



GESTIÓN DE REQUISITOS APLICADA A PROYECTOS PARA EL DESARROLLO DE SISTEMAS DE GESTIÓN POLICIAL

ESTUDIO APLICADO EN EL PROYECTO CICPC

Memoria Individual

**Trabajo de diploma para optar por el título de Master en Ciencias
Maestría de Informática Aplicada**

Autor: Ing. Nilet María Soto López

Tutor: MSc. Delly Lien González Hernández

Ciudad de La Habana, Cuba

Mayo, 2010

DECLARACIÓN DE AUTORÍA Y AGRADECIMIENTOS

Yo, Nilet María Soto López, con carné de identidad 81091220612, declaro que soy la autora principal del resultado que expongo en la presente memoria titulada: "Gestión de requisitos aplicada a proyectos para el desarrollo de sistemas de gestión policial", para optar por el título de Máster en Informática Aplicada.

Este trabajo fue desarrollado desde mayo del 2006 hasta diciembre del 2008 en estrecha colaboración con el Ing. Sasha Valdés Jiménez y con el equipo de estudiantes y profesores del Proyecto CICPC, quienes me reconocen la autoría principal del resultado expuesto en esta memoria.

Agradezco la colaboración de todas las personas que han trabajado con la mayor entrega en la realización del proyecto CICPC, permitiéndome con su confianza aplicar todas mis teorías e ideas.

Finalmente declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de la Habana a los 15 días del mes de Mayo del año 2010.

RESUMEN

Con la contratación de tres proyectos con el Ministerio del Poder Popular para Relaciones Interiores y Justicia (MPPRIJ) de Venezuela, para la informatización tecnológica de los organismos responsables de garantizar la justicia y la seguridad ciudadana, se abrió un nuevo campo de desarrollo de software en la Universidad de las Ciencias Informáticas (UCI), creando de esta forma un nuevo reto en la rama de la ingeniería de requisitos aplicada al desarrollo de software de gestión policial.

En este trabajo se demuestra la importancia de una correcta aplicación de la disciplina de ingeniería de requisitos, para lograr el éxito de los proyectos de software del dominio de la gestión policial. Teniendo como referencia los problemas de gestión enfrentados por los primeros proyectos contratados con la República Bolivariana de Venezuela (IP-UCI 2006); se realizó un análisis de las características particulares de contratación y desarrollo de estos proyectos, así como de los enfoques actuales para la aplicación de la gestión de los requisitos.

Como resultado se adoptó el modelo estándar de referencia propuesto por CMMI para el área de procesos de Gestión de Requisitos (REQM); se realizó una caracterización de los proyectos para el desarrollo de software de gestión policial; se identificaron un conjunto de buenas prácticas que fueron aplicadas en el Proyecto CICPC (Cuerpo de Investigaciones Científicas, Penales y Criminalísticas) y se logró la implementación de estas mediante la realización de actividades, procedimientos y mecanismos que pueden ser extendidos a nuevos contratos en este campo de acción.

Palabras clave: gestión de requisitos, ingeniería de software aplicada, software policial

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1	3
Los requisitos en la industria del software.....	3
La ingeniería de requisitos.....	4
La gestión de requisitos como área de apoyo al desarrollo del software.....	6
Normas y modelos aplicados al campo de la ingeniería y la gestión de requisitos	8
CAPÍTULO 2	11
La UCI como organización proveedora de software y su impacto en la gestión de requisitos.....	11
Los proyectos para el desarrollo de software de gestión policial en la UCI.....	11
Relación de la gestión de requisitos con el ciclo de vida de proyectos de desarrollo de software de gestión policial y otras áreas de gestión del proyecto	14
Buenas prácticas para aplicar en la gestión de requisitos de proyectos de desarrollo de software de gestión policial.....	16
Actividades de gestión de requisitos en proyectos de desarrollo de software de gestión policial.....	22
Procedimientos y mecanismos usados en la gestión de requisitos de proyectos de software de gestión policial.....	26
Entorno tecnológico que soporte la gestión de requisitos en proyectos de desarrollo de software de gestión policial.....	35
Resultados de la aplicación en el proyecto CICPC.....	37
CONCLUSIONES.....	38
RECOMENDACIONES.....	39
GLOSARIO DE TÉRMINOS Y ABREVIATURAS	40
BIBLIOGRAFÍA.....	41
ANEXO 1.	43

INTRODUCCIÓN

Como parte de los contratos de colaboración entre la República Bolivariana de Venezuela y la Universidad de las Ciencias Informáticas (UCI), a partir del año 2006 se comenzaron a firmar proyectos de informatización con el Ministerio del Poder Popular para Relaciones Interiores y Justicia (MPPRIJ) para beneficiar a algunas instituciones del gobierno cuya misión fundamental era contribuir a la “seguridad ciudadana” debido a la fuerte presión que generan los indicadores de criminalidad sobre el estado de la población en general.

Se abrió de esta forma un incipiente desarrollo en el campo del software de gestión policial en la UCI, donde inicialmente se solicitó ejecutar de forma independiente tres (3) proyectos de este carácter:

- El Sistema de Investigación e Información Policial (SIIPOL), para el Cuerpo de Investigaciones Científicas, Penales y Criminalísticas de Venezuela (CICPC).
- El Sistema de Gestión Penitenciaria (SIGEP) para la Dirección General de Custodia y Rehabilitación del Recluso (DGCR).
- El Sistema de Gestión Policial (SIGEPOL) para los Departamentos de las Policías de los estados, municipios y regiones.

Estos proyectos de forma general se negocian con restricciones que normalmente se obtienen luego de realizarse un levantamiento de requisitos; variables como el alcance, el tamaño, el tiempo, el costo y requisitos de calidad muy específicos, se prefijan al inicio del contrato. El primer gran desafío es lograr que estos proyectos sean viables dentro de las restricciones ya establecidas.

Estos proyectos se basan en sistemas construidos a la medida del cliente y del problema concreto que se quiere resolver en cuanto a su gestión. Que incorporan procesos complejos de obtención, transformación y recuperación de información, sumamente ajustados y regidos por las leyes y regulaciones vigentes del país. Se caracterizan por tener diseños arquitectónicos personalizados a partir de las condiciones tecnológicas que la institución pueda garantizar. Son además sistemas grandes, donde los usuarios tienen diferentes necesidades, lo que complejiza el proceso, hace más riesgosa la estabilidad del alcance y las funcionalidades del mismo, se necesitan más personas para desarrollar el proyecto y es más complicado gestionar el cambio de forma consistente sin afectar el cumplimiento de las necesidades de los usuarios.

Los requisitos son la base de cualquier proyecto de software, por lo que pueden ser usados para gestionar todo el proceso de desarrollo; una vez acordados y validados

pueden servir como “mapa de navegación” o el medio para llegar al destino final (Aurum and Wohlin 2005). El término de Ingeniería de Requisitos (IR) ha surgido para englobar los procesos de desarrollo y gestión de los requisitos durante todo el ciclo de vida de un software; en muchos casos de forma predeterminada se entiende solo como las actividades involucradas en el descubrimiento, documentación, validación y mantenimiento de los requisitos para un producto determinado. No obstante, con la introducción de prácticas de mejoras que han adoptado o propuesto muchas organizaciones líderes en la industria de software, se reconocen actualmente como áreas de procesos relacionadas pero diferentes: el desarrollo de los requisitos, y la gestión de requisitos como actividad de planificación, seguimiento y control que apoya a la primera (IEEE 1998) (Sawyer and Kotonya 2004; Sommerville 2005) (CMMI and Team 2006).

La presente investigación comenzó en el marco del desarrollo de uno de los proyectos de software de gestión policial, entre las diversas problemáticas a enfrentar se reconoce el siguiente **problema científico** ¿cómo lograr un soporte eficiente a las necesidades de planificación, monitoreo y control de cambios, para el desarrollo de los requisitos en proyectos del dominio de software de gestión policial en la UCI?

Teniendo en cuenta el rol de la autora en el proyecto, el **objeto de estudio** seleccionado para el problema planteado es la Ingeniería de Requisitos, y el **campo de acción** la gestión de requisitos para proyectos de desarrollo de sistemas de gestión policial.

Para solucionar el problema planteado el presente trabajo tiene como **objetivo general** implementar la adopción de buenas prácticas de gestión de requisitos en el dominio de los proyectos de software de gestión policial, a través de actividades, procedimientos, mecanismos y herramientas, adecuadas a las características de estos proyectos y de la organización que los ejecuta.

La **hipótesis** en este trabajo es que la implementación de buenas prácticas de gestión de requisitos, permitirá soportar eficientemente la planificación, el monitoreo y el control del cambio asociado al desarrollo de los requisitos en los proyectos del dominio de los sistemas de gestión policial en la UCI.

La memoria se estructura en dos capítulos. El primero aporta los fundamentos teóricos que avalan las posiciones y las propuestas que adopta la autora, relacionados con la gestión de requisitos como área de apoyo al proceso de desarrollo del software. En el segundo se describe la solución aportada en cuanto a la implementación de buenas prácticas de gestión de requisitos en el dominio del software de gestión policial.

CAPÍTULO 1

CONCEPTOS FUNDAMENTALES

LOS REQUISITOS EN LA INDUSTRIA DEL SOFTWARE

El término *requisito*, está asociado al proceso de desarrollo del software sin importar el modelo de desarrollo que se utilice o la metodología en concreto que se siga. El requisito es la forma esencial que se utiliza para describir un software desde varios puntos de vista. Se define un requisito como un atributo necesario en el sistema, una declaración que identifica una capacidad, característica o factor de calidad de un sistema con el fin de tener valor y utilidad para un cliente o usuario; mediante las cuales se pretende cumplir con determinadas necesidades o restricciones operativas, y que contribuyen a solucionar un problema en un entorno real (Young 2004) (Kontoya and Sommerville 2000).

La mayor parte de los problemas asociados al desarrollo de software están relacionados con los requisitos. Según un estudio realizado por el Standish Group en el año 1995 donde se incluyeron 8000 proyectos de software de 352 compañías, se demostró que los principales problemas que inciden en el fracaso de un proyecto están relacionados con los requisitos, ya sea porque están pobremente organizados o expresados, o porque representan necesidades irreales o alejadas de lo que realmente quieren los involucrados, o porque cambian demasiado rápidamente o a veces innecesariamente. En el último estudio publicado en el 2009 este escenario mantiene cifras similares de proyectos cancelados (24%), contra los que han logrado éxito (32%); y se presentan como factores para el éxito en orden de importancia: la comunicación con los involucrados, el soporte a la gestión del desarrollo, la definición de los objetivos del negocio y la optimización del alcance (TEMÜR 2009).

En la literatura existen múltiples propuestas para clasificar los requisitos, distinguirlos según su procedencia o según los niveles de descripción o de abstracción ya que, por ejemplo, no es lo mismo listar las necesidades del cliente de una forma entendible para el mismo, que hacer una definición más detallada de las funciones operativas del sistema (Young 2004).

De los múltiples materiales consultados (IEEE 2004; Sawyer and Kotonya 2004; Young 2004), hay varios elementos que sobresalen de forma común, y con las que esta autora coincide:

- Los requisitos son independientes del diseño, deben enseñar ¿qué? debe hacer el sistema más que el ¿cómo? debe hacerse.
- Se expresan en un lenguaje natural, pero pueden tener una representación abstracta de alto nivel o una descripción detallada y formal de la función del sistema, por lo que no hay una única manera de expresar los requisitos de un sistema.
- Se clasifican de muchas maneras, en las literaturas aparecen términos como: atributo, cualidad, capacidad, propiedad, restricción, necesidad, factor de calidad, etc. y todas son correctamente aceptadas ya que el término no es utilizado solamente para presentar una función o servicio de un software.
- Deben representar las necesidades del cliente y demás involucrados, por lo que el desafío consiste en capturar la esencia sin ambigüedades y de forma entendible para los clientes, los usuarios y el equipo de desarrollo.

LA INGENIERÍA DE REQUISITOS

El término *ingeniería de requisitos* (IR) ha surgido básicamente para englobar los procesos de desarrollo y gestión de los requisitos durante el ciclo de vida del software, debido precisamente a la importancia que tienen las actividades con los requisitos en la calidad del producto que se desarrolla.

En los inicios, el término *requisito* agrupaba todo lo que tenía que ver con la IR, porque lo único que se consideraba importante, era la declaración de lo que el sistema debía y no debía hacer. Sin embargo las investigaciones en esta rama han creado un cuerpo que conocimiento que incluye terminologías, técnicas, métodos, lenguajes, modelos, procedimientos, entre otras; y por tanto se han creado otros conceptos más abarcadores para el campo de los requisitos de un sistema.

Según Sommerville (2005), la IR es un proceso o disciplina que comprende todas las actividades que tienen que ver con la creación y el mantenimiento de los requisitos de un sistema; se enfoca por tanto en áreas como: el estudio de factibilidad, la obtención, análisis y especificación de los requisitos, la validación, la gestión y administración de los mismos; sin embargo, cuando propone su modelo en espiral donde se satisface un enfoque iterativo y con niveles crecientes de detalles en la especificación de un sistema, solo se incluyen aquellos subprocesos que tributan a la creación de un documento de requisitos correcto (obtención, especificación, y validación de requisitos). Por tanto no se observa con claridad el papel de la gestión de los requisitos en el proceso de desarrollo y

también se quedan fuera cuestiones relativas al proceso como el análisis y la negociación de los mismos, o la gestión de la calidad.

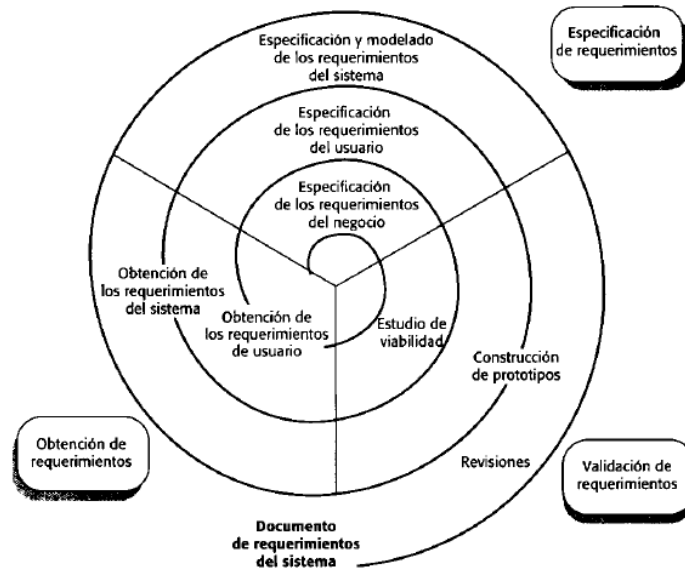


Figura 1: Modelo en espiral de los procesos de de la IR según Sommerville.

Aurum y Wohlin (2005), consideran que la IR es el proceso mediante el cual los requisitos para un producto de software son obtenidos, analizados, documentados y gestionados durante todo el ciclo de vida. La IR es actualmente una de las etapas más aceptadas del desarrollo del software, porque se encamina directamente a resolver el problema de definir el software correcto que necesita el cliente. Actualmente la IR ya se ha dividido en un conjunto de procesos que operan a distintos niveles de organización del producto y del proyecto como tal; además ya no solo se considera como la primera fase en el ciclo de vida del software, sino como un conjunto de actividades repetibles y estructuradas donde se manejan tanto aspectos ingenieriles como de gestión, durante todo el ciclo de vida del software.

Para la UCI, con menos de una década de creada, este concepto está aún inmaduro, por lo que muchos de los procesos asociados a la IR aún no se han logrado implementar e institucionalizar al menos parcialmente en los proyectos de desarrollo de software (CMMI_Consulting_Group 2009). Sin embargo ya se reconoce la necesidad de implantar lineamientos y un modelo de procesos que permita llegar al Nivel 2 propuesto por el Modelo Integrado de Madurez de Capacidades (CMMI).

LA GESTIÓN DE REQUISITOS COMO ÁREA DE APOYO AL DESARROLLO DEL SOFTWARE

Durante el desarrollo de los requisitos el cambio es inevitable. Las personas involucradas logran una mejor comprensión de lo que se quiere que haga el software, cambian las organizaciones, las tecnologías, etc. El proceso de gestionar estos cambios, según Sommerville (2005) se denomina gestión de requisitos.

Si la palabra “gestión” está relacionada a verbos como: organizar, comprender, controlar, planificar, y según la IEEE 12-207 los procesos de gestión son actividades genéricas o tareas que pueden ser empleadas por cualquier parte que tenga que gestionar sus respectivos procesos “claves” y son específicos por tanto a la forma en que la organización realiza los mismos, en correspondencia con la definición que aporta Sommerville, la autora considera como actividades de la gestión de requisitos, todas aquellas que permitan planificar, organizar y controlar los procesos de desarrollo de los requisitos (obtención, modelación, especificación, validación, etc.)

Por otro lado el Software Engineering Book Of Knowledge (SWEBOK) permite ampliar esta definición, al plantear que los procesos de gestión y mantenimiento son los “recursos” requeridos y consumidos durante el proceso de ingeniería de requisitos, y su principal propósito es enlazar las actividades de desarrollo (Ejemplo: la obtención, especificación, validación, etc.) con los intereses y cuestiones puramente organizativas y reales de un proyecto, como los costos, la calidad, el tiempo de desarrollo, los recursos humanos, el entrenamiento, las herramientas automatizadas, entre otras (IEEE 2004).

Concluyendo, se puede decir que una efectiva gestión de los procesos de IR incluye actividades, procedimientos y herramientas que permitan soportar de forma sistemática y automática los procesos de desarrollo de los requisitos en las distintas fases del ciclo de vida del software, y además incluye otros factores como los relativos a la organización y vías de comunicación específicas de cada proyecto.

Tomando como base la gráfica de Wiegers (2003), para los fines de la presente investigación se divide el dominio de la IR en desarrollo de requisitos y gestión de requisitos. Y a su vez, se divide el desarrollo de requisitos en cinco (5) subprocesos: elicitación, análisis y negociación, especificación, verificación y validación (Soto and Valdés 2010).

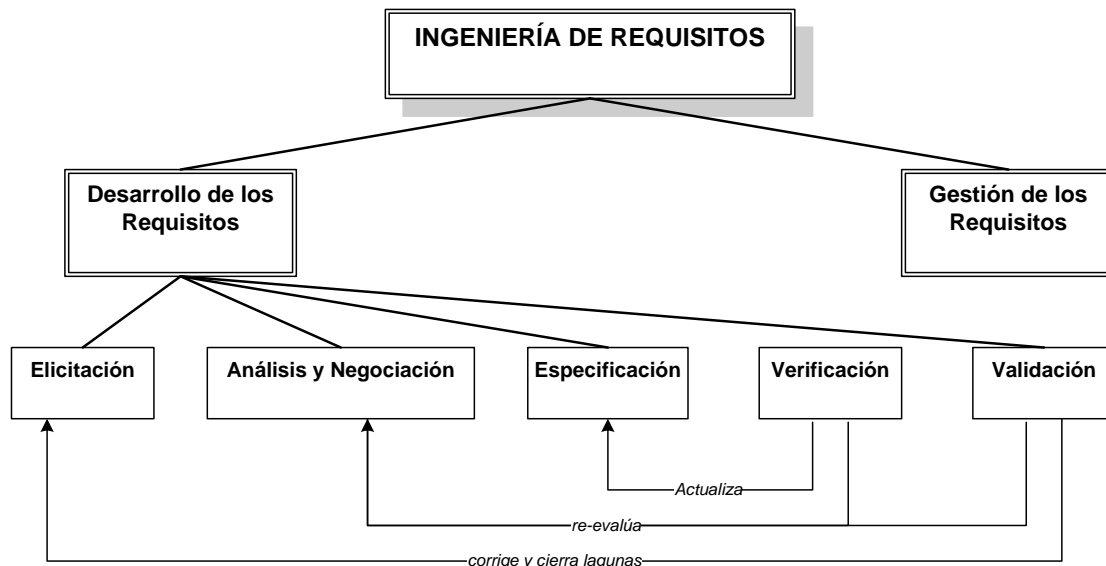


Figura 2: Dominio y sub-procesos de la IR.

Se pueden mencionar algunos problemas que hacen a la gestión de requisitos más difícil que otras áreas de gestión para aplicar en la UCI:

- La poca experiencia de las personas que trabajan en los proyectos de software de la universidad.
- El inevitable cambio asociado al proceso de desarrollo, siempre impacta en uno o más a los requisitos, haciendo más difícil el control del desarrollo, y la estabilidad de los artefactos que se generan para modelar y describir el sistema.
- Su aplicación está en dependencia del tipo de organización que desarrolla el software. Aunque se identifiquen cuestiones teóricas esenciales, la práctica está asociada al tipo de organización y características del proyecto que vaya a aplicarlas.
- Implica un trabajo adicional que no puede observarse su beneficio a corto plazo, ni por el cliente, ni por el equipo de desarrollo que no ve como avance otra cosa que no sea la cantidad de software que ya se ha programado.

Sin embargo también se pueden mencionar algunas ventajas o beneficios que aporta a los proyectos la gestión de los requisitos (Aurum and Wohlin 2005):

- Los requisitos son la base de cualquier proyecto de software, por lo que pueden ser usados para gestionar todo el proceso de desarrollo; una vez acordados y validados pueden servir como “mapa de navegación” o el medio para llegar al destino final.
- Tener una línea base de requisitos acordados, permite hacer planificaciones reales del desarrollo del sistema, si no se sabe explícitamente ¿qué? se debe hacer; cómo

saber ¿cuánto tiempo o esfuerzo se necesita?, o ¿cuándo se puede dar como concluido el proyecto y asegurar que lo logrado es totalmente lo que el usuario necesita?

- Los requisitos son consultados obligatoriamente cuando se quiere dar un informe del avance o % de completamiento del sistema, o cuando se necesita saber el impacto de los cambios que son inevitables que aparezcan en todo proceso de desarrollo, o cuando se quiere demostrar mediante pruebas que el sistema cumple con las necesidades de los usuarios.
- Los requisitos también permiten la identificación y el manejo de los riesgos desde etapas tempranas del proyecto; trazar el impacto y hacer planes de mitigación antes de que se incurra en altos costos o problemas para el proyecto.

NORMAS Y MODELOS APLICADOS AL CAMPO DE LA INGENIERÍA Y LA GESTIÓN DE REQUISITOS

Existen varios modelos de procesos con el fin de describir de forma genérica, cómo ocurren las actividades asociadas a la IR. Inicialmente Kontoya y Sommerville (2000) sugirieron un modelo conceptualmente lineal con iteraciones entre algunas actividades y Macaulay (1996) sugirió un modelo puramente lineal sin superposición, alineación o repetición de actividades. Loucopoulos y Karakostas (1995) describieron el modelo de IR como un proceso iterativo y cíclico, o sea, con un enfoque en espiral, el mismo presenta una secuencia de actividades que se realizan en iteraciones como resultado del gradual avance de los requisitos en el ciclo de vida del desarrollo de un software. Este modelo es similar a un segundo que posteriormente propusieron Kontoya y Somerville (2000) que utiliza las mismas actividades de su modelo lineal inicial pero representado en una espiral. Ver Figura 1.

Sin embargo, resultados de algunos estudios de procesos realizados en proyectos sobre el campo de la IR, indican que los enfoques iterativos e incrementales no representan la realidad en la práctica de los proyectos de software, ya que generalmente los proyectos siguen un modelo lineal hasta un determinado punto y posteriormente es que pasan a iterar entre algunas actividades.

Houdek y Pohl (2000) proponen que no existe un modelo monolítico (único) para el proceso de IR ya que las actividades están fuertemente relacionadas por lo que no existe división de tareas, con orden de precedencia o estructura.

Algunos autores (Kontoya, Somerville, Hofman y Lehner) afirman que no son muchas las organizaciones que han definido y logrado un proceso estándar de aplicación de las actividades de IR. Otros autores plantean que las organizaciones realmente usan los procesos y estándares definidos para la IR como “buenas prácticas” durante todo el ciclo de vida del software (Aurum and Wohlin 2005). Los campos de estudio de la IR también han alcanzado resultados conflictivos en el estado de los procesos y normas de IR propuestas por las organizaciones. Esto indica que la IR no está completamente “madura” en el sentido de que no hay un proceso que actualmente sea universalmente aceptado, y no hay ningún modelo de procesos de IR único o consensuado para la industria del software. No obstante, la conclusión no puede ser obviar la necesidad de lograr algún tipo de orden o estandarización de las actividades relacionadas a los requisitos, sino que deben utilizarse como base metodologías estándares de procesos, adaptadas a las necesidades específicas de cada proyecto.

Algunas normas conocidas y adoptadas a nivel mundial son el Modelo Integrado de Madurez de Capacidades (CMMI), las prácticas recomendadas por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) para la Especificación de Requisitos de Software y la Organización Internacional para la Estandarización (ISO).

El estándar de la IEEE/ANSI 830-1993 propone la estructura básica de un documento para los requisitos de un sistema, el cual aunque debe ser adaptado en función del tipo de sistema que desarrolle una organización, por lo menos sistematiza algunos elementos que son imprescindibles que existan con claridad en un sistema.

En CMMI se proponen diversas áreas de trabajo asociadas al desarrollo de software agrupadas en cuatro (4) categorías generales de procesos: Gestión de Procesos, Gestión de Proyectos, Ingenieril, Soporte. Las áreas de trabajo asociadas a los requisitos del software se encuentran en la categoría de Ingenieriles, y coincidiendo con el criterio de esta autora, se diferencian las actividades relacionadas al desarrollo, de las que son de apoyo o soporte a las primeras, siendo así que proponen dos áreas de procesos: Desarrollo de Requisitos (RD) y la Gestión de Requisitos (REQM).

En la propuesta de CMMI se tienen en cuenta otros elementos interesantes que se abordan en este trabajo, como la existencia de interacciones entre las áreas de procesos, siendo evidente el hecho de que la gestión de requisitos no es proceso que pueda realizarse de forma independiente o separada del resto de las actividades de un proyecto; y que independientemente de cómo se implemente de forma práctica el proceso, según

las características de cada organización, existen prácticas globales y específicas que pueden aplicarse a cada una de las áreas de procesos (CMMI and Team 2006).

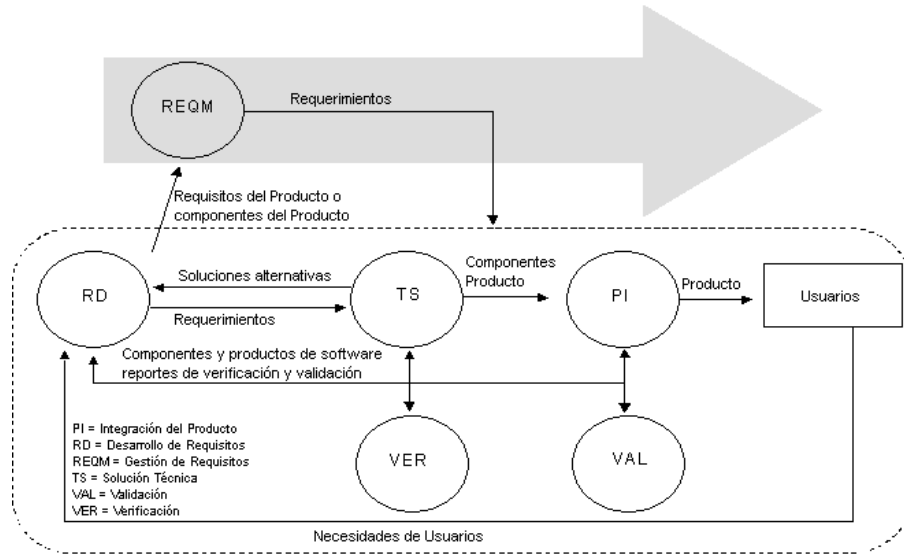


Figura 3: Relación de las áreas de procesos ingenieriles que propone CMMI.

La figura 3, brinda una vista de cómo interaccionan las seis (6) áreas de procesos ingenieriles, siendo representativo el caso de la Gestión de Requisitos (REQM) como un proceso que tiene como entrada (o punto de inicio) los resultados del Desarrollo de Requisitos (RD), aporta información de entrada a los restantes procesos ingenieriles y se desarrolla de forma paralela al proceso de desarrollo del software, o sea que el enfoque no es lineal (no antecede o sucede a otro proceso específico), iterativo o secuencial sino integrado o complementario a todas las demás áreas de procesos.

La autora está de acuerdo con el modelo propuesto por CMMI y se reafirma como conclusión que las actividades de la gestión de requisitos:

- No se realizan de forma secuencial con un principio y un fin definidos o de forma completa como una fase única antes de comenzar el desarrollo software, sino que comienzan con la firma del proyecto y continúan repitiéndose a todo lo largo del ciclo de vida como secuencia eventos dinámicos.
- Deben ser adaptadas a las características y contexto del proyecto y la organización.
- Tienen una relación directa con los resultados de las actividades de desarrollo de los requisitos, y puntos de contacto con otras áreas de la gestión como: la gestión del proyecto, los riesgos, la configuración y el aseguramiento de la calidad.

CAPÍTULO 2

DESCRIPCIÓN DE LA SOLUCIÓN APORTADA

LA UCI COMO ORGANIZACIÓN PROVEEDORA DE SOFTWARE Y SU IMPACTO EN LA GESTIÓN DE REQUISITOS.

En la literatura se reconoce que la manera en que los requisitos son gestionados depende principalmente del tipo de organización que debe aplicar los procesos de la IR. En la industria de software se dice que hay 3 tipos organizaciones para producir software (Hull, Jackson et al. 2005): las que compran sistemas y luego los usan para proporcionar una capacidad operativa necesitada (adquisición), las que se enfocan en el desarrollo y venta de productos, según las necesidades del “mercado” (productos), y las que responden a solicitudes a la medida de las necesidades de interesados de alto nivel (proveedores).

La UCI es una universidad productiva, cuya misión es producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación. La organización de la actividad productiva se hace principalmente por proveedores, donde a partir de una dirección comercial llamada ALBET, se crean proyectos a la medida de las necesidades de los clientes interesados. Algunas características que definen a este tipo de organización y que tienen importancia para la gestión de los requisitos son:

- 1- Se parte de una petición inicial que contiene un conjunto de requisitos de entrada (que no son relativos a funciones del sistema).
- 2- Se tiene que crear una propuesta que demuestre como se piensa llevar a cabo el proyecto teniendo en cuenta el conjunto de requisitos planteados por el cliente. El proyecto solo se logra contratar si el cliente queda convencido a priori de que la solución que se propone es la que necesita.
- 3- Se debe considerar para esta forma de organización dos enfoques, primero ¿cómo demostrar la factibilidad del proyecto?, y luego ¿cómo ejecutar el proyecto cumpliendo con lo pactado?

LOS PROYECTOS PARA EL DESARROLLO DE SOFTWARE DE GESTIÓN POLICIAL EN LA UCI.

Los Sistemas de Gestión de la Información Policial constituyen una herramienta indispensable en la eficiencia de las gestiones de cualquier institución policial. Debido al

incremento de los índices de ocurrencia de delitos en las sociedades actuales, y al impacto de estos en el estado de bienestar y seguridad de las poblaciones, se hace indispensable que cualquier país utilice o invierta en tecnología adecuada para lograr una gestión más eficiente del trabajo de los investigadores, funcionarios y demás colaboradores.

Al comienzo del proyecto se llevó a cabo una investigación del estado del arte del dominio de aplicación del software policial. Determinando que son numerosos los países que están llevando a cabo proyectos de modernización de la infraestructura tecnológica y el desarrollo de sistemas policiales más eficientes, incluyendo a Cuba. Se tienen informaciones generales sobre estos sistemas, sobre todo en cuanto a los beneficios que aportan, pero no existen publicadas descripciones técnicas de los mismos. Una parte del resultado de dicho estudio puede consultarse en la memoria colectiva (Soto and Valdés 2010).

A partir de la experiencia de la autora en la concepción de estos proyectos, se puede determinar como primer aporte que comúnmente en una institución policial, son aplicables sistemas que permitan:

1. La optimización de las labores de investigación, a fin de minimizar el tiempo que necesita un investigador para esclarecer un hecho delictivo.
2. La automatización de funciones de procesamiento, análisis y comparación de los datos (información) en el menor tiempo posible para lograr con eficiencia y eficacia la obtención de la mayor cantidad de resultados.
3. La gestión de información que valide con eficacia la toma de decisiones estratégicas y operativas del Estado, los Ministerios y las Instituciones en función del cumplimiento de la constitución y demás leyes.
4. La organización de los flujos de procesos de apoyo al trabajo policial, como puede ser la entrada y salida de correspondencia, asignación de los recursos policiales (humanos y logísticos), el acceso de personal a instalaciones policiales, las guardias, el patrullaje y el parte de novedades, etc.

Algunas características comunes que definen a los proyectos de este dominio de aplicación, y que en mayor o menor medida tienen impacto en la estrategia de gestión de requisitos que debe aplicarse para el desarrollo de los mismos son:

- Existe una distancia geográfica entre los involucrados del proyecto: ya que el ciclo de vida del software se realiza en dos lugares diferentes: el levantamiento de los requisitos y las pruebas de aceptación en el entorno del cliente y el resto de las

actividades en la sede del proveedor (Cuba). Esto dificulta la comunicación técnica presencial constante, lo cual impone un modo de organización tradicional del trabajo, o sea: acordar primero con claridad que hay que hacer, para después hacerlo.

- Existen restricciones de tiempo, costo, y tamaño del sistema que son parte de los criterios de calidad al comienzo del proyecto: la mayoría de los proyectos que se negocian para la producción de software en la UCI se realizan bajo el criterio inicial de un tamaño específico que se estima según alcance, experiencia con sistemas similares, sistemas existentes o términos de referencia que entrega el cliente. El costo y el tiempo se estiman a partir de ese tamaño teniendo en cuenta distintos elementos de gastos y estimaciones ya prefijadas por una base de cálculo construida para esos efectos por la dirección de producción de la UCI. Con esto se elabora una ficha técnica inicial y un contrato “marco” que indica de forma general los criterios o hitos para dar la aceptación de la terminación del proyecto.
- Se producen sistemas de gran tamaño: teniendo en cuenta el Caso de Uso (CU) como unidad de medida, se habla de un mínimo de 100 CU. De los proyectos desarrollados en la UCI, CICPC contrató 432 CU, SIGEP 390 y SIGEPOL 421. Normalmente los sistemas grandes, tienen una comunidad de usuarios diversa, con diferentes requerimientos y prioridades, lo que complejiza los procesos de elicitación y validación de los requisitos. Se necesita mucho tiempo (más de 10 meses) para terminar los proyectos lo que hace más riesgoso la estabilidad del alcance y las funcionalidades del sistema puesto que los involucrados no se mantienen en las mismas funciones hasta que el proyecto termina. Los recursos humanos superan la cifra de 140 personas, entre estudiantes y profesores, con diferentes niveles de conocimiento y especialización. Y se hace más complicado organizar el proceso y gestionar el cambio de forma correcta para que no se creen inconsistencias.
- Se debe garantizar un manejo confidencial y confiable de la información: por la implicación que tienen los datos sobre el estado legal de una persona, estas instituciones exigen que toda la documentación técnica generada sea entregada en forma de transferencia tecnológica al finalizar el contrato de modo que el cliente logre de manera independiente continuar el mantenimiento y desarrollo de nuevas funcionalidades una vez que haya concluido el contrato. Incluso se puede exigir la entrega de esta documentación al finalizar cada fase de desarrollo, para ser sometida a revisión por especialistas en sistemas; y definir de forma legal los “entregables” que espera recibir como constancia del conocimiento generado.

- Existe dependencia de los entornos donde de aplicarán estos sistemas con basamentos legales, reglamentarios, y vocabulario con significados propios del país o la región: esta característica hace que cada proyecto sea único en su tipo, y todas las actividades de la IR deben realizarse independientemente de que se tenga una mayor o menor experiencia en este tipo de proyectos. Además, si se realiza algún tipo de modificación a la legislación vigente del país durante el desarrollo del proyecto, la organización tendrá que cambiar procedimientos de trabajo, y por tanto, se hará necesario modificar el sistema inicialmente propuesto.

METODOLOGÍA DE DESARROLLO PARA LOS PROYECTOS DE SOFTWARE DE GESTIÓN POLICIAL

Por las características mencionadas en el epígrafe anterior, para los proyectos de software de gestión policial contratados en la UCI se decidió utilizar el Proceso Unificado de Racional (RUP) como metodología de desarrollo, donde la aplicación del estándar de CMMI y la propuesta concreta de este trabajo se integra perfectamente, ya que esta metodología se basa en el modelo espiral donde cada iteración cumple los objetivos definidos inicialmente según el grado de entendimiento que se tenga de los requisitos del sistema; esto permite desde la perspectiva de la gestión de los requisitos: planificar consistentemente según los objetivos a lograr en cada etapa, ver los avances que se hacen gradualmente, preparar los elementos que serán utilizados en las siguientes iteraciones, controlar, evaluar y resolver constantemente los problemas y los riesgos de no cumplir con el alcance pactado. Además el enfoque tradicional de esta metodología refuerza el empleo de mecanismos formales de comunicación entre los involucrados; por ejemplo, acordar primero con claridad qué hay que hacer, para después hacerlo.

RELACIÓN DE LA GESTIÓN DE REQUISITOS CON EL CICLO DE VIDA DE PROYECTOS DE DESARROLLO DE SOFTWARE DE GESTIÓN POLICIAL Y OTRAS ÁREAS DE GESTIÓN DEL PROYECTO

Como se plantea en el estándar de CMMI, las actividades propuestas de gestión de requisitos, se van a desarrollar de forma paralela y sistemática en todo el ciclo de vida del desarrollo del software, tratando que exista una relación estrecha entre las actividades de gestión que se realizan en un proyecto y las actividades ingenieriles.

Como elemento particular del ciclo de vida del software, se agrega el flujo de Estudio Preliminar, que no existe en la metodología RUP pero se hace necesario debido a las características que tiene la contratación de estos sistemas. Además debido a la existencia de una Dirección de Calidad en la UCI, en el flujo de trabajo de pruebas, antes de entrar a la fase de transición, el producto debe pasar por una prueba de liberación que lo ejecuta un agente externo al proyecto (un Dpto. de la Dirección de Calidad en la UCI).

En la figura 4, se propone considerar la gestión de los requisitos para el período de estudio preliminar donde se define cómo se realizará el proyecto, y la gestión de requisitos durante las distintas iteraciones del desarrollo del producto.

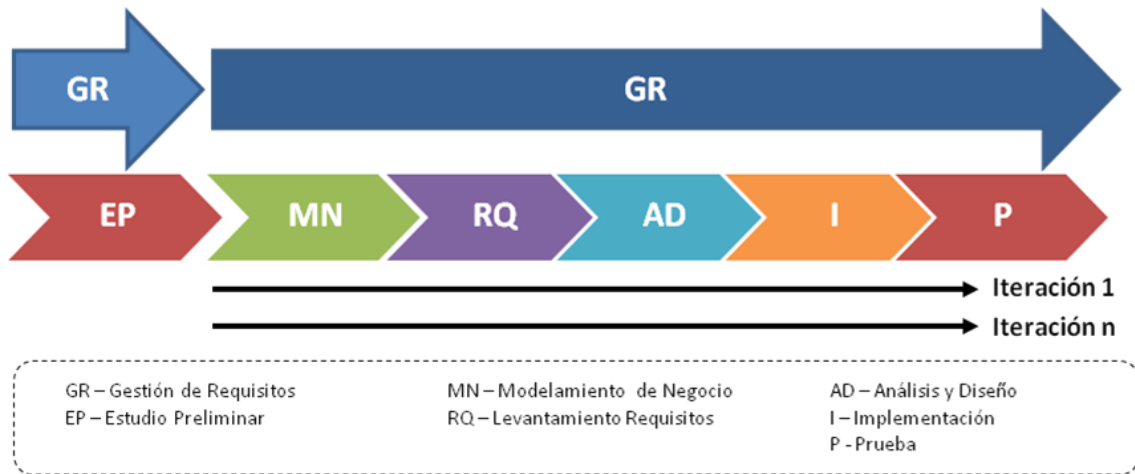


Figura 4: Aplicación de la gestión de requisitos en el ciclo de vida del software.

Las actividades que cubren el dominio de la gestión de requisitos están enfocadas hacia la estrategia de factibilidad/viabilidad, la planificación, el seguimiento, el control de los cambios e inconsistencias del desarrollo y la comunicación de los requisitos, entre los involucrados y el equipo de trabajadores del proyecto. En la memoria colectiva puede consultarse la organización de estas actividades en procesos integrados al dominio de la IR (Soto and Valdés 2010).

En la tabla 1 se identifican además los intercambios de información entre la gestión de requisitos y otras áreas de gestión que aportan al mismo objetivo: lograr que el proyecto termine en el tiempo y con los costos pactados, con un producto de calidad que satisfaga las necesidades reales del usuario y habiendo hecho un efectivo manejo de los recursos humanos y materiales para lograrlo.

Tabla 1. Mapa de relaciones internas entre las áreas de procesos de gestión del proyecto.

Áreas de Procesos	Gestión de Proyecto	Gestión de Configuración	Gestión de Riesgos	Aseguramiento de la Calidad	
G E S T I Ó N D E R E Q U I S I T O S	Entrada →	Requisitos de alto nivel. Tablas de seguimiento de los requisitos organizacionales y del proyecto hacia los CU. Listado de CU a desarrollar en cada Etapa.	Requisitos del producto organizados y documentados independientes por CU. Tablas de seguimiento o dependencia entre CU y entre otros elementos del desarrollo.	Requisitos de alto nivel. Listado de CU a desarrollar en cada Etapa.	Requisitos del proyecto Especificación de los CU. Requisitos no funcionales. Descripción de los procesos de IR.
	Salida ←	Disponibilidad y restricciones de tiempo y recursos para realizar los procesos de la IR. Iteraciones donde los CU se deben documentar para pasar al desarrollo.	Nomenclatura y políticas de almacenamiento de los ECS donde están representados también los CU y otros elementos del desarrollo.	Identificación de riesgos que inciden sobre el desarrollo exitoso de los CU a desarrollar en cada Etapa.	Inconsistencia, no conformidad y solicitudes de cambios en función de la mejora de la calidad de los procesos y artefactos de la IR.

BUENAS PRÁCTICAS PARA APLICAR EN LA GESTIÓN DE REQUISITOS DE PROYECTOS DE DESARROLLO DE SOFTWARE DE GESTIÓN POLICIAL

A partir del estudio bibliográfico realizado (Wieggers 2003) (Wieggers 2006) (CMMI and Team 2006) (Young 2004) la autora identificó un conjunto de buenas prácticas que son válidas para aplicar en los proyectos de desarrollo de gestión policial, teniendo en cuenta las características particulares de estos proyectos y orientado específicamente a la eficiente gestión de los requisitos. Estas prácticas fueron aplicadas eficientemente en el proyecto CICPC y se explican a partir de la experiencia de esta autora y de la ventaja que aportó al proyecto.

BP 1. Preparar el entorno para la informatización:

Esta práctica es aplicable sobre las áreas donde se realizará el levantamiento de los requisitos (elicitación) y permite que los mismos futuros usuarios, mediante la definición y

descripción de sus procesos, estandaricen el “deber ser” de su organización. Esto permite que cuando se haga la elicitación de los requisitos del cliente los analistas puedan enfocarse en el ¿Cómo?, porque previamente se sabe el ¿Qué? se hace en cada área.

BP 2. Utilizar un enfoque de planificación por Metas:

Según Wiegers, cuando un Proyecto tiene fecha fija de entrega, la estimación relevante no es el tiempo que se necesita para hacer los requisitos, sino cuánta funcionalidad (no en número sino en valor de uso) se puede realizar con calidad en ese tiempo fijo. Para aplicar este enfoque, se debe dividir el proyecto en varias fechas finales (Etapas) y a partir de ahí determinar la cantidad de funciones a implementar, como si cada etapa fuera un mini-proyecto con un ciclo completo de vida del software. Cada etapa debe estar asociada a una “Meta” o “Alcance”. Una “Meta” es la combinación resultante de tener en cuenta las funcionalidades del software (las que mayor valor de uso tengan para el usuario), los riesgos, los recursos del proyecto y la cantidad de tiempo con que se cuenta. Es importante que al constituir un mini-proyecto, en la práctica solo se realicen un número reducido de etapas (entre 2 o 3), porque el tiempo y esfuerzo que implica usar la metodología RUP hace que se necesite disponer de varios meses para completar las iteraciones que se necesiten por cada etapa.

BP 3. Utilizar un enfoque de descripción del sistema por Casos de Uso (CU):

Esta práctica complementa la anterior, porque precisamente el valor de uso de un sistema se puede expresar en CU, y beneficia la gestión de los requisitos porque permite enfocar mejor el trabajo de elicitación, análisis, modelación, especificación y validación. Permite planificar y controlar el avance de las restantes etapas del desarrollo del software (diseño, implementación y prueba), sobre la perspectiva de ¿qué? debe poder hacer el usuario sobre el sistema, más que en saber ¿cuáles? son las características o las funcionalidades que el producto debe tener. Esta práctica es beneficiosa también para el control de los cambios, ya que cada CU agrupa uno o más requisitos, lo que disminuye el esfuerzo de controlar 1000 requisitos, al equivalente de 200 o 300 CU. Una desventaja con el uso de esta práctica es que los CU mayormente representan los requisitos funcionales del sistema, por lo que habría que incluir otros mecanismos para que no se dejen de tener en cuenta otros tipos de requisitos.

BP 4. Manejar el desarrollo de los requisitos del producto para múltiples versiones:

Como el producto de software nunca se entrega completo cuando se trabaja por Etapas; y la comprensión de los requisitos se adelanta al resto del desarrollo, en la práctica se van a tener CU cuya versión corresponde a la iteración actual, y otras versiones que son para

posteriores entregas. También se pueden tener CU propuestos (especificados) pero no aprobados, CU que se eliminan o se agregan, o nuevas características asociadas a un CU que no están concebidos en la entrega actual, etc. Por tanto, hay que lograr un entorno adecuado para la administración de los requisitos donde se tenga en cuenta que la descripción del CU no necesariamente debe estar en correspondencia exacta con la iteración actual.

BP 5. Limitar los cambios que sean por adición/eliminación de CU durante una etapa:

La modificación de la cantidad total de CU en una Etapa no debe ser una práctica común, esta decisión debe precederse de un análisis riguroso de impacto, costo/beneficio y viabilidad. Sin embargo, tampoco se puede evitar la necesidad de cambiar los requisitos en la medida en que aumenta la comprensión del problema a resolver; por tanto, lo que plantea esta práctica es tratar de controlar y filtrar todo lo que no sea una necesidad de la que dependen directamente los CU que deben desarrollarse en la etapa. Existen variantes que pueden adoptarse para evitar la adición de CU durante una etapa, por ejemplo, se documenta y se modela, pero no se implementa hasta que no se desarrollen todos los CU previamente pactados. También se puede negociar con el cliente y solicitar que se muevan para la siguiente etapa algunos CU que sean menos valiosos que los que se desean agregar a la etapa actual.

BP 6. Documentar los requisitos (CU) del sistema de forma independiente:

Esta práctica facilita fundamentalmente el control de los cambios y la gestión de los requisitos de un sistema; o sea, en vez de tener un único documento físico o digital donde se describa en detalle el comportamiento del sistema se debe tener un documento por fragmento de funcionalidad (CU). Es necesario además apoyar esta práctica con la realización de Diagramas de CU para visualizar las dependencias, y los totales de CU por subsistemas.

BP 7. Diseñar e implementar varios procedimientos de control de cambios:

Tan inevitable como es el cambio es igualmente la riqueza de formas y momentos en los que se puede presentar. Es recomendable tener establecidos varios procedimientos para realizar los cambios; en dependencia de la formalidad, por ejemplo: pueden existir cambios informales, semi-formales, y formales, donde el grado de formalidad lo da el ente que genera la necesidad del cambio. Otro enfoque puede estar en dependencia de la dirección desde - hasta donde se ejecute el cambio, por ejemplo de la especificación de CU hacia la implementación del sistema, o desde la implementación hacia la especificación de CU. En ambos casos el objetivo a cumplir es el mismo: mantener la

integridad y consistencia del sistema durante todo el proyecto para que no se alarguen los cronogramas por causa de los cambios.

BP 8. Designar personas en el Proyecto (máximo 3) que sean los “Líderes del Producto”: Estas serían las personas de mayor experiencia en los procesos de la IR, en un papel homólogo al del Líder del Proyecto. Su misión sería mantener enfocado al equipo de desarrollo en la visión del Producto. Esto implicaría por ejemplo, que esas personas dirijan técnicamente los procesos de obtención, análisis, modelación, especificación y validación de los requisitos; y que al mismo tiempo gestionen su evolución y control durante todo el proyecto. Los líderes del producto (analistas principales) tendrían por tanto la visión total del sistema, representarían el canal de comunicación entre el cliente y el equipo de desarrollo, pero no como un simple mediador o traductor, sino como el representante de los intereses del cliente para con el equipo de desarrollo y viceversa; por tanto tendrían el poder de decisión de la organización de los CU en las distintas versiones, la potestad para decidir/aprobar si un requisito se cambia, se agrega o se elimina, revisarían todas las especificaciones de los CU para asegurar que estén de acuerdo con la visión global del sistema, y todas las demás acciones que tributan a la gestión de los requisitos. Esta práctica también puede traer algunos problemas a la par de los beneficios, por ejemplo: si es un sistema grande, tiene que ser más de una persona, porque es físicamente imposible que una sola persona centre todo el trabajo que esto implica y hay que evitar a toda costa los “cuellos de botella” en la producción; aunque tengan el mayor poder de decisión sobre el producto, deben consultar con los restantes especialistas técnicos del proyecto porque al final son estos los que ejecutarán las soluciones. Por último deben ser personas comprometidas y estables hasta la entrega del Producto, ya que sobre todo en proyectos grandes, la fluctuación de personal entrando y saliendo es normal, pero si estas personas se mantienen constantes el impacto negativo es menor en la calidad del sistema y el cumplimiento de tiempo pactado.

BP 9. Trabajar con requisitos de alto nivel:

Esta práctica permite hacer una administración más efectiva de los requisitos de un sistema, ya que hace que el equipo trabaje con el concepto ampliado de que los requisitos no solo representan las necesidades de los usuarios (requisitos funcionales y no funcionales), sino que también representan las necesidades de la organización y de los demás involucrados. Esta práctica puede servir para identificar inconsistencias, incompletitudes o contraposiciones de los requisitos de alto nivel, con los de menor nivel desde etapas tempranas del proyecto, tratando de que exista una alineación en los

distintos niveles, logrando así una estrategia más exacta para planificar el ciclo de vida del sistema. Niveles no es lo mismo que tipos de requisitos, pero hay relación entre estos. Al efecto se proponen: Requisitos Organizacionales, Requisitos del Proyecto, y Requisitos del Producto.

Los requisitos organizacionales: son los beneficios que se esperan obtener con la utilización del software en los niveles estratégicos, tácticos y operativos de la institución. Podrían comenzar con la frase “con el nuevo sistema automatizado se busca.....” y concatenar con un verbo infinitivo o sentencia que exprese un cambio y el impacto del mismo. Normalmente estos requisitos surgen cuando se hace un inventario de los problemas reales que enfrentan las distintas áreas, pero teniendo cuidado al filtrar esta información, porque no todos los problemas pueden resolverse con un sistema (ejemplo los problemas de infraestructura de comunicación, de gestión del suministro de recursos, etc.), y otros que sí pueden resolverse con un software pero que ya existen herramientas profesionales para resolverlos. Se pueden encontrar ejemplos de estos requisitos en la sección de la definición del proyecto técnico de CICPC de la memoria colectiva.

Requisitos del Proyecto: son los procesos que serán automatizados y que sirven como definición de alcance del proyecto (frontera sobre la cual se realizará el análisis de las funciones y datos para el sistema); también el listado de restricciones impuestas a la solución o requisitos de calidad generales que incidan sobre la estrategia posterior del desarrollo del sistema, pueden ser de tipo administrativas, técnicas o de otro tipo que el cliente esté interesado que se respete. Se pueden encontrar ejemplos de estos requisitos en la sección de estudio preliminar del proyecto y del producto de la memoria colectiva.

Requisitos del Producto: serían el listado organizado de los CU del sistema, la descripción de esos CU, así como requisitos no funcionales y reglas de negocio asociadas a los mismos.

BP 10. Definir de forma preliminar y explícita los requisitos de calidad y las restricciones impuestas a la solución:

De acuerdo con la práctica anterior, los requisitos a nivel de proyecto son difíciles de obtener en el proceso de elicitación de requisitos porque no salen de forma natural en la recopilación de información de las áreas a informatizar. Realizar esta práctica de forma intencional y preliminar (antes del levantamiento de requisitos) permite entre otras cosas: explorar características que incidan de forma negativa en los tiempos de diseño e implementación, consiguiendo llegar a acuerdos para lograr cosas posibles dentro de la

realidad; ahorrar tiempo en la etapa de levantamiento de requisitos; y realizar una efectiva gestión de riesgos asociados al desarrollo del software.

BP 11. Identificar e interactuar con diferentes tipos de involucrados:

En un proyecto hay personas involucradas con distintos niveles de interés y puntos de vista; muchos software fracasan, porque han tenido en cuenta solamente las necesidades de un grupo de interesados. Por ejemplo: a un investigador le interesa qué datos deben incluir en la aplicación, pero al gerente del proyecto le interesa que se cumpla el alcance pactado para el proyecto, etc. Se recomienda que se identifiquen los siguientes tipos/grupos de involucrados: Clientes (las personas que pagan por el avance del proyecto), Usuarios (los que van a usar el sistema), Expertos Funcionales (los que tienen una mayor experiencia en cuanto a la misión, visión y funciones de las áreas donde trabajan), Reguladores del Proyecto (los que representan al cliente para monitorear el avance del proyecto y velar porque se cumplan sus necesidades, así como garantizar condiciones para que el equipo de desarrollo haga su trabajo), Reguladores del Producto (ingenieros de sistema que usualmente son contratados para las áreas de tecnología de las instituciones policiales, los cuales intervienen en la toma de decisiones técnicas asociadas al sistema), Reguladores del Negocio (expertos o reguladores legales que garantizan que el sistema no incurra en violaciones de lo que está establecido en sus leyes o reglamentos).

BP 12. Establecer una estructura organizacional de trabajo a partir de los requisitos del sistema:

Todo lo que se tenga que programar en el proyecto se traduce a CU, sea o no parte del software a entregar. Esta práctica permite que de forma obligatoria todos los roles involucrados en el proyecto revisen los CU del sistema constantemente, ya que el trabajo se desglosa a partir de los mismos y la entrada principal de las actividades del desarrollo lo constituyen por tanto los CU. Además al asignar el trabajo a personas o equipos específicos en un plan de trabajo, se están declarando “dueños de CU” lo que facilita la planificación y la consistencia del sistema en la medida en que avance el proyecto y se realicen varias iteraciones. Ver Cronograma de Desarrollo de la iteración 2 de Elaboración donde se evidencia la organización a partir de los CU a desarrollarse en dicha etapa (Soto and Valdés 2010).

BP 13. Establecer diferentes vías de participación de los involucrados durante todo el desarrollo del proyecto:

Esta práctica se recomienda sobre todo porque estos proyectos tienen a los clientes y al equipo de desarrollo geográficamente separados. Inicialmente solo se considera obvia la participación del cliente durante el estudio preliminar y la captura de los requisitos, sin embargo la práctica demuestra que se necesitan formas de comunicación y definición de determinados involucrados para actividades relacionadas al desarrollo de la IR. Ejemplo: para las revisiones y validaciones de las especificaciones de los CU, para la toma de decisiones técnicas, para la aclaración de temas no entendidos totalmente, etc. Para aplicar esta práctica se puede desarrollar un “Plan de participación de involucrados”, o acordar explícitamente antes de comenzar cada etapa del proceso de IR como se realizará de forma práctica el intercambio, actualización y respuesta a las necesidades de los involucrados. Ver como ejemplo la bitácora de comunicación durante el desarrollo de las aplicaciones (Soto and Valdés 2010).

BP 14. Usar plantillas o patrones:

Esta técnica, recomendada por varios autores (Durán A. 1999; Escalona 2002), tiene por objetivo describir el sistema mediante el lenguaje natural pero de una forma estructurada. Una plantilla tiene una serie de campos y una estructura predefinida, aunque se debe seguir usando para ello el lenguaje del usuario. Las plantillas eliminan parte de la ambigüedad al estructurar la información. Y se pueden crear fácilmente listas de chequeo para verificar la calidad. Sin embargo, si el nivel de detalle elegido es demasiado estructurado, el trabajo de rellenar las plantillas y mantenerlas, puede ser demasiado costoso.

ACTIVIDADES DE GESTIÓN DE REQUISITOS EN PROYECTOS DE DESARROLLO DE SOFTWARE DE GESTIÓN POLICIAL

En correspondencia con el ciclo de vida del desarrollo de los proyectos de software de gestión policial y las prácticas antes mencionadas se describen las actividades de gestión de los requisitos; siendo responsabilidad del(los) analista(s) principal(es) la realización de las mismas. En la tabla 2 se enuncian las actividades, el propósito que aclara la necesidad de realizar cada una y se referencia la(s) buena(s) práctica(s) que se logra implementar con cada actividad.

Tabla 2. Actividades de gestión de requisitos.

Etapa de Estudio Preliminar			
No.	Actividad	Propósito	Referencia
1	Eliminar ambigüedades o inconsistencias, identificando o describiendo claramente cada uno de los requisitos de entrada del proyecto.	Conocer el nivel de informatización que existe en la institución, establecer y ajustar los objetivos del proyecto. Las técnicas recomendadas son la entrevista a determinados tipos de involucrados (clientes, reguladores del proyecto y del producto) y el cuestionario principalmente a los expertos funcionales. Ver cuestionario aplicado a los expertos funcionales en la sección de estudio preliminar del Proyecto de la memoria colectiva.	BP.9 BP.10 BP.11
2	Definir claramente el alcance del proyecto.	Hacer una derivación de las necesidades del cliente a partir de los requisitos de entrada del proyecto mediante la aplicación de cuestionarios a las áreas a informatizar, basado en los requisitos a nivel de organización. La técnica recomendada es la de Taller o Seminario de requisitos.	BP.1 BP.9 BP.11
3	Preparar una propuesta técnica que responda a como satisfacer cada uno de los requisitos de entrada del proyecto.	Con apoyo del equipo técnico, crear si es necesario diferentes proposiciones coherentes. La estructura de este documento debe lograr como mínimo describir una arquitectura básica del sistema y todos los requisitos de alto nivel, para que sean revisados, validados y firmados por representantes de varios tipos de involucrados (cliente, regulador del proyecto, regulador del producto, regulador del negocio).	BP.9 BP.10 BP.11
4	Definir las etapas del proyecto y qué criterios de medida se revisarán para determinar el progreso del mismo.	Incluir la propuesta de las metas a lograr por etapas en el informe técnico que sea revisado, validado y firmado por representantes de los distintos tipos de involucrados. Incluir los CU a entregar por cada etapa como anexo al Plan de Desarrollo que firmen ambas partes.	BP.2 BP.5
5	Obtener validación técnica del	Revisión y validación de la viabilidad y rapidez con que puede lograrse la propuesta técnica realizada. No solo es importante que el cliente esté conforme,	BP.12

CAPITULO 2. DESCRIPCIÓN DE LA SOLUCIÓN APORTADA

	equipo de desarrollo.	sino también debe asegurarse que el equipo esté de acuerdo en que es posible construir dicha propuesta.	
6	Almacenar y organizar la documentación.	Para mantener el proyecto enfocado en las metas se debe disponer para consulta todo el conocimiento que se haya creado en la etapa del estudio preliminar, como: minutas de reuniones, entrevistas, cuestionarios, informes técnicos, etc.	
Etapa de desarrollo del software			
1	Preparar las áreas a informatizar.	Para poder garantizar un efectivo y rápido levantamiento de requisitos para el producto.	BP.1 BP.11 BP.14
2	Capacitar a los distintos tipos de involucrados que participen en procesos de IR.	Tener en cuenta tres tipos de entrenamiento; uno orientado a las habilidades y conocimientos que deben tener los analistas del proyecto, otro orientado a los miembros del equipo de desarrollo (arquitectos, diseñadores, programadores, probadores, etc.), y un último orientado a los involucrados (clientes, usuarios, y especialistas en sistemas de la institución).	BP.3 BP.6 BP.11 BP.13 BP.14
3	Definir las políticas para la gestión de los requisitos en el proyecto.	Determinar cómo se realizará la identificación, planificación, monitoreo, actualización y evaluación de los requisitos del producto.	BP.3 - 8 BP.12
4	Planificar las actividades del desarrollo de los requisitos del sistema como procesos.	Lograr un plan detallado de cualquier objetivo del proyecto cuyo resultado tenga que ver con el desarrollo de los requisitos del sistema, ejemplo: plan de modelación del negocio, plan de realización de propuesta del sistema, plan de especificación de una versión, etc. Se deben detallar las actividades necesarias a realizar para lograr el objetivo hasta el nivel donde se puedan asignar a una sola persona.	BP.12
5	Planificar y ajustar los objetivos de las iteraciones en función de los requisitos de entrada del proyecto.	Priorizar regularmente los CU por iteraciones en función de variables técnicas y de impacto para el cumplimiento de los objetivos del sistema.	BP.2 BP.4 BP.12

CAPITULO 2. DESCRIPCIÓN DE LA SOLUCIÓN APORTADA

6	Monitorear el avance del desarrollo de los requisitos del sistema.	Realizar con frecuencia semanal un inventario por cada proceso, indicando los CU planificados para la semana, los CU reales terminados en la semana, y el porcentaje de cumplimiento del plan. Se puede realizar un desglose además por equipos de desarrollo para identificar los responsables de los CU que faltan para lograr el cumplimiento del plan. Esto permitirá al líder del proyecto realizar estimaciones más reales del tiempo y esfuerzo para lograr cumplir en los plazos previstos.	BP.12
7	Lograr el entorno tecnológico adecuado para la administración de los requisitos.	Garantizar con el uso de herramientas automáticas que el equipo de proyecto acceda a la planificación actualizada, que se pueda realizar el monitoreo, que se realicen las actualizaciones consistentemente y se respeten todas las políticas de trabajo establecidas en el proyecto. Se ha demostrado en la práctica que muchas herramientas entorpecen el trabajo, y que el fácil acceso, la actualización constante y la integración de las herramientas tecnológicas es más importante que tener la última versión de una herramienta para la gestión de requisitos que muy pocas personas sepan usar.	BP.4 BP.6 BP.7 BP.12
8	Gestionar los cambios sobre los CU durante todo el proyecto.	El cambio es inevitable, por tanto no se debe tomar una posición de rechazo, sino hacer los cambios de forma organizada y sincronizar todos los artefactos (documentos, diagramas, software, etc.) que tengan que ver con el mismo.	BP.4 - 7
9	Tracear el impacto y la gestión de inconsistencias.	Evitar las inconsistencias, mediante la creación de tablas de seguimiento, y mantenerlas actualizadas como medio de consulta cada vez que se quiera realizar un cambio de nivel medio/alto o periódicamente para verificar posibles inconsistencias creadas en el proceso. También mediante el uso de diagramas de representación visual del sistema.	BP.6 BP.9
10	Auditar y realizar inspecciones regularmente.	Revisar y aprobar todos los documentos relacionados con los requisitos del sistema, creados o modificados como parte de las actividades ingenieriles.	

PROCEDIMIENTOS Y MECANISMOS USADOS EN LA GESTIÓN DE REQUISITOS DE PROYECTOS DE SOFTWARE DE GESTIÓN POLICIAL

La realización de los siguientes mecanismos y procedimientos, permiten lograr algunas de las actividades descritas en el epígrafe anterior de forma tal que se concrete la implementación de las buenas prácticas de gestión de requisitos anteriormente mencionadas.

Actividad: Definir las etapas del proyecto y qué criterios de medida se revisarán para determinar el progreso del mismo.

Procedimiento de planificación del proyecto enfocada en las metas (objetivos).

Responsable: líder del proyecto y del producto (analista principal)

Para este procedimiento se toma como referencia lo definido por Wiegers (2006) para proyectos largos de software de gestión:

- R 1.El por ciento de tiempo que se necesita para la captura de requisitos, es aproximadamente un 9% del total estimado.
- R 2.El por ciento de CU arquitectónicamente significativos para sistemas grandes es de aproximadamente un 30% del total.
- R 3.Los CU se incrementan a razón de 1 al 3 % por mes durante la etapa de desarrollo por necesidades del cliente o de los desarrolladores.

Teniendo en cuenta estos tres principios, para el proyecto técnico a elaborar en el estudio preliminar, se realiza el enfoque de la división del proyecto por metas según lo siguiente:

1. Descontar del tiempo total de duración del proyecto, las semanas que deben dedicarse a la captura de requisitos (calcular según R1). Para el tiempo restante delimitar etapas (de no menos de 6 meses). Cada etapa será un mini-proyecto que incluye desde la captura de requisitos hasta las pruebas de aceptación.
2. En la primera etapa solo comprometer el 30% de los CU pactados para el proyecto (calcular según R2) aunque en ese momento no se sepa cuáles son.
3. En el análisis y negociación de los requisitos del producto, se deja un margen de CU con respecto al total de CU pactados para acomodar los cambios que surjan como necesidad durante el desarrollo (calcular según R3).
4. Cuando concluya la captura de requisitos, con el listado validado de CU se prepara una propuesta donde se filtren los CU que respondan a la meta que se acuerde con el cliente. La propuesta debe tener en cuenta además, todos los siguientes criterios de valor:

- a. Arquitectónicamente significativos.
 - b. Tributa directamente al cumplimiento de algún requisito organizacional.
 - c. Responde a algún requisito del Proyecto.
 - d. No existe riesgo medio o alto que impida su implementación.
5. Presentar la propuesta a los representantes de los involucrados: cliente, reguladores del proyecto y reguladores del producto.

Se puede consultar un ejemplo de la selección de los CU por etapas en la sección de desarrollo de requisitos del producto de la memoria colectiva (Soto and Valdés 2010).

Actividad: Preparar las áreas a informatizar.

Procedimiento para preparar las áreas a informatizar.

Responsable: especialista en procesos y equipo de analistas.

1. Realizar un diagnóstico mediante un instrumento de evaluación que mida: organigrama, perfiles de cargos por puestos de trabajo, funciones por unidades organizativas, identificación y clasificación de procesos claves y de apoyo, relaciones internas entre procesos y externas entre instituciones, reglamento interno, entre otros aspectos.
2. Realizar un informe de resultados y discutirlo con el cliente.
3. Acordar con el cliente una estrategia de completamiento y definición de la información que falta, la misma puede ser como sigue:
 - a. Identificación de los procesos claves, procesos de apoyo y procesos estratégicos.
 - b. Revisión y análisis de la documentación existente en función de los procesos.
 - c. Mapeo de procesos, sub procesos, ficha de procesos, elaboración de mapeo de relaciones internas y externas.
 - d. Identificación de la base de información necesaria para sustentar los procesos.
 - e. Refinar mediante la validación con los especialistas funcionales y directores de las áreas.
 - f. Revisión final por reguladores del negocio para la adaptación al marco legal.
 - g. Validación de los mapas de los procesos.
 - h. Entrega de la documentación de las áreas a informatizar al analista principal.
4. Revisión de la documentación entregada (se pueden hacer etapas de entrega por tipos de procesos, para comenzar el trabajo del equipo de analistas en paralelo).

Este procedimiento aumenta su aporte cuando los resultados son usados para preparar la matriz de contactos para las entrevistas (expertos funcionales). Para entregar el plan de entrevistas con tiempo de antelación a los reguladores del proyecto, y si es posible

circular una citación formal al entrevistado que indique la fecha, hora, propósito y condiciones que debe tener para la misma.

Actividad: Capacitar a los distintos tipos de involucrados que participen en procesos de IR.

Procedimiento para entrenar a los analistas que realizarán la elicitación de los requisitos.

Responsable: equipo de analistas.

1. Realizar un inventario de funcionalidades de los sistemas existentes mediante el estudio de los manuales de usuario o inspección directa mediante la manipulación de los mismos (en caso de que no existan manuales).
2. Elaborar un glosario de términos de negocio.
3. Realizar talleres con especialistas funcionales (ejemplo: investigadores, criminalistas y médicos legales) de organismos homólogos en Cuba. (se puede utilizar el glosario de términos del negocio para dirigir los talleres a propósitos concretos).
4. Talleres con especialistas informáticos de organismos homólogos en Cuba. (identificar requisitos funcionales candidatos y no funcionales representativos).
5. Validación del levantamiento de funcionalidades de los sistemas existentes.
6. Estudio de la base legal de la institución a informatizar.
7. Estudio de la documentación obtenida como parte de la preparación de las áreas a informatizar.

Actividad: Definir las políticas para la gestión de los requisitos en el proyecto.

Mecanismo de identificación de los CU del sistema.

Al ser tantos CU se recomienda una identificación lógica, y no física en función del nombre del fichero. Se identificarían mediante una organización lógica en tres niveles: Objetivo/Sub-objetivo/Funcionalidad, así quedan alineados con el alcance del sistema definido al inicio del proyecto. Objetivo → Proceso Clave o de Apoyo que se automatiza; Sub-objetivo → Parte del proceso que se automatiza; Funcionalidad → CU (debe incluirse el nombre de la entidad de información principal que se inserta, se actualiza, se consulta o se elimina en el sistema). Se puede consultar el ejemplo de aplicación de este mecanismo en la sección de desarrollo de requisitos del producto de la memoria colectiva (Soto and Valdés 2010). No importa que un CU sea incluido o extendido de otro, su identificación debe ser relativa al objetivo que cumpla en el sistema.

Mecanismo de realización de tablas de seguimiento para la trazabilidad.

La trazabilidad es la habilidad de seguir la vida de un requisito desde su origen hasta su liberación del proceso de desarrollo. Teniendo en cuenta la BP.3 las matrices deben referenciar las dependencias de los CU del sistema con respecto a otros elementos. Si se

utiliza un repositorio solo se necesita implementar una trazabilidad vertical (Dahlstedt and Persson 2005). Usando un mismo tipo de dependencia, con un solo nivel de profundidad. Esto se logra teniendo en cuenta un único propósito para cada tabla de trazabilidad; o sea, tan simple de realizar y tan fácil de entender y manipular como sea posible por todos los involucrados. Para ello se recomienda utilizar tablas de seguimiento para los CU. En la memoria colectiva se pueden consultar algunas tablas de seguimiento con diferentes propósitos (Soto and Valdés 2010).

Tabla 3. Propuesta de las tablas de seguimiento a realizar.

Caso de Uso	Dependencia	Responsable	Propósito
Tributa al cumplimiento.	Requisitos organizacionales	Líder Proyecto	Manejo riesgos, planificación y toma de decisiones.
Responden a	Requisitos del Proyecto	Líder Proyecto	
Automatiza parcial o totalmente.	Procesos Negocio (CUN)	Analista Principal	Lograr y mantener la organización del sistema.
Un rol del negocio utiliza esta función.	Trabajador de Negocio	Analista Principal	
Incluye, Extiende o Generaliza.	Caso de Uso.	Analista o Dueño de CU.	Gestionar los cambios.
Crea, Actualiza, Elimina o Consulta.	Entidades (datos).	Cualquier rol	
Asociado al cumplimiento	Requisito No Funcional.	Desarrolladores o Dueño de CU.	
Genera	Salidas impresas.	Analista o Desarrollador.	Gestionar los cambios.

Para confeccionar una tabla de seguimiento se recomienda:

1. Ubicar en las filas los elementos que sean agrupadores del par de seguimiento (ejemplo entre CU y Procesos de Negocio, los que agrupan son estos últimos).
2. Identificar los elementos de las filas que sean altamente dependientes, y buscar todos los elementos del otro par que tengan algún tipo de dependencia con esos primeros (con solo el 20% de los elementos más dependientes identificados, se van a cubrir entre el 60 y 70% del total de dependencias que existen).
3. Buscar por los nombres y declaraciones de los elementos, aquellos pares que se identifiquen igual, ya que es posible que exista algún tipo de dependencia entre estos.
4. Finalmente revisar y confirmar que los elementos restantes sean singulares (no tienen ninguna relación).

Procedimiento de trabajo con los Requisitos del sistema.

Responsable: líder del producto (analista principal)

1. Por medio de tablas u hojas de cálculo crear un artefacto que contenga todos los CU que se han identificado, organizados según la lógica Objetivo/Sub-Objetivo/CU.
2. Autorizar a una única persona (Analista Principal) para que pueda actualizar dicho artefacto.
3. Adicionar en columnas atributos que deben mantenerse actualizados. Ver la tabla 4.
4. Ubicar en un repositorio, asegurar que todo el proyecto tenga acceso y que la única información verídica se considere la que aparezca en dicho fichero.
5. No deben adicionarse o eliminarse CU de una etapa en curso, o la iteración en que corresponde desarrollarse el CU en la etapa. (si por cuestiones excepcionales el cliente necesitara CU que no están previstos para la etapa en curso, se declara una etapa intermedia superior, que se tratará de completar cuando se tengan asegurados todos los CU de la etapa en curso.
6. Los asignados como dueños de los CU son los que crearán, modificarán o eliminarán los artefactos y componentes que se desarrollen durante el ciclo de vida para ese CU (documento de especificación, prototipo no funcional, diccionario de entidades de datos, clases, casos de prueba, diagramas, etc.)
7. También se declararán los revisores/aprobadores de cada CU, que en este caso son los líderes del producto y algunos otros especialistas que tengan experiencia en el análisis, modelación y especificación de sistemas.
8. Todos los procesos de IR que se planifiquen deben incluir una actividad de revisión y aprobación que en dependencia del cronograma, puede realizarse en talleres con los revisores/aprobadores, o por inspección técnica directa de estos al resultado final.

Tabla. 4 Atributos que describen los CU identificados para el sistema.

Atributo	Significado	Valores
Impacto Arquitectónico	Indica los CU que deben priorizarse en las primeras iteraciones de una etapa.	Alto, Medio, Bajo (Ver métricas para la clasificación de CU en la memoria colectiva)
Complejidad	Indica dificultad de las tareas de desarrollo asociadas, y permite controlar el total de CU que fueron pactados según complejidad.	
Etapas	Indica prioridad para el desarrollo	Etapas 1, Etapas 2, Etapas 3
Iteración	Indica, dentro de una Etapa, el orden con que se planificará para	(número de iteraciones)

	realizar las actividades ingenieriles.	
Estado	Indica el por ciento de avance que representa el CU para el proyecto. (se usará cuando se quieran entregar informes)	Propuesto (en especificación) Aprobado (validado y aprobado para implementar) Incorporado (que ya se programó) Validado (que ya pasó por una revisión de aceptación del cliente)
Analista **	Indica quien es el especialista en esa funcionalidad del sistema, y por tanto el que mejor preparado está para cualquier actividad de la IR que tenga que ver con ese CU.	Nombre del responsable de la creación y especificación del CU.
Asignado **	Indica el equipo dueño del CU para la planificación de las iteraciones, para las correcciones de errores, etc.	Nombre del responsable del equipo de desarrollo que se encarga de la implementación del CU.

***En momentos puntuales por estrategia del proyecto algún equipo o analista, puede asumir de forma temporal el desarrollo o actualización de estos artefactos tratando de adelantar o cubrir atrasos de cronograma.*

Procedimiento para distribuir los CU por iteraciones en una Etapa.

Responsable: planificador, líder del proyecto y del producto (analista principal)

Para este procedimiento es muy útil construir primero la tabla de seguimiento de relaciones de inclusión, extensión y generalización o usar los diagramas de CU. Tener en cuenta los siguientes grados de prioridad:

- (1er Nivel). Los CU que sean arquitectónicamente significativos.
- (2do Nivel). Los CU donde se implemente la inclusión en el sistema de alguna de las entidades del negocio y todos los CU relacionados a este por inclusión.
- (3er Nivel). Los CU para consultar, eliminar, modificar datos de entidades del sistema (ejemplo el estado de una entidad).

También hay que considerar que el total de CU por iteraciones, deben seguir cierta proporcionalidad, pero intencionalmente dar mayor peso a la 1era iteración. Ejemplo con tres (3) iteraciones: 40% – 30(35)% – 20(25)%.

De acuerdo a esta proporcionalidad habrá CU de 2do nivel que puedan realizarse en la 1era iteración, pero otros tendrán que realizarse en la 2da o incluso en la 3era iteración para que el total de CU sea equitativo. Se puede consultar un ejemplo de la distribución de los CU por iteraciones para la etapa 1 en la sección de planificación de la gestión de requisitos de la memoria colectiva (Soto and Valdés 2010).

Actividad: Planificar las actividades del desarrollo de los requisitos del sistema como procesos.

Procedimiento para planificar los procesos de la IR.

Responsable: planificador del proyecto y el líder del producto (analista principal)

El propósito es obtener un plan detallado de cualquier proceso que se deba desarrollar sobre los requisitos del sistema, ejemplo: modelar el negocio; levantamiento de requisitos de una etapa; especificación los CU de una versión; modificación la línea base de los CU de una versión, etc. Para lograr el plan detallado:

1. Identificar el núcleo de actividades sistemáticas que según dependencia y orden se necesitan para completar el proceso, tratando de que el producto del trabajo a realizar esté lo más desagregado posible y se pueda verificar.
2. Recoger las políticas o reglas generales de cómo se debe entregar cada producto de trabajo y si hay algún tipo de prioridad entre los CU.
3. Asociar un tiempo estimado de tiempo requerido para cada actividad según tipo de recurso (Ejemplo: 2 horas un Profesor, 4 horas un Analista con experiencia, etc.)
4. Inventariar los recursos: PC (cantidad física y cantidad de turnos de trabajo disponibles), humanos (cantidad y tipo de recurso) y tiempo disponible (según cronograma del proyecto).
5. Teniendo en cuenta la BP.12 se esquematiza un plan donde horizontalmente se ubican los CU que se incluyen en el objetivo, con su prioridad (si es un criterio a tener en cuenta), y verticalmente se ubica el tiempo (mes, semana y día) que se disponga.
6. Distribuir las actividades teniendo en cuenta, la disponibilidad del tipo de recurso humano que se necesita (analista o dueño del CU), y del recurso tecnológico (PC).
7. Realizar las adecuaciones necesarias si con la cantidad de recursos disponibles la planificación se extiende por más tiempo de lo que está disponible. Se negocia con el Líder del Proyecto para aumentar la cantidad de personas o PC disponibles para esas actividades.
8. Realizar un entrenamiento previo donde se explique el proceso, las políticas o reglas a seguir, la plantilla (si existe) de los productos del trabajo, la organización de los mismos en el repositorio del expediente del proyecto, etc.
9. Informar la ubicación física y accesible (repositorio) del plan detallado para que cada recurso pueda consultar diariamente la asignación del trabajo.

Se puede consultar un ejemplo de un plan detallado para el proceso de especificación de los CU de la etapa 1 en la sección de planificación de la gestión de requisitos de la memoria colectiva (Soto and Valdés 2010).

Actividad: Monitorear el avance del desarrollo de los requisitos del sistema.

Procedimiento para monitorear los procesos de la IR.

Responsable: planificador del proyecto y el líder del producto (analista principal)

1. Extraer del plan detallado las actividades, e informar a cada persona que corresponde para el día en curso. Hacer una breve reunión al inicio de la jornada laboral o circularlo en una tabla con Persona/Actividad/CU.
2. En un horario fijo establecido al final del día, se solicita reportar el cumplimiento de cada actividad. Cada persona actualiza la tabla en el repositorio, o informa personalmente o por correo al planificador.
3. Actualizar con este reporte el plan detallado y realizar la tabla Persona/Actividad/CU del siguiente día, agregando como actividades adicionales las que no pudieron completarse en el día planificado.
4. Medir avance contra el plan en periodos de una semana, de modo que se puedan tomar medidas correctivas en caso de algún recurso que no esté trabajando con eficiencia, o modificar el plan detallado del tiempo restante, si la realidad demuestra que la actividad requiere más tiempo de lo que se planificó inicialmente.

Actividad: Gestionar los cambios sobre los CU durante todo el proyecto.

Procedimiento de control de cambios.

Responsable: líder del producto, analistas, desarrolladores, diseñadores de BD, etc.

Los cambios sobre los requisitos de un sistema, pueden ocurrir en momentos donde deben aplicarse procedimientos de distintos niveles de formalidad, aunque en principio se aplican los pasos básicos que teóricamente se conocen. Ver Anexo 1 para consultar el proceso de forma general.

Momento: Cuando hay periodos de aprobación de artefactos por el cliente.

1. La solicitud llega formalizada en un documento.
2. Debe recibirla y responderla siempre un líder del producto y debe contar con el criterio de los dueños del(los) CU involucrado(s) si es una petición de cambio de complejidad media o alta.
3. Si la solicitud de cambio es para perfeccionar el sistema, no para corregir un error; se mantiene la funcionalidad actual (CU) y se agrega(n) CU (incluido o extendido) donde

- se haga efectiva la mejora. Esto es importante ya que estratégicamente el cambio aquí no representaría atraso en el cronograma.
4. No se acepta un cambio hasta que no se defina exactamente que debe modificarse, y concretamente que solución se quiere (si existen variantes para resolver la petición).
 5. Documentar cual es el cambio concreto que se realizará (en un lenguaje que entienda el cliente), la complejidad del cambio y en cuál etapa será efectivo (actual o siguiente). Esto debe quedar aprobado por el solicitante.
 6. Registrar cada orden de cambio indicando: nombre del CU, ¿qué hay que cambiar?, ¿donde?, analista y dueño de CU responsable de ejecutar dicha orden. Dejar registrado esto en una hoja de cálculo, o en una herramienta automática.
 7. No se divulgan las órdenes de cambio hasta que no se hayan revisado y dado respuesta a todas las peticiones que se refieran al mismo CU (en la práctica no llegan o se responden todas las no conformidades o peticiones de cambio de un CU al mismo tiempo).
 8. Planificar detalladamente el proceso de realizar los cambios (ver procedimiento de planificar los procesos de la IR), porque cuando se contabilice, un mismo dueño de CU tendrá pendiente varias órdenes de cambio de distintos CU.
 9. Para realizar el cambio, si implica la modificación de varios artefactos se prioriza el artefacto que está siendo aprobado por el cliente (ejemplo, si es aceptación del software, se cambia primero este y se audita; posteriormente se modificaría la documentación, los diccionarios de datos, los diagramas, y todo lo que tenga relación de dependencia con este CU).
 10. Si el artefacto que se debe cambiar está en su línea base, se hace primero una copia de trabajo (versión o rama), y los cambios se hacen sobre la copia. La revisión (auditoría) también debe hacerse sobre esa versión.
 11. El cambio se documenta directamente sobre el artefacto modificado, por ejemplo: en tablas de control de cambios, en los registros del repositorio; y en el estado de la orden de cambio donde hay que indicar que ya está cumplida.
 12. La auditoría depende del artefacto modificado (si es software pruebas de regresión, si es un documento inspección directa o lista de chequeo), y pueden intervenir distintos revisores (el cliente, el analista principal, los reguladores del producto). También se documenta en el artefacto, con la fecha y el nombre de la persona que lo aprobó.
 13. Finalmente se integra la copia de trabajo con el original, o se sustituye (eliminando el original).

Momento: Cuando hay periodos de aprobación de artefactos por calidad para liberación.

1. La solicitud llega formalizada en un documento, pero siempre es sobre corrección de defectos (no conformidad).
2. Debe recibirla y responderla siempre el dueño del CU (analista o jefe de equipo de desarrollo asignado). Se consulta solo con el analista principal en caso necesario.
3. La solicitud se responde directamente mediante la realización del cambio (no hay que documentar una respuesta de qué cambio se va a realizar).
4. Se registran órdenes de cambio solo si es necesario: porque el responsable de ejecutar el cambio no es la persona que recibió la solicitud, o porque estén implicados varios artefactos que se necesitan actualizar, que tienen relación con el CU que está incorrecto.
5. El cambio se ejecuta inmediatamente, comenzando por las de mayor complejidad.
6. El cambio se documenta directamente sobre el artefacto modificado (por ejemplo en tablas de control de cambios, y en el registro del repositorio). También se debe actualizar el documento donde se recibió la solicitud formalmente, y si se generaron órdenes de cambio hay que indicar que ya están cumplidas.
7. Cuando se concluyan todos los cambios sobre el artefacto revisado, se entrega una nueva versión del mismo y el documento de no conformidades respondido.
8. La auditoría la realiza siempre el ente de calidad por la misma vía en que detectó la no conformidad.

ENTORNO TECNOLÓGICO QUE SOPORTE LA GESTIÓN DE REQUISITOS EN PROYECTOS DE DESARROLLO DE SOFTWARE DE GESTIÓN POLICIAL

Las herramientas que se utilicen deben tener como principio fundamental la integración y la facilidad de acceso de los miembros del proyecto al uso de la información que se maneja. Más que proponer una determinada herramienta por sus características funcionales y técnicas, lo que se necesita es que responda a propósitos específicos de la IR en un proyecto.

Tabla 5. Herramientas que apoyan las actividades de gestión de requisitos.

Propósito	Herramienta	Observaciones
Planificación	Profesional para gestión de	La primera es muy útil para planificaciones macros de mucho tiempo (meses); la segunda se recomienda para planificación detallada de los

CAPITULO 2. DESCRIPCIÓN DE LA SOLUCIÓN APORTADA

	proyectos. Hojas de cálculo	procesos de la IR (semanas).
Monitoreo y Control	Profesional para gestión de proyectos. Hojas de cálculo Repositorio y Sistema para la gestión y seguimiento de errores.	Se debe poder realizar registros de cumplimiento de tareas en ambas herramientas de planificación, por tanto igualmente se pueden obtener con ellas reportes de estado y cumplimiento (tablas y gráficos). Importante que exista un sistema para asignar tareas (sin un plan detallado), cuando hay que controlar el total de trabajo a cumplir en alguna etapa, y se quiere que se haga en el menor tiempo posible, esto se logra visualizando los tickets cerrados y abiertos contra el total de tickets por componente asignado.
Trabajo colaborativo	Herramienta CASE para Modelado Repositorio y Sistema para la gestión y seguimiento de errores.	Se usa en la elaboración de modelos abstractos de entendimiento del sistema, mediante el uso de diagramas de representación visual. Se logra trazar la organización lógica de los CU del sistema, a una estructura física de almacenamiento, y al mismo tiempo se puede lograr el cumplimiento de todas las políticas de trabajo con los requisitos del sistema.
Control de Cambios	Hojas de cálculo Herramienta de Gestión de Requisitos. Sistema para la gestión y seguimiento de errores.	Se registran y mantienen actualizadas las tablas de rastreo de relaciones entre los CU y demás elementos seleccionados; y los valores de los atributos definidos para cada CU. Se obtienen mediante filtros las dependencia entre elementos para realizar análisis e informes de impacto, y determinar y registrar todas las órdenes de cambio que se necesitan.
Control de inconsistencias	Hojas de cálculo Herramienta de Gestión de Requisitos.	Para analizar de forma sencilla las relaciones de los CU con los elementos que le dieron origen como los requisitos del proyecto y los organizacionales; o aquellos elementos que se derivaron de los CU.

En el proyecto CICPC se utilizaron: Microsoft Project, Microsoft Excel, el TRAC como herramienta online de gestión y seguimiento de errores que viene integrado con un repositorio, sobre el cual se trabajaba con un cliente dedicado llamado Tortoise. También se trabajó con el Rational Requisite Pro como herramienta de gestión de requisitos, pero no se potenció su uso por parte de la dirección del proyecto y por tanto solo se usó por pocas personas, utilizando en su lugar el Microsoft Excel para el mismo propósito. Para una mayor información consultar la sección de herramientas de la memoria colectiva.

RESULTADOS DE LA APLICACIÓN EN EL PROYECTO CICPC

La implementación de lo presentado en esta memoria aportó un beneficio palpable en la eficiencia del proceso de desarrollo, y consiguiente éxito de la entrega del sistema SIIPOL. Tanto el cliente como los miembros del equipo de desarrollo del proyecto CICPC, reconocen la ventaja de la aplicación de estas actividades de gestión de requisitos, para trabajar de una forma organizada, eficiente, y consecuente con los acuerdos que se tomaron al comienzo del proyecto.

En el periodo comprendido entre mayo del 2006 y diciembre del 2008 el proyecto no fue auditado por la dirección de calidad, pero como constancia de lo anteriormente expuesto se cuenta con los informes de los diagnósticos aplicados a todos los proyectos de la Universidad (entre ellos CICPC), con informes de dos revisiones realizadas en el año 2008, y con encuestas realizadas al cliente y a los miembros del equipo de desarrollo en varias ocasiones, con el fin de retroalimentar a los dos analistas principales del proyecto, sobre las mejoras que se estaban introduciendo en las áreas de la IR. Para abundar en el resultado obtenido consultar la memoria colectiva.

Entre los resultados más contundentes, está el criterio generalizado (más del 80%) del equipo referente a que los factores que más incidieron en el éxito del proyecto son precisamente: la claridad y especificación detallada que existía de los requisitos, el control sistemático que se hacía del progreso del trabajo a partir de los CU, y la definición clara de las responsabilidades de las personas dentro del proyecto (analista principal, dueños de CU, equipos de desarrollo por módulos, etc.)

También se demostró mediante los diagnósticos y las encuestas aplicadas que más del 85% del proyecto dominaba, o al menos sabía dónde encontrar: la documentación asociada a los requisitos, los CU que había que desarrollar o modificar en la versión en curso, el procedimiento establecido para responder a las no conformidades o peticiones de cambio del cliente, la planificación asociada a los procesos de la iteración en curso, a que objetivo del negocio respondía cada CU, y estaban al tanto del estado de avance para declarar completa una versión según el estado de terminación de los CU.

Además en los anexos de la memoria colectiva pueden consultarse las Actas de Aceptación del cliente como constancia del entendimiento y aprobación de los contenidos de la documentación entregada.

CONCLUSIONES

La IR constituye una disciplina indispensable para desarrollar un proyecto de software con calidad y eficiencia.

La implementación de buenas prácticas de gestión de requisitos en proyectos de software de gestión policial, garantiza una base adecuada para lograr cumplir con las expectativas de los distintos involucrados y mejorar el trabajo del equipo de desarrollo.

La aplicación de buenas prácticas de gestión de requisitos, y la utilización de los mecanismos, procedimientos y herramientas que permiten la implantación exitosa de estas, basadas en las características particulares de este tipo de proyectos y de la organización que los contrata, constituye un aporte al desarrollo del software en la UCI.

Esta propuesta está en correspondencia con el programa de mejoras que lidera la Dirección de Calidad, sobre las áreas de administración de requisitos de software (REQM), según el modelo de referencia de CMMI.

La construcción y aceptación de un sistema informático de gestión policial (SIIPOL) cuyo proceso de desarrollo y gestión de requisitos se realizó siguiendo lo propuesto en este trabajo, avala la efectividad de lo propuesto.

RECOMENDACIONES

Se recomienda complementar el resultado de los logros obtenidos en el proyecto CICPC con la publicación de una propuesta técnica de referencia de los sistemas de software para la gestión policial, porque además de los aportes realizados en el proceso de desarrollo y gestión de los requisitos, también constituye un logro el haberse diseñado un sistema capaz de responder a las necesidades de una organización policial cuando solo se podía contar como referencia con la experiencia de los especialistas cubanos en el Sistema Automatizado Jurídico Operativo (SAJO).

Una propuesta técnica general de este tipo de sistema, aportaría ventajas a la hora de contratar proyectos similares con clientes de otros países, ya que seleccionarían datos y funcionalidades de su interés desde la etapa de estudio preliminar, haciendo más sencilla la posterior captura de requisitos; y permitiría además al equipo de desarrollo trabajar con componentes ya reutilizables y prediseñados, acortando el tiempo de implementación de la solución.

Finalmente se recomienda utilizar este trabajo para complementar el entrenamiento de los recursos humanos que asuman el rol de analistas en los proyectos de desarrollo de software para la gestión policial, e incluirlo en los objetivos instructivos a lograr en algún curso del perfil de seguridad ciudadana de la Facultad 2.

GLOSARIO DE TÉRMINOS Y ABREVIATURAS

Artefacto: se refiere cualquier elemento de información que se produzca como parte del proceso de desarrollo del software, por ejemplo: documentos, diagramas, software, especificaciones, etc.

Buena Práctica: Es una experiencia sistematizada y documentada que tiene como fundamento la aplicación de actividades, métodos, o elementos innovativos que agreguen ventajas a un proceso.

Caso de Uso: una técnica empleada en la elicitación de los requisitos del producto, donde se agrupan los fragmentos de funcionalidad que tengan valor para alguno de los roles que representan los usuarios finales.

Elicitación: esta palabra es un préstamo lingüístico del idioma inglés, y se utiliza para dar a entender el grado de complejidad del proceso de identificar los requisitos de un sistema, porque la traducción correcta al español es *obtención*, pero este término da la idea de que el usuario te brinda el listado de requisitos, cuando en la realidad esta actividad es mucho más difícil.

Línea base: una especificación o producto que se ha revisado/aprobado, y de ahí en adelante sirve como base para un desarrollo posterior, por tanto debe cambiarse solamente a través de procedimientos formales de control de cambios (IEEE) 610.12/1990.

Mecanismo: Manera de producirse una actividad, una función o un fenómeno.

Procedimiento: Forma específica en que se lleva a cabo una actividad o proceso.

BIBLIOGRAFÍA

- Aurum, A. and C. Wohlin (2005). Requirements Engineering: Setting the Context. Engineering and Managing Software Requirements. A. Aurum and C. Wohlin. Berlin Heidelberg, Springer-Verlag: 1-15.
- CMMI and P. Team (2006). Software Engineering Institute. Carnegie Mellon. CMMI® for Development, Version 1.2. Pittsburgh, PA 15213-3890, Carnegie Mellon University.
- CMMI_Consulting_Group (2009). Nivel de implementación e institucionalización del nivel 2 de CMMI en la UCI., SIE. Center Mexico. Documento de carácter confidencial. . Presentación de Resultados. Fase 1. Programa de Mejoras UCI.
- Dahlstedt, Å. G. and A. Persson (2005). Requirements Interdependencies: State of the Art and Future Challenges. Engineering and Managing Software Requirements. A. Aurum and C. Wohlin. Berlin Heidelberg, Springer-Verlag: 95-115.
- Delgado, R., K. Pérez, et al. (2008). Lineamientos de Calidad de Software. IPL3000: 2008. D. d. Calidad. Universidad de Ciencias Informáticas: 14.
- Durán A., B., B., Ruiz, A., Toro M. (1999). A Requirements Elicitation Approach Based in Templates and Patterns. Taller de Ingeniería de Requisitos. Buenos Aires. Argentina.
- Escalona, M. J., Mejías, M., Torres, J. (2002). "Methodologies to develop web information systems and comparative analysis." Informatique 2/2002 de I/I.
- Houdek, F. and K. Pohl (2000). Analyzing Requirements Engineering Processes: A Case Study. 11th International Workshop on Database and Expert Systems Applications (DEXA'00). Greenwich, London, U.K., IEEE computer society: 5.
- Hull, E., K. Jackson, et al. (2005). Requirements Engineering. London Berlin Heidelberg, Springer.
- IEEE (1998). Standard Glossary of Software Engineering Terminology -Description. 610.12. IEEE.
- IEEE (2004). SWEBOK. Guide to the Software Engineering Body of Knowledge. EEUU, IEEE Computer Society.
- IP-UCI (2006). Memoria no publicada del Taller de experiencias de la solución del producto. Ciudad de la Habana, Universidad de las Ciencias Informáticas: 6.
- Jacobson, I., G. Booch, et al. (2000). El Proceso Unificado de Desarrollo de Software, Pearson Education, S. A.
- Kontoya, G. and I. Sommerville (2000). Requirements Engineering: Process and Techniques, John Wiley & Sons.

- Leffingwell, D. and D. Widrig (2000). *Managing Software Requirements. A Unified Approach*, Addison Wesley
- Loucopoulos, P. and V. Karakostas (1995). *System requirements engineering*. Europe, McGraw-Hill Book Company
- Macaulay, L. A. (1996). *Requirements Engineering*, Springer - Verlag.
- Sawyer, P. and G. Kotonya (2004). *Software Requirements. SWEBOK. Guide to the Software Engineering Body of Knowledge*. EEUU, IEEE Computer Society.
- Sommerville, I. (2005). *Ingeniería del Software*. Madrid, Pearson. Addison Wesley
- Soto, N. and S. Valdés (2010). *Ingeniería de requisitos aplicada al sistema de investigación e información policial (SIIPOL)*. Memoria Colectiva. Facultad 8. Universidad de las Ciencias Informáticas. Ciudad de la Habana. Cuba.
- Spence, I. and L. Probasco (2000) *Traceability Strategies for Managing Requirements with Use Cases*. Rational Software White Paper Volume, 31.
- TEMÜR, N. (2009). *Towards a new model of Public Private Partnership for E-Government Project Implementation*. International Conference on e-Government. Antalya - TURKEY.
- Wieggers, K. E. (2003). *Software Requirements*. Redmond, Washington, Microsoft Press.
- Wieggers, K. E. (2006). *More About Software Requirements: Thorny Issues and Practical Advice*. Redmond, Washington Microsoft Press © 2006.
- Young, R. R. (2004). *The Requirements Engineering Handbook*, ARTECH HOUSE, INC.

ANEXO 1.

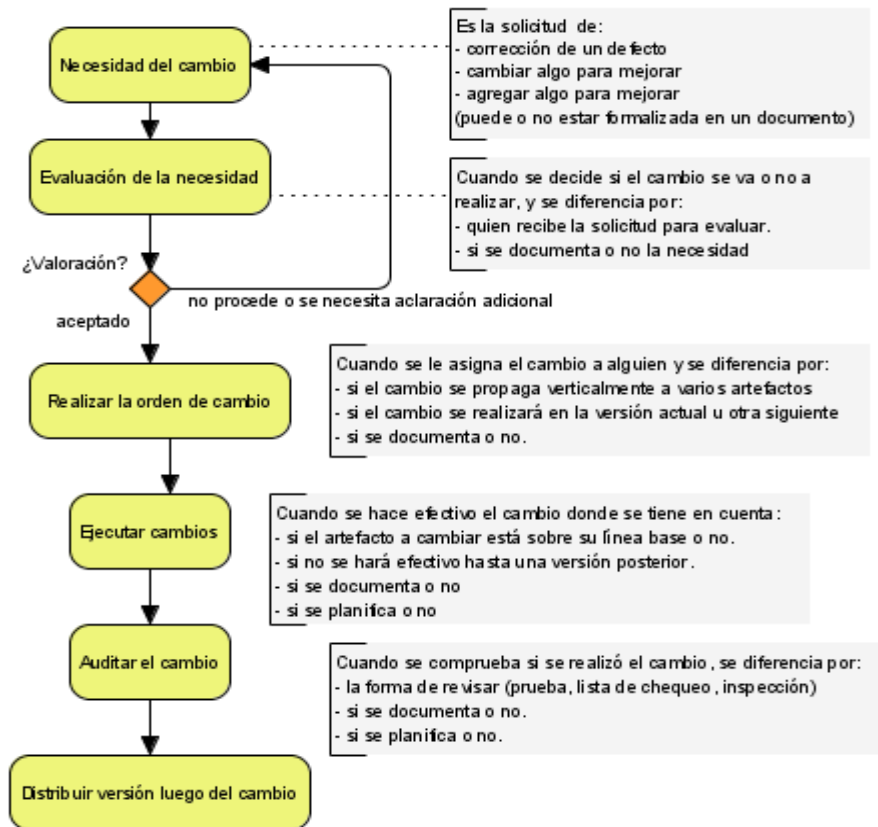


Figura 5. Proceso genérico de gestión de los cambios