

DIRECCIÓN DE FORMACIÓN POSTGRADUADA

Memoria individual

**DESARROLLO DE REQUISITOS
PARA PROYECTOS DEL DOMINIO
DE APLICACIÓN DE LA GESTIÓN
POLICIAL**

ESTUDIO APLICADO EN EL PROYECTO CICPC

Memoria presentada en opción al título de Máster
en Informática Aplicada

Autor: Ing. Sasha Valdés Jiménez

Tutor: MSc. Michael González Jorrín

Ciudad de La Habana, Junio 2010

“Año 52 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que soy el autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente con este trabajo.

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

AUTOR

Ing. Sasha Valdés Jiménez

TUTOR

MSc. Michael González Jorrín

RESUMEN

El Desarrollo de los Requisitos, como parte de la Ingeniería de Requisitos, dedica sus esfuerzos al estudio de la elicitación, análisis, especificación, verificación y validación de requisitos del software. Su aplicación en los procesos de producción ofrece ventajas sustanciales al entendimiento de la funcionalidad y caracterización de una aplicación informática, garantizando construcciones sólidas frente a correctas interpretaciones de las necesidades de usuarios y clientes, ejerciendo como un proceso de comunicación en las áreas de ingeniería de software entre los involucrados y el equipo de desarrollo.

El presente trabajo fundamenta su aporte en el establecimiento de un modelo para el Desarrollo de Requisitos de manera que disminuyan las incongruencias de los grandes proyectos de gestión entre lo que se desea, lo que se necesita y lo que al final se produce. El modelo se construye a partir de un proyecto productivo dirigido al área de la gestión policial. Incorpora elementos de estándares, normas y modelos de calidad unificando la Arquitectura de Información como parte del proceso de desarrollo de software. Define a su vez las competencias por roles y selecciona las mejores técnicas y herramientas para llevar a cabo el proceso.

ÍNDICE DE CONTENIDOS

| | |
|---|----|
| INTRODUCCIÓN | 1 |
| CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA | 3 |
| REQUISITOS EN LA INDUSTRIA DE SOFTWARE..... | 3 |
| INGENIERÍA DE REQUISITOS..... | 5 |
| ESTÁNDARES, NORMAS Y MODELOS APLICABLES AL DR | 6 |
| IR EN EL SOFTWARE DE GESTIÓN POLICIAL | 9 |
| CAPÍTULO 2. MODELO DE PROCESOS PARA EL DESARROLLO DE REQUISITOS | 12 |
| DEFINICIÓN DEL MODELO | 12 |
| DESARROLLO DE REQUISITOS DEL CLIENTE | 14 |
| DESARROLLO DE REQUISITOS DEL PRODUCTO | 19 |
| ROLES | 22 |
| TÉCNICAS | 23 |
| HERRAMIENTAS..... | 26 |
| RESULTADOS DE LA APLICACIÓN EN EL PROYECTO CICPC..... | 27 |
| CONCLUSIONES GENERALES | 29 |
| RECOMENDACIONES..... | 30 |
| REFERENCIAS BIBLIOGRÁFICAS | 31 |
| BIBLIOGRAFÍA CONSULTADA | 34 |

INTRODUCCIÓN

*“El nuestro es un mundo donde la gente no sabe lo que quiere
y está deseando atravesar el infierno para conseguirlo.”*

(Don Marquis)

En el marco de la colaboración entre los países de la Alianza Bolivariana para los pueblos de Nuestra América (ALBA) en el 2006, entre la República Bolivariana de Venezuela y Cuba se firma el Proyecto de Modernización del Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC). Uno de sus componentes es el desarrollo de una aplicación informática capaz de sustituir al antiguo Sistema Integrado de Información Policial por un nuevo Sistema de Investigación e Información Policial (SIIPOL) construido sobre modernas plataformas tecnológicas. Con la construcción del nuevo software, se esperaba mejorar el nivel de respuesta a las necesidades de seguridad del ciudadano venezolano, siendo parte de sus objetivos el control y organización del trabajo correspondiente a la gestión de la investigación penal, criminalística y forense, facilitando el análisis de información en cada una de las dependencias de la organización.

El proyecto de software se enmarca en un complejo escenario comprometido con objetivos políticos de un fuerte impacto social, caracterizado por la construcción de una aplicación informática ajustada a las necesidades del cliente, fuertemente regulada por la legislación vigente. Las definiciones técnicas, financieras y funcionales se ven condicionadas por un alto nivel de incertidumbre producto del gran tamaño del negocio a informatizar, la variedad de procedimientos institucionalizados, la redistribución de funciones en los puestos de trabajo por insuficiencia del personal y el nivel de adquisición en recursos tecnológicos y de despliegue.

La diversidad de criterios existentes sobre la colaboración cubana en áreas claves del gobierno, asociada al cambio de paradigmas por el proceso revolucionario existente en Venezuela e influenciada por la natural resistencia al cambio sobre la aplicación informática actual, afecta negativamente la obtención y validación de la información. Influyen además, la lentitud de la información, la ausencia de una cultura infotecnológica adecuada de sus trabajadores y la indefinición de la negociación con una gran cantidad de organizaciones externas para el intercambio de datos.

Es la Universidad de las Ciencias Informáticas (UCI) la entidad responsable del proyecto de software, aportando al escenario elementos de riesgo enfocados a los tiempos dedicados contra los realmente necesarios para la producción, los niveles de conocimiento y la poca experiencia tanto de profesores como de estudiantes en roles de especialistas frente al desarrollo de un software con el tamaño y

complejidad del propuesto. Preocupa además, la necesidad de la comunicación técnica, de negociación oportuna y actualización de la información frente a una distancia geográfica considerable, siendo imposible para su mitigación el financiamiento de la estancia de todos los miembros suficientes en cada etapa del proyecto por el presupuesto previamente fijado. Los proyectos de la comunidad universitaria varían en las nociones acerca del entorno, la metodología y el alcance, creando confusión y diversidad de criterios entre los líderes acerca de la estrategia más adecuada para software de gestión policial (IP-UCI, 2006). En la aplicación de las estrategias desarrolladas en la UCI, han influido los conocimientos empíricos de desarrollo de software, el estudio y la interpretación personal de los procesos medulares de la metodología aplicada, el conocimiento compartido de otros proyectos y la investigación de las mejores técnicas, patrones y prácticas sobre el tema sin haberse definido un proceso común de desarrollo que garantice la mitigación total de los riesgos planteados sobre el cliente y el equipo de desarrollo. (IP-UCI, 2006).

Para este tipo de software, las condiciones reflejadas afirman los criterios de uso de metodologías tradicionales¹, justificados por un extendido plazo de tiempo para el desarrollo, la objetividad sobre robustas construcciones arquitectónicas, la participación de un alto volumen de recursos tanto humanos como materiales, la imposibilidad de incorporación constante de los expertos del negocio como parte del equipo de desarrollo y en consecuencia, la estructuración de la información en esquemas documentales precisos (Sommerville, 2005). Se selecciona frente a estas condiciones el Proceso Unificado de Rational (RUP, por sus siglas en inglés) sobre las bases de dos modelos generales de desarrollo de software: el modelo en espiral y el incremental (Pressman, 2006). Se obtiene así una ejecución organizada e iterativa de las tareas sobre pequeños incrementos, permitiendo una gestión temprana de los riesgos, el control esperado sobre los recursos, el logro escalonado de hitos funcionales enfocados a la arquitectura y dirigidos por los requisitos de software en su expresión de casos de uso (Sommerville, 2005).

A partir del escenario antes expuesto y unido a la selección de la metodología, se identifica como una agravante para el éxito del proyecto las incongruencias entre lo que se desea, lo que se necesita producir y lo que se produce a la vista de todos los involucrados, encontrando una posible solución en los fundamentos de la Ingeniería de Requisitos (IR).

Los requisitos son la base del desarrollo de un producto, a partir de su correcta especificación se edifican las posteriores etapas dentro del ciclo de vida de un producto. El cambio inherente a los mismos determina un impacto en el cumplimiento de variables como costo, tiempo y alcance de un proyecto. El desarrollo ingenieril de los requisitos, entendido como la elicitación, el análisis y negociación, la

¹ Las metodologías tradicionales son también conocidas como pesadas, en contraposición a las denominadas ágiles.

documentación y la validación (Sawyer & Kotonya, 2001), unido a la gestión de los requisitos, forman parte de la IR. Esta es reconocida como una disciplina de la Ingeniería de Software concerniente al desarrollo, comunicación y administración de los requisitos de software (Mead, 2003).

El DR permite una correcta interpretación de las necesidades del cliente, propicia el análisis de alternativas y crea un espacio de negociación para el compromiso de la realización de los requisitos. Además de esto, garantiza la documentación de lo acordado sirviendo como base para la validación y común acuerdo, ya sea para el cliente como para el equipo de desarrollo (Pressman, 2006).

En el marco de esta investigación, se identifica como **problema** entonces: Los procedimientos utilizados en la UCI en proyectos del dominio de aplicación de la gestión policial no satisfacen una buena interpretación y modelación de los requisitos entre los involucrados del negocio y el equipo de desarrollo.

El **objeto de estudio** es el proceso de Ingeniería de Requisitos.

El **objetivo general** del presente trabajo es definir un modelo para el Desarrollo de Requisitos en proyectos del dominio de aplicación de la gestión policial. El **campo de acción** se enfoca a los procesos de Desarrollo de Requisitos como parte de la Ingeniería de Requisitos para software de gestión policial.

La **hipótesis** de la investigación sostiene que definiendo un modelo para el Desarrollo de Requisitos en proyectos del dominio de aplicación de la gestión policial, se garantiza una buena interpretación y modelación de los requisitos entre los involucrados del negocio y el equipo de desarrollo.

Los **métodos de investigación científica** utilizados fueron el método hipotético-deductivo para la definición de la hipótesis de la investigación, el método sistémico para lograr una integración consensuada de todos los elementos del modelo, el método histórico-lógico y el dialéctico para el análisis crítico y comparativo tanto de la bibliografía como de los resultados alcanzados, el método analítico-sintético al descomponer el problema de investigación en los elementos claves de solución, analizar cada caso en particular y sintetizar como una solución integrada, el método de la entrevista para obtener la información relevante para la investigación y los métodos estadísticos para el análisis de los datos recogidos en sesiones de trabajo, entrevistas y cuestionarios.

La estructuración de los contenidos de la memoria se inicia en el Capítulo 1 con la presentación de la información referente al estado del arte. En el Capítulo 2 se describe el aporte fundamental de la investigación sobre el modelo propuesto. En las Conclusiones, se discute los principales resultados y algunas recomendaciones sobre el proceso definido.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

*"La parte más difícil de construir de un sistema software es decidir qué construir. [...] Ninguna otra parte del trabajo afecta más negativamente al sistema final si se realiza de manera incorrecta. Ninguna otra parte es más difícil de rectificar después."
(Brooks, 1995)*

Requisitos en la industria de software

El término requisito en detrimento del término requerimiento es adoptado por el autor a partir de la (NC-ISO9000, 2001) y se define como una capacidad, característica o factor de calidad de un sistema mediante el cual se pretende cumplir con determinadas necesidades o restricciones operativas, aportando un valor y una utilidad para un cliente o usuario dentro del marco de solución de un problema en un entorno real (Young, 2004) (Sawyer & Kotonya, 2001).

Gran parte del éxito de un proyecto en la industria de software se debe al buen uso y gestión de los requisitos. El StandishGroup, en un estudio realizado en 2009 sobre proyectos de software gubernamentales y económicos, muestra un 24% con fallo por cancelación o uso nulo, y solamente un 32% con terminación exitosa. El 67% de estos proyectos poseen deficiencias funcionales. Entre los factores de fracaso medidos en el Chaos Report se identifican los requisitos incompletos, poca vinculación del usuario, expectativas poco realistas y cambio continuo en los requisitos y sus especificaciones (TEMÜR, 2009).

IAG Consulting revela, en un estudio realizado en 2008 a 100 compañías norteamericanas que han desarrollado amplios y estratégicos proyectos para la adición de nuevas funcionalidades en la organización, que las empresas con requisitos mal especificados gastan por proyecto como promedio \$2.24 millones más que lo planificado, entregando menos del 70% de las funcionalidades previstas. Asimismo declara que organizaciones con pobres estudios de requisitos y débil análisis de negocio tienen tres proyectos fallidos por cada uno exitoso, solamente el 32% de las compañías emplean prácticas que hacen probable el éxito del proyecto. Sobre el 40% de los presupuestos para software, equipo y servicios profesionales externos son consumidos en la producción de requisitos pobres por el uso de analistas promedios. Estos resultados son evitados por organizaciones que constantemente utilizan buenas prácticas en la elicitación de requisitos y el análisis de negocio desde el momento en que conciben el proyecto (IAG, 2009).

En el informe de evaluación de la producción realizado por Calisoft en la UCI en el año 2008, una de las áreas identificadas de mayor interés, atención y riesgo fue la concerniente a la IR. Dicha encuesta se le realizó a la mayoría de los proyectos productivos en todas las áreas de la universidad, identificando como factores de peso en el desarrollo de software la definición de los requisitos y la participación de los usuarios (Calidad-UCI, 2008).

Sobre esta base, el autor considera que para el cumplimiento de los objetivos propuestos en un proyecto de software sobre los costos y tiempos establecidos es determinante: la identificación correcta y completa de los requisitos dentro del alcance de la solución, la vinculación activa del cliente y el usuario, la mitigación del cambio y la inclusión de buenas prácticas en el desarrollo y la gestión de requisitos.

Los requisitos tienen a ser clasificados de varias maneras. La presente investigación adopta la clasificación establecida en FURPS+ según los atributos de calidad del software de Funcionalidad (Functionality), Usabilidad (Usability), Confiabilidad (Reliability), Rendimiento (Performance) y Soporte (Supportability); el + representa clasificaciones adicionales no contempladas en las anteriores (Grady, 1992) (ISO/IEC-25000, 2005)². Se adopta además la establecida en (CMMI, 2002) para la clasificación por audiencia en requisitos del cliente, también vistos como del usuario o del negocio, y requisitos del producto, también vistos como del sistema.

Todo requisito debe ser analizado como una descripción de la funcionalidad o característica que representa, cumpliendo con un conjunto de propiedades que permitan su mejor interpretación y convivencia dentro del conjunto de requisitos del software (Young, 2004). Las propiedades adoptadas por el autor a partir de la bibliografía consultada en (IEEE830, 1998)(Young, 2004) para una buena especificación de requisitos se listan en la Tabla 1.

| | | | |
|---------------|---|-------------|---|
| Correcta | expresa una funcionalidad que el sistema realmente deba cumplir | No ambigua | expresa una única interpretación |
| No redundante | no se repite su funcionalidad en otro requisito | Factible | es posible su implementación bajo el tiempo y el costo prrefijado |
| Concisa | está declarado de manera simple | Consistente | no debe entrar en conflicto con otra especificación de requisito |
| Completa | describe en el sistema lo que deba hacer o tener la funcionalidad o | Estable | mide el nivel de cambios que puede sufrir el requisito basado |

² La norma que inicialmente clasificó los requisitos en atributos de calidad fue la ISO/IEC 9126, sin embargo esta fue incorporada por la 25000 junto a otras contribuyentes al sistema de calidad.

| | | | |
|-------------|--|--------------------------|--|
| | característica que representa | | en la experiencia de eventos venideros en la organización, funciones y personas que manipulen el sistema software |
| Necesaria | todos los requisitos no tienen la misma prioridad en el desarrollo de software, unos son críticos para el ciclo de vida de la aplicación, otros son deseables y los últimos opcionales | Verificable | puede medir que el sistema cumple la funcionalidad o característica a través de un proceso de costo efectivo realizado por una máquina o persona |
| Modificable | cualquier cambio sobre el requisito puede ser realizado de manera completa, sencilla y consistente, manteniendo su estructura y estilo | Traceable | el origen del requisito es claro y facilita la referencia a cada requisito en el futuro desarrollo o la documentación relacionada |
| Localizable | posible de localizar en un componente diseñado | Independiente del diseño | no plantea una solución de aplicación específica a una tecnología |

Tabla 1. Propiedades de los requisitos.

Ingeniería de requisitos

La IR es descrita por (STEP, 1991) (Dorfman, y otros, 1990) como la aplicación disciplinada de principios científicos y técnicos para desarrollar, comunicar y administrar requisitos. En (Loucopoulos, 1989) se define como el proceso sistemático de desarrollar requisitos a través de un proceso iterativo de análisis de problema, documentación de los resultados observados y la validación de los resultados. Leite ofrece por su parte dos puntos de vista: usuario y equipo de desarrollo, y usa una combinación de métodos, herramientas y actores (Leite, 1987). En (Wiegers, 2006) (Mead, 2003) se ofrece una definición bastante completa, donde introduce el dominio de la gestión de requisitos complementando las actividades de desarrollo ingenieril, apoyado por (Sommerville, 2005) que además de caracterizarlo como proceso de comunicación entre los involucrados le adiciona una actividad de estudio preliminar donde se garantiza la factibilidad del proyecto. Novedoso es el aporte de este último hacia un modelo en espiral de la IR, con iteraciones de las actividades ingenieriles durante el ciclo de vida del requisito. En (CMMI, 2002) se identifican claramente las áreas de procesos correspondientes a la gestión y al desarrollo de requisitos,

presentando además conocimientos de interés en áreas de procesos correspondientes a la verificación y la validación.

Para los fines de la presente investigación y tomando como base las definiciones antes expuestas se propone la IR como (1) la disciplina de la Ingeniería de Software concerniente a la administración, desarrollo y comunicación de los requisitos entre los involucrados y el equipo de desarrollo, (2) caracterizada tanto por el dominio de la Gestión de Requisitos hacia el estudio preliminar, la planificación, el seguimiento, el control de los cambios y las inconsistencias, (3) como por el dominio del Desarrollo de Requisitos (DR) responsable de la elicitación, análisis y negociación, documentación, verificación y validación, (4) sobre un modelo iterativo e incremental y (4) un enfoque a la calidad.

Estándares, normas y modelos aplicables al DR

Para el establecimiento del objetivo de esta investigación se hace necesario el estudio de modelos aplicables al DR. Los modelos de desarrollo estudiados sobre el entorno empresarial nacional demuestran la preferencia por métodos ágiles centrados en los evolutivos incremental y en espiral. En esta posición se encuentran como puntales unidades del Ministerio del Interior, HAVASOFT del Ministerio del Azúcar, DESOFT-Sancti Spíritus (Pardillo Castañeda, y otros, 2007) y TECNOMÁTICA del Ministerio de la Industria Básica (TECNOMÁTICA, 2009). En estos modelos la IR no es vista como un proceso y su perspectiva ágil no centra su atención en la definición del DR como un área de especial interés en el levantamiento y modelación de la información.

Por otra parte el uso de metodologías tradicionales en empresas como DESOFT, perteneciente al Ministerio de la Informática y las Comunicaciones, que abre su abanico a soluciones complejas dirigidas a varios sectores de la sociedad, se ve orientada hacia una variante ajustada de RUP definida en la casa matriz y aplicada en las diferentes sucursales del país. En igual situación la Unidad de Compatibilización e Integración de Software para la Defensa (UCID) perteneciente al Ministerio de las Fuerzas Armadas, la UCI y sus centros regionales (Calidad-UCI, 2008) apuestan por el Proceso Unificado y su enfoque a estándares de calidad. Aunque el modelo de producción de estas entidades se acerca a las necesidades de la investigación, prevalecen las tendencias ágiles de las actividades correspondientes a la IR cuando los proyectos se ejecutan en el marco nacional, un ejemplo de esto es el modelo INeR propuesto en (Pérez, 2007) para el dominio de los proyectos bioinformáticos.

La experiencia acumulada hacia el 2006 en el desarrollo de megaproyectos con características similares al del SIIPOL en la UCI identificaban como riesgos comunes y sin una respuesta acertada: el levantamiento, la negociación, la aprobación de información en un lugar geográficamente alejado de donde se transforma en una aplicación informática, la experiencia productiva de los recursos humanos

bajo un sistema docente universitario, los presupuestos que imposibilitan una mayor vinculación del cliente y el equipo de desarrollo, el uso de nuevas tecnologías y las amenazas constantes de sobregiro de tiempo, costo y alcance (IP-UCI, 2006).

Hasta este punto, se refuerza la utilidad de la creación de un modelo para el DR en el dominio del software de gestión policial, que no encuentra soluciones completas en la experiencia nacional acumulada hasta el período que posibiliten mejorar la efectividad en la modelación e interpretación de requisitos entre proveedores de información y equipo de desarrollo.

En el marco teórico la presente investigación se identifica con el modelo evolutivo incremental y en espiral de (Sommerville, 2005), utilizando para su descripción el enfoque de la Ingeniería de Software según (Pressman, 2006) basado en procesos, métodos y herramientas, desarrollado por personas y orientado a la calidad.

En función de esto, para la guía de procesos y las fichas de descripción de los métodos se estudiaron las propuestas por (Rumbaugh, y otros, 2000) a través de disciplinas, flujos de trabajo, actividades y artefactos, considerándose más orientadas a buenas prácticas que a guías formales de ejecución. Por otra parte, el modelo MOPROSOFT, que basa sus definiciones en lo establecido por la ISO 9000:2001, niveles 2 y 3 de CMMI V.1.1, SWEBOK, el PMBOK y la norma ISO/IEC 15504 Tecnologías de la Información – Evaluación de Procesos, emplea conceptos, guías y un patrón documental más cercano a los intereses de especificación y clasificación (Oktaba, y otros, 2005). Para la modelación de los procesos, se descubre como novedosa y de amplio soporte en la industria la Notación y el Modelo de Procesos de Negocio (BPMN), más propia al conocimiento de procesos que los tradicionales diagramas en el Lenguaje Unificado del Modelado (UML), sustituyendo a los diagramas de flujo e información (OMG, 2007).

Basado en los criterios anteriores el autor elaboró una ficha para la especificación de los procesos, métodos y responsabilidades de las personas. Se puede analizar en la Memoria Colectiva (Soto & Valdés, 2010).

Para el enfoque a la calidad del DR, se investigaron los modelos, normas y estándares orientados al desarrollo de software de mayor reconocimiento en la industria de software aplicables al dominio de gestión.

De los estudiados, el Modelo Integrado de Madurez y Capacidad (CMMI, por sus siglas en inglés) ofrece preparación del capital humano, excelencia en la comunicación, calidad en la información, disminución de los tiempos de respuesta, abaratamiento de los costos y mejoras en la producción. La categoría de procesos ingenieriles que ofrecen una particular utilidad a la presente investigación son: Gestión de

requisitos, Desarrollo de requisitos, Verificación y Validación. Estos tienen como alcance las actividades de desarrollo y mantenimiento, compartidas a través de la ingeniería de software y de sistemas en un escenario de mejora de proceso orientada a productos (CMMI, 2002).

La Organización Internacional para la Estandarización (ISO, por sus siglas en inglés), mantiene la familia de normas ISO 9000, que establece un sistema de gestión de calidad para los procesos de una organización (NC-ISO90003, 2006). De la familia, dos normas se identifican como útiles en la investigación actual:

- ISO 9000: Principios y conceptos, lineamientos para su selección y utilización.
- ISO 9001: Modelo de aseguramiento de la calidad, aplicable al diseño, desarrollo, fabricación, instalación y servicio.

En nuestro país y como parte de la colaboración de distintas organizaciones, entre ellas la UCI, se establece la *NC-ISO 90003:2006 Ingeniería de software - directivas para la aplicación de la NC-ISO 9001:2001 al software de computación* en la que la presente investigación encuentra definiciones y métodos prácticos en la aplicación del DR.

El Instituto de Ingenieros Electrónicos y Eléctricos (IEEE, por sus siglas en inglés) posee una serie de estándares de interés para la IR:

- *IEEE Std 830-1998 Prácticas recomendadas para especificaciones de requisitos de software*, establece las guías para una buena especificación de requisitos, abordando su naturaleza, entorno, características, evolución, prototipado, integración en el diseño y los requisitos del proyecto (IEEE830, 1998). Propone además una estructura para el documento teniendo en cuenta los factores antes mencionados y las políticas de calidad para el control de cambios y registros históricos del documento.
- *IEEE Std 1074-1997 Estándar para el desarrollo de los procesos del ciclo de vida del software*, describe en su anexo 3.1 el grupo de actividades correspondientes al desarrollo de los requisitos del producto (IEEE1074, 1997). Las clasifica en (1) definir y desarrollar requisitos de software, (2) definir las interfaces de los requisitos y (3) su priorización e integración.
- *IEEE Std 610.12-1990 Glosario Estándar para la Terminología de Ingeniería de Software*, ofrece una ayuda para el común nombramiento en el lenguaje técnico (IEEE610, 1991).

- *IEEE Std 1012-1998 Estándar para la verificación y validación de software*, ofrece sus consideraciones acerca de la garantía de calidad del producto en revisiones hacia lo interno en el cumplimiento de los requisitos del sistema y hacia lo externo en la satisfacción de las necesidades del cliente (IEEE1012, 1998).

El autor encuentra en las tres propuestas anteriores soluciones interesantes para la clasificación, identificación, modelación y respaldo de buenas prácticas de las actividades a realizar para un buen DR, en total correspondencia con los trabajos sugeridos por (Mead, 2003) (Young, 2004) (Wiegers, 2006) (Sawyer & Kotonya, 2001) (Sommerville, 2005). Se sostiene que la estandarización durante la instanciación en un proceso real garantiza un enfoque de experiencia en la Calidad y la Ingeniería de Software.

IR en el software de gestión policial

El software de gestión policial, bajo apreciación del autor, debe ser considerado una herramienta elemental en la eficiencia de las gestiones de cualquier institución de carácter policial. Más aún con los índices actuales de hechos delictivos y criminales originados por las diferencias sociales, culturales, económicas, desviaciones psicológicas y consecuencias políticas, por citar algunos. La inversión científico técnica de los gobiernos en el logro de una gestión más eficiente en el trabajo, tiene su remuneración en la disminución del impacto del delito, el aumento del control, la previsión y el sano ejercicio de la justicia. Este tipo de software tiene sus aplicaciones fundamentales en las áreas de:

- Gestión de procesos de investigación, jurídicos, penales, legales y gubernamentales.
- Identificación biométrica para la identificación o verificación de la identidad de una persona.
- Mantenimiento y control de la información relativa al comportamiento penal, criminal, social de un individuo.
- Procesamiento, cruce y análisis de datos para la obtención de información veraz y oportuna en una investigación, el control estadístico y la toma de decisiones jurídicas o estratégicas.
- Posicionamiento Global por satélite para la protección ciudadana o de objetivos estratégicos.

Las características que distinguen a un software policial del resto de programas informáticos se plantean a partir de los criterios identificados por los asesores del MININT en el estudio preliminar del negocio. Ambos definen un conjunto de requisitos, para los cuales se reseña el análisis correspondiente al proceso en descripción.

- La seguridad de acceso: Aspecto que obliga a levantar cuidadosamente qué roles tienen acceso a qué información. Este proceso puede resultar complejo si se considera el volumen de entidades identificadas en un software de gestión policial³.
- La confiabilidad de la información: Aspecto referido tanto a la información que se recibe en el levantamiento como en la que brinda el software. Para el primer caso obliga a reiterados procesos de revisión, contraste y aprobación de la información suministrada por niveles jerárquicos diferentes en la organización. En el segundo se logra con la precisión de cálculo y el seguimiento de la información procesada por los distintos roles.
- La rapidez de la gestión: Se involucran aspectos tecnológicos, de Arquitectura de Información, de reingeniería de procesos, de ingeniería de patrones y optimización computacional.
- El almacenamiento de información, los tiempos de acceso, el rendimiento y la capacidad de procesamiento: Se consideran aspectos sobre la selección de la tecnología adecuada para Sistemas Gestores en Base de Datos, la conectividad, el análisis y la transmisión de datos, entre otras.
- El control sobre la información emitida: Se considera la clasificación de la información obtenida y el rastro de sus proveedores, clientes directos y terceros involucrados.
- La complejidad arquitectónica y funcional: Los sistemas de este tipo tienen altas restricciones relacionadas el aseguramiento del acceso, la confidencialidad, fiabilidad de la información y disponibilidad. Estas restricciones complementan las definiciones arquitectónicas en un grado tanto o mayor que los requisitos funcionales arquitectónicamente significativos.
- La cantidad y concurrencia de usuarios: Es requisito indispensable la garantía del acceso y el tráfico ordenado de la información sin tendencia a bloqueos ni fallos frente al uso concurrente de un número significativo de usuarios.
- La integración con fuentes de datos externas a la organización: Debe tenerse en cuenta la interoperabilidad a través de diferentes sistemas informáticos así como la seguridad y rapidez del canal de comunicación.
- El manejo de tecnología hardware asociada de propósito general o específico para entrada/salida de datos complejos: La vinculación de hardware para el reconocimiento de patrones, movilidad,

³ Para el sistema actual 168 tablas con 92 nomencladores y una media de 15 atributos.

identificación de individuos, comunicaciones, entre otras funciones, agilizan tareas de recopilación o transmisión de altos volúmenes de datos, garantizando fiabilidad y seguridad en la transacción.

La IR si bien no es responsable de la ejecución directa de muchas de estas características del software de gestión policial, posee los medios para asegurar su inclusión y garantizar su estricto cumplimiento. En todo levantamiento de información, análisis de solución, conversaciones de negociación ya sea orientado a reducir complejidad de una propuesta, fijar el marco arquitectónico de una posible reutilización, definir el entorno de negocio, fijar las variables del sistema o sus atributos de calidad deben tenerse en cuenta los requisitos funcionales, no funcionales de dominio o técnicos que cubran los aspectos anteriores.

CAPÍTULO 2. MODELO DE PROCESOS PARA EL DESARROLLO DE REQUISITOS

“Buenas personas con un buen proceso siempre actuarán mejor que buenas personas sin procesos.”
(Grady Booch)

Definición del modelo

En la vista en espiral de la IR modelada por el autor en la Figura 1 sobre las propuestas realizadas por (Sommerville, 2005), se describe el ciclo de vida de un requisito. El diagrama divide las cinco etapas del DR en un recorrido cíclico iterativo a partir del orden que produce incrementos de éxito en el establecimiento de los requisitos. Las divisiones de color indican el inicio del proceso desde el estudio de las necesidades del cliente, el usuario y el negocio hasta el logro esperado de la caracterización del producto donde el esfuerzo fundamental se centra en descubrir y entender el entorno en que se construye. El paso por cada etapa de la IR otorga un estado superior a la especificación de requisitos del cliente, siendo el deseado la clasificación de correctos, completos y formalmente descritos. El DR se identifica a través de una línea en espiral que recorre todas las etapas, mientras que la GR se modela como una actividad paralela al desarrollo durante toda su duración.

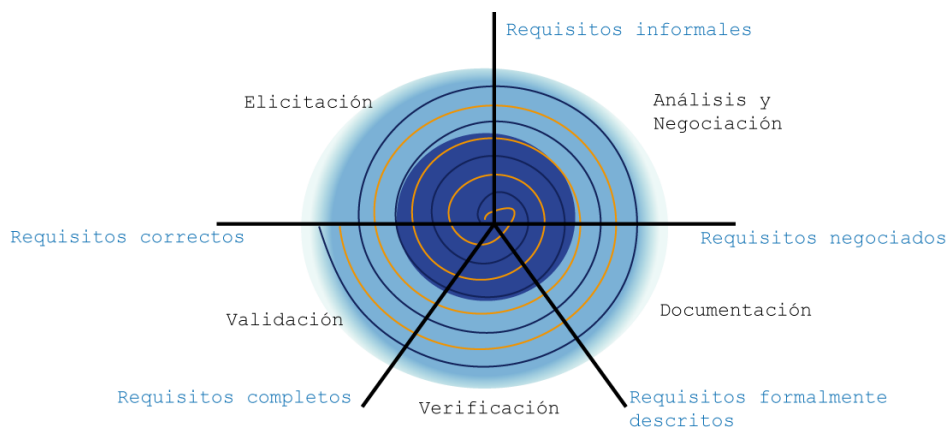


Figura. 1 Vista en espiral de la IR.

El modelo de procesos propuesto desglosa para el DR las etapas del ciclo de vida en subprocesos, los describe y los agrupa en dos procesos operativos orientados al desarrollo de requisitos del cliente y del producto. Independientemente se establece la GR como proceso ingenieril de mantenimiento de software,

estrechamente vinculado al DR (CMMI, 2002). La Figura 2 muestra el diagrama de procesos a través de la notación BPMN.

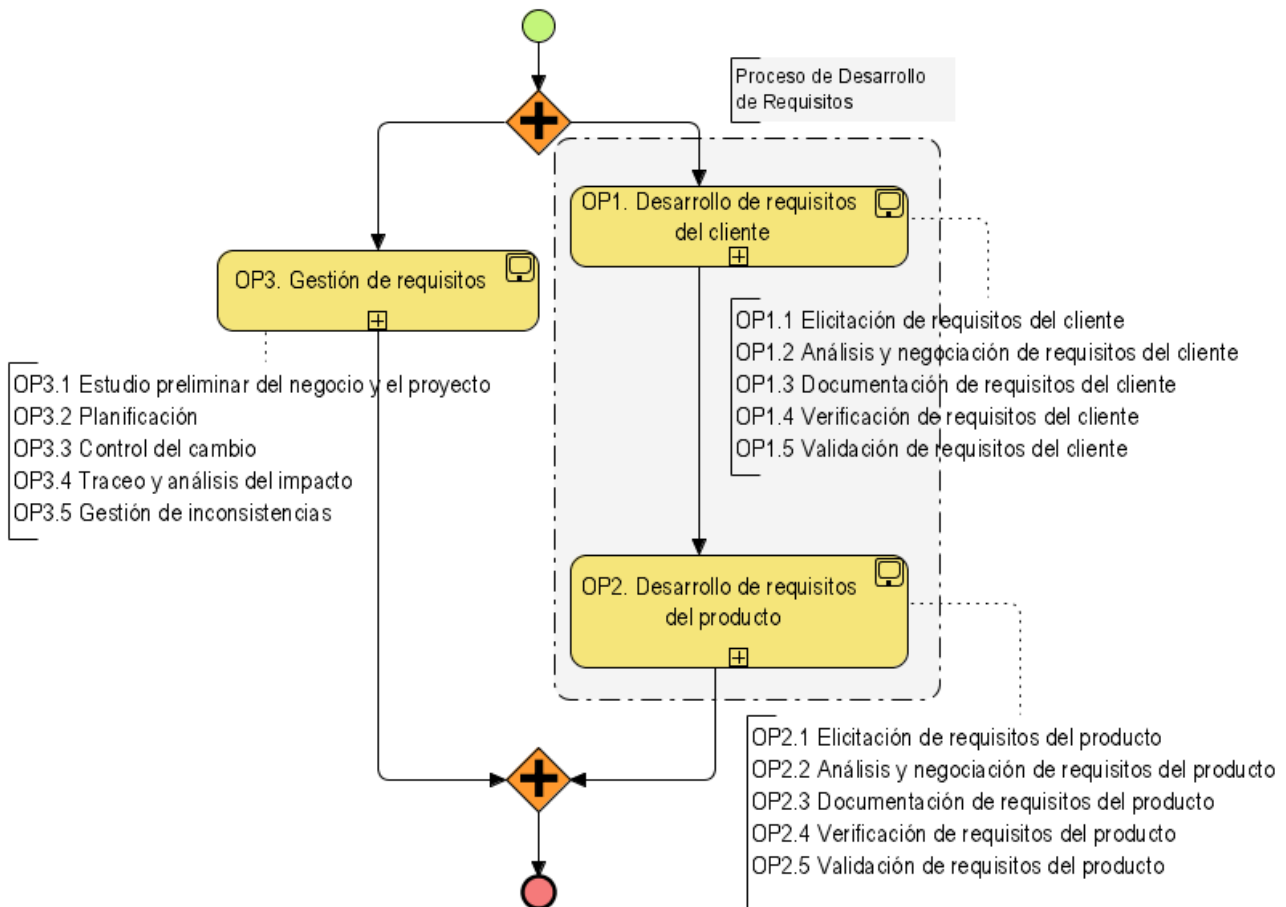


Figura. 2 Proceso de IR representando los dominios de la GR y del DR.

La descripción de cada proceso expuesta en la Memoria Colectiva (Soto & Valdés, 2010) cumple con las prácticas y objetivos específicos establecidos por (CMMI, 2002) en el nivel 3 de la representación por etapas, instanciando artefactos y buenas prácticas de RUP en un formato concreto de guía a partir de la ficha de procesos personalizada tomada de (Oktaba, y otros, 2005). Aplica además conceptos y lineamientos de los estándares y normas presentado en el capítulo anterior para la verificación y la validación, el establecimiento de un vocabulario estándar en la ingeniería y el control documental.

En similitud al RUP, se considera la aplicación del DR en un estudio del negocio y un estudio de requisitos del sistema, contemplados en las categorías de desarrollo de requisitos del cliente y el producto respectivamente. Se han nombrado de esta manera para entrar en consenso con la nomenclatura

consultada en (CMMI, 2002) (NC-ISO90003, 2006) (IEEE830, 1998) (Sommerville & Sawyer, 1997) (SWEBOK, 2004) (Young, 2004), de manera que sirvan para la ampliación del estudio a otras metodologías de software sin contradecir la establecida en el proyecto actual.

Desarrollo de requisitos del cliente

El propósito fundamental del proceso **Desarrollo de requisitos del cliente**, Figura 3, es el logro del común acuerdo entre los involucrados y equipo de desarrollo en el entendimiento de los requisitos que plantean los primeros, enfatizando en el funcionamiento del negocio dentro y fuera de la organización. A la par del consenso en la nomenclatura, se considera por el autor que aunque muchos investigadores referencien el término “requisitos del cliente” como requisitos del usuario, del dominio o del negocio, quien finalmente decide qué requisitos se deben implementar es el cliente, luego se le atribuye a este la categoría del estudio inicial sobre las necesidades y características del producto.

Este proceso comienza con la ejecución del **estudio preliminar del negocio y el proyecto**, que si bien en varias metodologías se describe como parte de la gestión del proyecto, Sommerville lo coloca como una actividad de la IR que decide la factibilidad del alcance del proyecto bajo los costos iniciales. Sus objetivos se centran en el diagnóstico de la disponibilidad del negocio para enfrentarse a la automatización, la identificación y aceleración del modelado de la información que circula en el negocio, el establecimiento de los términos y responsabilidades de las partes para el exitoso levantamiento de la información y el análisis crítico de la factibilidad de la realización del proyecto ante las condiciones iniciales de la negociación (Sommerville, 2005).

Es importante analizar con el cliente en este subproceso los resultados de investigaciones realizadas alrededor de las necesidades preliminares identificadas y la vía de solución tecnológica más apropiada según el equipo de desarrollo. En este momento de la concepción del software el cliente sabe lo que quiere pero no lo que necesita, por lo que la presentación temprana de ofertas en materia de tecnología, plataformas, lenguajes y soluciones deviene en ventajas para el equipo de desarrollo, tales como: la reutilización de software y sus componentes, el dominio de tecnologías, la certificación de conocimientos aplicados y la disminución de riesgos. Deben analizarse además detalles sobre los objetivos, riesgos, oportunidades y debilidades del negocio, teniendo como resultado final un ajuste del alcance, del tiempo, el costo y una delimitación de las estructuras físicas inmersas en el levantamiento de información de la organización.

El rechazo del proyecto por la evaluación del negocio no necesariamente significa la suspensión del contrato, como muchos interpretan de manera dogmática de los postulados de RUP, sino la reformulación

del problema y objetivos iniciales de manera que permitan un cambio factible en la visión del proyecto. Estas decisiones son tomadas por la Alta Gerencia, por lo que la información del proceso sirve para dar soporte a la necesidad de encontrar nuevas vías de solución.

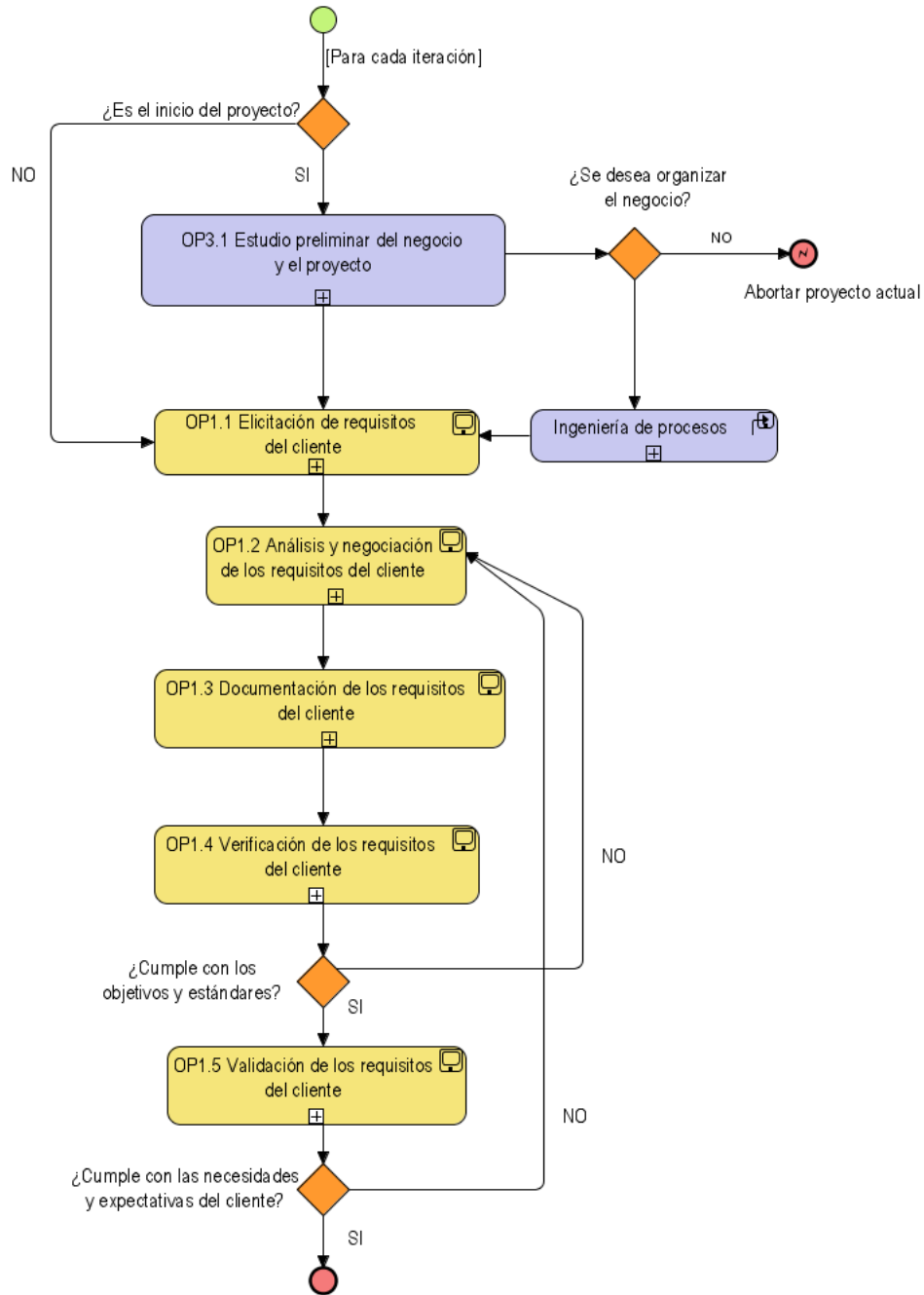


Figura. 3 Subproceso de Desarrollo de requisitos del cliente.

De la misma manera, se hace imprescindible la discusión de la preparación de la organización para una automatización a partir de los resultados del informe de Evaluación de las áreas del negocio, llevando a aprobación alguna reestructuración necesaria de los procesos. Esta discusión debe ser llevada al equipo de desarrollo, para la identificación y ajuste de la viabilidad de los compromisos iniciales.

La Ingeniería de procesos es un área de la gestión empresarial vinculada a la Ingeniería de Software por la vista completa que ofrece acerca del funcionamiento del negocio a informatizar, teniendo como responsabilidades la identificación, el establecimiento y la mejora de los procesos de la organización. En el proyecto no se contaba el tiempo para la realización cabal de las actividades de la Ingeniería de procesos, tanto así que la reingeniería desde el estudio del negocio no era aceptada a menos que tuviera un fundamento en las necesidades del software. Es por esto que el modelo propuesto obtiene de la Ingeniería de procesos la identificación, relación y descripción de procesos elementales de la organización, vincula al rol analista como auxiliar en las actividades del ingeniero de procesos del proyecto y facilita la transición de los conocimientos al sistema con la especificación puntual de procesos vista como casos de uso del negocio.

A consideración del autor y a partir de la experiencia adquirida en el presente trabajo, la especificación de un caso de uso del negocio es similar a una ficha de procesos, con menos datos de interés para la Ingeniería de procesos y una visión de relaciones características del RUP que imposibilita que ingenieros de procesos lo identifiquen como un artefacto completo para su modelación. Esto provoca que muchos proyectos repitan la información a la hora de adaptar los procesos a casos de uso del negocio, sin tomar en consideración que a partir de un proceso bien descrito se puede especificar directamente casos de uso del sistema.

La solución a este dilema se encuentra en la preparación de ingenieros informáticos con la visión de los temas de la ingeniería de procesos para la gestión. La utilidad de la formación de estos conocimientos en los analistas de sistema se materializó en etapas posteriores al inicio del proyecto; el ingeniero de procesos no fue necesario luego de la identificación inicial y el analista de sistema asimiló sus competencias directamente sobre el dominio de casos de uso del negocio. Se obtuvo con esto una entrada al desarrollo de requisitos del producto de procesos ordenados, definidos, completos, correctos y optimizados: agilizando el desarrollo, garantizando una base sólida e institucionalizada del cómo hacer las cosas, reduciendo ampliamente el margen del cambio.

La identificación de casos de uso de negocio y artefactos afines al modelo propuesto por RUP a partir de los procesos, fue responsabilidad de la **elicitación de requisitos del cliente**, desarrollado por analistas de sistema. La recopilación de información para el entendimiento del problema que el software debe

resolver y la resolución de conflictos detectados en la información obtenida permitió identificar necesidades, entender el negocio como si se formara parte de él, identificar la automatización y dominar el problema.

No sólo se hizo necesario levantar los procesos de negocio, sino realizar una investigación por las áreas en busca de soluciones alternas dadas por los trabajadores para agilizar su trabajo. Estas soluciones se materializaron en sistemas de trabajo personalizados, software de soluciones parciales, nuevos formularios, reorganizaciones de funciones por área ante su carencia de personal, entre otros. Necesario fue además la aprehensión de sistemas legados o existentes en la organización como actividad vital para el levantamiento de una aplicación que cumpliera todas las expectativas de un cliente; esto porque cualquier conjunto de requisitos de un nuevo sistema siempre será comparado con las funcionalidades del anterior.

El **análisis y negociación de los requisitos del cliente** tiene como propósito el entendimiento cabal de los requisitos y la modelación en base a mejora, permitiendo identificar usuarios potenciales del software y sus objetivos para mantener o mejorar el criterio de éxito en la respuesta del negocio. El análisis multiplica mediante el intercambio colaborativo el conocimiento del negocio en los desarrolladores para lograr una visión más clara de los requisitos que debe cumplir el software. Por su parte, la negociación establece un común acuerdo entre clientes, usuarios y equipo de desarrollo ante posibles inconsistencias.

El uso de representaciones gráficas acerca del funcionamiento, como diagramas de procesos o actividades, ayudan al usuario a entender los objetivos y las necesidades de información del especialista del proyecto ganando tiempo en la negociación. Sin embargo, muchas veces basta con la demostración del dominio del problema y el proceso en discusión para cerrar una negociación de manera exitosa. Sobre este punto, quedó demostrado que la creación de estos diagramas no puede ser una actividad rígida en el proceso, por el tiempo que cuesta su modelación didáctica frente a clientes que no lo necesiten y frente a un equipo de desarrollo que centrará su atención en los casos de uso del sistema.

Otro factor importante en este subproceso es el compromiso del equipo del proyecto con el desarrollo de los requisitos, entendido como el análisis técnico que avala la posibilidad de implementación de un requisito (CMMI, 2002). Si el equipo de proyecto no es capaz de realizar o entender un requisito debe ser negociada con el cliente una variación de su contenido. Un requisito que no es aprobado por el cliente ni por el equipo de desarrollo en una negociación al nivel de este subproceso, debe ser tratado como una inconsistencia a resolverse en una instancia superior. La determinación y aprobación del cambio de alcance y objetivos de un proyecto corresponden a la Alta Gerencia, aún cuando la causa sea descubierta en el análisis de requisitos directamente con el cliente.

En la priorización de requisitos, es fundamental el criterio arquitectónico, donde los primeros en construirse deben ser escogidos por las necesidades de construcción de las bases arquitectónicas de software y de procesos (IBM, 2006). Un segundo factor de éxito es el beneficio o necesidad que le reporta al cliente, bajo la perspectiva de que en un software de fabricación incremental, debe poner en funcionamiento inmediato aquellos requisitos que le permitan ganar en menor plazo su inversión.

En la **documentación de los requisitos del cliente** se define como propósito crear una base documental que sirva de estudio, reutilización y toma de decisiones en momentos posteriores a través de la estructuración de la documentación técnica y de soporte, los mecanismos de constancia formal de lo acordado, elaborado y aprobado por ambas partes. Para esto se aplicaron los patrones establecidos por las normas ISO y se personalizaron los modelos de obtención de la información para artefactos propuestos por el RUP (IBM, 2006).

El volumen de información que se produce en un proyecto bajo una metodología tradicional es amplio, cada elemento puede resultar en un factor determinante ante una negociación. Es por eso la necesidad de mantener en orden su nomenclatura, actualización de cambios, distribución y pautas documentales. La calidad visual y de contenidos de un documento ante un cliente influye en la imagen de la empresa que lo presenta, tanto o más como el dominio de la información levantada lo tranquiliza en el curso de las negociaciones.

Se recomienda considerar durante la entrega de documentos, cuando el volumen de información es elevado para ser impreso, una entrega en formato digital con sus correspondientes medidas de seguridad que garanticen la autenticidad e integridad de la información. De igual manera, no se debe presentar mucha información en un único documento que haga difícil su manipulación por los medios tecnológicos, que pueda ser revisado por diferentes funcionales o que necesite de personas excluyentes para su aprobación. Se debe dividir la información en series de documentos correspondientes a un mismo volumen de acuerdo al alcance de la información, a quién va dirigido y los niveles de aprobación correspondiente. Esto hará que sea más sencillo de leer, revisar y corregir.

Con la **verificación de requisitos del cliente** se garantiza un nivel adecuado en la estructura y contenido técnico de la documentación mediante el cumplimiento del estándar de documentación del proyecto, el ajuste de los contenidos a los objetivos y el alcance del proyecto, la correcta aplicación del conocimiento técnico en la documentación y el uso correcto de la taxonomía en la clasificación, las normas de redacción y ortografía.

En el análisis de los defectos es crucial la participación de los responsables por el equipo de desarrollo, pues muchos defectos pueden ser descartados antes del dictamen final de la revisión por limitaciones de la tecnología o falta de comprensión del revisor.

La verificación se desarrolla en la UCI por Calisoft, quien garantiza el proceso de revisión a gran escala; Como grandes desventajas en el proceso institucionalizado por la universidad se encuentran la demora de las liberaciones producto del uso de estudiantes en entrenamiento, con bajo nivel de compromiso hacia el proyecto y una variación constante por la docencia. Una solución probada de buenos resultados es la inclusión de un inspector del equipo de revisión central como supervisor o responsable directo del equipo de calidad interna del proyecto, de manera que utilice los recursos humanos más identificados con el resultado de la actividad sobre una tendencia ágil de revisión.

Con la **validación de requisitos del cliente** se obtiene el compromiso con la validez de la información levantada. Para esto es crucial la participación de expertos con el suficiente poder de acción como para avalar procedimientos y orientar cambios necesarios en el proceso del negocio y el producto en gestación. Generalmente estos roles son ocupados por grandes jefes de la organización, cuyo tiempo no les permite un estudio profundo de los requisitos expuestos. En su sustitución puede sugerirse el empleo de expertos en el negocio de confianza al mando, de manera que sus decisiones validen el comportamiento levantado y sus sugerencias sean analizadas por los superiores en un corto plazo. Obtener el compromiso del cliente y los miembros necesarios es casi tan importante como obtener el correspondiente al equipo de proyecto frente a un requisito complejo.

Desarrollo de requisitos del producto

Una vez determinados los requisitos del cliente, los subprocesos fundamentales de la IR se aplican nuevamente orientados al discernimiento del software, orientando sus postulados a un móvil de fuerte análisis por parte del equipo de desarrollo. La representación del proceso de **Desarrollo de requisitos del producto** se diagrama en la Figura 4.

En la **elicitación de requisitos del producto** se trabaja en la definición de los requisitos con los que debe cumplir el software a través de casos de uso de RUP, la descomposición de los requisitos del cliente en derivados del producto y el esclarecimiento de los requisitos del producto para el equipo de desarrollo. Este subproceso no tiene tanta duración en el ciclo de vida como el correspondiente al análisis si se realizó una buena elicitación de información en el Desarrollo de requisitos del cliente. Su función radica en la aclaración de posibles dudas e inconsistencias durante la materialización en requisitos del producto. Un razonamiento válido aportado en la investigación es que para proyectos donde no existe negocio y se

modela directamente el dominio, la elicitación de requisitos del producto puede ser un instrumento valioso en el orden de ejecución y logros consecutivos a obtener.

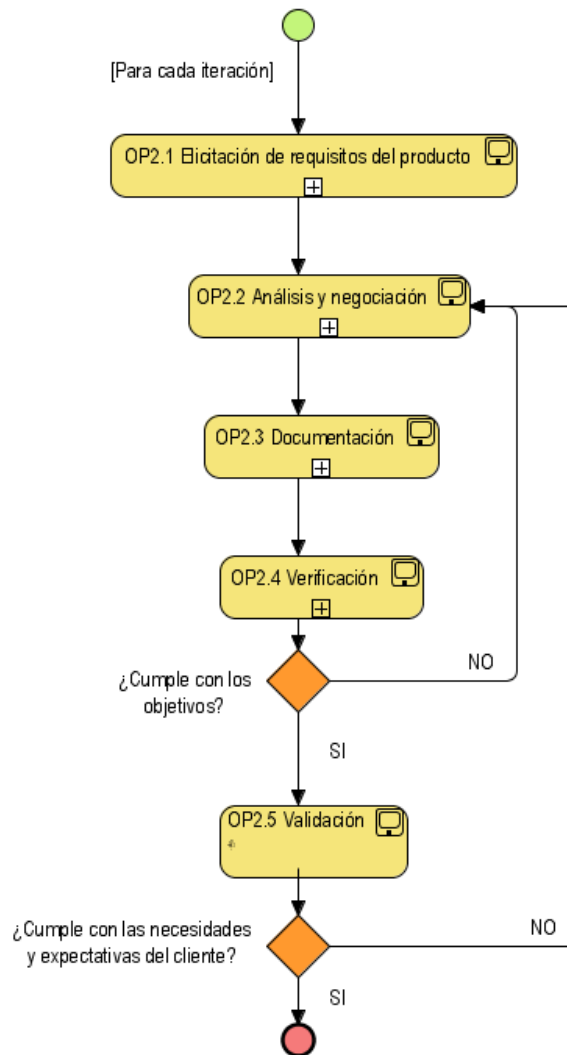


Figura. 4 Subproceso de Desarrollo de requisitos del producto.

El **análisis y negociación de requisitos del producto** se orienta a definir los casos de uso del sistema y sus relaciones con actores, entidades, requisitos funcionales, no funcionales y otros casos de uso. Además de definir el prototipo de interfaces de usuario del sistema enfocados a las necesidades de los mismos, se establece un común acuerdo entre clientes, usuarios y equipo de desarrollo ante posibles inconsistencias; se establecen los principios de la Arquitectura de Información en la modelación del sistema y se multiplica el conocimiento del negocio en los desarrolladores para lograr una visión más clara de los requisitos que debe cumplir el software durante el proceso de construcción.

Junto a la validación de requisitos del producto, es este el proceso de mayor importancia en la determinación del comportamiento del software. Requiere de un alto nivel de experiencia, de conocimiento sobre sistemas homólogos, de creatividad, de conocimientos en plataformas de desarrollo, arquitecturas de software, de información y técnicas de reutilización. Es un momento de razonar, intercambiar en talleres entre los miembros del equipo de análisis, las soluciones independientes de manera que converjan a un único estilo de funcionamiento. Lamentablemente, es la actividad que menos se valora por muchos jefes y gerentes de proyecto por no tener un resultado inmediato en la gestión; es vital para los analistas y administradores de proyecto respetar este tiempo de análisis pues en dependencia de su resultado menos se tendrá que iterar en busca del completamiento de las soluciones adoptadas.

Los subprocesos de **documentación, verificación y validación de requisitos del producto** mantienen los mismos principios que frente a los requisitos del cliente orientando sus técnicas a un objetivo diferente.

Según Pressman la primera versión del producto es siempre un prototipo y la única decisión a tomar al respecto es si entregarla al cliente o desecharla (Pressman, 2006). La validación temprana de prototipos del producto mitiga el efecto predicho por Pressman. Dada la complejidad de la tecnología y la insuficiencia de programadores en la capa de interfaz seleccionada⁴, los prototipos utilizados fueron estáticos y se modelaron sobre la herramienta Office Visio 2003. Se fabricó para esto una galería de componentes bajo las pautas de diseño del proyecto y se modelaron las interfaces auxiliándose de un mapa de navegación para la interpretación de las funcionalidades, las validaciones y las restricciones del negocio. En la Memoria Colectiva (Soto & Valdés, 2010) se ofrecen múltiples ejemplos de estas interfaces y la aplicación de las pautas de Arquitectura de Información.

Esta construcción de prototipos estáticos de interfaz de usuario y los reportes o salidas del sistema para la posterior validación, así como la precisa descripción de las entidades en el glosario de términos son relevantes para la documentación. Son la clave de una correcta interpretación en la etapa de construcción por desarrolladores de interfaces de usuario y la capa de datos.

Uno de los aspectos en los que se hizo necesario una refinación escalable, fue la especificación de casos de uso, que bajo el principio de documentar y luego estructurar, produjo especificaciones en extremo largas y difíciles de entender, siendo en un final posibles de separar en varias especificaciones relacionadas. El uso de prototipos no funcionales en este aspecto, facilitó la comunicación tanto con usuarios como con el equipo de desarrollo, por lo que progresivamente el esfuerzo en la especificación se vio dirigido a la descripción de datos, las funcionalidades básicas y al buen diseño de prototipo.

⁴ En el proyecto se utilizó el framework Java Server Faces, JSF, sobre el lenguaje de programación Java.

En la validación, el instrumento fundamental para la comunicación entre analistas y clientes, usuarios y equipo de desarrollo fue el caso de uso, específicamente de las secciones dedicadas a los prototipos y el diccionario de datos. No significa esto que los escenarios fueran texto muerto frente al cliente, pues siempre se recurrían a la lectura como estudio previo o certificación a la hora de plasmar su firma, sino que en la demostración del uso y aplicación de los requisitos implícitos en el caso de uso de manera comunicativa, los mecanismos antes dichos disminuían los tiempos del proceso garantizando igual o mayor aceptación.

Para el registro de defectos y su respuesta, los mecanismos burocráticos sobre las bases documentales se sustituyeron por la herramienta informática Trac. Mediante su acceso, consulta y exportación de datos se garantizaba el dominio de las interrogantes a la totalidad de participantes, dando soporte a la gestión contractual.

A manera de conclusiones de este epígrafe, el autor considera que el modelo propuesto adapta la vista general del proceso unificado a una solución particular, posible de aplicar con mayor sentido práctico y de obtener buenos resultados en el proceso de desarrollo de software.

Roles

La definición de los roles, es un elemento que defiende el modelo seleccionado para el DR. El porte de la investigación sobre estos se fundamenta con la caracterización del analista de sistema, el arquitecto de información y al analista principal, estos últimos no aparecen en la metodología aplicada. Las definiciones de estos roles se encuentran desglosadas en la Memoria Colectiva (Soto & Valdés).

Es novedosa la responsabilidad que el autor le concede al **analista principal**, definiéndolo como un rol ingenieril con las competencias de la administración de proyectos. Su papel en el desarrollo de software es administrar las actividades correspondientes a los requisitos, manteniendo un dominio completo sobre las definiciones, el estado, la prioridad y la lógica del funcionamiento como un todo del software. El analista principal no puede perder la perspectiva de la arquitectura del negocio ni su transformación continua en un sistema funcional, debe dominar la totalidad de requisitos, su ubicación, dependencia y contribución al objetivo del software, debe ser visto como el líder del producto.

El **arquitecto de información** si bien existe en las definiciones dadas por (Díaz, y otros, 2007) (Morville, 2000) (Ronda, 2006) en aplicaciones Web, afianza los criterios de utilidad y vinculación al software de gestión en el resultado de la presente investigación con las responsabilidades del diagnóstico de la audiencia y el establecimiento de requisitos de usabilidad. Se le otorga además, el mérito del estudio de la distribución espacial de componentes sobre las interfaces de usuario, el tipo, la calidad y la cantidad de la

información que se muestra, así como la definición de los niveles de información de la aplicación. Este rol juega un papel de asegurador de la calidad durante la verificación de los requisitos.

La principal observación realizada sobre el rol del **analista de sistema** es el conocimiento necesario de la arquitectura del software a medida que se va definiendo y de los “frameworks” de la tecnología empleada. Este aspecto regula mucha de las ideas aplicables a otros tipos de software y bajo otras tecnologías durante la resolución de problemas quizás triviales, más aún cuando el equipo de desarrollo es joven y en entrenamiento. No importa que los requisitos se hayan modelado lo más independiente al diseño final, aún en un prototipo no funcional las ideas que se muestren pueden casarse con una solución en la mente del usuario y mucho más riesgoso, en la del mismo desarrollador.

La definición de las habilidades y capacidades de los roles fundamenta su mayor aporte en el discernimiento de los planes de capacitación a aplicarse en la formación de los recursos humanos necesarios para la continuidad del proyecto. Esta clasificación puede ser vista en la Memoria Colectiva (Soto & Valdés, 2010).

Técnicas

Las técnicas aplicadas para el DR se dividen en tres dominios de aplicación: la elicitación de la información, el análisis y documentación de requisitos y en las mediciones de validez, calidad y organización de los requisitos (Soto & Valdés, 2010).

Con el uso del **cuestionario** (Ortíz, y otros, 2006) se inició el proyecto, permitiendo una valoración del estado del negocio en sus inicios, la caracterización de la audiencia y las necesidades preliminares de la organización. A lo largo del desarrollo sirvió para medir el nivel de cumplimiento de las expectativas para los involucrados y el equipo de desarrollo, así como la identificación de vías para la resolución de inconsistencias generales al negocio.

La elicitación de la información utiliza en la investigación la **entrevista** (Durán, 2002) como técnica principal. En escenarios como el expuesto en la situación problemática el intercambio comunicativo es esencial para la determinación de la verdadera necesidad y el discernimiento entre las soluciones propias del usuario y las nunca institucionalizadas por factores organizacionales deficientes; es la contradicción entre “el día a día” y “el deber ser”. Es vital en este tipo de organización tener el punto de vista por separado de varios usuarios y del cliente, siendo seleccionados entre los trabajadores operativos, los ejecutivos y los gerenciales. Los talleres JAD y las tormenta de ideas entre personas de diferente rango en la organización no son recomendables, salvo para las aprobaciones de las propuestas, porque

generalmente el criterio del de mayor rango (el gerencial) prevalece por encima de quien se enfrenta al problema en cuestión (operativo).

La **tormenta de ideas** (Durán, 2002) es muy efectiva en el logro de consensos sobre requisitos del cliente ejecutados de diferentes formas en las organizaciones, siempre que se realice entre profesionales del mismo rango y con basta experiencia en el tema. Es mucho más efectiva que los talleres JAD donde el orden y la disciplina llegan a cansar al auditorio terminando por no ponerse de acuerdo o posponiendo el encuentro sin soluciones encontradas. En la polémica generada por la aceptación de criterios de la tormenta de ideas se llega de manera más rápida al resultado esperado, inclusive para en el análisis de requisitos entre analistas y los restantes miembros del equipo de desarrollo.

El **JAD** (Durán, 2002) es una técnica recomendada para las validaciones sobre todo para los altos niveles de la organización, donde la información ya se presenta organizada, con una propuesta tentativamente sólida y con elementos puntuales de riesgo identificados. Tiende a consumir el menor tiempo posible de sus involucrados, siendo la clave de su éxito la buena preparación y comunicación de los temas a intercambiar.

La **observación documental** (Ortíz, y otros, 2006) se considera la técnica de mayor aporte a la identificación de entidades y atributos del negocio. Una cantidad considerable de modelos, plantillas, formularios, “scripts” de bases de datos, impresiones de pantallas y reportes del sistema legado permitieron el completamiento del diccionario de datos del software dejando en un menor grado las definiciones para la entrevista. Así mismo el análisis de procedimientos, procesos, diagramas y documentos reglamentados en una organización, como la policial estudiada, contribuyó a la aprehensión de conocimientos jurisdiccionales y su modelación en los requisitos en un menor espacio de tiempo.

En el análisis y documentación de requisitos la práctica demostró el buen uso de los patrones de casos de uso orientados a las relaciones, al proceso y a su estructura. Las definiciones del **CRUD** (Cockburn, 2000) y la **Reusabilidad** (Övergaard, y otros, 2004) permitieron la creación de una cultura de interpretación de funcionalidades en el usuario/cliente y de facilidad de reutilización para la modelación y construcción al equipo de desarrollo. La **Reusabilidad interna** (Övergaard, y otros, 2004) estratificó especificaciones extensas en subflujos independientes con un mayor manejo para el entendimiento y la asignación en desarrollo, así como la **Adición** (Övergaard, y otros, 2004) diferenció los niveles de dependencia entre funcionalidades aprobadas e incrementos, permitiendo el control sobre las extensiones de requisitos dentro del proceso de DR sobre una nueva línea espiral.

En los patrones orientados al proceso (Bramble, 2001) el éxito se debe al establecimiento de **Pequeños equipos de especificación** donde un analista llevaba el control de la entrevista y otro auxiliaba en la sustanciación de la información, obteniéndose luego un conjunto de aspectos comunes durante la revisión por pares y el intercambio de puntos de vista diferentes en la construcción de los subsistemas que estandarizaban la visión de los incrementos. La inclusión en el proceso de análisis y negociación de los requisitos del cliente, el usuario y terceros portadores o consumidores de información hicieron del patrón **Participación de la audiencia** un aliado en el logro de requisitos completos a la par de su negociación, mitigando el cambio posterior y reduciendo tiempos de modelación. Los **Equipos balanceados** garantizaron la especialización del conocimiento desde la perspectiva de cada rol y la defensa de sus intereses fomentó la edificación de requisitos correctos.

En los patrones orientados a la estructura (Bramble, 2001) se consideran de mayores resultados para la identificación de varios requisitos como casos de uso así como para la agilidad en la aprobación, la complementación de un **Objetivo único** con una **Intensión visible del actor**, facilitando el reconocimiento y la comunicación frente al cliente y el equipo de desarrollo el **Intención revela el nombre**. La exhaustiva fragmentación del modelo fue controlada a través de **Escenarios y no fragmentos** quien defiende los criterios de (Larman, 1999) de especificar antes de estructurar centrándose en describir de manera completa la intención de uso del artefacto. El empleo en el lenguaje de **Transacciones con valor al usuario** regulado por el principio de **Legible y entendible** y **Neutral a la tecnología** permitió una especificación posible de implementarse en cualquier escenario tecnológico, fácil de entender el cómo y el valor del caso de uso, contribuyendo al éxito de los procesos de verificación, negociación y validación de los requisitos.

Las mediciones sobre el DR sobre los criterios de **aceptación del requisito** y **selección de proveedores**, si bien se ejecutaron en sus inicios de una manera empírica, se le propone un fundamento matemático elemental en el cuerpo de la Memoria Colectiva (Soto & Valdés, 2010). El primero se construye a partir de las propiedades de los requisitos referenciadas en el Capítulo 1 y el segundo a partir de las características que debe tener un trabajador del negocio de mayor importancia a la hora de sustanciar una información para sistemas de gestión policial, como la experiencia, compromiso, seriedad, conocimientos legales, informáticos, creatividad, entre otros. Se adopta como un hecho que cualquier trabajador no tiene aptitudes para comunicar la información necesaria del negocio, produciendo en muchos casos la extensión del tiempo y la creación de ruidos en la información; asimismo, no toda información es un requisito del sistema y el incumplimiento de alguna de las propiedades arroja resultados indeseables en el DR. Para la **clasificación de casos de uso** y **la priorización de requisitos** se utiliza como base el método de estimación por puntos de casos de uso y se analizan factores esenciales en sistemas de

gestión como la repetición y dependencia de acciones, la experiencia ganada, el esfuerzo de realización basado en su complejidad y el beneficio directo al cliente.

Se puede concluir que las mediciones propuestas, aunque simples y flexibles, proveen una guía en la definición de criterios necesarios para el ordenamiento de los requisitos de software, utilizando como base modelos tradicionales de gestión de software.

Herramientas

En la selección de las herramientas que complementen el modelo de DR, se tuvieron en cuenta las que contribuyeran a la interpretación y a la modelación de los requisitos. En este sentido se sustenta que herramientas simples, sin muchas restricciones de funcionamiento ni complejidad en el llenado o flujo de los datos, llevan a cabo de manera más efectiva al desarrollo de proyectos complejos. La afirmación se fundamenta en el uso del repositorio **Subversion** (SVN) y su cliente **TortoiseSVN** para el control de todo artefacto generado durante el DR, la validación de su integridad, su autenticidad y el seguimiento histórico del cambio. En su complemento, el **Trac** garantiza la asignación de tareas, el tratamiento de defectos en los productos de trabajo y el control de su resolución por versiones de entrega.

Visual Paradigm para la modelación en el UML de los diagramas correspondientes a la metodología y **Microsoft Office Visio 2003** para el prototipado de interfaces de usuario se consideraron como las herramientas de mayor fortaleza en el diseño para el modelo adoptado. En la selección del Visio influyó el nivel de integración del paquete ofimático de Microsoft, permitiendo modificar las interfaces de usuario desde el documento de especificación de casos de uso hecho en **Microsoft Office Word 2003** durante las validaciones con el usuario. Además de esto, la facilidad de personalización de los componentes y bibliotecas visuales posibilitó la configuración de un diseño gráfico en el prototipo similar al resultado final. Para la protección de la documentación y la declaración de autenticidad hacia una entrega al cliente se empleó el formato digital portable sobre **Adobe Acrobat 9.0**. Es necesario señalar que el uso de las herramientas propietarias no está condicionada por un criterio de oponencia al uso de tecnologías libres, en la Memoria Colectiva (Soto & Valdés, 2010) se expone una comparación entre varias funcionalidades de las herramientas que refuerzan el criterio de uso.

En conclusiones del epígrafe, las herramientas seleccionadas complementan el logro de los objetivos trazados en la investigación bajo un tiempo y costo esperado, sirviendo de instrumento para la modelación, especificación y control de los aspectos necesarios y la consecuente interpretación de sus resultados.

Resultados de la aplicación en el proyecto CICPC

La aplicación del proceso aportó beneficios significativos en el desarrollo del proyecto, siendo su máxima expresión la aprobación final de una versión funcional del sistema, de común acuerdo entre los involucrados y el equipo de desarrollo.

En los Anexos de la Memoria Colectiva (Soto & Valdés, 2010) se exponen las actas de aceptación por parte del cliente/usuario y de la Alta Gerencia de los principales resultados de la investigación expresados en las especificaciones de requisitos suplementarios, el proyecto técnico como aceptación de la visión, los procesos elementales del negocio y los casos de uso del producto. La aceptación se les presentó como un estado de entendimiento del contenido de los documentos, de los procesos del proyecto y el común acuerdo de su implementación; se cuenta además con una evaluación objetiva del proceso desde la perspectiva del cliente.

En estudios estadísticos realizados sobre el equipo de desarrollo, expuestos en detalle en la Memoria Colectiva (Soto & Valdés, 2010), se obtuvieron resultados alentadores en cuanto a los niveles de interpretación, negociación y entendimiento de los requisitos sobre el 75% de los roles que consumen los artefactos de salida del desarrollo de requisitos. Así mismo la valoración de los roles fundamentales de desarrollo, análisis y soporte sobre las características del modelo, determina el grado de beneficio de su aplicación en el proyecto con un promedio alto de aceptación. La Figuras 5 y 6 muestran gráficos que refuerzan estas afirmaciones.

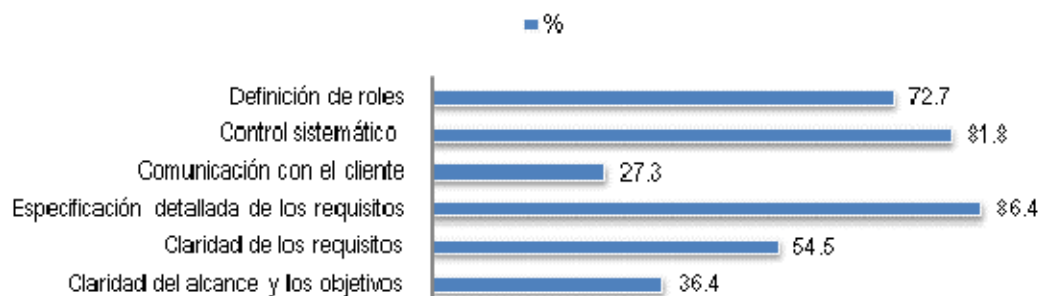


Figura 5. Factores de incidencia del éxito del proyecto.

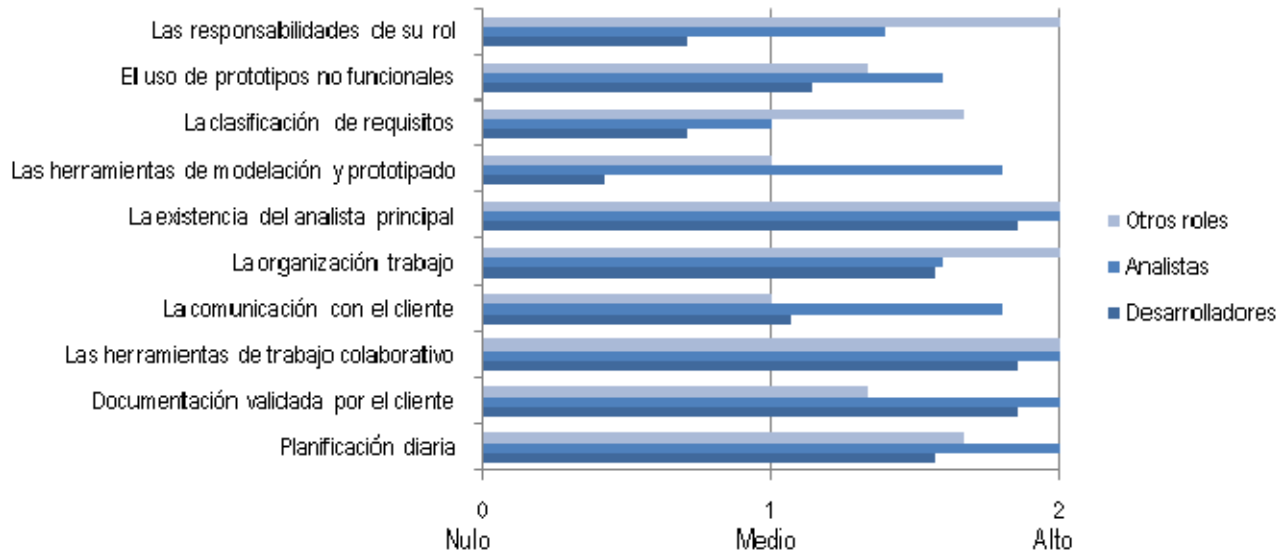


Figura 6. Grado de beneficio de factores por roles de interés.

Durante los años 2006 y 2008 no se realizaron auditorías al proyecto CICPC por Calisoft, sin embargo en el resultado de las revisiones efectuadas a la adherencia del proceso se evidencian el cumplimiento de las buenas prácticas de los estándares de calidad establecidos para el desarrollo de requisitos, esto incluye la definición de planes de gestión, de documentación específica para el registro de la información, el chequeo y control de los acuerdos, la aplicación de la ingeniería en el levantamiento de información sobre soportes documentales de servicio al cliente y al equipo de desarrollo. Además se puede constatar que más de un 85% de los miembros del equipo dominan la existencia de los elementos del modelo, conocen los procedimientos y aplican sus principios y lineamientos en su trabajo diario.

CONCLUSIONES GENERALES

*“La gente olvida cómo de rápido hiciste un trabajo
pero siempre recuerda cómo de bien lo hiciste”
(Howard Newton)*

El autor concluye que el modelo de DR propuesto garantiza una buena interpretación y modelación de los requisitos, sirviendo como puente de comunicación en la eliminación de las inconsistencias de la información entre los involucrados del negocio y el equipo de desarrollo durante el proceso de construcción de un software en el dominio de la gestión policial.

Con el modelo propuesto se tiene garantía de la elicitación, el análisis, la documentación y la validación de los requisitos del software a través de un método ordenado que controla el flujo de datos manejado. La incorporación del usuario, el cliente y otros involucrados del negocio abre un espacio para la negociación inmediata de los requisitos mucho antes de su validación final, reduciendo el margen de los cambios. El cumplimiento de estándares y principios de calidad aplicables a la organización y al producto complementan el enfoque de calidad, incorporando mejoras a la actividad de validación institucionalizada en la UCI.

La incorporación del estudio del negocio y el proyecto permite la preparación del escenario de desarrollo y el establecimiento de los requisitos técnicos, derivados de las características del software policial. La naturaleza iterativa del modelo permite el descubrimiento y negociación de estos requisitos a medida que se obtengan de las necesidades de los involucrados; no obstante, una identificación temprana contribuye a una mejor definición del alcance y viabilidad del proyecto.

El modelo propuesto establece en su estructura un procedimiento para la gestión del conocimiento en roles de poca experiencia en el desarrollo de software de gestión policial. Se considera una guía para la creación de una cultura laboral que garantiza la comunicación dentro de un equipo de desarrollo. Su aplicación permite una modelación escalable de los requisitos de software como base a una plena comunicación entre las partes para el logro de los objetivos del cliente y la resolución exitosa del problema.

RECOMENDACIONES

A manera de recomendación el autor propone:

- Estudiar la Ingeniería de procesos y aterrizar sus conceptos directamente sobre la visión de la IR, de ser posible.
- Incluir indicadores y metas cuantitativas por proceso, de manera que se haga más precisa la evaluación de la adherencia al proceso.
- Considerar el modelo propuesto como parte del proceso Desarrollo de Requisitos de CMMI en su nivel 3, como parte del proceso de mejora de la UCI.
- Utilizar estos conocimientos para complementar el entrenamiento de los recursos humanos que asuman el rol de analista en sistemas de gestión policial, e incluirlo en los objetivos instructivos a lograr en algunos de los cursos del perfil de Seguridad Ciudadana.

REFERENCIAS BIBLIOGRÁFICAS

Bramble, Adolph. 2001. *Patterns for effective use cases.* 2001.

Brooks, F. P. 1995. *The Mythical Man–Month: Essays on Software.* s.l. : Addison Wesley, 1995.

Calidad-UCI. 2008. *Informe anual de producción.* [Electrónico] Ciudad de La Habana : UCI, 2008.

CMMI. 2002. *Capability Maturity Model Integration. Stage Representation.* Pittsburgh : SEI Carnegie Mellon, 2002. CMU/SEI-2002-TR-011.

Cockburn, Alistair. 2000. *Writing effective use cases.* s.l. : Addison-Wesley, 2000.

Díaz, Lic. Yimian de Lyz Contreras y Amador, Lic. Soleydi Rivero. 2007. *Diseño del sistema de gestión de información del centro de estudios de medio ambiente y recursos naturales (cemarna) de la universidad de pinar del río.* Pinar del Río : Universidad de Pinar del Río, 2007.

Dorfman, Merlin y Thayer, Richard H. 1990. *Standards, Guidelines, and Examples of System and Software Requirements Engineering.* s.l. : Los Alamitos : IEEE Computer Society Press, 1990.

Durán, Amador & Bernárdez, Beatriz. 2002. *Metodología para la Elicitación de Requisitos de Sistemas Software.* Sevilla : Universidad de Sevilla, 2002. LSI–2000–10.

Grady, Robert. 1992. *Practical Software Metrics for Project Management and Process Improvement.* s.l. : Prentice-Hall, 1992. 0137203845 .

IAG. 2009. Press Release: Business Analysis Benchmark 2008. *IAG Consulting.* [En línea] 2009. [Citado el: 07 de Julio de 2009.] <http://www.iag.biz>.

IBM. 2006. *Rational Unified Process.* [Ayuda electrónica Rational Modeler] 2006.

IEEE1012. 1998. *IEEE Std 1012-1998 Standard for Software Verification and Validation.* New York : s.n., 1998.

IEEE610. 1991. *IEEE Standard Glossary of Software Engineering Terminology.* Spring 1991 Edition. 1991. IEEE Standard 610.12–1990.

IEEE830. 1998. *IEEE Recommended Practice for Software Requirements Specifications.* Software Engineering Standards Committee of the IEEE Computer Society. 1998. ISBN 0-7381-0448-5, SS94654.

- IP-UCI. 2006.** Memoria no publicada del Taller de experiencias de la solución del producto. Ciudad de la Habana, UCI, Noviembre, 2006.
- Larman, Craig. 1999.** *UML y patrones. Introducción al análisis y diseño orientado a objetos.* s.l. : Prentice Hall, Segunda Edición, 1999. 9701702611.
- Leite, Julio Cesar. 1987.** *A Survey on Requirements Analysis.* Department of Information and Computer Science, University of California at Irvine. 1987. Advanced Software Engineering Project Technical Report .
- Loucopoulos, P. & Champion. 1989.** R.E.M. Knowledge-Based Support. *Information and Software Technology.* Abril de 1989. págs. 123-125.
- Mead, Nancy R. 2003.** *Requirements Engineering for Survivable Systems.* s.l. : Carnegie Mellon University, 2003. CMU/SEI-2003-TN-013.
- Morville, Rosenfeld & Peter. 2000.** *Arquitectura de la Información para el WWW.* s.l. : MacGraw Hill, 2000.
- NC-ISO9000. 2001.** *NC-ISO 9000:2001 Sistemas de Gestión de la Calidad. Principios fundamentales y vocabulario.* Ciudad de La Habana : Oficina de Normalización, 2001.
- NC-ISO90003, 2006.** *NC-ISO 90003:2006 Ingeniería de software - directivas para la aplicación de la NC-ISO 9001:2001 al software de computación.* Ciudad de La Habana : s.n., 2006.
- Oktaba, Hanna, Esquivel, Claudia Alquicira y Ramos, Angélica Su. 2005.** *Modelo de Procesos para la Industria de Software.* Facultad de Ciencias, Universidad Nacional Autónoma de México. 2005.
- OMG. 2007.** *OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2.* 2007.
- Ortíz, Frida y García, María Del Pilar. 2006.** *Metodología de la investigación. El proceso y sus técnicas.* México : Limusa, 2006. 9789681860752.
- Övergaard, Gunnar y Palmkvist, Karin. 2004.** *Use Cases Patterns and Blueprints.* s.l. : Addison Wesley Professional, 2004. pág. 464. 0-13-145134-0.
- Pardillo Castañeda, Martha Karina y Pérez Teruel, Karina. 2007.** *Referencia para la aplicación de la metodología ágil Programación Extrema a proyectos DESOFT SS.* [Electrónico] Ciudad de La Habana : UCI, 2007. 005.1-Par-R-TD-0220-07.

- Pérez, Karina. 2007.** *Modelo de Referencia para la IR en proyectos de Bioinformática INeR*. Ciudad de La Habana : s.n. UCI
- PMI. 2009.** *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Fourth Edition*. Project Management Institute, USA, ISBN: 9781933890517.
- Pressman, Roger S. 2006.** *Ingeniería de Software. Un enfoque práctico*. México : McGraw-Hill, 2006.
- Ronda, León Rodrigo. 2006.** *Algunas técnicas para los arquitectos de información*. Ciudad de La Habana : s.n., 2006.
- Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 2000.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Pearson Educacion, 2000.
- Sawyer, P. & Kotonya, G. 2001.** Software Requirements. [ed.] CA: IEEE Computer Society. *Guide to the Software Engineering Body of Knowledge, Trial Version*. s.l. : Los Alamitos, 2001, Cap. 2, págs. 9-34. <http://www.swebok.org>.
- Sommerville, I. & Sawyer, P. 1997.** *Requirements Engineering: A Good Practice Guide*. s.l. : John Wiley & Sons, 1997. págs. Cap. 1-2.
- Sommerville, Ian. 2005.** *Ingeniería del software*. Madrid : Pearson Education, 2005. 84-7829-074-5.
- Soto, Nilet, & Valdés, Sasha. 2010.** *Sistema de Investigación e Informacion Policial. Aplicación de la Ingeniería de requisitos*. Memoria colectiva, UCI, 2010.
- STEP, Software Test & Evaluation Panel. 1991.** Requirements Definition Implementation Team . *Operational Requirements for Automated Capabilities*. [Draft Pamphlet]. 23 de Abril de 1991.
- SWEBOK. 2004.** *Software Engineering Body of Knowledge*. s.l. : IEEE, 2004.
- TECNOMATICA. 2009.** *IAC-I Guía para el desarrollo de software*. [Electrónico] Ciudad de La Habana : MINBAS, 2009.
- Temür, Nuri. 2009.** *Towards a new model of Public Private Partnership for E-Government Project Implementation*. eGovShare 2009: International Conference on e-Government, Antalya, Turkía. 9 de diciembre de 2009.
- Young, R. 2004.** *The Requirements Engineering Handbook*. Boston, London : ARTECH HOUSE, 2004. 1-58053-266-7.

BIBLIOGRAFÍA CONSULTADA

Berry, Daniel M. 1992. *Academic Legitimacy of The Software Engineering Discipline*. MSE Project, Software Engineering Institute. s.l. : Carnegie Mellon University, 1992. Technical Report. CMU/SEI-92-TR-34.

Ford, G. 1990. *SEI Report on Undergraduate Software Engineering*. Software Engineering Institute. Pittsburgh : Carnegie Mellon University, 1990. Technical Report. CMU/SEI-90-TR-3, DTIC: ADA223881.

Furet Díaz, Sandra, Guzmán Hernández, Yailen y Rondón Hernández, Yaneida. 2008. *Aplicación de Mejora de Proceso a Metodologías de Software en el área de Modelado de Negocio y Gestión de Requerimiento*. [Electrónico] Ciudad de La Habana : UCI, 2008. 005.12-Fur-M-TD-1672-08.

Garret, Jesse James. 2002. *The Elements of User Experience*. New York : New Riders Publishing, 2002.

Gibbs, W. Wayt. 1994. *Software's Chronic Crisis*. Septiembre de 1994, Scientific American, pág. 86.

IEEE1042. 1993. *IEEE Std 1042-1987 (Reaff 1993), IEEE Guide to Software Configuration Management*. New York : s.n., 1993.

IEEE1074. 1997. *IEEE Std 1074-1997 Standard for Developing Software Life Cycle Processes*. New York : s.n., 1997.

ISO13407. 1999. *ISO 13407:1999 Human centred design for interactive systems*. New York : s.n., 1999.

La Calle, Alberto. 2005. Alberto La Calle. *Prototipos*. [En línea] 2005. [Citado el: 10 de Octubre de 2009.] http://www.albertolacalle.com/hci_prototipos.htm.

Leyva, Ailyn López y Cuesta, Walber Mengana. 2009. *Aplicación del Modelado del Universo de Discurso a la Metodología TOT*. [Electrónico] Ciudad de La Habana. : UCI, UCI, 2009. 005.1-Lop-A-TD-1895-09 .

Maner, Walter. 1997. *Prototipado*. [En línea] 1997. [Citado el: 10 de Octubre de 2009.] <http://www.sidar.org/recur/desdi/traduc/es/visitable/maner/Prototipado.htm>.

Mejía, Pedro. 2005. Proceso de Ingeniería de Requerimientos Aplicado a Proyectos de Software. 2005. Sección de Computación. CINVESTAV-IPN.

Mojena, Elizabeth Ramírez y Castellanos, Orestes Martínez. 2009. *Análisis del proceso de Gestión de Quejas en la Empresa Correos de Cuba utilizando la Metodología TOT.* [Electrónico] Ciudad de La Habana : UCI, 2009. 005.12-Ram-A-TD-1909-09 .

Preuss, Deborah Hartmann. 2006. InfoQ. *Interview: Jim Johnson of the Standish Group.* [En línea] C4Media Inc, 25 de Agosto de 2006. [Citado el: 12 de Mayo de 2010.] <http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS>.

Quiroga, Lourdes Aja. 2002. *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones.* 2002.

Randall W., Jensen y Charles C., Tonies. 1979. *Software engineering.* Englewood Cliffs, New Jersey : Prentice-Hall , 1979. 0-13-822130-8.

Randell, Brian & Naur, Peter. 1969. *NATO Software Engineering Conference.* [ed.] NATO SCIENCE COMMITTEE (Comitee de Ciencias de la OTAN). Garmisch, Germany : s.n., 1969.