar que por sí sola no es una metodología de desarrollo o de administración.

El LSD ofrece principios que son aplicables a cualquier ambiente para mejorar el proceso de desarrollo de software. Sus principales objetivos son: Desarrollar software en un tercio del tiempo, con un tercio del presupuesto y con un tercio de la tasa de defectos. En fin LSD no es más que un conjunto de creencias comunes de como dirigir el cambio y la mejora.

El proceso dirigido por Lean Software Development posee 7 principios básicos:

1. Eliminar los residuos: Analizar cuales actividades y recursos son indispensables para llevar a cabo la producción del software.
2. Ampliar el aprendizaje.
3. Retrasar los compromisos.
4. Liberar rápido.
5. Facultar al equipo.
6. Construir integridad intrínsecamente.

7. Ver el todo.

Entre otros factores, estos principios permiten usar LSD como un marco de trabajo o como un grupo de líneas que dirigen las mejoras a los problemas que presenta el mundo del desarrollo de software en la actualidad.

En el caso del primer principio, mencionado con anterioridad, existen 7 tipos de residuos fundamentales durante el desarrollo:

- Características adicionales.
- Requerimientos.
- Pasos adicionales.
- Búsqueda de Información.
- Errores no capturados por las pruebas.
- Espera para la adopción de decisiones, incluyendo a los clientes.
- Liberaciones.

La práctica básica que aplica el LSD es las cajas de tiempo o Time Boxes, esta práctica consiste en fijar la iteración con una fecha final, que no se puede cambiar; si los requerimientos seleccionados para el time-box no pueden entregarse, entonces se disminuye el alcance. Esta práctica no debe ser empleada para presionar a los desarrolladores con trabajo de largas jornadas, sólo es una vía para agilizar el trabajo. Es importante aclarar que la duración de los time-boxes debe ser elegida para cada proyecto.

Otra de las buenas prácticas de LSD es que proporciona una vía para realizar los cambios fácilmente, a través de la creación y ejecución de pruebas en los diferentes estados del proceso de desarrollo. También se puede afirmar que otro de los aspectos positivos de la presente metodología es el tiempo que se dedica a la preparación del personal que se ocupará de desarrollar el software.

1.4.4 Adaptive Software Development.

Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes de software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone

tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

Esta metodología da forma a las fases básicas de la gestión ágil en:

ESPECULACIÓN, compuesta por 5 pasos:

1. Inicio para determinar la misión del proyecto.
2. Determinación del marco temporal del proyecto.
3. Determinación del número de iteraciones y la duración de cada una.
4. Determinación del objetivo de cada una de las iteraciones.
5. Asignación de funcionalidad a cada iteración.

COLABORACIÓN: Desarrollo concurrente del trabajo de construcción y gestión del producto.

APRENDIZAJE: Consiste en la revisión en cada iteración de los siguientes aspectos:

- Calidad, con criterios de cliente.
- Calidad, con criterios técnicos.
- Funcionalidad desarrollada.
- Estado del proyecto.

Las características básicas de ASD son:

- Trabajo orientado y guiado por la misión del proyecto.
- Basado en la funcionalidad.
- Desarrollo iterativo.
- Desarrollo acotado temporalmente.

- Guiado por los riesgos.
- Trabajo tolerante al cambio.

Sobre la presente metodología se puede decir que es una de las que incorpora la gestión de proyectos dentro de sus procesos.

1.5 Modelos de Calidad.

En la actualidad la implantación de líneas de producción lleva aparejado el uso de algún modelo de calidad existente debido a la necesidad de garantizar la calidad tanto del producto final como de los diversos componentes de software.

Un *modelo de calidad* es un conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos. Es importante tener en cuenta que los modelos de calidad te indican qué hacer y no cómo hacerlo, ya que dependen de la metodología que se use y de los objetivos del negocio.

Los modelos más usados en el mundo son: CMMI, Normas ISO y Métrica 3. CMMI (Capability Maturity Model Integration, siglas en inglés) está orientado a la mejora de procesos en diferentes niveles de madurez, teniendo en cuenta las características específicas de cada proyecto. En el caso de las Normas ISO, están orientadas al proceso del ciclo de vida del software de manera general. Métrica 3 está orientado al modelo y la implementación del software, aunque se considera un poco obsoleto como modelo de calidad.

En el proyecto Nova Linux no se cuenta con un modelo de calidad definido, de ahí que no existe una base para la realización del presente estudio.

Es de suma importancia conocer que para el proceso de selección del modelo de calidad a utilizar en cualquier proyecto de desarrollo se deben tener en cuenta las características del mercado al cual estará dirigido el producto, y más específicamente el modelo de calidad que predomina en dicho mercado.

1.5.1 Capability Maturity Model.

Es un modelo de evaluación de los procesos de una organización. Este modelo establece un conjunto de prácticas o procesos clave agrupados en Áreas Clave de Proceso (KPA, siglas en inglés). Para cada área de proceso define un conjunto de buenas prácticas que habrán de ser:

- Definidas en un procedimiento documentado.

- Provistas (la organización) de los medios y formación necesarios.
- Ejecutadas de un modo sistemático, universal y uniforme (institucionalizadas).
- Medidas.
- Verificadas.

A su vez estas Áreas de Proceso se agrupan en cinco "niveles de madurez", de modo que una organización que tenga institucionalizadas todas las prácticas incluidas en un nivel y sus inferiores, se considera que ha alcanzado ese nivel de madurez.

Los niveles son:

1. *Inicial*. Las organizaciones en este nivel no disponen de un ambiente estable para el desarrollo y mantenimiento de software. Aunque se utilicen técnicas correctas de ingeniería, los esfuerzos se ven minados por falta de planificación. El éxito de los proyectos se basa la mayoría de las veces en el esfuerzo personal, aunque a menudo se producen fracasos y casi siempre retrasos y sobrecostes. El resultado de los proyectos es impredecible.
2. *Repetible*. En este nivel las organizaciones disponen de prácticas institucionalizadas de gestión de proyectos, existen unas métricas básicas y un razonable seguimiento de la calidad. La relación con subcontratistas y clientes es gestionada sistemáticamente.
3. *Definido*. Además de una buena gestión de proyectos, a este nivel las organizaciones disponen de correctos procedimientos de coordinación entre grupos, formación del personal, técnicas de ingeniería más detalladas y un nivel más avanzado de métricas en los procesos. Se implementan técnicas de revisión por pares (peer reviews, nombre en inglés).
4. *Gestionado*. Se caracteriza porque las organizaciones disponen de un conjunto de métricas significativas de calidad y productividad, que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos. El software resultante es de alta calidad.
5. *Optimizado*. La organización completa está volcada en la mejora continua de los procesos. Se hace uso intensivo de las métricas y se gestiona el proceso de innovación.

Así es como el modelo CMM establece una medida del progreso conforme avanza, en niveles de madurez. Cada nivel a su vez cuenta con un número de áreas de proceso que deben lograrse. El

alcanzar estas áreas o estadíos se detecta mediante la satisfacción o insatisfacción de varias metas claras y cuantificables. Con la excepción del primer Nivel, cada uno de los restantes Niveles de Madurez está compuesto por un cierto número de Áreas Claves de Proceso.

1.5.2 Capability Maturity Model Integration. (CMMI).

CMMI es la integración de los modelos *CMM-SW, SE-CMM, IPD-CMM* y surge como una alternativa del modelo CMM, después de estudios realizados que permitieron reparar algunos conflictos existentes en el mismo. Es un modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software y está menos sujeto a interpretaciones ya que cada práctica dentro de él esta ampliamente documentada. En los últimos años este modelo se ha convertido en referencia, como un estándar "de-facto" internacional para la Industria de la Tecnología de Información, siendo actualmente uno de los modelos con mayor reconocimiento y aceptación a nivel mundial.

Áreas de proceso

Las áreas de proceso que ayuda a mejorar o evaluar CMMI son 22 en la versión que integra desarrollo de software e ingeniería de sistemas (CMMI-SE/SW) y 25 en la que cubre también integración de producto (CMMI-SE/SW/IPPD).

Vistas desde la representación continua del modelo, se agrupan en 4 categorías según su finalidad: Gestión de proyectos, Ingeniería, Gestión de Procesos y Soporte a las otras categorías. Vistas desde la representación escalonada, se clasifican en los 5 niveles de madurez. Al nivel de madurez 2 pertenecen las áreas de proceso cuyos objetivos debe lograr la organización para alcanzarlo, al igual que sucede con los niveles 3, 4 y 5.

El modelo para software (CMM-SW) es un modelo de procesos para el desarrollo y mantenimiento de sistemas de software, diseñado sobre los siguientes criterios:

- La calidad de un producto o sistema es consecuencia directa de los procesos empleados en su desarrollo.
- Las organizaciones que desarrollan software presentan un atributo denominado madurez, cuya medida es proporcional a los niveles de capacidad e institucionalización de los procesos que emplean en su trabajo.

Este modelo establece 5 niveles de madurez para clasificar a las organizaciones, en función de qué

áreas de procesos consiguen sus objetivos y se gestionan con principios de ingeniería. Es lo que se denomina un modelo escalonado, o centrado en la madurez de la organización.

La visión continua de una organización mostrará la representación del nivel de capacidad de cada una de las áreas de proceso del modelo.

La visión escalonada definirá a la organización dándole en su conjunto un nivel de madurez del 1 al 5.

Niveles de madurez definidos en SW-CMM

Nivel 1: Inicial

Los resultados de calidad obtenidos son consecuencia de las personas y de las herramientas que emplean. No de los procesos, porque o no los hay o no se emplean.

Nivel 2: Repetible

Se considera un Nivel 2 de madurez cuando se llevan a cabo prácticas básicas de gestión de proyectos, de gestión de requisitos, control de versiones y de los trabajos realizados por subcontratistas. Los equipos de los proyectos pueden aprovechar las prácticas realizadas para aplicarlas en nuevos proyectos.

Nivel 3: Definido

Los procesos comunes para desarrollo y mantenimiento del software están documentados de manera suficiente en una biblioteca accesible a los equipos de desarrollo. Las personas han recibido la formación necesaria para comprender los procesos.

Nivel 4: Gestionado

La organización mide la calidad del producto y del proceso de forma cuantitativa con base a métricas establecidas. La capacidad de los procesos empleados es previsible, y el sistema de medición permite detectar si las variaciones de capacidad exceden los rangos aceptables para adoptar medidas correctivas.

Dentro de la Gestión de un Proyecto:

- Cubre las actividades relacionadas con la planificación, seguimiento y control.

- Proporciona mecanismos para establecer, mantener y monitorizar acuerdos con los clientes y proveedores.
- Provee mecanismos para establecer y mantener un entorno de colaboración entre equipos.
- Proporciona un método común para gestionar el proyecto de forma cuantitativa y anticipándose a los problemas.

Nivel 5: Optimizado

La mejora continua de los procesos afecta a toda la organización, que cuenta con medios para identificar las debilidades y reforzar la prevención de defectos. Se analizan de forma sistemática datos relativos a la eficacia de los procesos de software para analizar el coste y el beneficio de las adaptaciones y las mejoras.

Se analizan los defectos de los proyectos para determinar las causas, y se realiza un mapeado sobre los procesos. Es el nivel más alto de CMM por el momento.

1.5.2.1 SCAMPI.

SCAMPI (Standard CMMI Appraisal Method for Process Improvement, significado en inglés) en español “Método de Apreciación para Mejora de Proceso”, es el método de evaluación oficial para CMMI desarrollado por el Instituto de Ingeniería de Software de Carnegie Mellon University (SEI, siglas en inglés) que permite determinar en su ámbito más completo (SCAMPI clase A) el nivel de capacidad y/o madurez de una organización y/o área interna de desarrollo de software.

El método de evaluación SCAMPI clase A es un método muy robusto que consta de 3 fases y 13 procesos. Las fases de planeación y preparación son consideradas dentro de este método como las fases y actividades críticas del SCAMPI, y dependiendo de que ésta sea llevada a buen término o no radicará el éxito del método.

Para la fase de planeación primero es necesario tener un claro entendimiento de las necesidades de la organización y el compromiso del patrocinador para llevar a cabo el proyecto SCAMPI. Durante esta fase se revisan aspectos de infraestructura, logística, alcance de la evaluación, determinación del equipo evaluador, selección e identificación de los proyectos muestra, agenda detallada para los períodos de readiness review (revisión documental) y on-site (entrevistas y presentaciones), entre otros aspectos.

Durante la fase de preparación la organización prepara la evidencia objetiva (documentación) que será considerada por el equipo evaluador (SCAMPI Team). Tal documentación es un mapeo de la ejecución (salida) de los procesos de la organización (proyectos, etc.) a las prácticas del CMMI. Esta fase es muy crítica y se debe entender correctamente ya que esta documentación es una base de información importante y fundamental que utilizará el equipo evaluador, y no es trivial de generar ya que se requiere de un entendimiento del contexto y objetivos del proyecto (y la organización), los procesos de la organización, y del modelo CMMI.

1.5.3 Métrica 3.

Es una metodología de planificación de Desarrollo y Mantenimiento de Sistemas de Información. Ofrece a las organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software.

Tiene como objetivos:

- Proporcionar o definir Sistemas de Información (SI) que ayuden a conseguir los fines de la Organización mediante la definición de un marco estratégico para el desarrollo de los mismos.
- Dotar a la organización de productos de software que satisfagan las necesidades de los usuarios. Más importancia al Análisis de Requisitos.
- Mejorar la productividad de los departamentos de Sistemas y TIC.
- Facilitar la comunicación y entrenamiento de los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto.
- Facilitar la operación, el mantenimiento y el uso de los productos de software obtenidos.

Este método contempla el desarrollo de SI para las distintas tecnologías y los aspectos de gestión que aseguran que un proyecto cumple sus objetivos en términos de calidad, costo y plazos. Además conserva la adaptabilidad, la flexibilidad y la sencillez, así como la estructura de actividades y tareas y facilita a través de interfaces la realización de los procesos de apoyo u organizativos: Gestión de Proyectos, Gestión de Configuración, Aseguramiento de la Calidad y Seguridad.

Métrica 3 descompone cada uno de los procesos en actividades y estas a su vez en tareas. Para cada tarea se describe su contenido: principales acciones, productos, técnicas, prácticas y participantes.

Procesos principales:

- Planificación de Sistemas de Información (PSI).
- Obtención de un marco de referencia para el desarrollo de SI que responda a los objetivos estratégicos de la organización.
- Desarrollo del Sistema de Información.
- Mantenimiento del Sistema de Información (MSI).
- Obtención de la nueva versión de un SI desarrollado con Métrica versión 3, a partir de las peticiones de mantenimiento de los usuarios que realizan con un motivo de un problema detectado en el sistema, o por la necesidad de una mejora del mismo.

Este modelo describe:

- Pasos a seguir en el desarrollo.
- Conjunto de productos finales a desarrollar.
- Conjunto de técnicas para obtenerlo
- Papeles (roles) de los participantes.
- Modo de implantación.
- Proyectos de distintos tamaños.

Además dicho modelo cubre el desarrollo estructurado y el Orientado a Objetos. La estructura de dicho modelo brinda información interesante sobre su forma de aplicación (Figura 1).

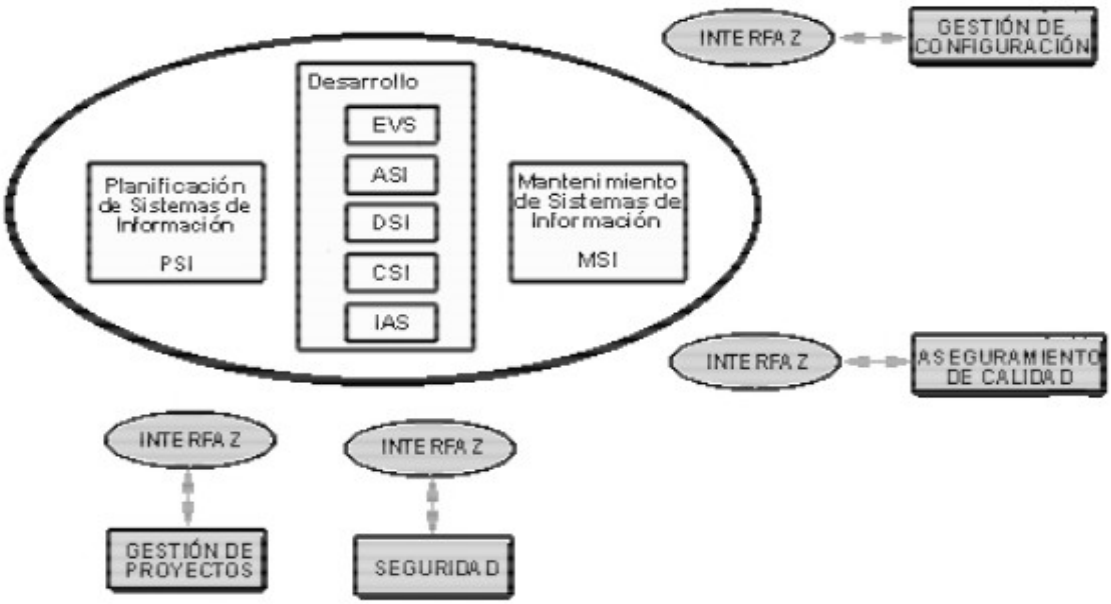


Figura 1.

1.5.4 Norma ISO 9001:2000.

La Norma ISO 9001:2000 elaborada por la Organización Internacional para la Estandarización, especifica los requisitos para un sistema de gestión de la calidad que puede utilizarse para su aplicación interna por las organizaciones, para certificación o con fines contractuales.

La actual versión de ISO 9001 data de diciembre de 2000, por ello se expresa como ISO 9001:2000.

La norma anteriormente mencionada es un método de trabajo, considerado entre los conocedores como “muy bueno”, incluso se dice que “es el mejor para mejorar la calidad y satisfacción de cara al consumidor.” La versión actual ha sido adoptada como modelo a seguir para obtener la certificación de calidad. Y es a lo que tiende, y debe de aspirar toda empresa competitiva, que quiera permanecer y sobrevivir en el exigente mercado actual.

De manera general, dicha norma, mejora los aspectos organizativos de una empresa, además propone unos sencillos, probados y geniales principios para mejorar la calidad final del producto mediante sencillas mejoras en la organización de la empresa que a todos benefician.

Estructura de la Norma ISO 9001:2000

La Norma está estructurada en capítulos, los cuales están distribuidos de la siguiente forma:

Capítulo 1: Guías y descripciones generales: No se enuncia ningún requisito.

1.1 Generalidades.
1.2 Reducción en el alcance.

Capítulo 2: Normativas de referencia.

Capítulo 3: Términos y definiciones.

Capítulo 4: Sistema de gestión: Contiene los requisitos generales y los requisitos para gestionar la documentación.

4.1 Requisitos generales.
4.2 Requisitos de documentación.

Capítulo 5: Responsabilidades de la Dirección: Contiene los requisitos que debe cumplir la dirección de la organización, tales como definir la política, asegurar que las responsabilidades y autoridades están definidas, aprobar objetivos, el compromiso de la dirección con la calidad, etc.

5.1 Requisitos generales.
5.2 Requisitos del cliente.
5.3 Política de calidad.
5.4 Planeación.
5.5 Responsabilidad, autoridad y comunicación.
5.6 Revisión gerencial.

Capítulo 6: Gestión de los recursos: La Norma distingue 3 tipos de recursos sobre los cuales se debe actuar: RRHH, infraestructura, y ambiente de trabajo. Aquí están contenidos los requisitos exigidos en su gestión.

6.1 Requisitos generales.
6.2 Recursos humanos.
6.3 Infraestructura.
6.4 Ambiente de trabajo.

Capítulo 7: Realización del producto: Aquí están contenidos los requisitos puramente productivos, desde la atención al cliente, hasta la entrega del producto o el servicio.

7.1 Planeación de la realización del producto y/o servicio.
7.2 Procesos relacionados con el cliente.
7.3 Diseño y desarrollo.
7.4 Compras.
7.5 Operaciones de producción y servicio
7.6 Control de dispositivos de medición, inspección y monitoreo

Capítulo 8: Medición, análisis y mejora: En este capítulo se sitúan los requisitos para los procesos que recopilan información, la analizan, y que actúan en consecuencia. El objetivo es mejorar continuamente la capacidad de la organización para suministrar productos que cumplan los requisitos. El objetivo declarado en la Norma, es que la organización busque sin descanso la satisfacción del cliente a través del cumplimiento de los requisitos.

8.1 Requisitos generales.
8.2 Seguimiento y medición.
8.3 Control de producto no conforme.
8.4 Análisis de los datos para mejorar el desempeño.
8.5 Mejora.

El objetivo principal de la misma radica en satisfacer al consumidor, además de estar orientada a garantizar la calidad de los procesos que se emplean en el desarrollo de software.

1.6 Conclusiones.

En nuestros días el desarrollo de software a alcanzado un lugar significativo en el mercado a nivel mundial; fenómeno causado por el incremento de la producción de tecnologías y por las necesidades sociales. Este desarrollo trae aparejado que el software sea visto como un producto, por lo que la aparición de las factorías de software y las líneas de producción le ha dado a este tipo de producción esquemas y métodos industriales con el fin de mejorar la calidad de los productos de manera general; además permite mejorar los métodos y sistemas de creación de los mismos.
En Cuba, se están llevando a cabo un grupo de acciones con la finalidad de elevar la producción de software e incorporar al país en este mercado, pero en el día a día se presentan un grupo de

problemas que dan al traste con estas intenciones, los mismos están dados por las prácticas empleadas, que no logran satisfacer las necesidades del mercado. En la Universidad de las Ciencias Informáticas, institución de referencia en esta industria, también prevalecen estos problemas; un ejemplo de esta situación lo constituye el proyecto Nova, el cual está desarrollando una distribución cubana personalizada, basada en Gentoo Linux, que colabora con la migración del país al software libre.

La idea del uso de una línea de producción en el proyecto Nova Linux, ha conllevado al estudio de un grupo de metodologías de desarrollo ágiles, las cuales responden de forma eficiente a los cambios, tan frecuentes durante la elaboración de una distribución Linux, en especial Nova, la cual posee una filosofía de producción creativa. El hecho de que se tenga en consideración la elección de un modelo de calidad, es de vital importancia ya que será implementado con la finalidad de que documente, verifique y certifique la calidad de esta imagen como producto rentable, capaz de satisfacer las necesidades del cliente.

2

Capítulo 2. Identificación y descripción de procesos implicados en el desarrollo de Nova Linux.

2.1 Introducción.

Durante el desarrollo del proyecto Nova Linux se llevan a cabo un grupo de procesos que permiten la culminación de los productos que se llevan a cabo. Para la implantación de una línea de producción se hace necesario el conocimiento pleno de dichos *procesos*, los cuales hacen referencia a lo que se conoce como proceso productivo, que no es más que transformar entradas (insumos) en salidas, (bienes y/o servicios) por medio del uso de recursos físicos, tecnológicos, humanos, etc. Un proceso productivo incluye acciones que ocurren en forma planificada y producen un cambio o transformación de materiales, objetos o sistemas, al final de los cuales se obtiene un producto.

La documentación disponible en el proyecto Nova no posee el nivel de especificación necesario para lograr una comprensión exacta del flujo de procesos que se ejecutan durante el desarrollo de cada una de las versiones estables del producto, referentes a cada una de las personalizaciones que se desarrollan. A pesar de esta situación, se puede contar con una breve descripción de los mismos, reconocidos como hitos dentro del proceso productivo actual.

El principal objetivo del presente capítulo es, a través de elementos que permitan llevar la descripción de los procesos actuales a un mayor grado de especificidad, detectar pasos innecesarios, repeticiones y otros obstáculos durante la creación de una personalización de Nova Linux.

2.2 Gestión por procesos.

El término gestión por procesos se refiere fundamentalmente a la forma de gestionar toda la organización basándose en sus Procesos, siendo definidos estos como una secuencia de actividades orientadas a generar un valor añadido sobre una entrada para conseguir un resultado, y una salida que a su vez satisfaga los requerimientos del cliente.

La gestión por procesos es ampliamente utilizada hoy en día debido a que permite una mejora continua de las actividades que se desarrollan, también permite reducir la variabilidad innecesaria, así como elimina las ineficiencias asociadas a la repetitividad de las actividades, permitiendo la optimización del empleo de los recursos disponibles en un proyecto.

Los autores de la presente investigación basan sus descripciones en la metodología antes

mencionada, aplicando sus paradigmas en pos de lograr un nivel superior de detalle en el proceso de liberación empleado en el proyecto Nova.

Primeramente se debe tener claramente definida la misión del proyecto, o sea, se debe identificar el objetivo fundamental del mismo, su razón de ser. (Ver Figura 2)

Figura 2.

El proyecto Nova no posee un único cliente, ya que sus personalizaciones están destinadas a diversas ramas de la sociedad, ejemplo de ello es el ministerio de las Fuerzas Armadas Revolucionarias (MINFAR), así como la Empresa de Telecomunicaciones de Cuba SA (ETECSA); también existe una propuesta de emplear una de las personalizaciones en el entrenamiento de los estudiantes élite para las competencias internacionales de programación; adicionalmente a esto el Sistema Operativo Nova Linux es empleado en la docencia dentro de la universidad. A modo de conclusión se puede afirmar que el principal cliente de este grupo de desarrollo es el propio centro al que pertenece, debido a que es este último el que permite la diversificación de clientes que el proyecto posee.

Es de vital importancia tener bien definido que es lo que se hace dentro del proyecto. Nova Linux es un Sistema Operativo (SO) encargado de brindar las funcionalidades fundamentales de este tipo de productos, incluyendo los programas imprescindibles para el funcionamiento de la computadora y que a su vez forman parte de la interfaz que permite la comunicación entre los usuarios y la máquina. Luego de quedar en **versión estable** se le adicionan los diversos programas que a petición de los clientes conforman una personalización.

Para la realización de este SO se ejecuta como proceso principal un proceso de liberación, el cual permite la entrega del producto final. Según las fuentes de documentación del proyecto que compete, el proceso que se desarrolla está basado fundamentalmente en la Ingeniería de Liberaciones, de ahí su nombre.

La metodología de gestión por procesos indica la necesidad de realizar una descripción exhaustiva de los procesos que forman parte del desarrollo de las personalizaciones de Nova, así como una clasificación de los mismos. Según la metodología empleada los procesos pueden ser clasificados en tres grupos fundamentales:

- *Procesos estratégicos*: Son aquellos que orientan y dirigen los procesos claves y de soporte.
- *Procesos clave*: Son la razón de ser del proyecto, el objetivo principal de actividad.
- *Procesos de soporte*: Son aquellos que apoyan a uno o más de los procesos claves.

El proceso de liberación antes mencionado puede ser clasificado como Clave dentro de la gestión por procesos, aún así es necesario aclarar que incluye subprocesos pertenecientes a las tres categorías anteriormente mencionadas.

En la Tabla 3 se identifican los diversos subprocesos que componen el proceso de liberación de Nova Linux, así como también se realiza una breve descripción de los mismos y son clasificados atendiendo a los criterios establecidos con anterioridad en este capítulo. Estos subprocesos constituyen hitos dentro del desarrollo de cualquier sistema operativo, aunque están en su mayoría adaptados al proceso de desarrollo de Nova.

Procesos	**Descripción**	**Clasificación**
Reunión Inicial de planeamiento	Se crea el Cronograma del **Release**, se definen las características iniciales y otras informaciones pertinentes. A partir de la reunión se hace un levantamiento de requerimientos y funcionalidades para el nuevo sistema.	Estratégico
Culminación del levantamiento de requerimientos y	Se congela el documento de los requerimientos. A partir de ésta fecha no se aceptarán nuevas funcionalidades o requerimientos.	Estratégico

funcionalidades		
Corte en la versión de repositorio	Se selecciona una versión del repositorio a trabajar para construir los binarios del release. A partir de éste momento se comienza a construir los binarios y el sistema base a utilizar. Los paquetes nuevos serán agregados o actualizados en dependencia de su impacto en el sistema (se cubre una funcionalidad o riesgos de seguridad).	Clave
Aseguramiento de la Calidad/ Pruebas de construcción	Se le realizan pruebas a cada uno de los paquetes binarios, y se chequea el correcto funcionamiento de cada una de las aplicaciones incluidas.	Clave
Actualización de la documentación	Se crea o actualiza la documentación sobre el release (versión del núcleo, funcionalidades añadidas, cambios respecto a versiones anteriores, manual de instalación)	Soporte
Se congela el sistema.	Se congela el sistema, para crear una **versión Beta** a distribuir a la comunidad para un proceso amplio de prueba. Se descongela el sistema. Se comienza la etapa de corrección de errores. (Esta etapa puede repetirse en dependencia de la cantidad y complejidad de los defectos encontrados)	Clave
Información sobre el Beta	Se redacta la información del sistema Beta, y se revisa hasta que quede lista para publicar.	Soporte
Se publica sistema Beta	Se sube el **ISO** del sistema Beta a un servidor público, se actualiza la pagina principal del sitio y se envía la noticia a los sitios externos (softwarelibre.uci.cu)	Estratégico
Actualización de la documentación	Se actualiza la documentación sobre el release (versión del núcleo, funcionalidades añadidas, cambios respecto a versiones anteriores, manual de instalación)	Soporte

Se congela el sistema (Final)	Se congela el sistema por última vez. A partir de ésta fecha no se aceptan actualizaciones al sistema, se comienza la etapa de configuración y personalización de la interfaz.	Clave
Se prepara liberación (etapa final)	Se revisa por última vez el sistema, se comprime y crea el CD de instalación.	Clave
Se publica el release (a los miembros del proyecto)	Se suben todos los recursos (sistema, manuales etc.) para el servidor donde se va a publicar, pero sin acceso público a los mismos.	Estratégico
Publica liberación	Se hacen públicos todos los recursos, se actualiza página principal del sitio y se envía la noticia a los sitios externos (softwarelibre.uci.cu)	Estratégico

Tabla 3.

2.3 Descripción del Proceso de Liberación dentro del proyecto Nova Linux.

Describir un proceso es exponer ordenadamente las fases del mismo, indicando que sucede en cada fase y cómo sucede. Dicha descripción debe ser clara y ordenada con el fin de que los desarrolladores sean capaces de entenderla. El proceso de Liberación del proyecto Nova Linux cuenta con un grupo de estos subprocesos o fases que hacen posible llegar al final del desarrollo de las diversas personalizaciones.

La primera fase es la *Reunión inicial de planeamiento,* la cual tiene como finalidad lograr un entendimiento entre los desarrolladores sobre el trabajo a realizar y el tiempo del que se dispone para ello. Forman parte de este proceso el Líder del subproyecto y el equipo de trabajo asignado a la tarea. Durante el desarrollo de este subproceso se cuenta con una máquina a disposición tanto del Líder como del equipo de trabajo para lograr una visualización de los parámetros a tratar en dicha reunión. También es necesario el empleo de herramientas de planificación como el Planner, que pertenece a la suite del GnomeOffice**,** permitiendo dejar constancia del proceso, así como una planificación estricta de las diversas fases. Durante la reunión primeramente se crea el Cronograma del Release, con el fin de cronometrar y organizar el trabajo en todo momento; luego se definen las características iniciales del producto y otras informaciones pertinentes, con el objetivo de brindarle el conocimiento necesario a los desarrolladores; como último paso se hace un levantamiento de requisitos y funcionalidades para el

nuevo sistema, aunque esto último se hace de forma muy básica, se debe lograr situar al equipo de desarrollo en el nuevo producto.

El segundo subproceso a llevar a cabo es la *culminación del levantamiento de requisitos y funcionalidades*; el cual tiene como principal objetivo llegar a un acuerdo entre el cliente y el equipo de desarrollo sobre las características del producto. En el presente proceso se deja redactado el documento que recoge las diversas funcionalidades que poseerá el sistema final, sirviendo a la vez de contrato entre las partes. Durante este proceso participan fundamentalmente el Líder del subproyecto, el cliente y el analista del sistema, siendo necesario tener claro en ambos casos las posibilidades reales que se tienen respecto al producto. En el final de esta fase se congela el documento de los requerimientos con el fin de imposibilitar cambios que puedan afectar la estabilidad del proyecto, o sea, a partir de ésta fecha no se aceptarán nuevas funcionalidades.

La próxima fase a poner en práctica es el *corte en la versión del* ***repositorio***, el cual tiene como finalidad fundamental seleccionar tanto la versión del **stage1** que se empleará, como el **árbol de portage**, este último posee aquellos paquetes que serán incluidos como parte del sistema operativo que se va a liberar. La tecnología fundamental empleada durante este proceso es:

- La **Sindicación RSS**: Permite conocer el estado de los paquetes desde http://gentoo-portage.org.

En el proceso de corte participa el mantenedor de paquetes, que es el encargado de compilar y construir los binarios que serán empleados.

Dentro de este subproceso primeramente se hace una investigación sobre los paquetes binarios que responden a los requisitos establecidos anteriormente, buscando las versiones más estables de los mismos, así como las más actualizadas. Luego se busca el stage1 sobre el que se va a trabajar, y por último el árbol de portage que se va a incorporar a la versión del stage1 seleccionado. Es importante esclarecer que estas versiones seleccionadas deben mantenerse mientras el sistema este en desarrollo, aunque salgan nuevas versiones que se adapten mejor a los requerimientos. Todo este proceso de selección o corte se realiza con el objetivo de lograr que la versión del sistema que se va a liberar sea estable, así como para cumplir los requerimientos funcionales que se establecieron para el sistema. Luego se comienza la generación de los **paquetes binarios**. La principal herramienta en este proceso es el **sistema de instalación de Gentoo Linux**.

La cuarta fase es el *Aseguramiento de la Calidad y las Pruebas de construcción*; la misma se realiza

con el fin de garantizar un producto final con calidad ya que en ella se prueban los diversos componentes del sistema en construcción, siendo este es el mejor momento para realizar esta tarea y reparar errores de forma anticipada. Las principales herramientas que se emplean en esta fase son el **midnight comander**, los diversos editores que existen para GNU/Linux y las salidas de los compiladores, en versiones que quedan a selección del Especialista de Sistema Base. Es importante aclarar que este proceso se lleva a cabo de forma colaborativa entre el equipo de desarrollo y la comunidad de software libre, tanto dentro del centro, como fuera de él. En este proceso se trabaja sobre dos sistemas; un primero que posee el sistema de generación de binarios y donde se realizan las **pruebas unitarias**, y un segundo que es el sistema de liberación, el cual no puede poseer la suciedad generada por la fase de prueba y que a su vez debe ser virgen, o sea, debe tener la menor cantidad de **logs** posibles, generados por el proceso de compilación y el manejo del sistema por parte de los desarrolladores. Las pruebas unitarias que se le realizan a los paquetes son hechas en el primer sistema, y a medida que los paquetes son probados se van integrando al segundo. Los paquetes nuevos son agregados o actualizados en el segundo sistema en dependencia de su impacto en el mismo, o sea, para llevar a cabo este último paso se realiza un análisis sobre si los paquetes cubren una funcionalidad o riesgos de seguridad. Durante estas pruebas se busca principalmente incompatibilidades, errores que ocasionan el mal funcionamiento de los paquetes, además de verificar la agilidad que poseen los mismos durante su uso; siempre velando por el cumplimiento de los requisitos. Todo esto con el objetivo de evitar errores a escalas tan bajas, lo cual permite brindar una garantía sobre el producto que se está desarrollando. En caso de presentarse problemas con alguno de los requisitos, para no comprometer al resto de la paquetería se emplea el **overlay** del proyecto, que no es más que un árbol de portage con funcionalidades y parches relacionados con el error anteriormente mencionado, dicho árbol es construido por los propios desarrolladores. Las pruebas y el proceso de integración son llevados a cabo empleando, al igual que en el paso anterior, el sistema de instalación de Gentoo Linux.

El siguiente paso es la *Actualización de la Documentación*; el mismo es realizado por el programador, con el fin de lograr que la nueva versión del producto este correctamente documentada. El primer paso dentro de este proceso es crear o actualizar la documentación sobre la liberación, o sea, la versión del núcleo, las funcionalidades añadidas, los cambios respecto a versiones anteriores y el manual de instalación del producto. Esta es una forma eficiente de garantizar que al terminar el sistema se cuente con una gran parte de la documentación que se requiere, resaltando en la misma las diferencias respecto a las versiones anteriores del sistema.

El sexto paso consiste en el *congelamiento del sistema*, realizado por el Especialista en Sistema con el objetivo de realizar pruebas en el mismo y comprobar como cumple con los requisitos establecidos, dichas pruebas se realizan a nivel de usuario. En el centro se emplea como herramienta para este fin el Gforge, que es el sitio donde los usuarios indican los problemas que le han surgido, y donde el desarrollador se informa al respecto. El primer paso es congelar el sistema, para crear una versión Beta a distribuir en la comunidad de software libre dentro de la UCI para un proceso amplio de prueba; o sea, se pone a disposición de los usuarios un sistema sin configurar ni personalizar, con el fin de que el propio usuario sea capaz de hacerlo, y de esta forma envíe información sobre errores que puedan resultar de el uso de aplicaciones, e incluso del propio sistema operativo. El segundo momento es el descongelamiento del sistema, para la realización de mejoras. Por último se comienza la etapa de corrección de errores, esta etapa puede repetirse en dependencia de la cantidad y complejidad de los defectos encontrados.

Al concluir con el congelamiento del sistema y su respectiva rehabilitación es necesario brindar *Información sobre el Beta* que constituiría el siguiente proceso a realizar, este paso se lleva a cabo con el objetivo de brindar a los usuarios y desarrolladores datos suficientes para comprender el funcionamiento básico del sistema, dando información precisa sobre la paquetería incorporada en la versión, y es llevado a cabo por el programador. Al igual que cualquier proceso de documentación las herramientas están relacionadas con la realización de documentos. Para ello primero se redacta la información de dicho sistema Beta, revisándola hasta que quede lista para su publicación; esta revisión se debe fundamentalmente a los errores que reportan los usuarios, ya que a medida que se reparan los mismos si existe algún cambio trascendente debe ser informado.

EL próximo paso es la *publicación del sistema Beta,* realizada por el Administrador de la infraestructura con la finalidad de dar a conocer el nuevo sistema, ya sin errores, pero aún sin configurar, ni personalizar. Lo primero dentro de este proceso es subir el ISO del sistema Beta a un servidor público, luego se actualiza la página principal del sitio y como paso final se envía la noticia a los sitios externos, en el caso de la Universidad de las Ciencias Informáticas, el sitio de software libre (softwarelibre.uci.cu).

Posteriormente se realiza lo que se conoce como *Actualización de la documentación*, este paso se repite realizado esta vez por el Líder del Grupo de proyecto con el objetivo de actualizar la documentación después de haber realizado cambios en el sistema. O sea, en este caso solo se actualizan aquellos elementos que han cambiado desde el último proceso de actualización. Para ello

primeramente se revisa la documentación existente y luego se llevan a cabo los cambios pertinentes.

Luego *se congela el sistema*. Al igual que en el caso anterior se repite uno de los procesos que ya se habían ejecutado, con el fin de lograr una versión estable del producto, evitando de esta manera cambios innecesarios en pos de un correcto funcionamiento, dedicando los esfuerzos de esta etapa a las configuraciones y a la interfaz del sistema. En este proceso participa activamente el Especialista en Sistema. En este caso un primer paso lo constituye el congelamiento del sistema por última vez; o sea, a partir de ésta fecha no se aceptan actualizaciones o mejoras del producto final. Durante este proceso se comienza la etapa de configuración y personalización de la interfaz. Para este último paso se emplean herramientas que permiten el trabajo con imágenes, ejemplo de ellas el Gimp y el Inkscape, al igual que en casos anteriores, las mismas son seleccionadas por los desarrolladores. En el caso de las configuraciones que se realizan, las mismas deben ser hechas a través de los editores que se emplean en estos casos.

El décimo paso consiste en *preparar la liberación*, con el fin de entregarla a los clientes en un formato adecuado, asegurando no obstante la calidad del sistema; en este subproceso participan el Especialista en Sistema Base y el Especialista en LiveCD. En este caso primero se revisa por última vez el sistema, evitando de esta manera que se libere el mismo con errores; luego el mismo es comprimido, garantizando la optimización del espacio para crear el disco de instalación; por último se crea el CD de instalación; al cual, a través de diversos scripts, se le incorpora la personalización de Nova Linux divida en tres paquetes, que permiten agilizar el proceso de instalación.

El penúltimo paso es la *publicación del release a los miembros del proyecto*, en este caso es un paso que garantiza el conocimiento de los desarrolladores con respecto al producto que acaban de desarrollar; esta tarea es ejecutada por el Líder del Grupo de proyecto. Primero se suben todos los recursos relacionados al sistema, incluyendo los manuales de soporte al usuario, al servidor donde se van a publicar, pero sin acceso público a los mismos, solo los miembros del equipo de desarrollo pueden acceder a ellos.

Por último se realiza una *Liberación Pública*, realizada por el Administrador de la infraestructura con el fin de que todos los usuarios tengan acceso al sistema. O sea se hacen públicos todos los recursos; luego se actualiza la página principal del sitio y se envía la noticia a los sitios externos mencionados con anterioridad.

Una vía de comprensión para este complejo proceso de liberación lo constituye el diagrama que lo

describe, este tipo de diagrama se conoce como diagrama de flujo, y no es más que una representación gráfica de la secuencia de pasos que se llevan a cabo para cumplimentar cualquiera de los procesos que se llevan a cabo.

La simbología empleada para la representación de un diagrama de este tipo en la presente investigación está basada en la norma ANSI (American National Standard Institute, siglas en inglés). Dentro de la norma existe una gran diversidad de símbolos para la realización de diagramas de flujo, los cuales pueden ser vistos en el Anexo 1 de la investigación. Además del empleo de los conectores, ampliamente conocidos, para indicar la secuencia de pasos a seguir. No existe necesidad de emplear el resto de los símbolos debido a la poca complejidad de los procesos que se ejecutan dentro del proyecto Nova.

El diagrama de flujo referente al Proceso de Liberación se presenta a continuación en la figura 3; los diagramas pertenecientes a los diversos subprocesos se encuentran en el Anexo 2 de la investigación:

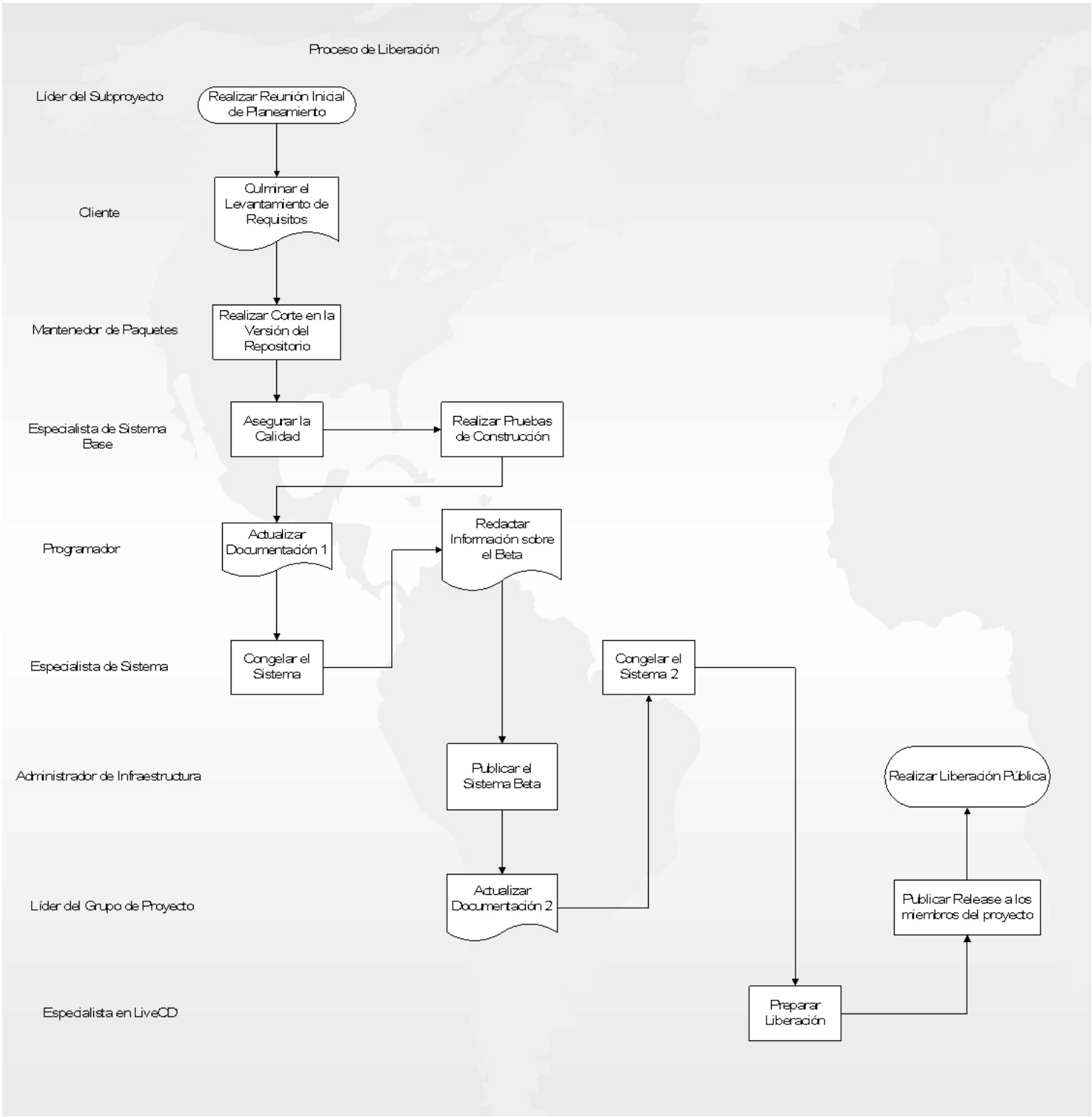

Proceso de Liberación
Líder del Subproyecto
Realizar Reunión Inicial de Planeamiento
Cliente
Culminar el Levantamiento de Requisitos
Mantenedor de Paquetes
Realizar Corte en la Versión del Repositorio
Especialista de Sistema Base
Asegurar la Calidad
Realizar Pruebas de Construcción
Programador
Actualizar Documentación 1
Redactar Información sobre el Beta
Especialista de Sistema
Congelar el Sistema
Congelar el Sistema 2
Administrador de Infraestructura
Publicar el Sistema Beta
Realizar Liberación Pública
Líder del Grupo de Proyecto
Actualizar Documentación 2
Publicar Release a los miembros del proyecto
Especialista en LiveCD
Preparar Liberación

Figura 3.

2.4 Identificación de problemas y oportunidades para la mejora del proceso de liberación en Nova.

Durante el análisis y descripción del proceso de Liberación que se lleva a cabo en el proyecto Nova Linux es posible la identificación de un grupo de problemas durante la ejecución de las fases establecidas. Un estudio realizado por los autores de la presente investigación arroja un grupo de aspectos que se encuentran recogidos en las buenas prácticas de la Ingeniería de Liberaciones y que deben ser tenidos en cuenta durante el desarrollo de cualquiera de las personalizaciones de Nova Linux:

- El proceso de liberación debe estar automatizado.
- Debe existir un proceso de seguimiento de las liberaciones, o sea, debe ser posible la comparación entre ellas, estableciendo diferencias y similitudes.
- El proceso de liberación debe estar integrado con SCM (Software Configuration Management, siglas en inglés), en español Gestión de configuración de software, lo cual permite el control de versiones.
- Las formas de manejar actualizaciones, archivos obsoletos, así como verificar su correcto funcionamiento deben formar parte del proceso.
- Las vías para la creación de las diversas formas de distribución deben estar reflejadas con claridad en el proceso.

La descripción del proceso de Liberación en subcapítulos anteriores permite, teniendo en cuenta los aspectos anteriormente mencionados arribar a conclusiones sobre ciertas dificultades que se presentan durante el desarrollo de Nova. Es importante clarificar que durante la investigación se concretó la falta de un equipo de calidad dentro del Grupo de desarrollo, debido fundamentalmente a que no existe un modelo de calidad implantado dentro de la organización. Dicho equipo ha sido suplantado por una liberación que se realiza a los miembros del equipo de desarrollo con el fin de que se reporten errores en el sistema, aún así esto no ha arrojado buenos resultados. A este respecto se evidencia además la falta de un proceso de calidad vinculado a los procesos que se realizan, lo cual garantizaría la calidad en el proceso general y por ende en el producto final.

El proceso de liberación dentro de Nova Linux no está automatizado, de forma general y a pesar de la

existencia de un kit de instalación, los subprocesos que se realizan dentro de la Ingeniería de Liberaciones son realizados empleando el sistema de instalación de Gentoo Linux, o sea, se realizan de forma manual, con la presencia a tiempo completo del equipo de desarrollo.

El control de versiones está implementado a través de **Subversion,** a pesar de ello los desarrolladores no se encuentran familiarizados con este proceso, que además no está acoplado de forma estratégica con el proceso de Liberación establecido. También es posible apreciar que la documentación no se encuentra del todo vinculada al sistema de control de versiones implementado. Es importante valorar que el sistema solo se encuentra funcionando para una parte del proyecto, la totalidad de los sistemas creados no pasan por un proceso de control de versiones, dificultando la reutilización de código. Tampoco se encuentran dentro de este proceso los archivos de configuración, las versiones del kernel utilizado, además de la paquetería probada, la cual debería ser pilar fundamental dentro del mismo.

Uno de los mayores problemas que se identifican dentro del proceso de release, es que no se encuentran reflejadas dentro de él las diversas vías para la creación de las diferentes formas de distribución que pueden desarrollarse dentro del proyecto. Esto provoca la documentación de los diferentes procesos aunque los mismos sean idénticos.

Algunos de los problemas detectados pertenecen a otra parte importante del proceso de desarrollo, la metodología de desarrollo de software. Actualmente eXtreme Programming rige el ciclo de vida del proyecto Nova Linux, aún así no se respetan los roles que propone dicha metodología, además de existir un nivel de especialización demasiado elevado en los desarrolladores, lo cual provoca la pérdida de la responsabilidad conjunta sobre los resultados. Cada uno se encarga de la tarea que debe desarrollar, sin preocuparse por como afecta al resto del equipo. Otro error relacionado con la metodología es la no correspondencia entre esta y la documentación que se genera, violando de esta manera uno de los aspectos vitales dentro de un proyecto de desarrollo.

2.5 Metodología ágil seleccionada para la implantación de la Línea de Producción.

En el capítulo anterior se explica en detalle las razones para el uso de una metodología ágil en la implantación de una LPS para Nova Linux. Las razones de mayor peso concuerdan en la necesidad de usar un proceso de construcción iterativo que a su vez permita entregar software funcional a la mayor brevedad posible y que además permita fortalecer la comunicación y la colaboración entre los miembros del equipo de desarrollo.

Durante la sistematización para la selección de la metodología de desarrollo de software a emplear se tiene en cuenta cuan adaptable puede ser la misma, ya que cualquier personalización de Nova se ve sujeta a cambios durante su construcción, de ahí que una metodología de desarrollo tradicional resulte poco adecuada para dicho proyecto. Es necesario recalcar que debido a la escasez de recursos materiales y humanos la metodología debe poseer pocos roles y artefactos a generar.

De forma general la metodología que se adapta a la perfección a los requerimientos anteriormente expuestos es Scrum, considerada como la segunda en el ranking de agilidad vigente, además de poseer un grupo de características que permiten que el proceso de desarrollo de Nova Linux sea óptimo.

Scrum está diseñada para proyectos con requisitos inestables, además de que incluye, a diferencia de otras metodologías, la gestión de software, aspecto importante a tener en cuenta para la aplicación de una LPS, debido a la necesidad de lograr un producto final con la calidad requerida.

Esta metodología no se queda en el “cómo” de las prácticas, sino que trabaja desde el “por qué” para descartar, modificar o incorporar, según las características de la empresa y los proyectos, mejoras que permitan una mayor calidad del producto.

Este modelo, al igual que el resto de las metodologías ágiles trabaja sobre la premisa de que la calidad y la eficiencia de la producción se deben sobre todo a los procesos empleados; de ahí que halla resultado satisfactoria su elección, ya que la presente investigación esta directamente enfocada a los procesos que se llevan a cabo en el proyecto que compete.

El funcionamiento general de Scrum es perfectamente adaptable e inteligible. Ver Figura 4.

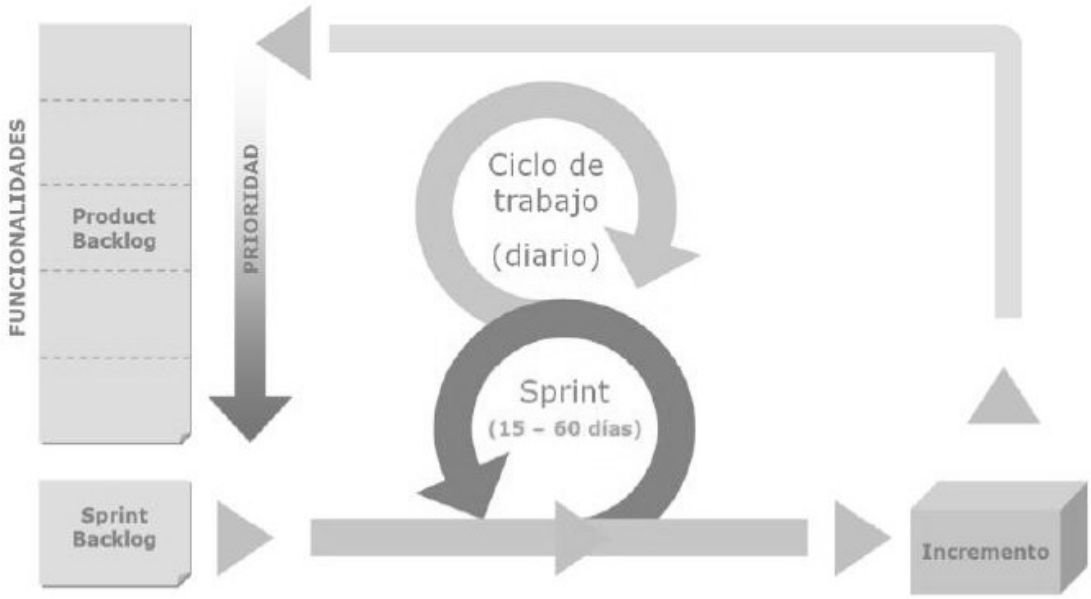


Figura 4.

El desarrollo con Scrum se realiza a través de *Sprint*, que no son más que las iteraciones que se realizan en cualquier otra metodología, durando las mismas 30 días o menos, en dependencia de las necesidades. El resultado de cada una de estas iteraciones es un incremento ejecutable que es mostrado al cliente.

Durante el desarrollo de software basado en Scrum se realizan tres tipos de reuniones fundamentales: la reunión de planificación, las reuniones de seguimiento y por último la reunión de revisión, todas ellas relacionadas a cada uno de los Sprint. Ver Figura 5.

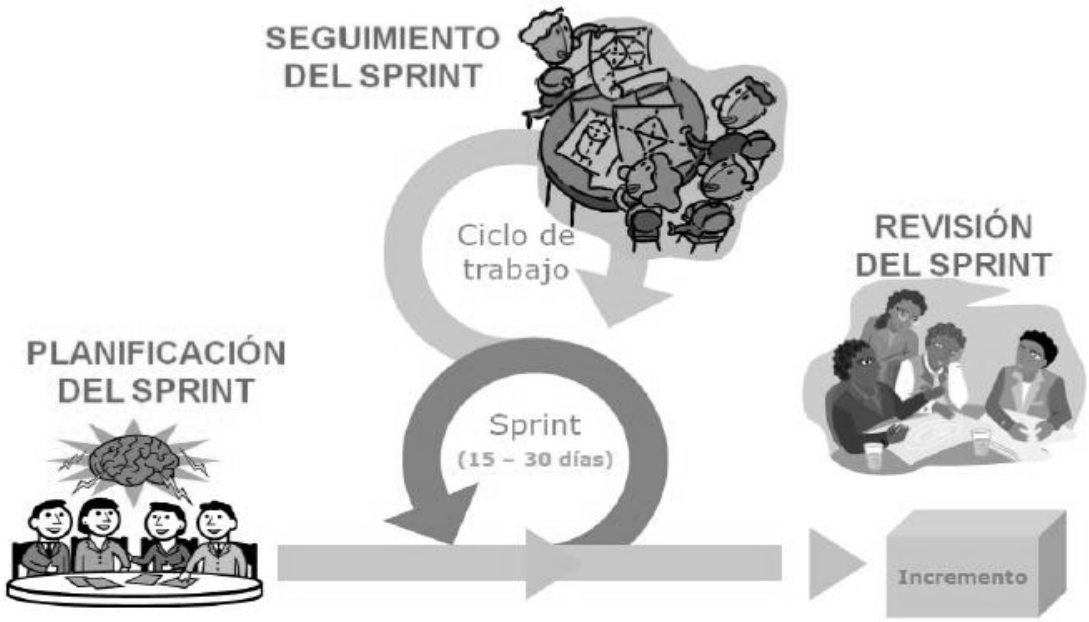


Figura 5.

En la figura 6 se pueden apreciar los artefactos que se tienen en cuenta durante la reunión de planeamiento de un Sprint y las diversas salidas que genera la misma.

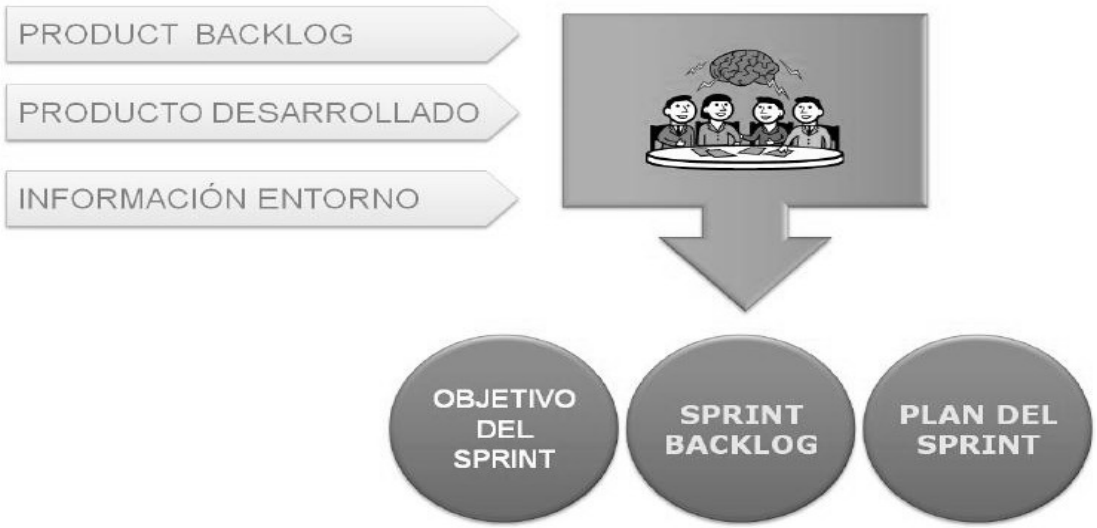


Figura 6.

Las reuniones a lo largo del proyecto son las verdaderas protagonistas, especialmente la reunión diaria (Daily Scrum) de 15 minutos que realiza el equipo de desarrollo para coordinación e integración. El

equipo de desarrollo es conocido en esta metodología como Scrum Team, y sólo debe estar conformado por un grupo de 5 a 9 personas. Las reuniones diarias deben ser celebradas, como su nombre lo indica, todos los días y con preferencia a la misma hora. En ella se agrupan el Scrum Máster y el equipo de desarrollo con el propósito de eliminar todos los impedimentos de rapidez para el grupo. Es importante aclarar que cualquier persona puede participar en la reunión, pero que solo pueden opinar los miembros del proyecto y el Líder del mismo. En estas reuniones cada uno de los integrantes del Scrum Team debe analizar ciertos aspectos relacionados con su comportamiento:

- ¿Que hice desde la pasada reunión?
- ¿Que voy a hacer entre ahora y la próxima reunión?
- ¿Existe algún problema para hacer lo que tengo planeado?

Esta es una forma de trazarse metas que quedan evidenciadas en el encuentro, además de rendir cuentas sobre el trabajo que se debió realizar con anterioridad.

El Scrum Máster o Líder del equipo de proyecto es el encargado de dirigir al equipo de desarrollo, eliminar los posibles impedimentos y constantemente trabajar para asegurar que el equipo posea las mejores circunstancias posibles para llevar a cabo las metas del sprint.

El primer momento del desarrollo con Scrum pone su peso en el *Product Owner* o Dueño del Producto, que no es más que el cliente, el cual para llevar a cabo ese rol debe poseer un conocimiento amplio de los procesos de negocio, además de ingeniería y marketing. En este primer momento el Product Owner compila todas las funcionalidades planificadas para el producto y prioriza aquellas que considera de vital importancia. El resultado de este trabajo llevado a cabo por el cliente es lo que se conoce como Product Backlog o Pila del Producto, que consiste en una lista de requisitos a cumplir y que está cambiando constantemente sus propiedades, siendo además es visible para toda la organización. Para crear este primer Backlog el cliente compila todos los requerimientos y las especificaciones que son la base de los cambios del producto, así como nuevas funciones y reparación de errores. Después de definidas estas metas, las mismas son divididas en segmentos. Paralelamente se realiza una lista priorizada donde el Product Owner toma las decisiones de forma personal.

Antes de cada Sprint las metas con más alta prioridad son transferidas a lo que se conoce como Sprint Backlog, que no es más que una lista de funcionalidades a tratar durante esa iteración, esta lista se redacta teniendo en cuenta las prioridades establecidas por el cliente en el Product Backlog. Durante

las discusiones entre los miembros del proyecto y los usuarios son determinadas las metas del sprint y las funcionalidades priorizadas son convertidas en tareas detalladas. En estos procesos es importante aclarar que el equipo se auto organiza y los miembros tienen responsabilidad conjunta de los resultados, además dentro del equipo no existen roles.

Al finalizar cada Sprint se realiza una reunión de evaluación conocida como Sprint Retrospective o Revisión del Sprint, donde se revisan las experiencias y se dan las conclusiones del ciclo de desarrollo. La base para estas reuniones de evaluación es una demostración durante la cual está ejecutándose el software frente a los usuarios.

Scrum propone dos herramientas fundamentales para la gestión, tanto para el dueño del producto como para el equipo de desarrollo; el gráfico Burn-Up y el gráfico Burn-Down respectivamente.

El gráfico Burn-Up presenta las versiones del producto previstas, las funcionalidades de cada una, la velocidad estimada, las fechas probables para cada versión, el margen de error previsto en las estimaciones y el avance real del producto. Un ejemplo de este tipo de gráfico puede verse en la figura 7.

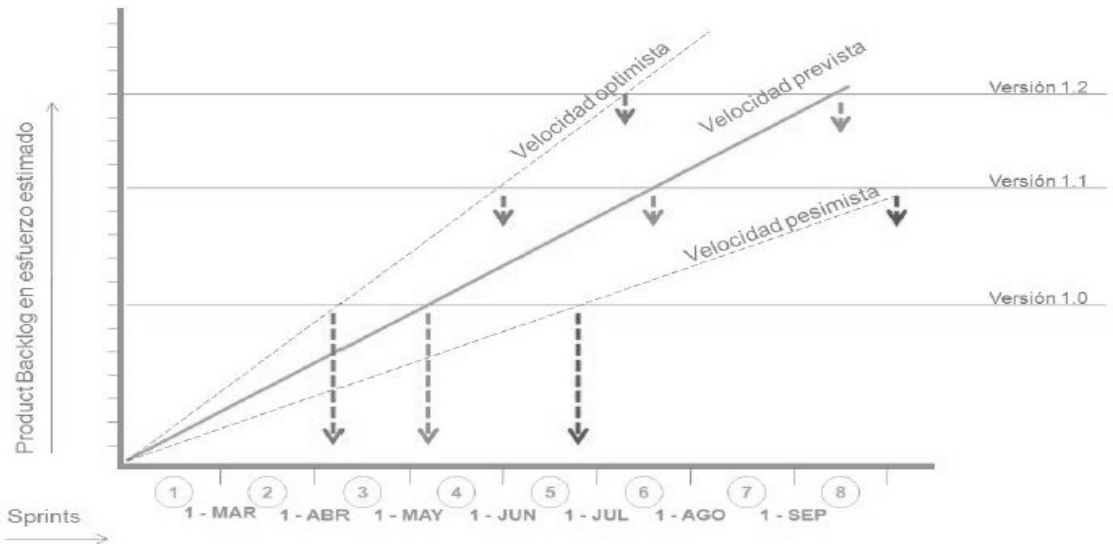


Figura 7.

El gráfico Burn-Down gestiona y sigue el trabajo de cada Sprint, o sea, es una representación gráfica del avance del Sprint. En la figura 8 puede verse un ejemplo de este gráfico en el cual la línea punteada en gris representa la evolución ideal del Sprint y la línea consecutiva representa la evolución

real diaria del mismo.

Figura 8.

Las ventajas que brinda esta metodología de desarrollo ágil son evidentes en lo anteriormente expuesto, de esta manera se considera que es la más adaptable a las necesidades actuales del proyecto Nova. Las razones pueden ser dadas relacionando la metodología con el proyecto, para de esta manera poner en evidencia la utilidad de la misma y los beneficios que reporta.

Para lograr un producto base con calidad se hace necesario emplear pocos recursos, tanto humanos como materiales, Scrum propone equipos de 5 a 9 personas. Además la necesidad de que se pueda entregar el producto en poco tiempo es imprescindible, dicha metodología garantiza tener funcionalidades vitales en 30 días. También Scrum incorpora la gestión de software, mejorando las condiciones de trabajo para el Líder del proyecto, como para el equipo que se encarga de llevar a cabo este proceso. La comunicación se ve mejorada con las reuniones diarias, al igual que el trabajo colaborativo, ya que no existen roles y todos son responsables de los resultados.

Existen otras muchas razones que permiten demostrar la utilidad de Scrum para la implantación de una LPS en el proyecto Nova Linux, de esta manera, queda seleccionada por excelencia por los autores de este trabajo dicha metodología.

2.6 Modelo de calidad seleccionado para la implantación de la Línea de Producción.

La aplicación de un modelo de calidad durante el desarrollo de Nova Linux es fundamental para garantizar la calidad de los procesos que se llevan a cabo. En el capítulo 1 se hace una disertación sobre algunos modelos aplicables al proyecto, después de un análisis en profundidad los autores han llegado a la conclusión de que el modelo mas adaptable a las necesidades de Nova es la Norma ISO 9001:2000.

A pesar de no ser el modelo más usado en el mercado, su aplicación se hace posible ya que participan en su aprobación representantes de los organismos nacionales de normalización y representantes del sector empresarial de países como: Argentina, Chile, Colombia, Costa Rica, Ecuador, España, Estados Unidos de Norte América, México, Perú, Uruguay y Venezuela. Igualmente participan representantes de COPANT (Comisión Panamericana de Normas Técnicas) y de INLAC (Instituto Latinoamericano de Aseguramiento de la calidad). Es posible apreciar que Latinoamérica favorece esta normativa y la emplea sobre todo en la Gestión y aseguramiento de la calidad en sus empresas.

De manera general, dicha norma, mejora los aspectos organizativos de una empresa, además propone unos sencillos, probados y geniales principios para mejorar la calidad final del producto mediante simples mejoras en la organización de la empresa que a todos benefician.

Su selección esta basada fundamentalmente en que es un marco para la gestión de la calidad de los procesos. Además de que su dominio de aplicación es muy amplio; o sea, se aplica de manera general al conjunto de actividades de una organización.

Esta Norma Internacional pueden utilizarla partes internas y externas, incluyendo organismos de certificación, para evaluar la capacidad de la organización para cumplir los requisitos del cliente, los reglamentarios y los propios de la organización. Además promueve la adopción de un enfoque basado en procesos cuando se desarrolla, implementa y mejora la eficacia de un sistema de gestión de la calidad, para aumentar la satisfacción del cliente mediante el cumplimiento de sus requisitos. De manera general enfatiza la importancia de:

- La comprensión y el cumplimiento de los requisitos.
- La necesidad de considerar los procesos en términos que aporten valor.
- La obtención de resultados del desempeño y eficacia del proceso.
- La mejora continua de los procesos con base en mediciones objetivas.

En la figura 9 se puede apreciar como funciona esta Norma:

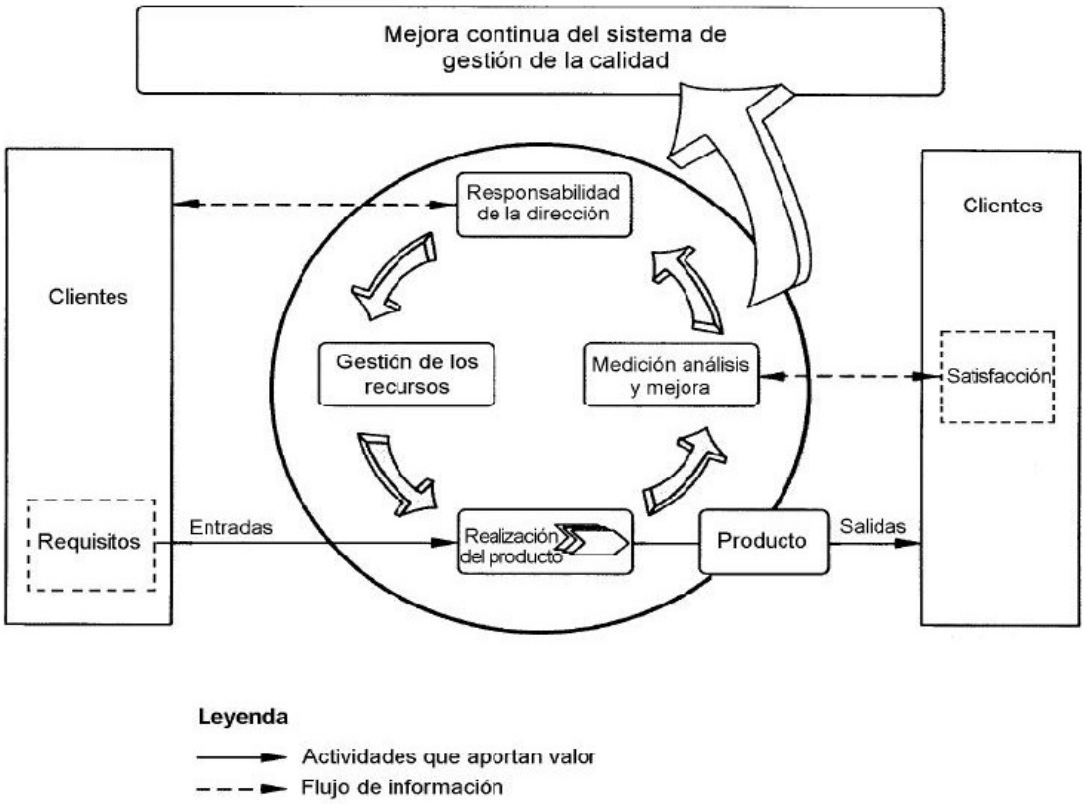


Figura 9.

Es importante aclarar que la Norma ISO 9001: 2000 es compatible con otros sistemas de gestión, facilitando de esta forma cumplir con los requerimientos e incorporarse a otros modelos dentro de la propia organización.

De manera general esta norma especifica los requisitos para un sistema de gestión de la calidad, además de que sus requisitos son genéricos y por lo tanto aplicables a todas las organizaciones sin importar su tipo, tamaño y producto suministrado.

En el documento oficial de este modelo se establecen un grupo de requisitos que debe cumplir la organización en pos de mejorar sus procesos. Los requisitos generales de la misma establecen que:

La organización debe establecer, documentar, implementar y mantener un sistema de gestión de la calidad y mejorar continuamente su eficacia de acuerdo con los requisitos de esta Norma Internacional.

La organización debe:

- Identificar los procesos necesarios para el sistema de gestión de la calidad y su aplicación a través de la organización.
- Determinar la secuencia e interacción de estos procesos.
- Determinar los criterios y métodos necesarios para asegurarse de que tanto la operación como el control de estos procesos sean eficaces.
- Asegurarse de la disponibilidad de recursos e información necesarios para apoyar la operación y el seguimiento de estos procesos.
- Realizar el seguimiento, la medición y el análisis de estos procesos.
- Implementar las acciones necesarias para alcanzar los resultados planificados y la mejora continua de estos procesos.

La organización debe gestionar estos procesos de acuerdo con los requisitos de la Norma Internacional. En los casos en que la organización opte por contratar externamente cualquier proceso que afecte la conformidad del producto con los requisitos, la organización debe asegurarse de controlar tales procesos. El control sobre dichos procesos contratados externamente debe estar identificado dentro del sistema de gestión de la calidad.

Parte de los requisitos fundamentales dentro de esta normativa son los que hacen referencia a la documentación, de manera general establecen que:

La documentación del sistema de gestión de la calidad debe incluir:

- declaraciones documentadas de una política de la calidad y de objetivos de la calidad,
- un manual de la calidad,
- los procedimientos documentados requeridos en esta Norma Internacional,
- los documentos necesitados por la organización para asegurarse de la eficaz planificación, operación y control de sus procesos, y
- los registros requeridos por esta Norma Internacional.

La extensión de la documentación del sistema de gestión de la calidad puede diferir de una organización a otra debido a:

- el tamaño de la organización y el tipo de actividades,
- la complejidad de los procesos y sus interacciones, y
- la competencia del personal.

Además la documentación puede estar en cualquier formato o tipo de medio.

Por esta misma vía se establecen en la Norma un grupo de restricciones con respecto a las formas de control que deben ser empleadas.

La alta dirección debe asegurarse de que la política de la calidad:

- es adecuada al propósito de la organización,
- incluye un compromiso de cumplir con los requisitos y de mejorar continuamente la eficacia del sistema de gestión de la calidad,
- proporciona un marco de referencia para establecer y revisar los objetivos de la calidad,
- es comunicada y entendida dentro de la organización, y
- es revisada para su continua adecuación.

Los diversos capítulos de la norma, así como sus subcapítulos recogen toda una filosofía para la gestión de la calidad, encontrándose en ella los aspectos fundamentales para lograr un producto final con calidad apoyado en los recursos humanos.

En el proyecto Nova Linux se debe establecer una política de calidad sencilla, de acuerdo a las dimensiones del mismo, la cual establece la realización de un grupo de pruebas a medida que se va desarrollando el producto.

Aplicar la Norma ISO 9001: 2000 en el proyecto favorece la organización de la política establecida con anterioridad, ya que las normativas recogidas en este modelo son de riguroso cumplimiento, estableciendo disciplina en todas las ramas del proyecto como organización.

A pesar de sus grandes ventajas es necesario realizar la inserción de un grupo de procesos

relacionados con la Norma, aspecto que permitirá llevar de la mano el proceso de liberación y la gestión de la calidad.

La norma establece un grupo de requisitos a tener en cuenta durante las revisiones. Nova necesita sobremanera un seguimiento de los procesos, a través de revisiones periódicas, lo cual es otro factor que favorece la implantación de este modelo.

La gestión de recursos de cualquier índole también es recogida dentro de los parámetros a seguir dentro del modelo. Este aspecto garantiza dentro del proyecto una equitatividad laboral, ya que hoy en día se satura de trabajo a los desarrolladores con mayores competencias.

Otro aspecto que se recoge en la Norma y que a su vez favorece su implantación como modelo es la obligación de la organización en la preparación de sus profesionales:

La organización debe:

- determinar la competencia necesaria para el personal que realiza trabajos que afectan a la calidad del producto,
- proporcionar formación o tomar otras acciones para satisfacer dichas necesidades,
- evaluar la eficacia de las acciones tomadas,
- asegurarse de que su personal es consciente de la pertinencia e importancia de sus actividades y de cómo contribuyen al logro de los objetivos de la calidad, y
- mantener los registros apropiados de la educación, formación, habilidades y experiencia.

En uno de sus capítulos la norma recoge lo referente a la infraestructura y al ambiente de trabajo, facilitando mejorar las condiciones de los desarrolladores, aspecto que la hace adaptable a la metodología de desarrollo seleccionada, que también debe garantizar este aspecto.

Es válido aclarar que a pesar de ser CMMI el modelo de calidad más usado en el mercado al cual va dirigido el producto; el mismo posee pequeños factores negativos que imposibilitan su uso en el proyecto que compete.

Dentro de estos factores se encuentra lo que podía ser tomado en cuenta como dimensiones del modelo. El mismo posee una documentación demasiado extensa para el establecimiento de una LPS,

ya que las mismas deben ser concisas. La incorporación de los diversos procesos para la implementación de CMMI provocaría un aumento considerable del porciento de procesos que se llevan cabo dentro del proyecto como organización.

Es necesario aclarar que debido al déficit de personal en el proyecto Nova se hace materialmente imposible dedicar un equipo de desarrollo de las dimensiones que lo requiere este modelo, además de que el mismo debe poseer una preparación del todo profesional al respecto, lo que imposibilita que pueda dedicarse personal que ya se encuentre vinculado al desarrollo.

La implantación de CMMI resulta en su mayoría compleja en cuanto a sus niveles y áreas de proceso, provocando que deba pararse la producción del sistema en pos de un correcto funcionamiento del modelo. O sea, según las propias recomendaciones para una buena implementación del mismo se puede afirmar que no es posible vincular todos los proyectos de la organización al comenzar la implantación del modelo; deben seleccionarse dos o tres proyectos piloto. La LPS para Nova Linux abarca todos los proyectos dentro del grupo de desarrollo, de ahí que CMMI no sea factible para la LP.

También es posible afirmar que CMMI necesita de las evaluaciones periódicas a través de SCAMPI, las cuales en caso de ser realizadas por profesionales resultan caras y en el caso contrario poco fiables.

Además se puede afirmar que la aplicación de la Norma ISO 9001:2000 garantiza que la propuesta se encuentre acoplada al nivel 2 o incluso 3 del mencionado modelo. De esta manera ha quedado descartado el modelo CMMI y se selecciona por excelencia la Norma ISO 9001:2000 para la gestión de la calidad de los procesos.

2.7 Conclusiones.

En el transcurso de este capítulo se han llevado a cabo los primeros pasos de lo que se conoce como Gestión por procesos. Un proceso no es más que transformar entradas (insumos) en salidas, (bienes y/o servicios) por medio del uso de recursos físicos, tecnológicos, humanos, etc. Este tipo de gestión es ampliamente utilizada debido a los resultados que ha reportado a las diversas empresas que la han puesto en práctica.

La identificación y posterior descripción de los procesos que se llevan a cabo en Nova Linux se ha realizado de forma concienzuda a través de diversas entrevistas a los actuales desarrolladores del proyecto. El proceso de Liberación es el encargado de brindar un resultado final, el mismo a su vez

está dividido en subprocesos que facilitan su comprensión. La diagramación de este proceso garantiza claridad en el mismo, además de hacer evidente los problemas que surgen durante el desarrollo del producto.

Para el establecimiento de una línea de producción de software en el proyecto Nova se hace necesaria la optimización del proceso de liberación, para ello, durante este capítulo se lleva a cabo una identificación de los problemas que se evidencian en dicho proceso, el cual se encuentra en correspondencia con las buenas prácticas de la Ingeniería de Liberaciones. Los principales problemas que se presentan están relacionados con omisiones de pasos fundamentales, así como inclusiones de procesos innecesarios en la misma.

Por último se realiza la selección de la metodología de desarrollo de software y del modelo de calidad a implantar junto a la LPS en el proyecto. La metodología seleccionada es Scrum, debido a un grupo de ventajas y elementos adaptables que posee. Así mismo el modelo de calidad seleccionado es la Norma ISO 9001:2000, que colabora de la misma manera con la metodología y con la gestión de la calidad.

3

Capítulo 3. Propuesta de Línea de Producción de software para el proyecto Nova Linux.

3.1 Introducción.

En el presente capítulo se hace referencia a la forma de optimizar el proceso de liberación en el proyecto Nova, así como las vías que los autores de la presente investigación proponen con el fin de erradicar los diversos problemas que existen dentro del mismo y que a su vez se manifiestan en capítulos anteriores.

El principal objetivo del capítulo radica en la realización de la propuesta para la reestructuración de la metodología de desarrollo de Nova en una Línea de Producción. La propuesta consta primeramente de soluciones a los principales problemas encontrados en el transcurso de la investigación, lo cual se contempla como un proceso de optimización, que es estrictamente necesario antes de la aplicación de la filosofía de una línea de producción.

Los principales problemas a optimizar radican en el no acoplamiento del proceso de liberación en su totalidad a un sistema de control de versiones; además de que no se encuentran unificadas las diversas vías existentes para la creación de las personalizaciones que se generan, lo cual no permite el cumplimiento del objetivo propuesto, además de que ambos aspectos violan los principios de la Ingeniería de Liberaciones.

La LPS propuesta es expresada según los criterios mundiales para el establecimiento de las mismas, o sea, a través de una serie de pasos que contemplan tanto la metodología de desarrollo de software, como el modelo de calidad a implantar y a su vez y como elemento más importante el proceso de liberación que permite la obtención del producto final.

3.2 Optimización del Sistema de Control de Versiones implementado en el proyecto Nova.

En el entorno que rodea el desarrollo de software se hace cada vez más frecuente, debido a su indiscutible necesidad, el uso de algún sistema de control de versiones; fundamentalmente los proyectos de amplias dimensiones en los cuales el trabajo colaborativo debe constituir su principal premisa. Esto proporciona ventajas indiscutibles como: la reducción del trabajo basado fundamentalmente en la reutilización de código e información, así como la reducción del tiempo de

desempeño.

De forma general el control de versiones se basa en la revisión o edición de un producto o componente del producto, teniendo en cuenta el estado en que se encuentra en un momento dado de su desarrollo o modificación. Por lo que un sistema de control de versiones realiza la gestión de los diversos cambios que se realizan sobre los elementos que componen algún producto, incluyendo incluso los cambios en la configuración del mismo. Este tipo de sistemas facilitan la administración de las distintas versiones de cada producto desarrollado así como las posibles personalizaciones realizadas. Este tipo de gestión aporta ventajas significativas como:

- El seguimiento histórico de los archivos y directorios a través de copias y renombrados.
- Las modificaciones (incluyendo cambios a varios archivos) son atómicas.
- Se envían sólo las diferencias en ambas direcciones.
- Maneja eficientemente archivos binarios.
- Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.

Las funcionalidades principales que ofrece son:

- Un mecanismo de almacenaje de los elementos que deba gestionar (como archivos de texto, imágenes, documentación).
- La posibilidad de realizar cambios sobre los elementos almacenados (es el caso de modificaciones parciales como: añadir, borrar, renombrar o mover elementos).
- Un registro histórico de las acciones realizadas con cada elemento o conjunto de elementos (normalmente pudiendo volver o extraer un estado anterior del producto).

Es importante especificar que aunque no es estrictamente necesario suele ser de mucha utilidad la generación de informes con los cambios introducidos entre dos versiones, informes de estado, marcados con nombres identificativos de la versión de un conjunto de ficheros, o de alguna otra forma.

Durante el Capítulo 2 de la presente investigación, uno de los principales problemas que presenta Nova dentro de su proceso de Liberación, es que no se versionan algunos elementos de vital

importancia en el desarrollo de las personalizaciones; entre ellos se encuentra las políticas para la construcción del software y algunos archivos de configuración del sistema. Además se presentan algunos problemas referentes a la forma de gestionar las versiones de algunos elementos como la documentación de las distribuciones. De forma general todas las áreas de trabajo del proyecto no se encuentran acopladas a un Sistema de Control de Versiones; aún en el caso particular del empleo de Subversion como sistema con esta finalidad dentro del mismo.

Subversion es un sistema de control de versiones libre y de código fuente abierto, permitiendo el manejo de ficheros y directorios a través del tiempo. Su funcionamiento radica en la existencia de un árbol de ficheros en un *repositorio* central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Esto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos. El acceso al repositorio es a través de la red, lo que permite que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus ordenadores.

La estructura que este sistema tiene en el proyecto esta basada en el almacenamiento centralizado, consta del ya mencionado repositorio central o servidor, el que se relaciona con otros 4 repositorios los cuales están distribuidos de acuerdo a ciertos criterios relacionados con las personalizaciones: Beta, Estable, Estándar y Laptop; en ellos se versionan diversos elementos como el kernel empleado, los archivos de configuración del sistema, el overlay del proyecto, la paquetería probada con anterioridad, entre otros. A la totalidad de estos elementos se puede tener acceso por medio del servicio WebSVN.

En las entrevistas realizadas para medir la factibilidad del uso del control de versiones por medio de Subversion dentro del proyecto se han detectado un grupo de deficiencias que deben ser resueltas en pos de futuras mejoras en el proceso de desarrollo de cualquier personalización de Nova. Uno de los principales problemas que se detectan lo constituye el uso inadecuado que se le ha dado al sistema elegido, en el cual trabajan pocos desarrolladores haciéndolo de forma muy esporádica. Otra falla detectada es que por medio del Subversion implementado no se están gestionando los archivos relacionados con la configuración del entorno de escritorio.

La idea de acoplar en su totalidad un sistema de control de versiones al proceso de desarrollo de Nova y que se ejecute eficientemente, está dada por la necesidad de mantener dentro del proyecto un registro histórico y minucioso de todo el trabajo que en él se realice, en pos de simplificar el mismo a la hora de pasar a documentar el proceso de desarrollo. A la par de mantener un registro de cada uno de las modificaciones realizadas en las diversas etapas por las que se transcurre con relación a la

compilación de paquetes, cambios de archivos y de configuración que facilite la reutilización de componentes; lo que conlleva a la disminución del tiempo y del costo de recursos humanos y tecnológicos.

Para lograr lo antes planteado es necesaria la reestructuración del actual sistema de control de versiones, sin violar su filosofía de manejo de datos. Por lo que se propone mantener los repositorios existentes, a los cuales se les agrega uno de configuración, donde radiquen los archivos de configuración de entorno de escritorio. Por último debe ser agregado un repositorio correspondiente a la **gestión documental**. En este caso debe colocarse en él no sólo la documentación referente a las diversas personalizaciones, sino además las políticas que permiten conocer como se desarrolla el producto y los diversos Release Backlog que genere la metodología seleccionada. Es importante recomendar el acoplamiento de la totalidad del proyecto a este sistema, además de incentivar a un trabajo colaborativo y continuado de los desarrolladores a este respecto.

3.3 Optimización del Proceso de Liberación empleado dentro del proyecto Nova.

La ingeniería de Liberaciones dentro de sus principios expresa la necesidad de documentar el proceso de liberación contemplando las vías de creación para diversos productos; teniendo en cuenta además que el proceso debe ser único.

En capítulos anteriores se evidencia esta dificultad dentro del proyecto Nova, ya que durante el proceso de liberación documentado no se contemplan las vías para llegar a las diferentes personalizaciones, provocando la existencia de una variedad de documentación al respecto, aunque sea idéntica; o sea, existe un proceso documentado para cada una de las personalizaciones.

Para solucionar este aspecto que de cierta manera viola los principios de la Ingeniería de Liberaciones y afecta el desarrollo fluido del proyecto se hace necesario realizar una propuesta del proceso de liberación, manteniendo los patrones establecidos por los desarrolladores y líderes del proyecto.

Las principales diferencias entre las distribuciones de Nova radican en dos procesos fundamentales:

- Corte en la versión del repositorio: Se selecciona la paquetería que se debe instalar teniendo en cuenta los requisitos establecidos por el usuario.
- Configuración y personalización de la interfaz: En dicho paso se personaliza la distribución según los requisitos que el cliente establece.

En ambos casos existe un factor común que influye directamente en la diferenciación del proceso de liberación para cada una de las distribuciones: El levantamiento de requisitos. Este subproceso es indispensable a la hora de establecer un contrato con el cliente, el cual se encarga de indicar a los desarrolladores las características que desea en su producto.

El proceso general de liberación cuenta con 13 subprocesos que definen pasos a seguir durante el desarrollo del producto. Como se puede apreciar en párrafos anteriores solo dos afectan la optimización del proceso general. En este caso la reutilización tanto de componentes como de otros recursos puede agilizar el desarrollo del producto final.

Es recomendable la optimización del proceso de liberación teniendo en cuenta la filosofía que implementan algunas herramientas como **Catalyst.** El proceso debe ser multifacético y debe permitir construir todos los aspectos de una Liberación de Nova sin necesidad de ir al principio del mismo. A modo de explicación se puede plantear la necesidad de que después de seleccionado los elementos a emplear en la personalización no halla necesidad de volver iniciar el proceso para una próxima versión de una personalización o para la creación de un LiveCD, se hace indispensable que solo se ejecuten cambios en la configuración del sistema. Este aspecto se vería facilitado por un sistema de control de versiones, así como el empleo de alguna herramienta que permita esta alta reusabilidad de componentes.

Una propuesta del proceso de liberación optimizado se muestra en la tabla 4, clasificando los procesos de acuerdo a los criterios que la metodología de gestión por procesos ofrece.

Procesos	**Descripción**	**Clasificación**
Reunión Inicial de planeamiento	Se crea el Cronograma del Release, se definen las características iniciales y otras informaciones pertinentes. A partir de la reunión se hace un levantamiento de requerimientos y funcionalidades para el nuevo sistema. Se congela el documento de los requisitos. A partir de ésta fecha no se aceptarán nuevas funcionalidades.	Estratégico
Corte en la versión de repositorio	Se selecciona una versión del repositorio a trabajar para construir los binarios del release. A partir de éste momento se comienza a construir los binarios y el sistema base a utilizar.	Clave

	Los paquetes nuevos serán agregados o actualizados en dependencia de su impacto en el sistema (se cubre una funcionalidad o riesgos de seguridad).	
Pruebas de construcción	Se le realizan pruebas a cada uno de los paquetes binarios después de vinculados al sistema, y se chequea el correcto funcionamiento de cada una de las aplicaciones incluidas.	Clave
Actualización de la documentación	Se crea o actualiza la documentación sobre el release (versión del núcleo, funcionalidades añadidas, cambios respecto a versiones anteriores, manual de instalación)	Soporte
Se congela el sistema.	Se congela el sistema, para crear una versión Alpha a distribuir a miembros de la comunidad para un proceso amplio de prueba. Se descongela el sistema. Se comienza la etapa de corrección de errores. (Esta etapa puede repetirse en dependencia de la cantidad y complejidad de los defectos encontrados)	Clave
Se publica sistema Beta	Se redacta la información del sistema Beta, y se revisa hasta que quede lista para publicar. Se sube el ISO del sistema Beta para un servidor público, se actualiza la pagina principal del sitio y se envía la noticia a los sitios externos (softwarelibre.uci.cu)	Estratégico
Actualización de la documentación	Se actualiza la documentación sobre el release (versión del núcleo, funcionalidades añadidas, cambios respecto a versiones anteriores, manual de instalación)	Soporte
Se congela el sistema (Final)	Se congela el sistema por última vez. A partir de ésta fecha no se aceptan actualizaciones al sistema, se comienza la etapa de configuración y personalización de la interfaz.	Clave
Se prepara liberación (etapa final)	Se revisa por última vez el sistema, se comprime y crea el CD de instalación.	Clave

Se publica el release (a los miembros del proyecto)	Se suben todos los recursos (sistema, manuales etc.) para el servidor donde se va a publicar, pero sin acceso público a los mismos.	Estratégico
Publica liberación	Se hacen públicos todos los recursos, se actualiza página principal del sitio y se envía la noticia a los sitios externos (softwarelibre.uci.cu)	Estratégico

Tabla 4.

Se recomienda la realización de un proceso denominado Aseguramiento de la Calidad de forma individual al proceso de liberación; con preferencia, el mismo debe ser realizado previamente al comienzo del proceso de release, este proceso tiene como principal objetivo realizar pruebas individuales a los diversos paquetes binarios que se van a utilizar en la próxima liberación, pero con anterioridad, lo cual agilizaría el proceso final, además de garantizar que la paquetería llegue en un estado de calidad superior al sistema. Este proceso debe ser llevado a cabo de forma independiente y constante.

En las figura 10 se muestra el diagrama de procesos que representa al proceso de Liberación optimizado de cualquiera de las personalizaciones producto del proyecto Nova.

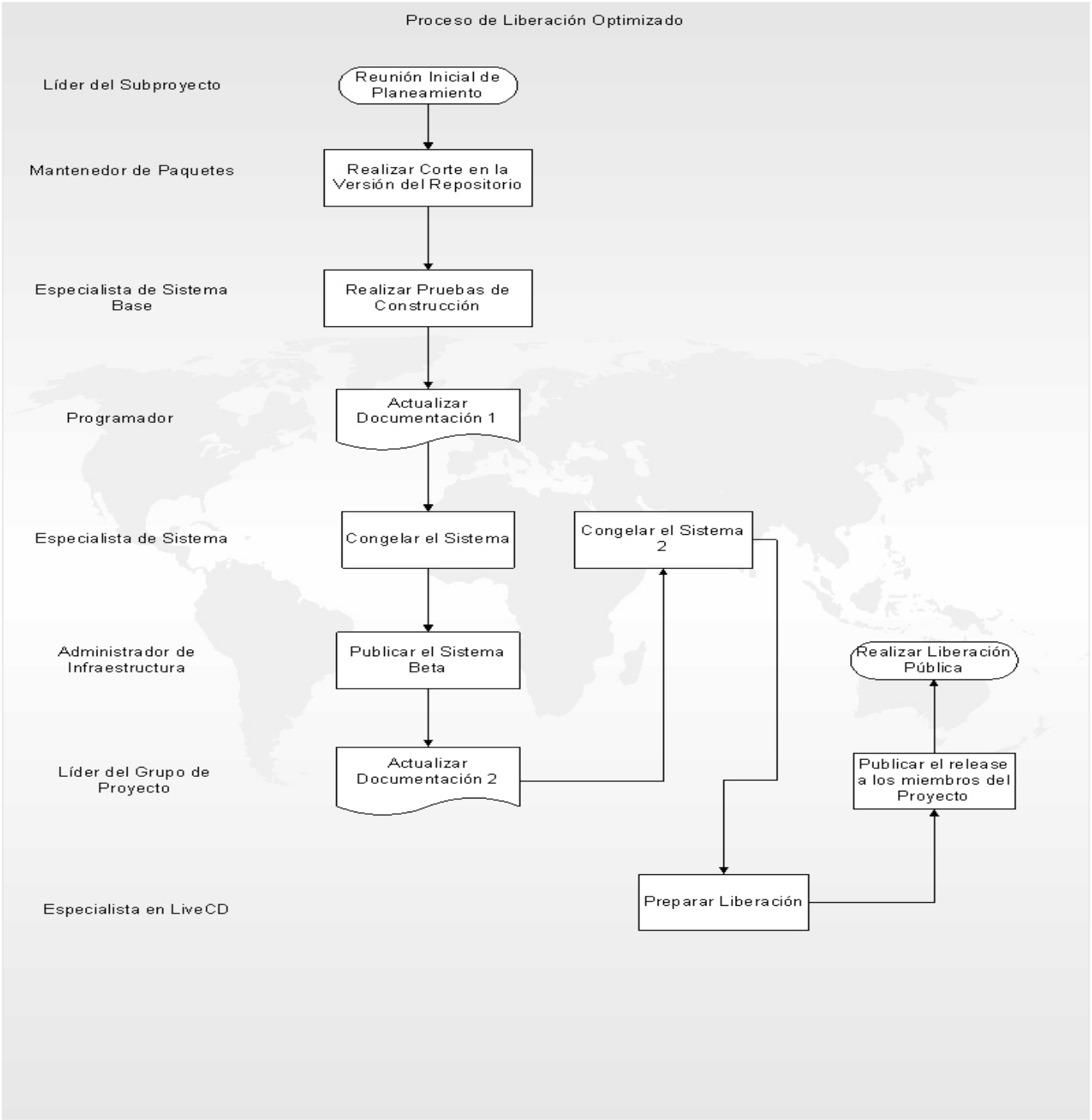


Figura 10.

Durante la optimización del proceso de Liberación para cualquiera de las personalizaciones que se llevan a cabo en el proyecto Nova se vieron afectados tres subprocesos del mismo:

- La reunión inicial de Planeamiento.
- El Aseguramiento de la Calidad y las pruebas de Construcción.
- Publicación del Sistema Beta.

Estos subprocesos sufren cambios con el fin de lograr un proceso general óptimo. En el caso de la Reunión Inicial de Planeamiento, a la misma se le incorpora el subproceso de levantamiento de requisitos. Como se había planteado anteriormente el aseguramiento de la calidad pasa a ser un proceso constante e independiente del proceso de liberación, o sea, el segundo subproceso queda compuesto solamente por la realización de las pruebas de construcción. En el último caso, al subproceso de publicación del sistema Beta se le incorpora la creación de información sobre el Beta, o sea, nuevamente se unen dos subprocesos en uno.

En las figuras 11, 12 y 13 se muestran los diagramas de flujo correspondientes a los subprocesos anteriormente mencionados con las optimizaciones correspondientes.

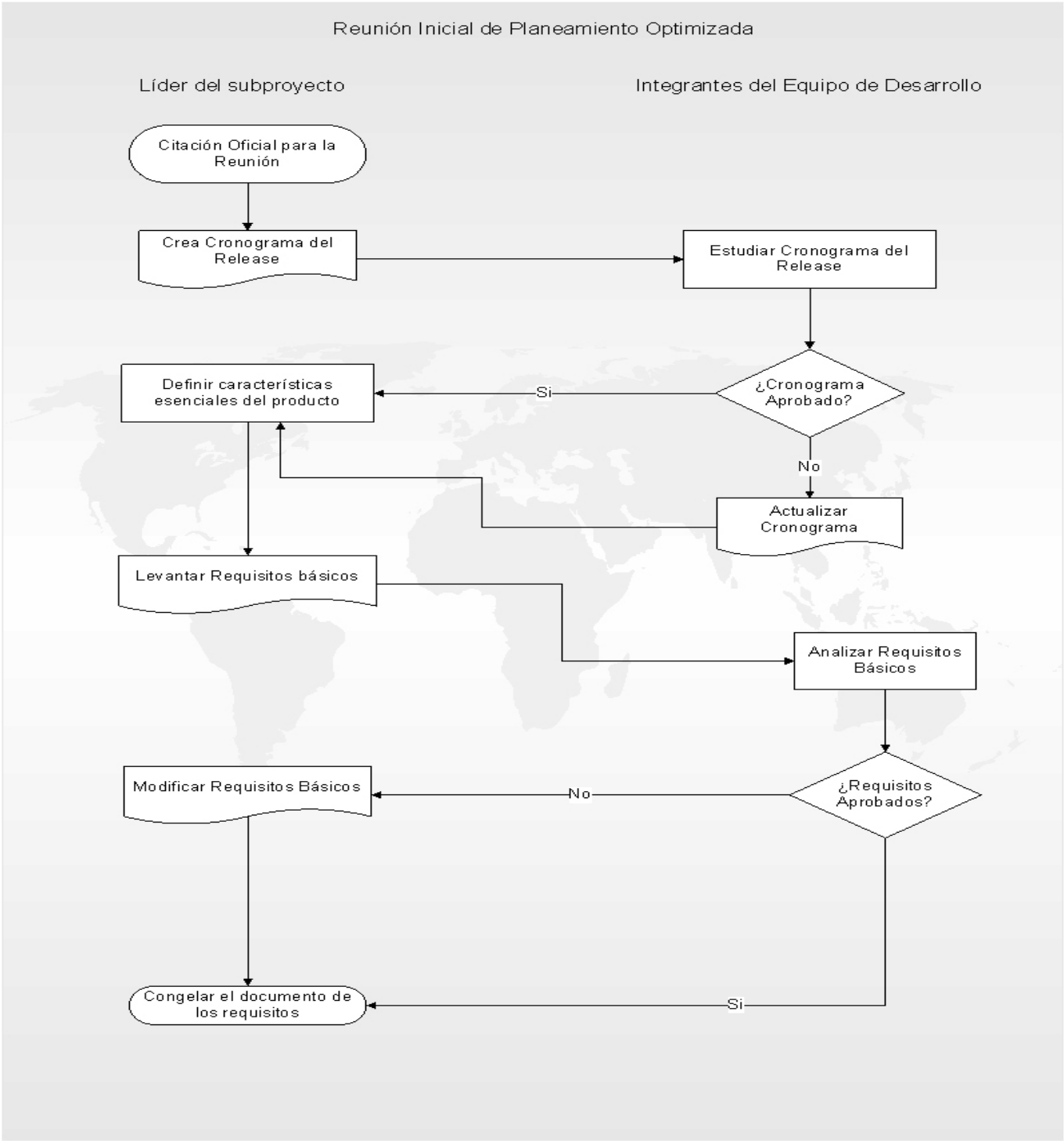


Figura 11. (Reunión Inicial de Planeamiento Optimizada).

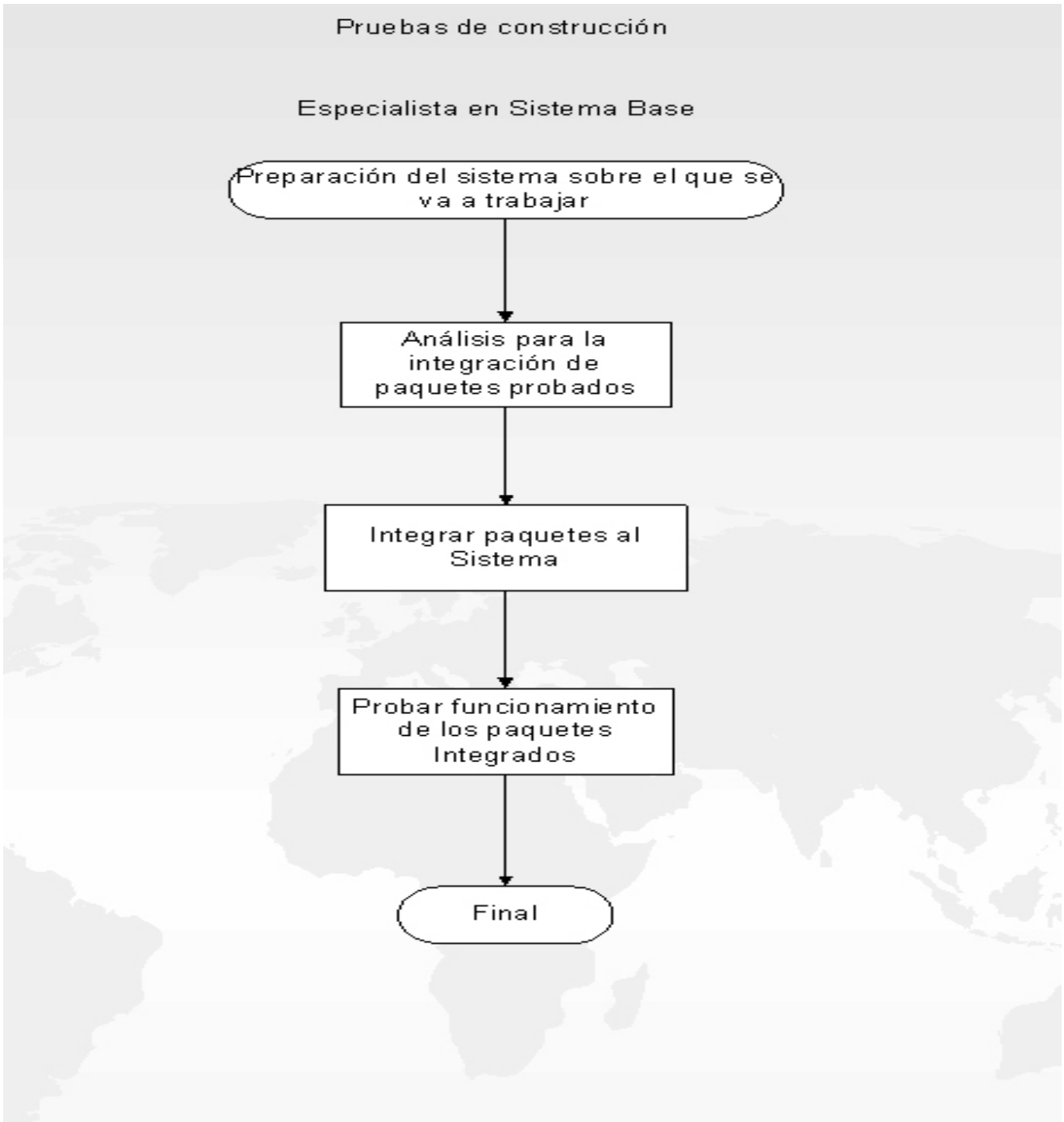


Figura 12. (Pruebas de Construcción Optimizadas).

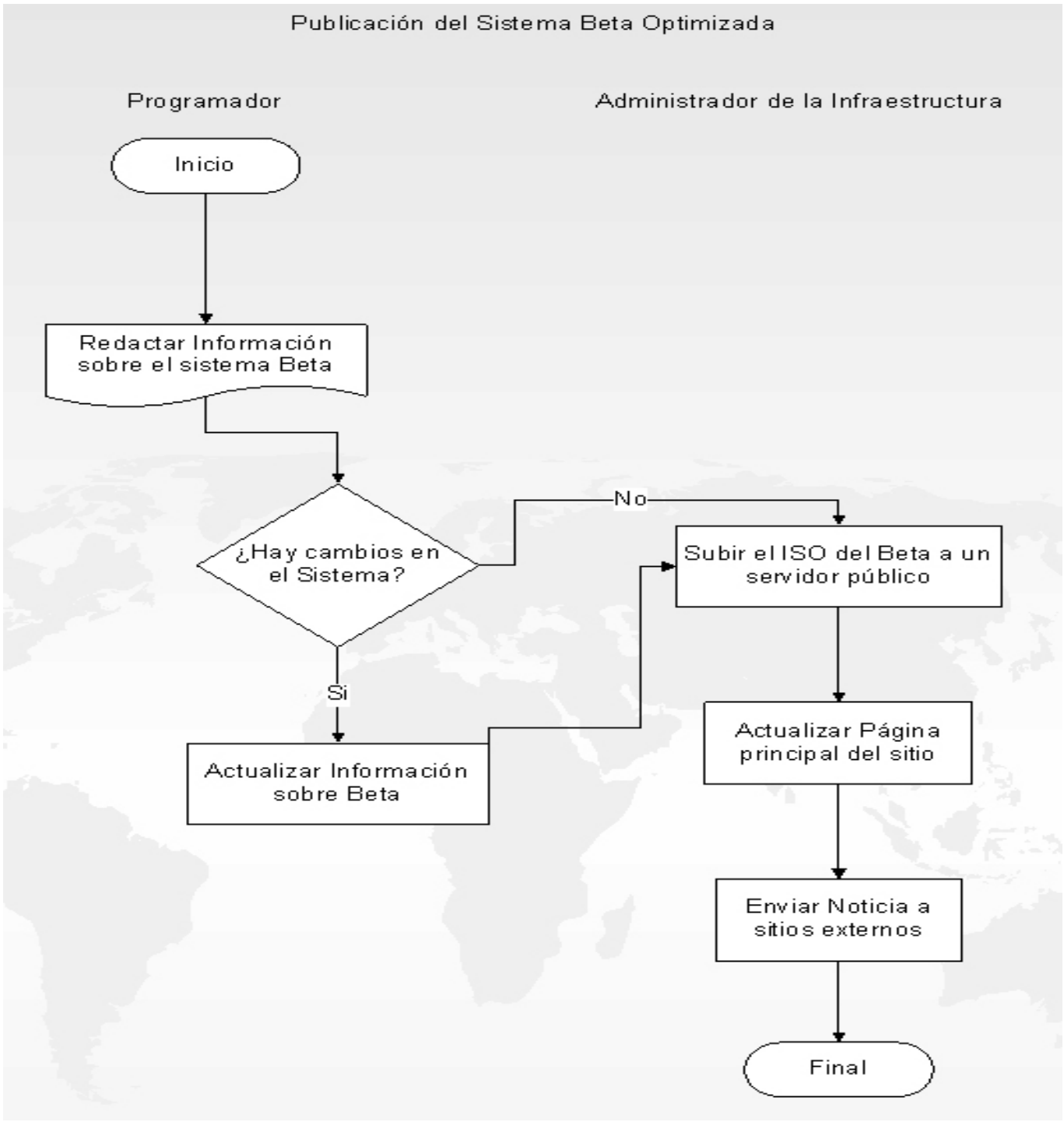


Figura 13. (Publicación del Sistema Beta Optimizada).

3.4 Línea de Producción para el proyecto Nova.

Para el establecimiento de la línea de producción objetivo de la presente investigación se hace necesaria la realización de un grupo de recomendaciones referentes tanto a la metodología de desarrollo de software seleccionada como al modelo de calidad a implementar.

La primera recomendación y la más importante es la capacitación en profundidad sobre la metodología de desarrollo y el modelo de calidad seleccionados, Scrum e ISO 9001:2000 respectivamente. Esta recomendación debe ser considerada como el eslabón fundamental para lograr el éxito en la implantación de ambos elementos, ya que es estrictamente necesario un conocimiento pleno al respecto por parte del equipo de desarrollo.

La propuesta consiste en la aplicación de un Scrum de Scrums, o sea, pequeños equipos de desarrollo Scrum que garantizan la realización de las diversas personalizaciones, y que a su vez forman parte de un equipo Scrum superior; de cierta manera se tendría en cuenta cada equipo de desarrollo como un miembro del equipo general del proyecto.

Scrum propone equipos de 5 a 9 personas, después del estudio realizado se concreta en la propuesta realizar la recomendación al proyecto Nova de crear equipos de desarrollo de 9 integrantes, teniendo en cuenta que debe existir un equipo por cada una de las personalizaciones que se llevan a cabo; además de un equipo que se encargue de compilar y testear los binarios para cada una de las personalizaciones, el mismo debe trabajar para el resto de los equipos, ocupándose del aseguramiento de la calidad del producto, proceso que se recomienda independizar en subepígrafes anteriores. El equipo de aseguramiento de la calidad debe ser puesto en funcionamiento con anterioridad a cualquier desarrollo personalizado que se proponga.

Dentro de los equipos y a pesar de no existir roles específicos dentro de la metodología se propone incorporar especialistas que contribuyan con la fluidez del proceso y hagan menos engorroso el trabajo a realizar. Un equipo debe estar conformado por:

1. Líder Scrum.
2. Especialista en Sistema.
3. Especialista en LiveCD.
4. Mantenedor de Paquetes (todos).
5. Programador (todos).

Es necesario especificar que a pesar de que se necesitan dos especialistas, el equipo en su totalidad debe poseer las competencias necesarias para los actuales roles de mantenedor de paquetes y programador.

La metodología propone además la realización de reuniones diarias durante el transcurso de cada Sprint que se realice, teniendo en cuenta en ellas el seguimiento y la medición de la conformidad del producto con los requisitos establecidos. Estas reuniones no fueron tenidas en cuenta dentro de la LPS debido a la complejidad de su representación en un proceso que debe ser realizado siguiendo la filosofía de pasos consecutivos. Aún así se recomienda realizar dichas reuniones, cada día a la misma hora, dirigidas por el Líder Scrum junto a su equipo de desarrollo, mientras dure el Sprint o iteración.

El grupo de desarrollo Nova debe definir a conciencia los clientes que formarán parte de los diversos equipos Scrum, en dependencia de la personalización que es de interés para cada uno de ellos. Esto es de vital importancia, ya que es una de las ventajas fundamentales del desarrollo ágil: la participación de los clientes como parte de los equipos de desarrollo.

Se hace imprescindible aclarar que durante la confección de la LP no se tienen en cuenta los nueve factores claves para la implantación de Scrum, ya que el proyecto que compete no sienta sus bases en la filosofía de empresas con oficinas especializadas. Por tanto la LPS propuesta posee sus pilares en el desarrollo básico de la metodología de desarrollo de software seleccionada.

El uso de lo que dentro de la metodología se conoce como Product Backlog se hace un poco ineficiente debido a la filosofía del producto que compete, de ahí que se proponga el uso de lo que se conoce dentro de Scrum como Release Backlog, que no es más que un Product Backlog pero restringido a un release o liberación del producto.

El modelo de calidad a implementar requiere de la creación de la documentación del sistema de gestión de calidad, o sea, la documentación requerida por la Norma ISO 9001:2000, la misma incluye fundamentalmente la política de calidad del proyecto, los objetivos de calidad generales de los diversos productos y los específicos a tener en cuenta en cada Sprint que se realice. Además la documentación debe incorporar un manual de calidad. Todo este proceso de creación debe estar regido por la normativa propuesta.

La creación de un Equipo Scrum que se encargue del buen funcionamiento del modelo de calidad implementado, así como de ejecutar los procesos referentes a esta parte tan importante de la LPS es indispensable. De esta manera se deben crear las condiciones para el buen funcionamiento de dicho

equipo, cuya única tarea debe estar vinculada a lo anteriormente expuesto. Es válido recalcar que este equipo se ocupará de los procesos de calidad de manera general en todo el proyecto.

Los líderes del proyecto deben determinar las competencias necesarias para el personal que realiza trabajos que afectan la calidad del producto. En caso de ser necesario debe llevarse a cabo el proceso de formación con el objetivo de preparar correctamente a los desarrolladores. De este tipo de actividades de formación la normativa recomienda mantener un registro minucioso.

Otra de las recomendaciones de importancia referentes al modelo de calidad a implantar es realizar verificaciones respecto al cumplimiento de los diversos requisitos establecidos con anterioridad (del producto, legales y reglamentarios).

Se hace imprescindible aclarar que la realización de cambios durante el Sprint, ya sean en el sistema de gestión de la calidad o en los requisitos debe ser estrictamente registrada.

Según los resultados arrojados por esta investigación el grupo de desarrollo Nova garantiza actividades posteriores a la entrega del producto. A pesar de ello las mismas deben ser refinadas en pos del cumplimiento de la Norma establecida.

La necesidad de proporcionar evidencia de la conformidad del producto con los requisitos especificados provoca que se haga imprescindible determinar las actividades de seguimiento y medición a realizar con este fin, así como los dispositivos necesarios para ello.

Se recomienda además la realización de auditorías internas a intervalos de tiempo relativamente cortos verificando el correcto funcionamiento del Sistema de gestión de la calidad establecido. Estas auditorías deben ser planificadas al inicio de cada Sprint.

Los autores de la presente investigación recomiendan al equipo de desarrollo de Nova realizar una revisión de los requisitos relacionados con cada uno de los productos antes de crear compromisos contractuales con el cliente. Esto garantizará la posibilidad de lograr un resultado final con calidad, ya que el equipo determinará su capacidad para llevar adelante la creación del producto especificado.

La Línea de Producción de Software debe quedar de la siguiente forma:

1. *Creación del Release Backlog*: Este proceso es realizado por el Propietario del Producto, el cual se encargará de desarrollar una lista de requisitos del producto. En dicha lista los requisitos deben estar ordenados por prioridad, la cual es establecida únicamente por el

propietario del producto. Dicha prioridad debe ser tenida en cuenta para la creación del Sprint Backlog.

2. *Reunión de Planificación del Sprint*: Esta es planificada y llevada a cabo por el Líder Scrum. En la misma debe ser creado el cronograma de trabajo. Además la tarea más importante dentro de esta reunión es la creación del Sprint Backlog, donde se determinarán los requisitos que debe tener la muestra funcional del producto al finalizar la presente iteración, indicando las tareas específicas que realizará cada integrante del equipo durante el proceso de desarrollo. En este momento deben ser establecidos los objetivos de calidad del producto final y del resultado que será arrojado por el Sprint; también deben ser establecidos los criterios de aceptación del release generado en la iteración. Se hace necesario además analizar los requerimientos legales y reglamentarios que deben ser incorporados en el Sprint Backlog, al igual que aquellos requisitos necesarios no establecidos por el cliente. También debe realizarse una planificación exhaustiva de las auditorías internas que serán realizadas al sistema de gestión de la calidad, determinando el momento en el que serán llevadas a cabo las mismas. Al Líder Scrum le queda la tarea de seleccionar las herramientas que empleará en el presente proceso. Esta reunión de planificación del Sprint es fundamental dentro del proceso de liberación, ya que en ella se establecen todos los patrones a seguir durante el desarrollo del release.

3. *Realizar corte en la versión del repositorio*: Este proceso es llevado a cabo por cualquiera de los integrantes del equipo Scrum, dicha tarea será asignada durante la reunión de planificación del Sprint. El principal objetivo de este proceso es seleccionar tanto la versión del stage1 que se empleará, como el árbol de portage, que a su vez posee aquellos paquetes que serán incluidos como parte del sistema operativo que se va a liberar. Dichos paquetes seleccionados deben haber sido testeados con anterioridad por el equipo de aseguramiento de la calidad. Luego esta paquetería debe ser compilada y debe procederse a la construcción de los binarios respectivos. Es importante esclarecer que las versiones que sean seleccionadas deben ser mantenidas mientras el sistema esté en desarrollo, aunque salgan nuevas versiones que se adapten mejor a los requerimientos. El equipo debe buscar alternativas para la automatización de este proceso.

4. *Realizar pruebas de construcción*: Esta tarea es ejecutada por el Especialista en Sistema que forma parte del Equipo Scrum. El primer paso del presente proceso es la preparación del sistema que se va a utilizar durante la integración de los paquetes, esta es una forma de

garantizar la limpieza del producto, ya que sobre él solamente se llevan a cabo pruebas de integración, verificando que los paquetes son compatibles unos con otros. Posteriormente deben ser analizados los paquetes para su integración, teniendo en cuenta si cumplen con requisitos funcionales o cubren riesgos de seguridad. Luego se lleva a cabo la integración de los binarios al sistema ya preparado y se prueba su funcionamiento en conjunto. En dependencia del resultado que arrojen estas pruebas se continúa el desarrollo o se regresa al proceso de corte en la versión del repositorio; también es posible corregir los errores a través de parches o empleando el overlay del proyecto.

5. *Actualizar documentación 1*: Es realizada por uno de los integrantes del equipo Scrum, designado con anterioridad. Este proceso se realiza con el fin de que el release correspondiente esté adecuadamente documentado. El primer paso que se lleva a cabo es la creación o actualización de la documentación de la liberación correspondiente, o sea, la versión del núcleo, las funcionalidades añadidas, los cambios respecto a versiones anteriores y el manual de instalación del producto. Esta es una forma eficiente de garantizar que al terminar el sistema se cuente con una gran parte de la documentación que se requiere, resaltando en la misma las diferencias respecto a las versiones anteriores del sistema.

6. *Congelar el Sistema 1*: Este proceso es realizado por el Especialista en Sistema miembro del equipo, con el objetivo de realizar pruebas en el mismo y comprobar como cumple con los requisitos establecidos, dichas pruebas se realizan a nivel de usuario. El primer paso es congelar el sistema, para crear una **versión Alpha** a distribuir en la comunidad de software libre dentro de la UCI para un proceso amplio de prueba; o sea, se pone a disposición de los usuarios un sistema sin configurar ni personalizar, con el fin de que el propio usuario sea capaz de hacerlo, y de esta forma envíe información sobre errores que puedan resultar del uso de las aplicaciones incorporadas a la imagen, e incluso del propio sistema operativo. El segundo momento es el descongelamiento del sistema, para la realización de mejoras. Por último se comienza la etapa de corrección de errores, esta etapa puede repetirse en dependencia de la cantidad y complejidad de los defectos encontrados.

7. *Auditar cumplimiento del modelo de calidad*: La auditoría es realizada por el equipo Scrum de Calidad. En este momento el equipo de calidad debe realizar una inspección al equipo de desarrollo con el objetivo de encontrar fallos en la aplicación del modelo de calidad implementado. En caso de encontrar productos que no cumplen con la normativa o violaciones

dentro del equipo de desarrollo al respecto, no es posible continuar con el desarrollo del producto, además de hacerse necesario los análisis pertinentes en pos de tomar medidas.

8. *Publicar el Sistema Beta*: Esta tarea es realizada por alguno de los miembros del equipo Scrum designado con este fin. En dicho proceso primeramente se redacta la información pertinente sobre el Beta, revisándola hasta que quede lista para su publicación; esta revisión se debe fundamentalmente a los errores que reportan los usuarios, ya que a medida que se reparan los mismos si existe algún cambio trascendente debe ser informado. El próximo paso es la publicación del sistema Beta con la finalidad de dar a conocer el nuevo sistema, ya sin errores, pero aún sin configurar, ni personalizar. Para ello se debe subir el ISO del sistema Beta a un servidor público, luego se actualiza la página principal del sitio y como paso final se envía la noticia a los sitios externos, en el caso de la Universidad de las Ciencias Informáticas, el sitio de software libre (softwarelibre.uci.cu).

9. *Actualizar documentación 2*: El presente paso es repetido con el objetivo de actualizar la documentación después de haber realizado cambios en el sistema. O sea, en este caso solo se actualizan aquellas cosas que han cambiado desde el último proceso de actualización. Para ello primeramente se revisa la documentación existente y luego se llevan a cabo los cambios pertinentes. Este proceso es llevado a cabo por alguno de los miembros del equipo designado con este fin.

10. *Congelar el Sistema 2*: Este paso es llevado a cabo por el Especialista en Sistema que integra el equipo de desarrollo Scrum y alguno de los integrantes del equipo destinado a este fin. Al igual que en el paso anterior se realiza la repetición de uno de los procesos, en este caso con el fin de lograr una versión estable del producto, evitando de esta manera cambios innecesarios en pos de un correcto funcionamiento, dedicando los esfuerzos de esta etapa a las configuraciones y a la interfaz del sistema. En este caso un primer paso lo constituye el congelamiento del sistema por última vez; o sea, a partir de ésta fecha no se aceptan actualizaciones o mejoras del producto final. Durante este proceso se comienza la etapa de configuración y personalización de la interfaz. Para este último paso se emplean herramientas que permiten el trabajo con imágenes que quedan a selección de los desarrolladores. Por último se realizan las configuraciones pertinentes a través del empleo de los diversos editores que se emplean en estos casos.

11. *Preparar Liberación*: Esta actividad es llevada a cabo por el Especialista en LiveCD. En este

caso primero se revisa por última vez el sistema, evitando de esta manera que se libere el mismo con errores; luego el mismo es comprimido, garantizando la optimización del espacio para crear el disco de instalación; por último se crea el CD de instalación, quedando de esta forma el sistema listo para su entrega.

12. *Revisión del Sprint*: Esta tarea es planificada y dirigida por el Líder Scrum con la participación de los desarrolladores en su totalidad y el cliente potencial del producto. En esta reunión debe mostrarse el producto a los clientes y deben ser revisados los criterios de aceptación del mismo, verificando su cumplimiento. Además deben ser analizadas las experiencias adquiridas durante el proceso de desarrollo. En el presente proceso debe quedar fijada la fecha para el inicio del próximo Sprint.

13. *Liberación Pública*: Este proceso es llevado a cabo por alguno de los integrantes del equipo definido con anterioridad. El primer paso es publicar el release a los miembros del proyecto en una última fase de prueba a nivel de usuario, junto con la imagen deben ser subidos al servidor toda la documentación referente al sistema. El próximo paso es realizar lo que se conoce como una liberación pública, o sea, se le da acceso a la comunidad de usuarios al sistema en el servidor. Además el producto debe ser entregado al cliente con las debidas formalidades.

De esta forma queda propuesta la LPS objetivo de la presente investigación. Para una visión más clara del proceso de Liberación optimizado y con la respectiva incorporación de los procesos referentes a la metodología de desarrollo y al modelo de calidad propuesto, se puede ver el diagrama de procesos correspondiente en la figura 14.

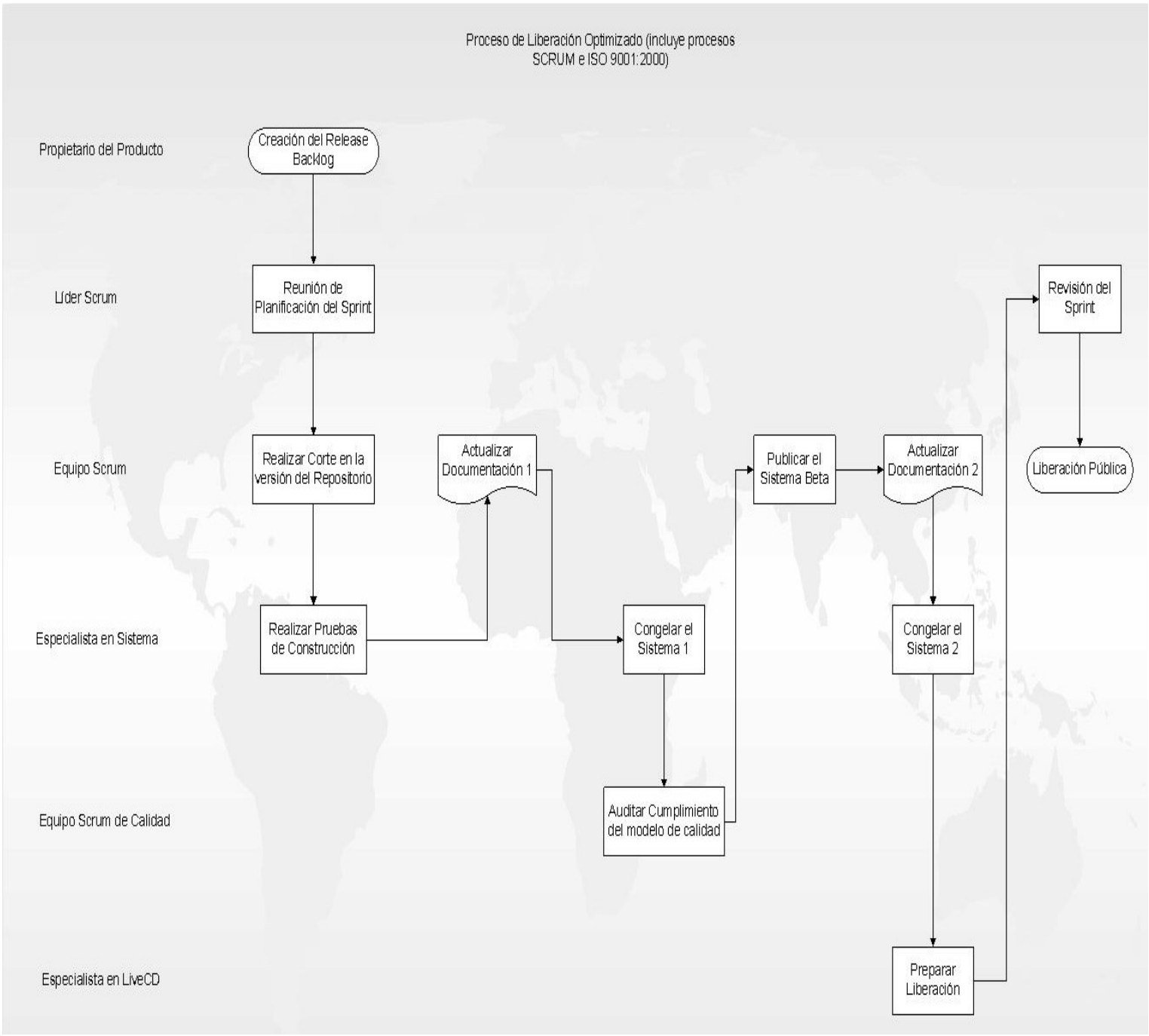


Figura 14.

El surgimiento de nuevos procesos dentro del desarrollo de las diversas personalizaciones de Nova hace necesario mostrar los diagramas de flujo correspondientes que brinda la metodología de gestión por procesos de forma eficiente en este caso. De esta manera en la figura 15 se puede ver el flujo de

actividades del proceso de Creación del Release Backlog.

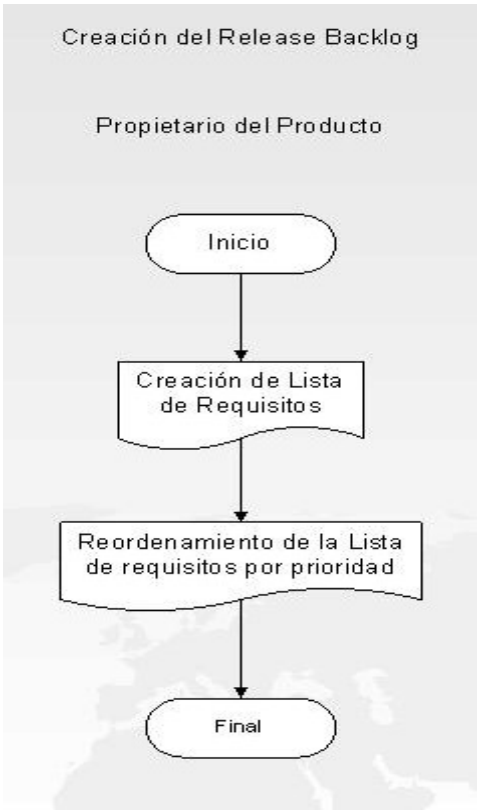


Figura 15 (Creación del release backlog).

En la figura 16 se aprecian las tareas específicas que se realizan durante el proceso de realización de la Reunión de planificación del Sprint.

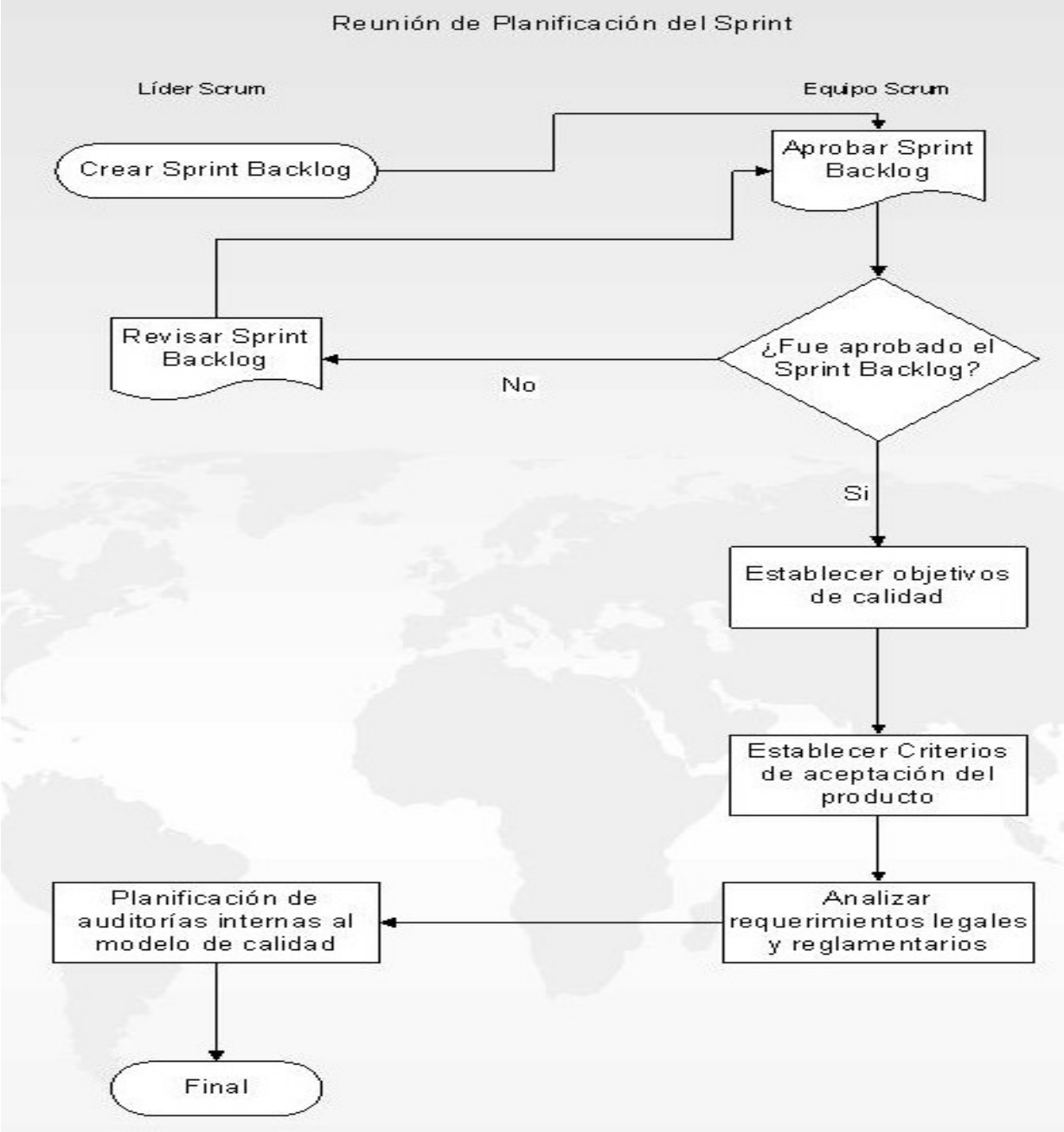


Figura 16. (Reunión de Planificación del Sprint).

En la figura 17 se pueden ver las diversas tareas que se llevan a cabo durante el proceso de auditoría del modelo de calidad.

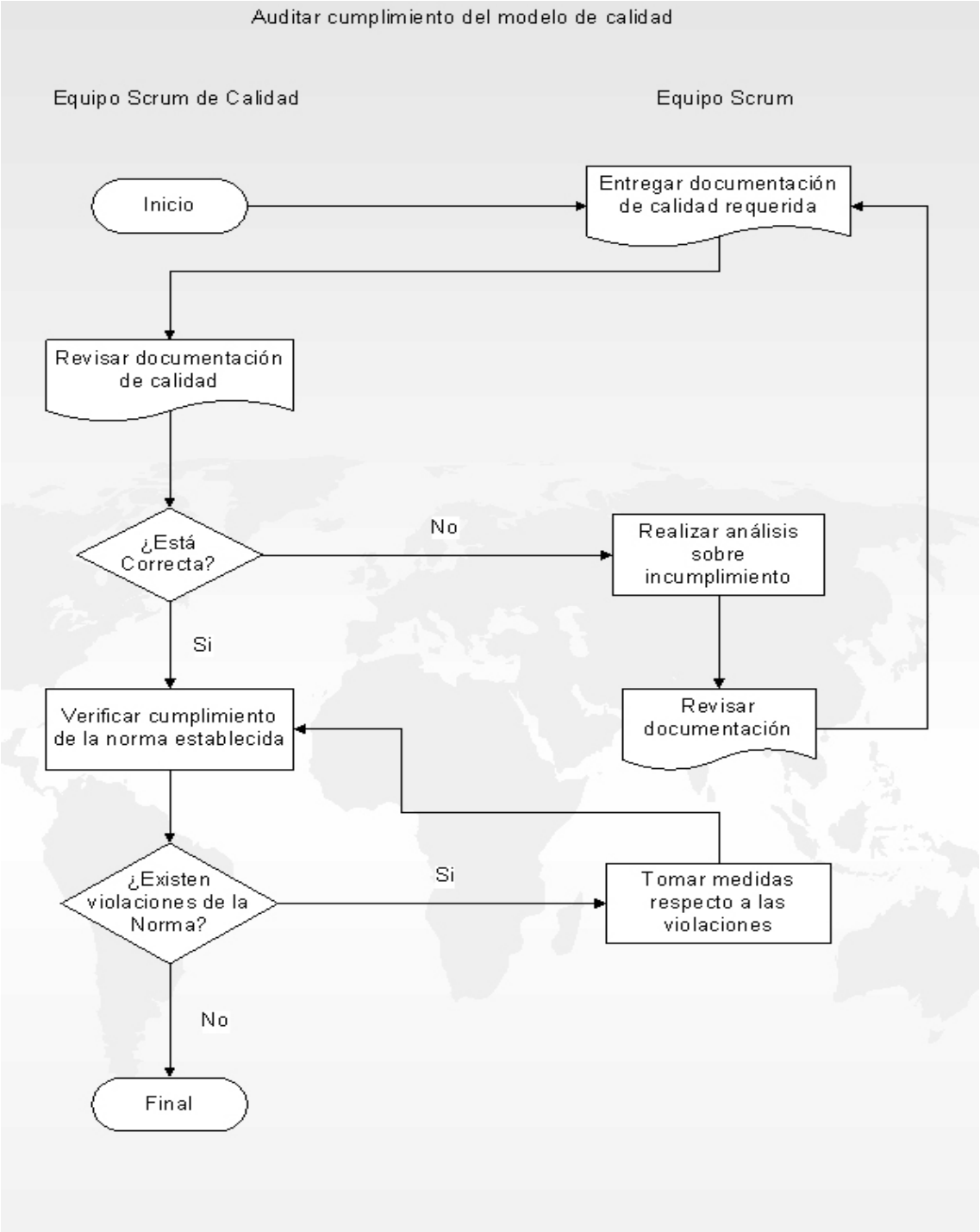


Figura 17. (Auditar cumplimiento de la Norma ISO 9001:2000)

El flujo de actividades del proceso de revisión del Sprint puede verse en la figura 18.

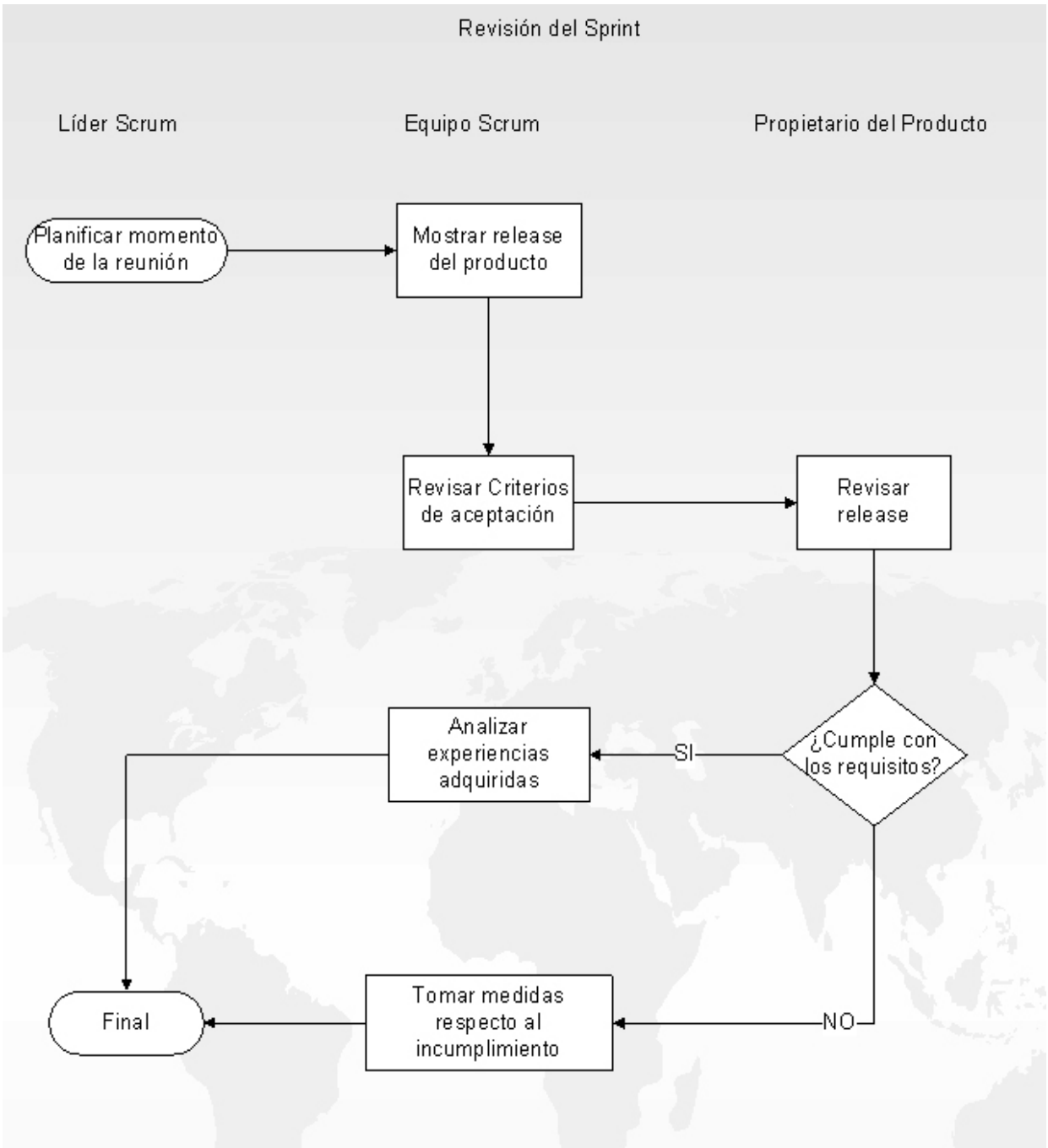


Figura 18. (Revisión del Sprint).

La propuesta de una Línea de Producción para el proyecto Nova Linux está totalmente respaldada por el nivel 2 del modelo de calidad CMMI. La realización de una lista de chequeo que permita soportar el criterio anterior se encuentra debidamente justificada, aún así se hace necesario la realización de una justificación en profundidad de cómo cumple la propuesta con las diversas prácticas genéricas

establecidas por CMMI para ser incorporadas al proceso de desarrollo.

La primera práctica a tener en cuenta durante la adhesión al nivel 2 de CMMI es *Establecer una política organizacional*. El cumplimiento de la misma se evidencia durante la reunión inicial de planeamiento donde se realiza un levantamiento de las expectativas que se tienen del proceso, las cuales son transmitidas a todos los interesados. El principal encargado de llevarla a cabo es el Líder Scrum. Estas expectativas quedan redactadas en forma de plan.

El próximo elemento a tener en cuenta es la *Planificación del Proceso*. En este caso se toma como referencia de cumplimiento el hecho de que la principal tarea del Líder Scrum es velar por la calidad del proceso; teniendo en cuenta de forma constante las necesidades para la ejecución correcta del mismo. Además durante la reunión inicial de planeamiento este es descrito, son establecidos sus objetivos y sus requisitos, se establecen las dependencias entre las diversas actividades que lo componen, se asignan las responsabilidades a cada uno de los miembros del equipo durante su ejecución, quedando establecidas las mismas en el Sprint Backlog. Teniendo en cuenta que además se planifica el proceso de capacitación necesario para los implicados en él.

También deben ser identificados durante esta parte los clientes potenciales que participaran durante el desarrollo como miembros del equipo. Deben ser planificadas las actividades de monitoreo y control y los objetivos de las mismas.

Todos estos elementos son controlados durante las reuniones diarias que se realizan mientras dura el Sprint.

El hecho de *Proveer Recursos* forma parte importante de las prácticas de este nivel. El aseguramiento de los recursos necesarios para la ejecución del proceso se encuentra dentro del plan, ya que Scrum así lo requiere, pero no dependen del proyecto y su gestión. Aún así se buscan soluciones eficientes en pos de no detener la producción; partiendo de este criterio es posible afirmar que se cumplen la mayoría de los elementos de esta práctica.

La *Asignación de Responsabilidades* constituye otra de las buenas prácticas recomendadas en el nivel 2 de CMMI. Durante la reunión de planificación de cada Sprint se asignan las responsabilidades pertinentes a cada uno de los miembros del equipo, además cada uno de ellos posee la autoridad suficiente para la toma de decisiones que puedan afectar su actividad dentro del proceso. Además al estar los miembros del proyecto presentes en la reunión es posible mostrar su acuerdo o desacuerdo con dicha asignación.

Un elemento importante a tener en cuenta es el *Entrenamiento o capacitación del personal*. Durante el proceso el Líder Scrum debe asegurarse de que existen las condiciones para la ejecución del mismo, en caso de necesitarse capacitación, la misma debe ser debidamente planificada y proporcionada con inmediatez. Aún así antes de comenzar con el proceso debe efectuarse un proceso de capacitación para el personal del equipo de desarrollo, teniendo en cuenta las competencias necesarias que requieren los mismos.

El proceso de *Gestionar Configuraciones* forma parte indiscutible de las prácticas establecidas para este nivel. En este caso se puede afirmar que se conserva la integridad de las tareas designadas a cada uno de los miembros del equipo de desarrollo, las mismas son controladas a diario en las reuniones realizadas durante el Sprint, analizando su avance y las necesidades para su ejecución. Además de mantener el debido control de versiones a través de Subversion, lo cual también permite el cumplimiento a cabalidad de la presente práctica.

Otro aspecto fundamental a tener en cuenta es *Identificar e involucrar a los interesados más relevantes*. Antes de iniciar el proceso Scrum solicita la identificación del cliente que formará parte del equipo de desarrollo, lo cual es contemplado en el plan, además de establecer en el mismo su nivel de incorporación al equipo; este cliente o interesado en el producto debe participar de forma activa en la toma de decisiones, así como en el resto de las actividades que se llevan a cabo durante el proceso. Dicha información es debidamente divulgada con todos los involucrados en el proceso.

Una de las prácticas más relevantes dentro del nivel 2 de CMMI es *Monitorear y controlar el Proceso.* En el caso de la propuesta el proceso es debidamente monitoreado y controlado por un equipo encargado de verificar el cumplimiento de la Norma especificada (Norma ISO 9001:2000). Además se controla el proceso teniendo en cuenta la planificación del mismo. En caso de identificar problemas deben aplicarse las respectivas medidas correctivas y darle seguimiento a las mismas.

Como aspecto a tener en cuenta se tiene la *Evaluación objetiva de la adhesión*. En este caso durante el proceso de monitoreo y control se evalúa como se adhiere el proceso a su respectiva planificación, descripción, etc.

Por último debe cumplirse con la práctica que recoge la *Revisión del estado con altos niveles de gestión*. En este caso el líder Scrum de la organización debe incorporarse junto a los diversos Líderes de los equipos de desarrollo y el equipo de calidad a la revisión de las actividades, estado y resultados del proceso específico que se lleva a cabo.

De esta manera queda justificado el cumplimiento de las diversas prácticas que el nivel 2 de CMMI propone en pos de lograr mejoras en el proceso de desarrollo. Las mismas están debidamente controladas a través de una lista de chequeo que se encuentra en el Anexo 3.

3.5 Conclusiones.

En el presente capítulo se ha llevado a cabo la tarea de optimizar aquellos procesos dentro del proyecto Nova que afectan sobremanera la calidad y eficiencia de la producción de las diversas personalizaciones que se desarrollan. La correcta implementación del un sistema de control de versiones, así como el buen uso de la metodología de desarrollo de software y el modelo de calidad seleccionados, Scrum e ISO 9001:2000, respectivamente; son factores fundamentales en la reestructuración del proceso de desarrollo de Nova en una Línea de producción.

La LPS propuesta consta de trece pasos consecutivos que garantizan un consumo mínimo de recursos durante la transición de uno a otro proceso. Los diversos diagramas de los nuevos procesos incluidos en la LPS muestran en detalle el flujo de actividades de estas nuevas tareas a realizar. En la diagramación no se tienen en cuenta algunos procesos, ya que estos fueron descritos en epígrafes anteriores dentro del presente capítulo. De esta manera se realiza una propuesta eficiente de una LPS para el proyecto Nova, recalcando que es ampliamente adaptable a las diversas personalizaciones que se construyen dentro del grupo de desarrollo.

Conclusiones:

El empleo en la actualidad de las Líneas de Producción de software se ha convertido en toda una filosofía; los beneficios que aporta son invaluables, y así ha quedado evidenciado durante la presente investigación. Esta tendencia puede resultar la solución para la Universidad de las Ciencias Informáticas respecto a los problemas que se presentan durante el desarrollo de software.

El grupo de desarrollo Nova resulta una experiencia piloto en este sentido, ya que la investigación realizada a este respecto puede ser adaptada y llevada a cabo en tiempo récord a cualquiera de los proyectos productivos existentes dentro del centro.

El análisis en profundidad realizado por los autores arroja un proceso de liberación optimizado y adaptado a un grupo de requisitos que unido a la metodología de desarrollo de software y al modelo de calidad seleccionados permiten agilizar la entrega de las diversas personalizaciones de Nova Linux, cumpliendo a su vez con los patrones de calidad requeridos; de esta forma se asegura una posible reestructuración del proceso de desarrollo de Nova en una Línea de Producción, en caso de una aceptación positiva de la misma.

El estudio exhaustivo realizado en la presente investigación brinda, como alternativa eficiente para transformar el proceso de desarrollo de Nova, a lo que se conoce hoy en día como metodología de gestión por procesos, la cual permite una sistematización de todos los procesos involucrados en el desarrollo de las diferentes personalizaciones de Nova Linux, ofreciendo además las operaciones a tener en cuenta durante la transformación. Después del estudio realizado a través de dicha metodología es posible establecer un orden para la carga de procesos que se llevan a cabo en el proyecto, permitiendo la creación de la LP; el orden establecido para la misma respeta en todo momento los criterios creativos de desarrollo que se efectúan en estos momentos en el proyecto Nova, además de tener en cuenta ciertos cambios factibles en el orden de los procesos fundamentales que permiten lograr un resultado satisfactorio.

La optimización realizada a los diversos procesos dentro del proyecto tiene en cuenta solo aquellos procesos que, debido a su ineficiencia o características específicas, permiten la realización de la misma. El patrón indicador de esta selección de procesos a optimizar es brindado por la metodología de gestión por procesos, además de una observación rigurosa del desarrollo dentro del proyecto y la realización de entrevistas minuciosas a los implicados directamente en la puesta en práctica de los procesos generales que se llevan a cabo.

La realización del estudio, análisis y descripción de los procesos que se están desarrollando de forma artesanal en Nova, así como la sistematización de las diversas vías de optimización para dicha carga de procesos y finalmente la realización de la propuesta de la Línea de producción de software que soluciona de forma parcial los problemas que se están presentando en el proyecto ha servido de provecho y amplio aporte de conocimientos a los investigadores. El cumplimiento a cabalidad de las tareas propuestas durante la investigación arroja como resultado un conocimiento profundo de los procesos que se llevan a cabo dentro del proyecto que compete, así como una eficiente optimización de los mismos y el establecimiento de un orden adecuado para su ejecución en pos de proporcionar un producto final con la calidad requerida; aspectos que son la base de la filosofía de las líneas de producción en el mundo.

Recomendaciones:

La realización de la presente investigación aporta un grupo de mejoras al proceso de liberación que se realiza en el proyecto Nova Linux; de esta manera, además de contribuir a la eliminación de los diversos errores humanos que pueden introducirse en el proceso de desarrollo, así como facilitar que los desarrolladores se centren más en entender las necesidades de los usuarios, se recomienda la ejecución de un proceso de informatización basado en la propuesta hecha en el presente trabajo. La informatización del proceso de liberación que se lleva a cabo en Nova permite agilizar la entrega de los productos con una calidad superior gracias tanto a la Línea de Producción establecida como a la posterior automatización de la misma.

La posibilidad de resolver problemas similares en otras ramas a través de la filosofía de las Líneas de producción es evidente. La realización de estudios en profundidad de los procesos de desarrollo de los diversos proyectos productivos dentro del centro, su optimización, su organización en instrucciones ordenadas y su posterior informatización beneficiaría tanto a clientes como a desarrolladores; adentrándolos con fuerza en la era moderna del desarrollo de software.

La puesta en práctica de la propuesta realizada durante la presente investigación debe ser llevada a cabo a través de una profunda preparación del grupo de desarrollo Nova, teniendo en cuenta los elementos referentes tanto a la metodología de desarrollo de software como al modelo de calidad a emplear, en este caso Scrum e ISO 9001:2000 respectivamente. El seguimiento preciso de las instrucciones que conforman la propuesta de línea de producción facilitan tanto el proceso de desarrollo como de informatización.

La recomendación de mayor importancia hecha por los autores del presente trabajo consiste en la implantación estricta de la propuesta, fundamentalmente debido a la poca utilidad de la misma en caso de ser fraccionada.

Referencias Bibliográficas:

1. Wikipedia. *Desarrollo de Software*. 2007 [citada 2007 27 de noviembre]; Disponible en: http://es.wikipedia.org/wiki/Desarrollo_de_software.
2. Wikipedia. *Crisis del Software*. 2007 [citada 2007 27 de noviembre]; Disponible en: http://es.wikipedia.org/wiki/Crisis_del_Software.
3. Wikipedia. *Release Building*. 2007 [citada 2007 1 de diciembre]; Disponible en: http://es.wikipedia.org/wiki/Release_Management#2..09Release_Building.
4. Casañola, I.Y.T. *Evento Virtual Informática Habana 2007*. 2007 [citada 2007 4 de diciembre]; Disponible en: http://www.informaticahabana.com/evento_virtual/.
5. M, F.C. *Genexus; Referente de Desarrollo en Latinoamérica*. 2004 [citada 2007 10 de diciembre]; Disponible en: http://www.technologies.cl/HXWNEW01.exe?1,1,30,NEW,111.
6. heinsohn. *Proyectos y Consultoría*. 2007 [citada 2007 11 de diciembre]; Disponible en: http://www.heinsohn.com.co/porfolio/proyectos-consultoria.shtml.
7. elEconomista. *Sadiel instalará una filial de software en la Bahía que tendrá más de 400 trabajadores en 2010*. 2007 [citada 2007 11 de diciembre]; Disponible en: http://www.eleconomista.es/empresas-finanzas/noticias/218471/05/07/.
8. Ing. Jenny Ruiz de la Peña, I.O.A.C., *Importancia de la Ingeniería de Software en la producción de software*, en *Revista Ciencia Holguín*. XIII.

Bibliografía Consultada:

1. Patricio Letelier, M.C.P. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. 2007 [consultada 2007 13 de diciembre]; Disponible en: http://www.willydev.net/descargas/prev/TodoAgil.Pdf.
2. Palacio, J. *Flexibilidad con Scrum*. 2007 [consultada 2007 13 de diciembre]; Disponible en: www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf.
3. Kumar, D.R. *Lean Software Development*. 2007 [consultada 2007 13 de diciembre]; Disponible en: www.projectperfect.com.au/downloads/Info/info_lean_development.pdf.
4. Palacio, J. *Gestión de proyectos ágil: conceptos básicos*. 2006 [consultada 2007 14 de diciembre]; Disponible en: http://www.navegapolis.net/files/s/NST-003_01.pdf.
5. Mantenimiento mundial. *Iso9001, 2000*. 2007 [consultada 2008 12 de enero]; Disponible en: www.mantenimientomundial.com/sites/mmnew/her/normas/Iso9001.pdf.

Anexos:

Anexo 1. Simbología de un diagrama de Flujo:

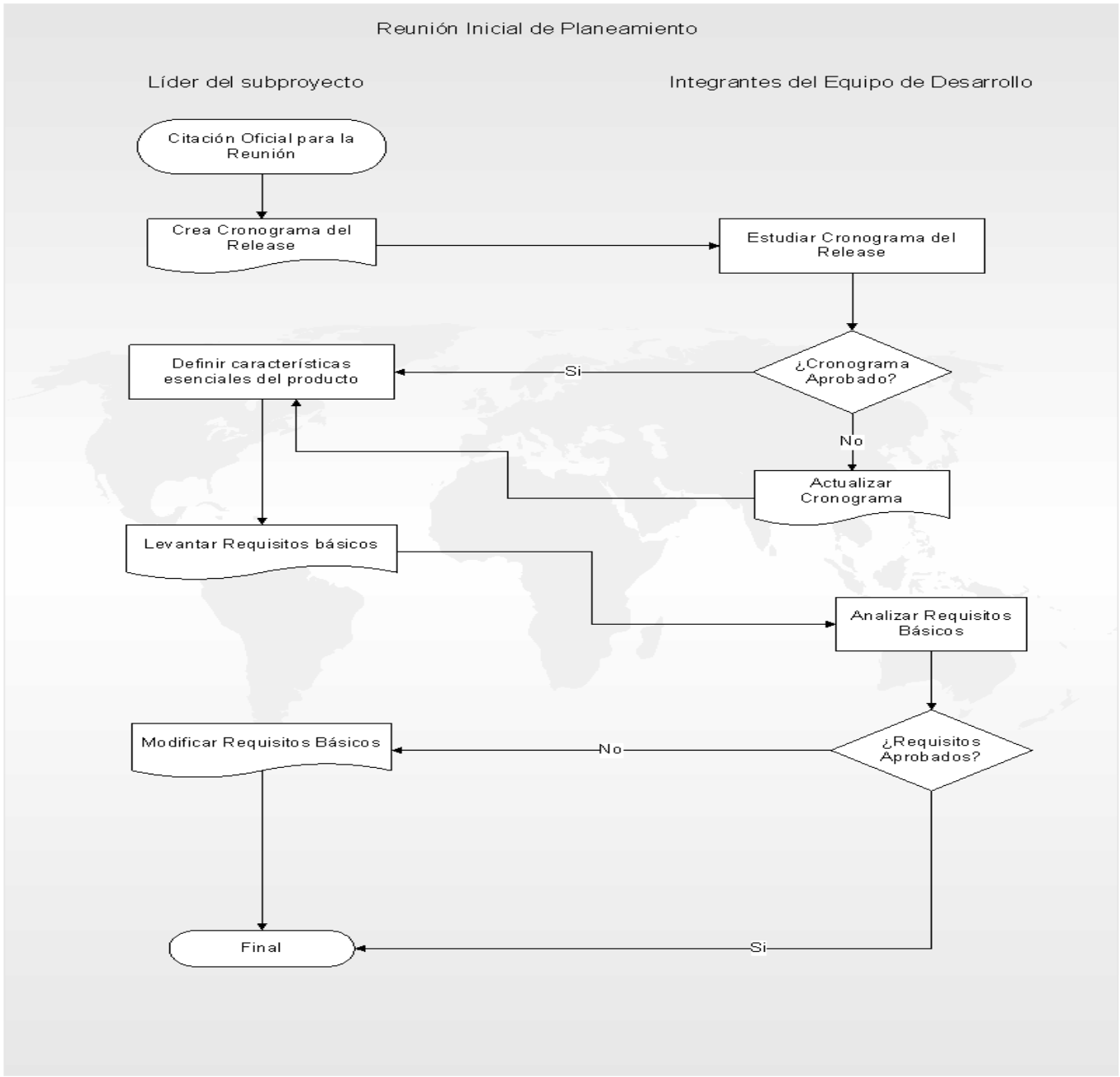
Inicio y fin de un proceso. Figura 1.1

Actividad o paso individual. Figura 1.2

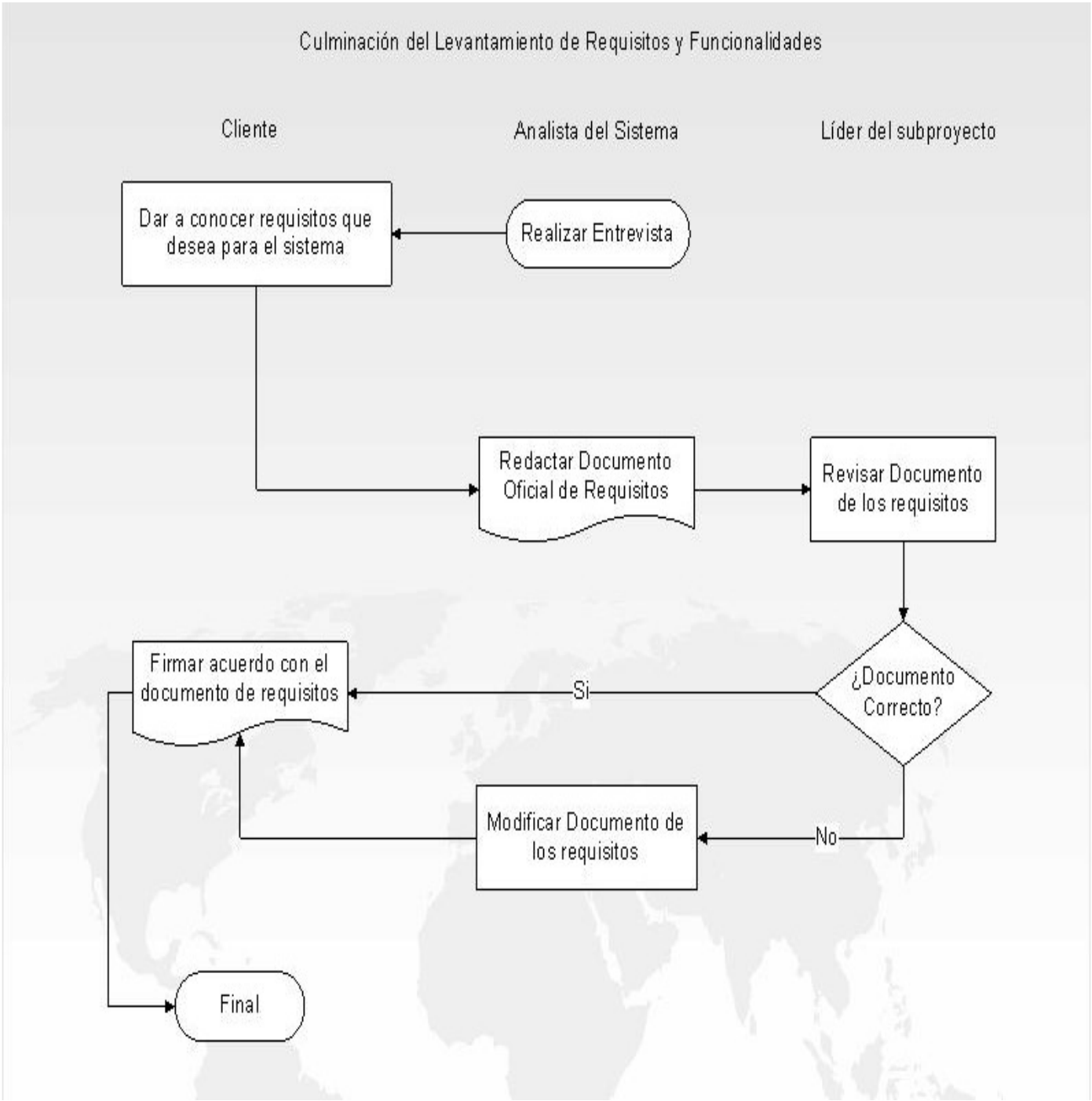
Punto o nodo de decisión. Figura 1.3

Documento. Figura 1.4

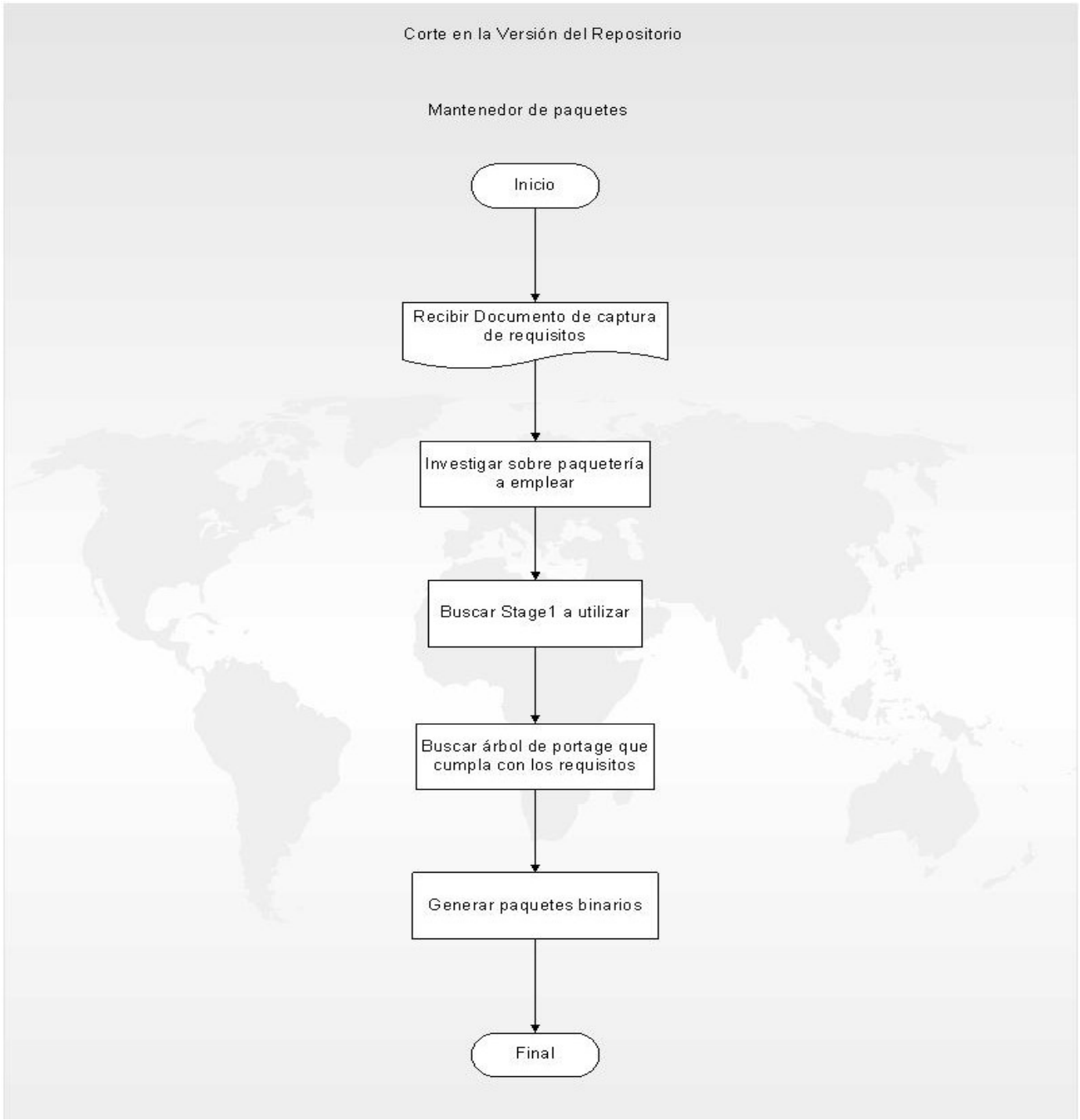
Anexo 2. Diagramas de Flujo del proceso de liberación actual.



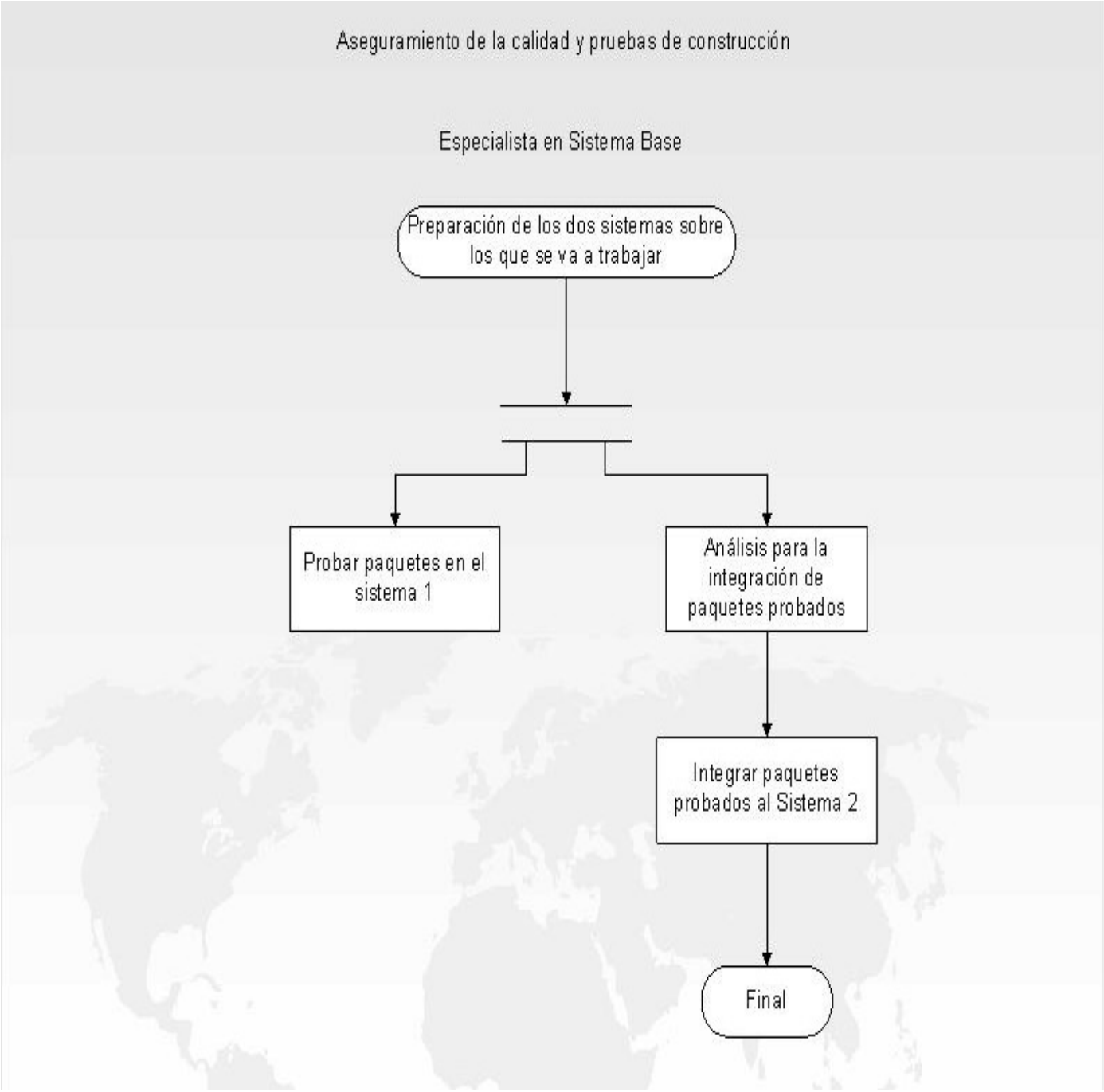
Proceso "Reunión Inicial de Planeamiento". Figura 2.1.



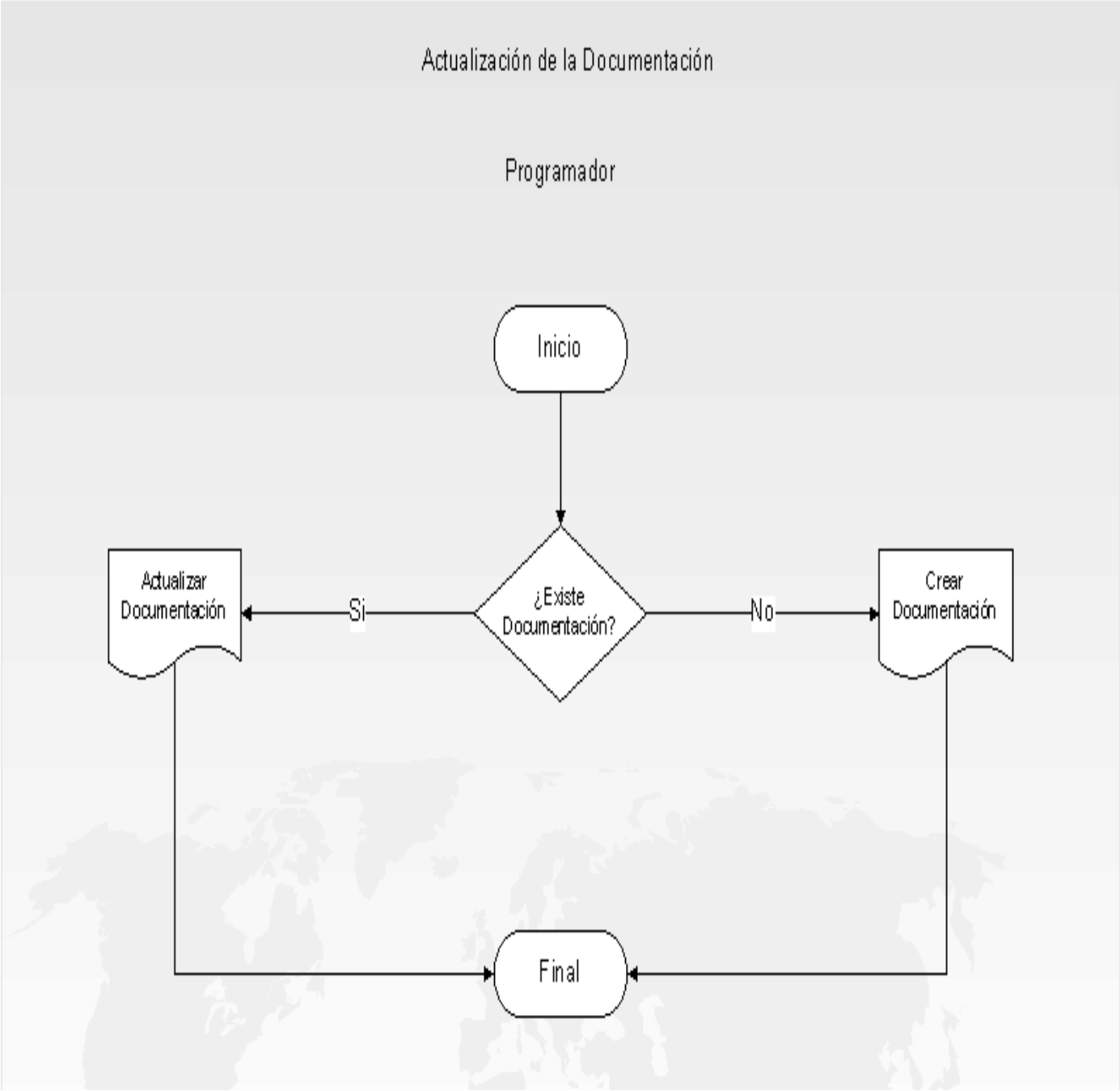
Proceso "Culminación del Levantamiento de Requisitos". Figura 2.2.



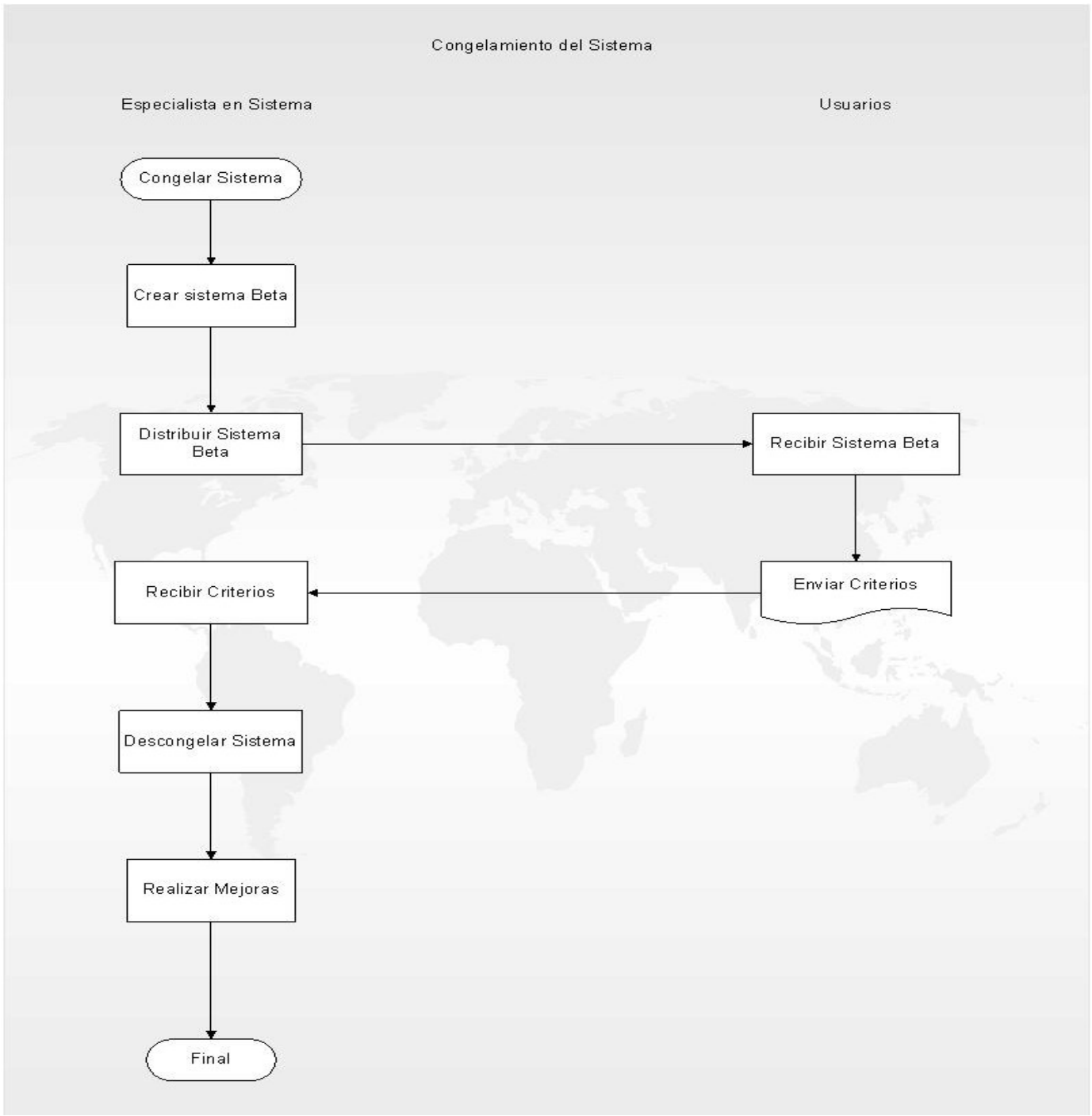
Proceso “Corte en la versión del Repositorio”. Figura 2.3.



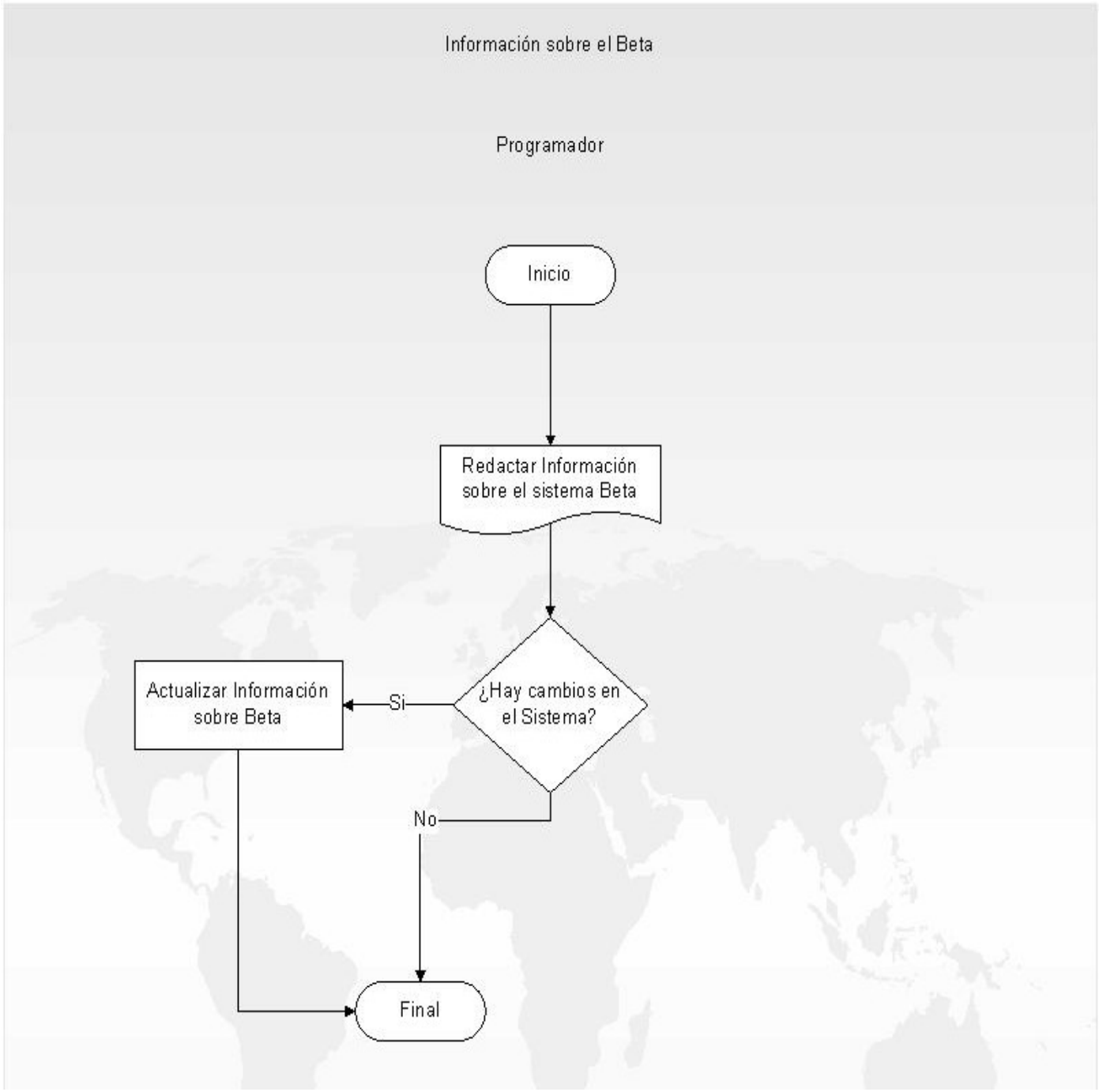
Proceso "Aseguramiento de la Calidad y Pruebas de Construcción" Figura 2.4.



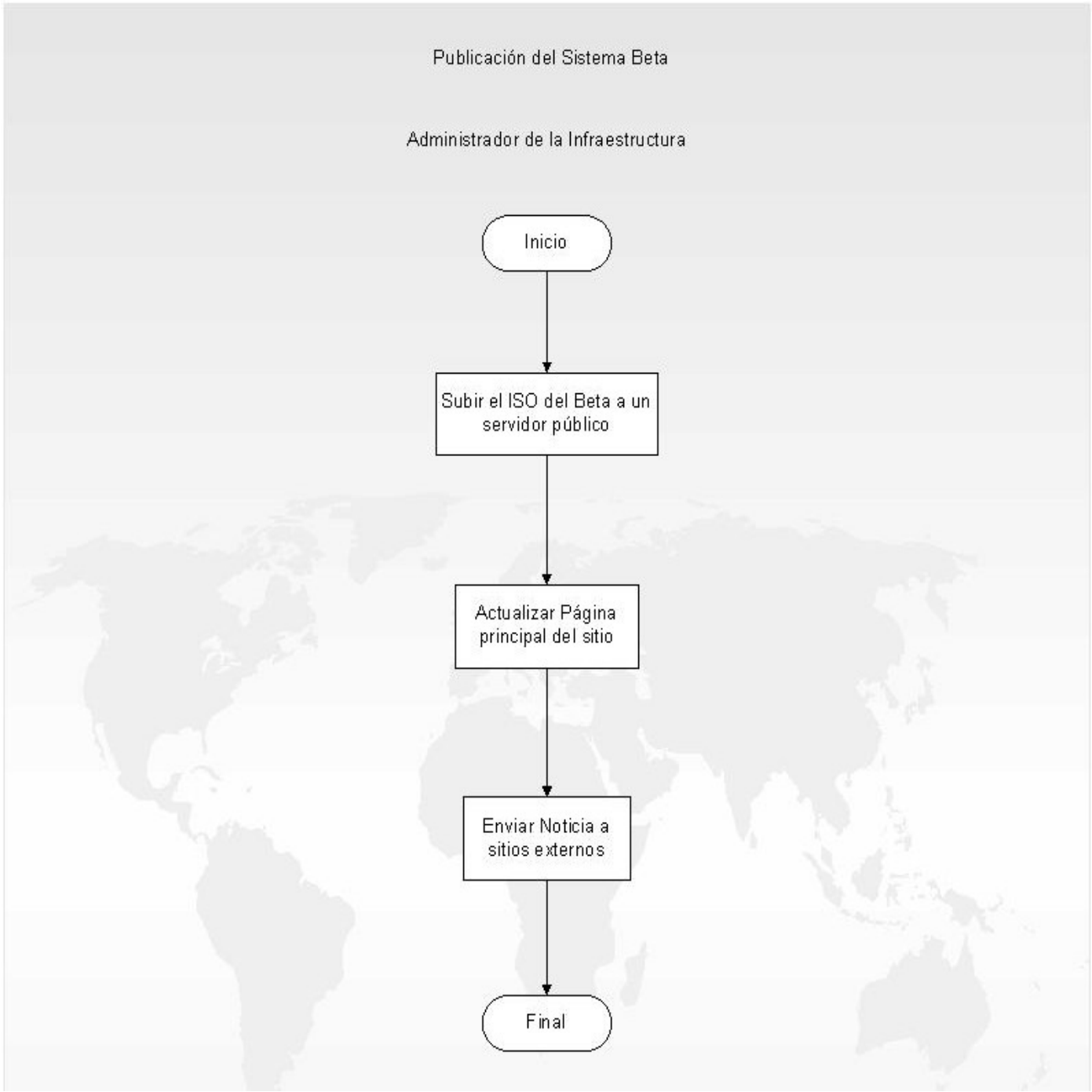
Proceso "Actualización de la Documentación 1". Figura 2.5.



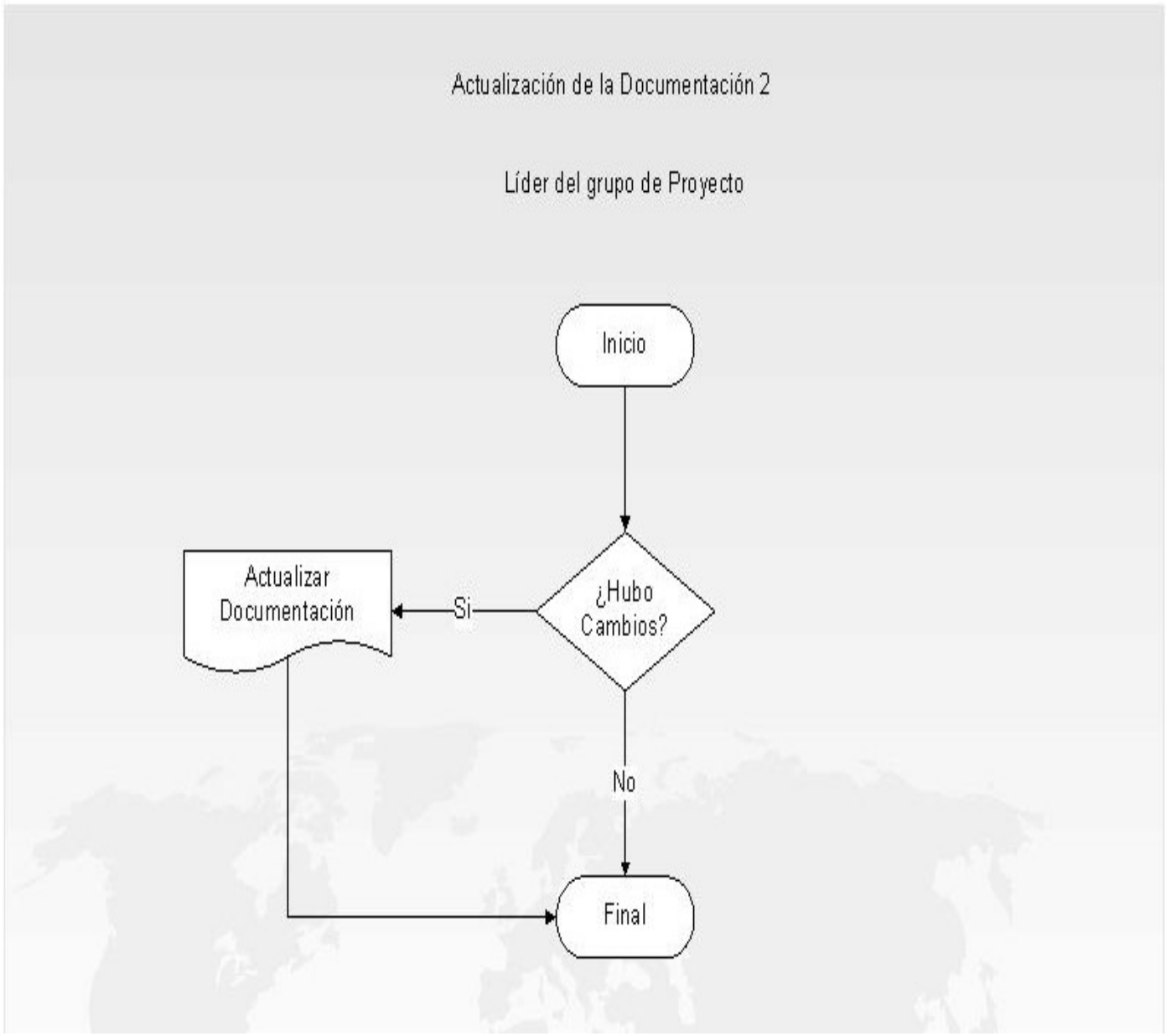
Proceso "Congelamiento del Sistema 1". Figura 2.6.



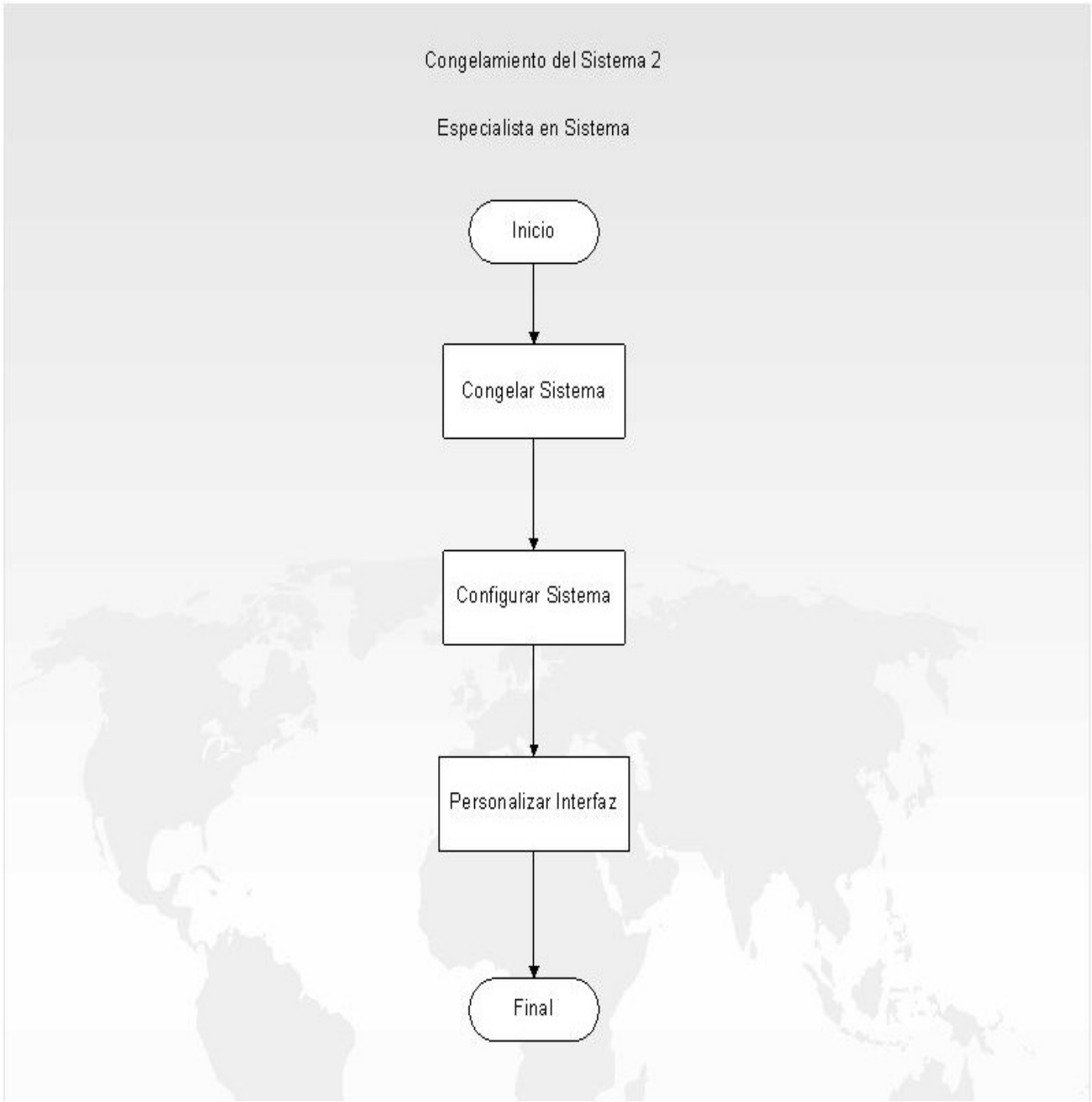
Proceso "Información sobre el Beta". Figura 2.7.



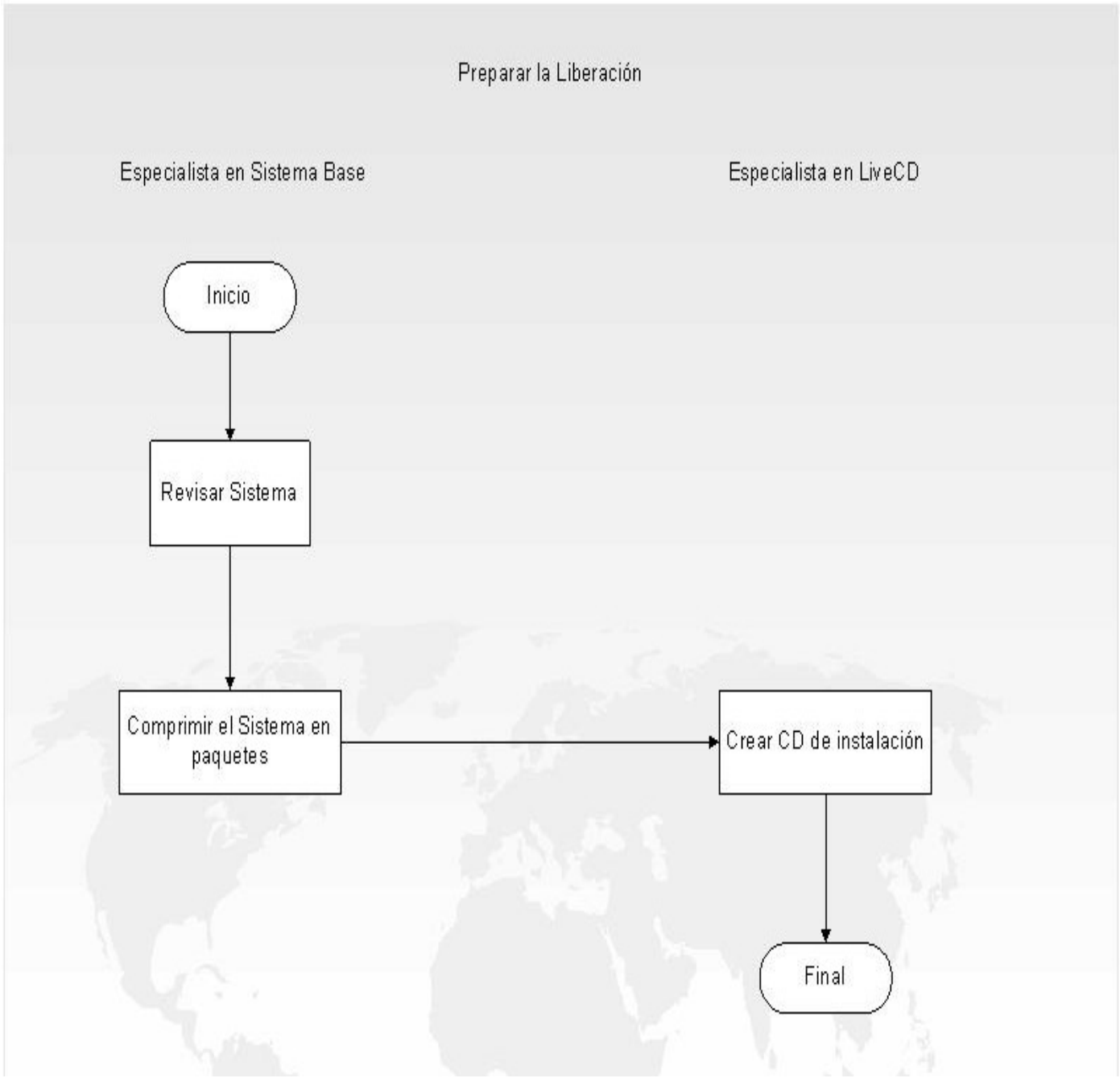
Proceso "Publicación del Sistema Beta". Figura 2.8.



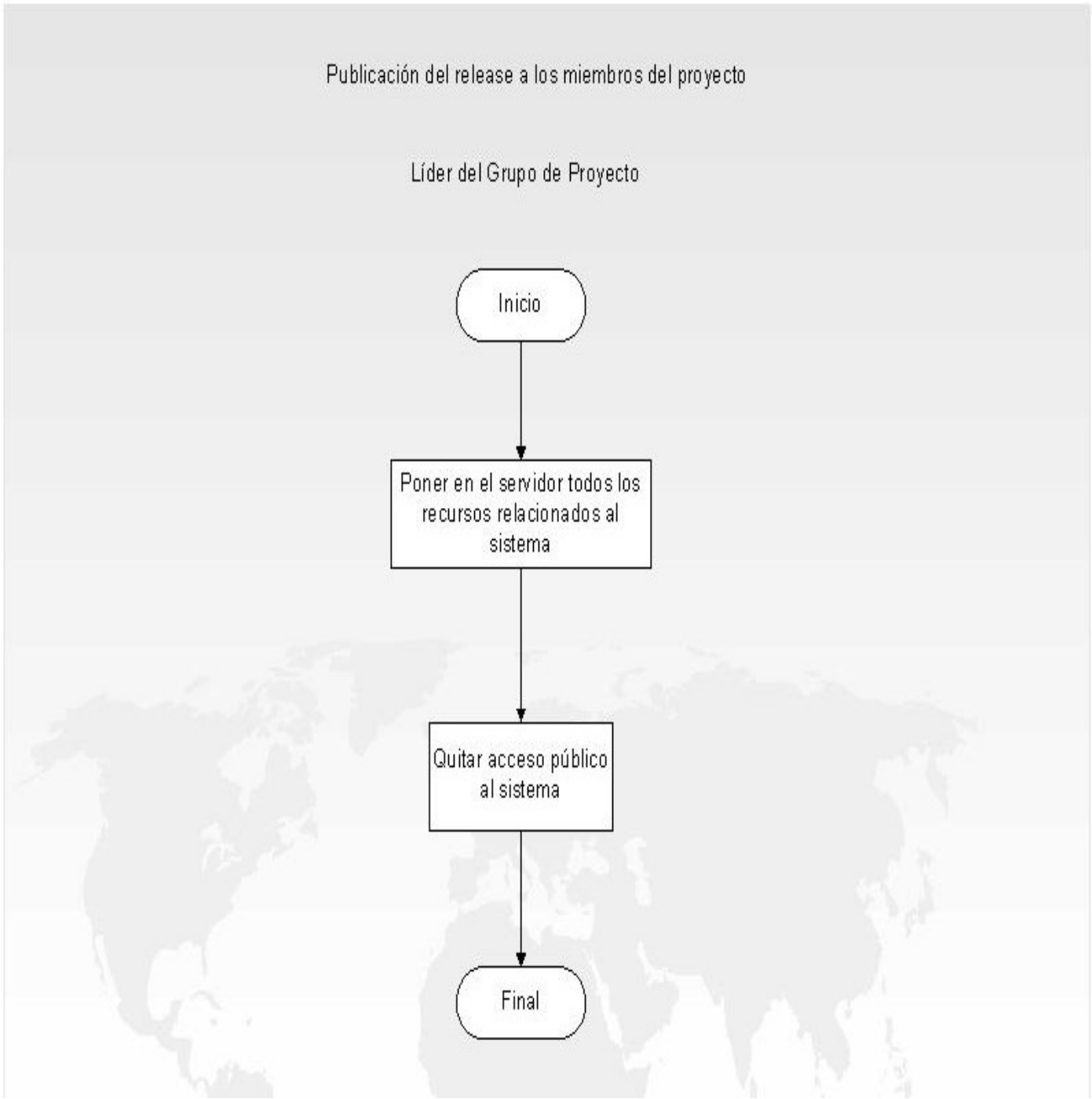
Proceso "Actualización de la Documentación 2". Figura 2.9.



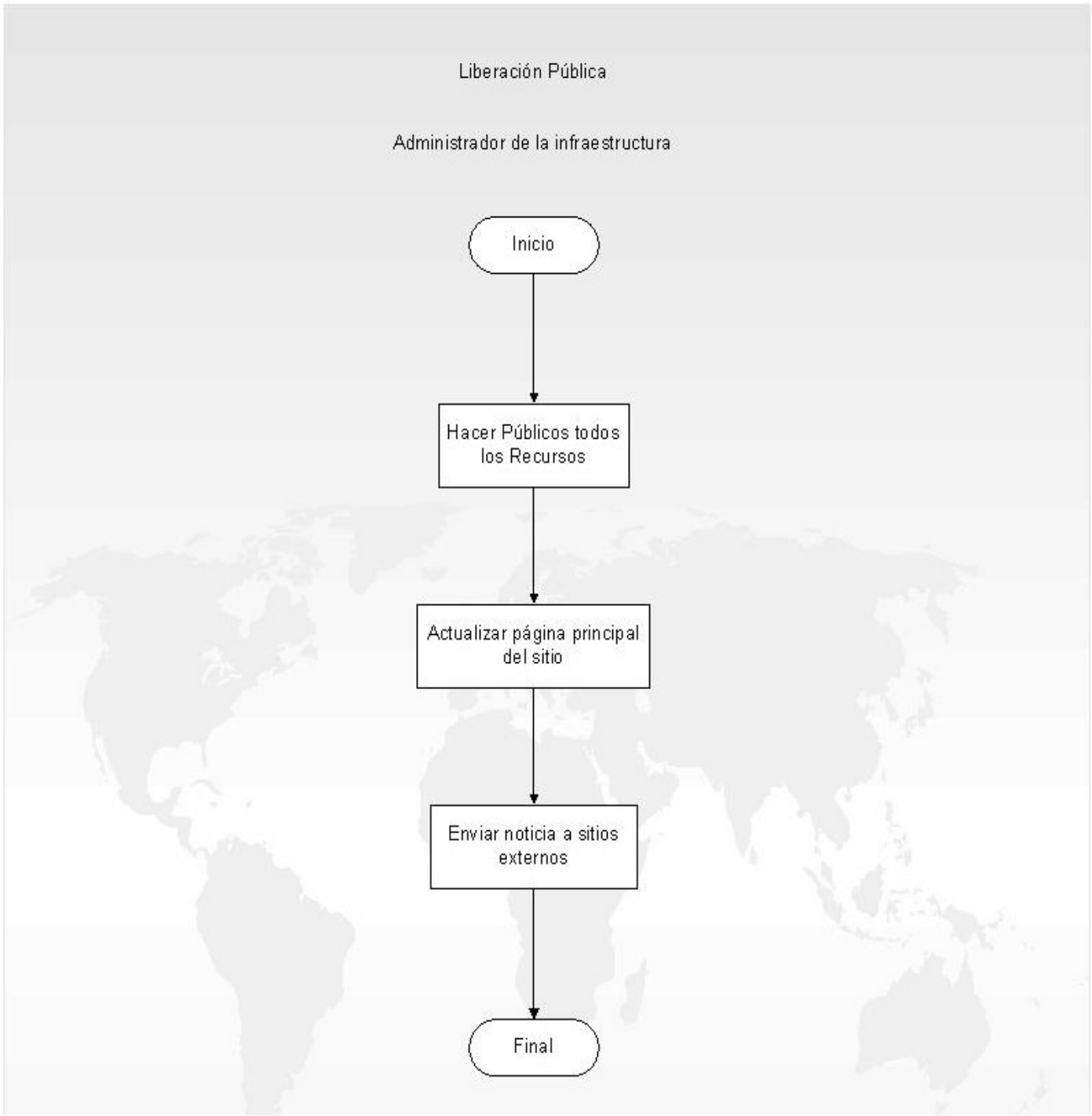
Proceso “Congelamiento del Sistema 2”. Figura 2.10.



Proceso "Preparar Liberación". Figura 2.11.



Proceso "Publicación del release a los miembros del proyecto". Figura 2.12.



Proceso “Liberación Pública”. Figura 2.13.

Anexo 3. Lista de Chequeo para adhesión de la propuesta al Nivel 2 de CMMI.

Número	Práctica CMMI Nivel 2	Evaluación	Observaciones
1	Establecer una política organizacional.	5	Reunión inicial de planeamiento
2	Planificar el Proceso	5	Reunión inicial de planeamiento, Reunión de Planificación del Sprint. Tareas del Líder Scrum
3	Proveer Recursos	4	Reunión inicial de planeamiento. Tareas del líder Scrum.
4	Asignar Responsabilidades	5	Reunión de planificación del Sprint.
5	Entrenar Personal	5	Tareas del Líder Scrum
6	Gestionar Configuraciones	5	Reuniones diarias durante el Sprint. Subversion
7	Identificar e involucrar a los interesados más relevantes	5	Metodología de desarrollo. Reunión inicial de planeamiento.
8	Monitorear y controlar el Proceso	5	Auditorías periódicas. Tareas del equipo de calidad.
9	Evaluar objetivamente la adhesión	5	Auditorías Periódicas. Reuniones diarias durante el Sprint
10	Revisar estado con altos niveles de gestión	5	Tareas de los Líderes Scrum. Tareas del equipo de calidad.

Aspectos a tener en cuenta para la lista de chequeo:

- ✓ En el cuadro número se contempla la numeración correspondiente a la práctica que se está evaluando. La numeración se rige por el orden establecido en el modelo CMMI.
- ✓ Como criterio de evaluación se toman las diversas prácticas genéricas establecidas por CMMI.
- ✓ En el campo observaciones se tendrán en cuenta aquellos criterios que permitan validar la evaluación correspondiente a la práctica; o sea, actuarán como una justificación.
- ✓ De forma general se puede asegurar el cumplimiento de la propuesta con el objetivo genérico del nivel 2 del modelo.

- ✓ En el campo evaluación se coloca la puntuación, según criterio de los investigadores, para la práctica específica. Las puntuaciones son dadas en el rango de 1 a 5:

Evaluación	Significado
1	No cumple con la práctica.
2	Cumple con esporádicos elementos de la práctica.
3	Cumple con aproximadamente un 50% de la práctica.
4	Cumple con la mayoría de los elementos de la práctica.
5	Cumple a cabalidad con la práctica.

- ✓ A modo de conclusión se puede afirmar que la propuesta se encuentra en el nivel 2 del modelo CMMI.

Glosario de términos:

Árbol de Portage: Es una colección de archivos (ebuilds) que contienen toda la información que Portage necesita para mantener el software (instalar, buscar, actualizar, etc.). Estos ebuilds radican por defecto en /usr/portage.

Bootstrap: Proceso mediante el cual se desarrolla un entorno cada vez más complejo a partir de otro más simple. También permite construir un sistema completo a partir de sus componentes base, o bien, de previas versiones precompiladas de los componentes.

Catalyst: Es una herramienta para la construcción de liberaciones, fácil de usar, administrar y mantener. Permite la creación de elementos personalizados.

Control de versiones: Es la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas.

Ebuild: Es una interfaz de bajo nivel para el sistema Portage.

Gestión documental: Se entiende por gestión documental el conjunto de normas, técnicas y prácticas usadas para administrar el flujo de documentos de todo tipo en una organización, permitir la recuperación de información desde ellos, determinar el tiempo que los documentos deben guardarse, eliminar los que ya no sirven y asegurar la conservación indefinida de los documentos más valiosos, aplicando principios de racionalización y economía.

Imagen ISO: Archivo que posee una copia idéntica de determinado sistema de archivos.

Línea de producción: Grupo de operaciones o instrucciones ordenadas, establecidas con anterioridad, por las cuales se someten uno o varios materiales para lograr un producto final con calidad. El gasto de recursos durante la transición de un proceso a otro es mínimo o casi nulo.

Log: Registro oficial de eventos durante un período de tiempo en particular. Para los profesionales en seguridad informática un log es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué un evento ocurre para un dispositivo en particular o aplicación.

Midnight comander: Es un gestor de ficheros ortodoxo para sistemas tipo Unix y un clon de Norton

Commander. Midnight Commander (mc) es una aplicación modo texto, su interfaz principal consiste en dos paneles que muestran el sistema de ficheros del sistema operativo.

Overlay: Es un árbol de paquetes para Portage que contiene ebuilds adicionales de Gentoo, creado por los propios desarrolladores.

Paquete binario: Colección de archivos construidos en un único archivo, el cual puede ser manejado mucho más fácilmente. Posee tanto los archivos requeridos por el programa para ejecutarse, como unos archivos especiales llamados “scripts de instalación”, los cuales copian los archivos en el lugar adecuado (además de otras cosas). Contienen código de máquina y por ello están construidos específicamente para algún tipo de ordenador o “arquitectura”.

Portage: Es la tecnología corazón de Gentoo, constituyendo su sistema de distribución de software. Emplea la colección de scripts que se encuentra en su árbol para crear e instalar las últimas versiones de paquetes de dicho Sistema Operativo. Portage también es un paquete para la construcción e instalación del sistema, permitiendo un alto nivel de personalización.

Pruebas unitarias: Es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego, con las Pruebas de Integración, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

Release: Se refiere a un producto final, preparado para lanzarse como versión definitiva a menos que aparezcan errores que lo impidan.

Repositorio: Es un almacén de paquetes (como una caja con piezas). Los repositorios pueden estar en internet, en un medio extraíble como un CD, en el disco duro, etc. Incluyen un fichero con las dependencias e información sobre cada paquete (para qué sirve, qué ficheros contiene, etc.) de todos los paquetes que contiene.

Sindicación RSS: Es un sencillo formato de datos que es utilizado para redifundir contenidos a suscriptores de un sitio web. Es una forma de distribución de información mediante la cual parte de una página web se pone a disposición para su uso desde otras páginas.

Sistema de instalación de Gentoo: Herramientas empleadas dentro de Gentoo para la gestión de paquetes (instalar, desinstalar, actualizar, etc.).

Stage1: Es un Linux muy básico en el que hay que hacer todo desde el principio. No viene con software de sistema, siendo necesaria la realización de un proceso de bootstrap.

Subversion: Es un software de sistema de control de versiones. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta en la línea de comandos.

Versión Alpha: Es la primera versión para la que el equipo de desarrollo decide implementar todas las funcionalidades especificadas en los requisitos. Es la primera versión del programa que se envía a los verificadores para probarla.

Versión Beta: Representa generalmente la primera versión completa del programa informático o de otro producto, que es probable que sea inestable pero útil para las demostraciones internas y las inspecciones previas de los clientes. Esta etapa comienza a menudo cuando los desarrolladores anuncian una congelación de las características del producto, indicando que solamente se harán pequeñas ediciones o se corregirán errores. Las versiones Beta están en un paso intermedio en el ciclo de desarrollo completo. Los desarrolladores las lanzan a un grupo de probadores beta (a veces el público en general) para una prueba de usuario. Los probadores divulgan cualquier error que encuentran y características, a veces de menor importancia, que quisieran ver en la versión final.

Versión estable: Es aquella versión del software que no está en peligro de sufrir cambios debido a errores.