

Universidad de la Ciencias Informáticas

Facultad 2



*Trabajo de Diploma para optar por el título de
Ingenieros en Ciencias Informáticas*

Título:

*“Método de evaluación del impacto de los cambios a los requisitos del
producto de software.”*

Autores:

Adriana Troche Robles

Yaneisi Acosta Navarro


Tutores:

Ing. Kariné Ramos Blanco

Ing. Dennis Neuland Agüero

Ciudad Habana, Cuba

Junio 2010



*“La gente no le tiene miedo al cambio,
sino a la incertidumbre que genera el cambio.”*

Alvin Toffler

Agradecimientos

De Adriana:

A mis padres por todos sus sacrificios, por llorar conmigo cuando estaba triste y reírse de mis locuras, por darme un buen consejo cada vez que lo necesitaba, por guiarme siempre por el buen camino y enseñarme a valorar las cosas de la vida. Muchas gracias por ser mis padres.

A mi abuelita Cachita, que aunque hoy no pudo estar conmigo se sentiría muy orgullosa de verme graduada.

A mi abuela Elsa y a mi abuela Fesita, por siempre estar a mi lado, darme todo su cariño, por cuidarme, malcriarme y por hacerme muy feliz.

A Javier por ser mi novio, mi amigo, mi compañero y porque siempre puedo contar con su apoyo y su todo su amor, te amo bebé.

A toda mi familia que siempre se preocuparon por saber como me iba en los estudios.

A mis tías, Maricel y Belkis y a mi tío Paquitín, son los mejores tíos del mundo.

A Osmel y Manuela por quererme como una hija.

A mis tutores Kariné y Dennis por dedicarnos tanto tiempo, por apoyarnos y defendernos, y estar siempre ahí para nosotras.

A Tata, mi loca compañera de tesis por contagiarme con su optimismo en los momentos difíciles.

A todos mis profesores y compañeros de grupo.

A todos los amigos que tuve el placer de conocer durante estos 5 años.

A las niñas del edificio 13, Diana, Yarle, Leydis, Vanesa, Yurelis, Jessica, Geny, Katy, Yelina, Lisbet, por ser las hermanas que nunca tuve y ser las mejores amigas del mundo.

A Fidel y a la Revolución por darme la oportunidad de estudiar en esta Universidad de Excelencia.

Y a todos los que en algún momento preguntaron ¿Cómo va la tesis?

De Yaneisi:

A mi mamá, gracias por siempre estar presente, por ser mi principal fuente de inspiración para avanzar en la vida, por el esfuerzo constante y por una vida de tantos sacrificios, te quiero, eres la mejor del mundo, te regalo mi título.

A mi otra mamá (mi abuela Esperanza) por todas las cosas buenas que ha hecho por mi, por quererme , malcriarme y fundamentalmente por su preocupación y apoyo en todo este tiempo.

A mi hermanita Yoania, gracias por todo el amor y cariño que siempre me has dado, siempre seré un ejemplo para ti, te quiero Yoa.

A mis queridos tíos Yobeida y Yunior, a ustedes les agradezco por estar siempre ahí para mi, brindándome todo el payo y la comprensión que nunca recibí de mi padre, la cual necesitaba para cumplir con mi sueño de convertirme en una ingeniera.

A mi querido novio Leynier Viquillon, gracias por ser el mejor amigo y amiga incondicional que tuve durante estos 5 años en la universidad, gracias por estar siempre ahí para escucharme, para secarme las lágrimas, por apoyarme, por elegirme a mi como tu novia,

gracias por tu amor y por todos los momentos que pasamos juntos en todos estos años y fundamentalmente por ayudarme en los momentos más difíciles de mi vida, nunca te olvidare.

A mis magníficos tutores Kariné y Dennis gracias por apoyarnos en todo momento, por su comprensión, dedicación absoluta, por su esfuerzo, por la confianza depositada en mí y por su ayuda a lo largo de todo este tiempo. Muchas gracias.

A mi amiga Nilsa por los tantos consejos que me diste y que me sirvieron de mucho.

A mis compañeros de aula y amigos en especial a mi compañera de tesis Adriana, Lisandra, Dayana, Yelina, Genisel, Yeni, Lianet, Ana Rosa, Katy, María, Cinolkis, Reinier y Samuel, le doy gracia a la vida por verme dado la oportunidad de conocerlos y por compartir juntos momentos de preocupación y alegría. Siempre los llevare en mi corazón.

A todas las personas que se preocuparon por el desarrollo de esta tesis.

A la Revolución y nuestro Comandante Fidel por darme la posibilidad de realizar mis sueños.

A todos gracias y les regalo la satisfacción de que ya soy Ingeniera.

Resumen

En el proceso de desarrollo de software los cambios son inevitables, estos pueden tener consecuencias negativas sino se hace un análisis del impacto que tendrá el cambio en las funcionalidades del sistema, en el costo, en el tiempo de desarrollo y en el esfuerzo del personal encargado.

La presente investigación surge con la necesidad de realizar un método que analice de forma objetiva el impacto de los cambios a los requisitos del software y a la vez que ayude al equipo de desarrollo a la hora de tomar una decisión sobre la factibilidad de los mismos. Con el desarrollo de este método se logra obtener un trabajo más organizado en los proyectos productivos permitiendo así el desarrollo exitoso de los mismos, además le ofrece al jefe de proyecto información valiosa en cuanto al cambio, tales como los ECS que se vieron afectados, obtiene un valor exacto del esfuerzo y el costo que implica realizarlo, la cantidad de requisitos afectados, la cantidad de personas implicadas y el tiempo de desarrollo necesario para implementar este cambio. Con esta información obtenida los miembros del Comité de Control de Cambio pueden reunirse, analizar, revisar y finalmente podrán aceptar o rechazar la Solicitud del Cambio realizada.

Índice de Contenido

DEDICATORIA.....	ERROR! BOOKMARK NOT DEFINED.
AGRADECIMIENTOS.....	II
DECLARACIÓN DE AUTORÍA.....	ERROR! BOOKMARK NOT DEFINED.
DATOS DE CONTACTO	ERROR! BOOKMARK NOT DEFINED.
RESUMEN	VIII
ÍNDICE DE CONTENIDO	1
ÍNDICE DE FIGURAS.....	4
ÍNDICE DE TABLAS	5
INTRODUCCIÓN.....	6
CAPÍTULO1: FUNDAMENTACIÓN TEÓRICA	10
1.1 INTRODUCCIÓN.....	10
1.2 CONCEPTOS FUNDAMENTALES.....	10
1.2.1 CMMI (Capability Maturity Model Integration)	10
1.2.2 Programa de Mejora	11
1.2.3 Gestión de Configuración del Software (GCS)	11
1.2.4 Elemento de Configuración del Software (ECS).....	11
1.2.5 Proceso de cambio	12
1.2.6 Control de cambios	12
1.2.7 Línea base de configuración.....	13
1.2.8 Análisis de impacto	14
1.3 ROLES ASOCIADOS AL COMITÉ DE CONTROL DE CAMBIOS (CCC).....	14
1.4 REQUISITOS.....	15
1.4.1 Clasificación de los requisitos.....	15
1.4.2. Trazabilidad de requisitos	16
1.4.2.1 Trazabilidad bidireccional.....	16
1.4.3 Matrices de trazabilidad	17
1.4.4 Guía de trazabilidad	17
1.4.5 Información de trazabilidad.....	19

1.5. HERRAMIENTAS PARA LA ADMINISTRACIÓN DE REQUISITOS	19
1.5.1 OSRMT (Open Source Requirement Management Tool)	20
1.5.2 Rational RequisitePro	21
1.5.3 Enterprise Architect.....	21
1.6 MÉTODOS DE ESTIMACIÓN.....	22
1.7 ANÁLISIS DE IMPACTO COMO PROCESO	24
1.8 CLASIFICACIÓN DE LOS MÉTODOS DE ANÁLISIS DE IMPACTO	26
1.9. MÉTODOS DE ANÁLISIS DE IMPACTO DE LOS CAMBIOS	27
1.9.1 Método de análisis de los cambios en software orientado a objetos	27
1.9.2 Método diseño relacional	28
1.9.3 Procedimiento para la Toma de decisiones en la Gestión de Cambios	29
1.10 VENTAJAS DEL ANÁLISIS DE IMPACTO.....	30
1.11 CONCLUSIONES	30
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN	31
2.1 INTRODUCCIÓN.....	31
2.2 IDENTIFICACIÓN DE LOS ELEMENTOS DE CONFIGURACIÓN DE SOFTWARE.....	31
2.3. LISTA DE LOS ELEMENTOS DE CONFIGURACIÓN DE SOFTWARE.....	31
2.3.1 Estructura de trazabilidad entre los elementos.	36
2.4 ARTEFACTO QUE SE GENERA DURANTE EL PROCESO DE GESTIÓN DE CAMBIOS.....	37
2.5. PROPUESTA DEL MÉTODO DE EVALUACIÓN DEL IMPACTO DE LOS CAMBIOS A LOS REQUISITOS DEL PRODUCTO DE SOFTWARE.....	37
2.5.1 Identificar la fuente del cambio	38
2.5.2 Identificar los elementos relacionados con el cambio.....	38
2.5.3 Construir la tabla de los elementos afectados con el cambio.....	39
2.5.4 Calcular el Factor de Peso (FP) para cada tipo de elemento	39
2.5.5 Calcular los ECS sin ajustar	45
2.5.6 Calcular los ECS ajustados	45
2.5.7. Calcular el Esfuerzo y el Costo que implica el cambio	50
2.5.7.1 Análisis de Esfuerzo que implica el cambio.....	51
2.5.7.2 Análisis de Costo que implica el cambio.....	51

2.6. TOMA DE DECISIONES RESPECTO AL CAMBIO..... 51

 2.6.1 Cantidad de requisitos a cambiar..... 52

 2.6.2 Esfuerzo del cambio 52

 2.6.3 Costo del cambio 52

 2.6.4 Riesgos que pueden surgir o afectarse con el cambio..... 53

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS..... 55

3.1 INTRODUCCIÓN..... 55

3.2. APLICACIÓN DEL PROCEDIMIENTO A UN CASO DE ESTUDIO 55

 3.2.1 Caso de estudio 55

 3.2.2 Aplicación del método 60

3.3 CONCLUSIONES 69

CONCLUSIONES GENERALES..... 70

RECOMENDACIONES..... 71

REFERENCIAS BIBLIOGRÁFICAS 72

ANEXO 1: DOCUMENTO DE SOLICITUD DE CAMBIO POR MEJORA. **ERROR! BOOKMARK NOT DEFINED.**

ANEXO 2: DOCUMENTO DE SOLICITUD DE CAMBIO POR ERROR..... **ERROR! BOOKMARK NOT DEFINED.**

ANEXO 3: FORMATO DE LA ENCUESTA APLICADA PARA OBTENER EL FACTOR DE LOS ECS.
..... **ERROR! BOOKMARK NOT DEFINED.**

GLOSARIO DE TÉRMINOS **ERROR! BOOKMARK NOT DEFINED.**

Índice de Figuras

Figura 1: Líneas Base.....	14
Figura 2: Trazabilidad bidireccional en CMMI	17
Figura 3: Relación entre los elementos a tracear en el proyecto	19
Figura 4: Matriz de trazabilidad en OSRMT	20
Figura 5: Matriz de trazabilidad en Rational RequisitePro.....	21
Figura 6: Matriz de trazabilidad con el Enterprise Architect.	22
Figura 7: Proceso de análisis de impacto	25
Figura 8: Relaciones entre los algoritmos de análisis de impacto de los cambios	27
Figura 9: Procedimiento para la Toma de decisiones en la Gestión de Cambios	29
Figura 10: Nueva guía de trazabilidad	36
Figura 11: Matriz de probabilidad de impacto	53
Figura 12: Relación entre los ECS afectados como se muestra en la OSRMT.....	61

Índice de Tablas

Tabla 1: Relación entre los ECS y sus ubicaciones.....	19
Tabla 2: Elementos de Configuración de Software.....	35
Tabla 3: Tabla de elementos afectados.....	39
Tabla 4: Peso de los factores de complejidad técnica.....	48
Tabla 5: Escala de los factores de complejidad técnica.....	49
Tabla 6: Peso de los factores de ambiente.....	50
Tabla 7: Elementos afectados con el cambio.....	62
Tabla 8: Tabla para organizar los datos con los que se calcula el Factor.....	63
Tabla 9: Criterios para dar el Valor Asignado en el cálculo del FCT.....	65
Tabla 10: Criterios para dar el Valor Asignado en el cálculo del FA.....	66
Tabla 11: Esfuerzo de los ECS.....	67

Introducción

En la actualidad el desarrollo de software se ha convertido en una rama importante en la economía de nuestro país. En la Universidad de las Ciencias Informáticas (UCI), se desarrollan una variedad de proyectos nacionales e internacionales que aportan resultados importantes. Para optimizar estos resultados se lleva a cabo un programa de mejora en los procesos de desarrollo de software.

En el desarrollo de software los cambios, debidos principalmente a modificaciones de requisitos y fallos, son inevitables. El análisis del impacto de estos cambios se encuentra dentro de la Gestión de Configuración de Software, esta es una actividad de protección que se aplica a lo largo de todo el proceso del software, esta identifica, controla, audita e informa de las modificaciones que invariablemente se dan al desarrollar el software una vez que ha sido distribuido a los clientes [1].

Al desarrollar un software se deben tener en cuenta lo que los clientes necesitan, por lo que se debe hacer un buen levantamiento de requisitos. Estos son una manera de comprender mejor el desarrollo de las necesidades de los clientes, por lo tanto, es esencial tener en cuenta los posibles cambios en los requisitos a lo largo del ciclo de vida del proyecto; los mismos pueden ocurrir por varias razones, porque al analizar el problema, no se hacen las preguntas correctas a las personas adecuadas, porque los usuarios cambiaron su forma de pensar o sus percepciones, o porque cambió el ambiente de negocio.

Cuando un cliente solicita un cambio se deben identificar los problemas y causas por los que se produce. Para esto los requisitos deben ser traceables, se debe conocer su origen, quien lo propuso, su necesidad, su relación con otros requisitos, y también su relación con el diseño y la implementación. Esta información es importante para saber qué tanto afecta un cambio a un requisito y poder analizar el impacto en el costo, las funcionalidades del sistema, el esfuerzo de las personas implicadas, y si es factible o no realizarlo.

Según entrevistas realizadas a algunos Jefes de Proyecto de la facultad 2 de la UCI; se llegó a la conclusión de que en algunos proyectos no se utiliza ningún método para evaluar el impacto de los cambios, a pesar de que se tiene conocimiento de que estos existen. Los proyectos que se encuentran dentro del alcance del programa de mejora, los que pertenecen a las facultades 15, 5, 7 tampoco utilizan ningún método objetivo para evaluar el impacto de los cambios. Solamente se siguen los pasos de

Gestión de Configuración para procesar los cambios que pueden ocurrir a lo largo del ciclo de vida del proyecto.

Todo lo antes mencionado nos lleva a la siguiente **Situación Problemática**:

Actualmente en algunos de los proyectos que se llevan a cabo en la UCI no se aplica ningún método que evalúe de forma objetiva el impacto de los cambios a los requisitos, esto afecta el desarrollo exitoso de los proyectos, ya que no se tiene en cuenta el impacto de los mismos. Con la solicitud de un cambio llega la necesidad ineludible de estimar el alcance de este, el tamaño, la complejidad, la planeación de las actividades que lo implementan y el esfuerzo del personal para realizarlas. Es necesario el desarrollo de un método que apoye el proceso de evaluación del impacto de los cambios, que se vuelve un tema indispensable en el desarrollo y mantenimiento de un software.

Después de analizar la situación expuesta, se define como **Problema a Resolver**:

¿Cómo evaluar de forma objetiva el impacto de los cambios a los requisitos del software en la UCI?

A partir del problema científico se puede inferir como **Objeto de Estudio** para esta investigación: el proceso de control de cambios en los requisitos del software; y como **Campo de Acción** el impacto de los cambios en los requisitos del software.

La presente investigación tiene como **Objetivo General** desarrollar un método de evaluación de impacto que permita valorar objetivamente cuánto afecta un cambio a los requisitos del software.

Para dar cumplimiento al objetivo planteado se han trazado las siguientes **Tareas de la Investigación**:

- Estudio de los conceptos fundamentales vinculados al análisis de impacto de los cambios.
- Estudio de los métodos de evaluación de impacto de los cambios a los requisitos propuesto en la bibliografía internacional.
- Estudio de las técnicas que existen para el análisis de impacto de los cambios en el proceso de desarrollo de software para hacer un análisis crítico y determinar su utilización.
- Estudio de las matrices de trazabilidad, para analizar la trazabilidad de los requisitos.
- Estudio de las herramientas vinculadas a la administración de cambios.
- Estudio de los métodos de estimación para el análisis de costos y esfuerzo para realizar una

propuesta en el método.

- Propuesta de un procedimiento para el análisis de impacto de los cambios en el proceso de desarrollo del software.

Para la ejecución de las tareas se utilizan los siguientes **métodos teóricos**:

- **Método Analítico-Sintético:** Mediante este método se analiza la información existente sobre el análisis de impacto para sintetizar los elementos más útiles e importantes para desarrollar la investigación y proponer un método acertado.
- **Método Histórico-Lógico:** Se utiliza para determinar las tendencias actuales y la evolución de los diferentes métodos que se utilizan para analizar el impacto de los cambios.

También se utilizaron los siguientes **métodos empíricos**:

- **Entrevista:** Se entrevistaron a varios jefes de proyecto para investigar la aplicación de métodos de evaluación de impacto de los cambios en los proyectos y así llegar a conclusiones que se ponen de manifiesto en la investigación.
- **Encuesta:** Se aplicó una encuesta a un grupo de expertos para obtener el valor del Factor que se le da a los Elementos de Configuración de Software, y el rango de los atributos de cada uno de los ECS que permiten medir la complejidad en Simple, Medio o Complejo.

Se propone como **Idea a Defender** que con el desarrollo de un método para evaluar de forma objetiva el impacto de los cambios a los requisitos del software se logra determinar cuánto puede afectar dicho cambio las funcionalidades del sistema, además el impacto en el costo del proyecto y en el esfuerzo de las personas encargadas de desarrollarlo.

El documento está compuesto por tres capítulos, que incluyen todo lo relacionado con el trabajo investigativo:

En el Capítulo I: Fundamentación Teórica, se hace un estudio del estado del arte, se tratan los fundamentos teóricos y conceptos sobre los temas necesarios para sustentar la investigación. Entre los temas tratados se encuentran la definición del proceso de Gestión de Configuración, Gestión de Cambios, Impacto de los Cambios, entre otros.

En el Capítulo II: Propuesta de Solución, se elabora el método de evaluación de impacto de los cambios

a los requisitos de software, se explican en detalles cada una de las actividades que deben llevarse a cabo para aplicarlo y los encargados de realizarlas para que los cambios se gestionen correctamente.

Finalmente, en el Capítulo III: Validación de resultados, se aplica el método propuesto a un caso de estudio.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En este capítulo se tratan conceptos y elementos teóricos que forman los pilares de esta investigación. A lo largo de este se verán reflejadas una serie de definiciones detalladas extraídas de las bibliografías que aportan mayor conocimiento del tema. Principalmente se utilizan las definiciones dadas por CMMI (del inglés Capability Maturity Model Integration) por ser el modelo utilizado actualmente en la UCI para llevar a cabo el programa de mejora en los procesos de desarrollo del software y algunos conceptos dados por Pressman en su libro de Ingeniería de Software.

1.2 Conceptos fundamentales

El proceso de análisis de impacto de los cambios se encuentra dentro del área de procesos de Gestión de Configuración de Software, esta se encuentra dentro de la categoría de Soporte de CMMI, para su entendimiento es necesario tener un conocimiento básico de algunas definiciones que ayuden a comprender la importancia del proceso antes mencionado.

1.2.1 CMMI (Capability Maturity Model Integration)

CMMI es un modelo de referencia para el crecimiento de capacidades y madurez que se enfoca tanto en procesos de Administración como de Ingeniería de Sistema y Software. Con la instauración de este modelo en la UCI se espera alcanzar beneficios como:

- Calendarios y presupuestos predecibles en los proyectos.
- Mejora del ciclo de vida dentro del desarrollo de software.
- Mayor productividad.
- Mayor calidad de los productos y servicios que ofrece la universidad a sus clientes y por ende la satisfacción de los mismos.
- Mejorar la moral del personal que labora en el centro.[29]

1.2.2 Programa de Mejora

Actualmente en la UCI se lleva a cabo un Programa de Mejora en los procesos de desarrollo de software basado en el modelo CMMI y con la contratación de los servicios de consultoría del SIE Center (del inglés Software Industry Excellence Center) del Tecnológico de Monterrey. El Programa de Mejora está encaminado a que la Universidad alcance en el 2010 una certificación internacional del nivel 2 del modelo CMMI. Hecho que la convertiría en la primera empresa cubana certificada con este modelo.[29]

1.2.3 Gestión de Configuración del Software (GCS)

Es el proceso de identificar y definir los elementos de configuración de un sistema, controlar los cambios de estos a través de su ciclo de vida, registrar y reportar el estado de los elementos y las solicitudes de cambios y verificar que los mismos estén completos y correctos. [2]

Es una disciplina que aplica supervisión y dirección técnica y administrativa a:

- Identificar y documentar las características funcionales y físicas de un elemento de configuración.
- Controlar los cambios a esas características.
- Registrar y reportar el procedimiento de los cambios y el estado de implementación.
- Verificar la conformidad con los requisitos especificados.[3]

En términos generales se podría decir que la GCS es una actividad de autoprotección que se aplica durante el proceso de desarrollo del software, ya que el cambio es algo inevitable y que puede surgir en cualquier momento, las actividades de esta pueden identificar y controlar el cambio, así como garantizar que este sea implementado correctamente e informar a todos aquellos que puedan estar interesados.

1.2.4 Elemento de Configuración del Software (ECS)

Cada uno de los componentes de la configuración del software se llama Elemento de Configuración del Software (ECS).

Según el modelo CMMI un elemento de configuración es una agregación de productos de trabajo designado para la administración de la configuración y tratado como una entidad individual en el proceso de administración de la configuración.[4]

1.2.5 Proceso de cambio

Para poner en perspectiva el análisis del impacto de los cambios, se necesita entender primero el proceso de cambio. Madhaji [5] define el proceso de cambio como:

- Identificar la necesidad de hacer un cambio a un elemento del ambiente.
- Definir el cambio adecuado relacionado con el conocimiento sobre el elemento.
- Evaluar el impacto del cambio en otros elementos del ambiente.
- Seleccionar o construir un método para procesar el cambio.
- Realizar los cambios a todos los elementos y que sus inter-dependencias sean resueltas satisfactoriamente.
- Guardar los detalles de los cambios para referencias futuras, y liberar el elemento cambiado detrás del ambiente.

Un problema importante a la hora de realizar un cambio es saber todos los factores que se afectan con el mismo y las consecuencias de implementarlo.

1.2.6 Control de cambios

La primera Ley de la ingeniería de sistemas establece: "Sin importar en qué momento del ciclo de vida del sistema nos encontremos, el sistema cambiará y el deseo de cambiarlo persistirá a lo largo de todo el ciclo de vida".

Existen cuatro fuentes fundamentales del cambio: [24]

- Nuevos negocios o condiciones comerciales que dictan los cambios en los requisitos del producto o en las normas comerciales.
- Nuevas necesidades del cliente que demandan la modificación de los datos producidos por sistemas de información, funcionalidades entregadas por productos o servicios entregados por un sistema basado en computadora.
- Reorganización, crecimiento o reducción del negocio que provoca cambios en las prioridades del proyecto o en la estructura del equipo de ingeniería del software.

- Restricciones presupuestarias o de planificación que provocan una redefinición del sistema o producto.

El control de cambio es un concepto relacionado con la metodología de desarrollo de software. Se trata de hacer el desarrollo de forma evolutiva, mediante cambios sucesivos realizados de una manera disciplinada. En un gran proyecto de desarrollo de software, el cambio no controlado lleva rápidamente al caos. La realidad del control de cambio en un contexto moderno de ingeniería de software ha sido resumida por James Bach: "El control de cambios es vital pero la fuerza que lo hacen necesario también lo hacen molesto. Nos preocupamos por el cambio porque una pequeña perturbación en el código puede crear un gran fallo en el producto, pero también puede reparar un gran fallo o habilitar excelentes capacidades nuevas. Nos preocupamos por el cambio, porque un desarrollador pícaro puede hacer fracasar el proyecto, sin embargo, las brillantes ideas nacidas en la mente de esos pícaros, y un pesado proceso de control de cambios puede disuadirle de hacer un trabajo creativo".[16]

1.2.7 Línea base de configuración

Una línea base es una agrupación de elementos de configuración que comparten un mismo estado. Los estados pueden ser cualquiera que le sirva a la organización para distinguirlos y trabajar con ellos:

- Oficiales
- Versión alfa, beta, liberación
- Referencia histórica

La IEEE (Estándar IEEE 610.12-1990) define una línea base como: Una especificación o producto que se ha revisado formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en adelante sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos formales de control de cambios.[25]

Las líneas base más los cambios aprobados a estas líneas base, constituyen la información de configuración actual. [6]

La Figura 1 muestra las líneas base definidas en el proceso de Administración de Configuración de la UCI; cada documento que resulta del proceso de software, se convierte en Línea Base, una vez que dicho documento se haya revisado, corregido y aprobado.

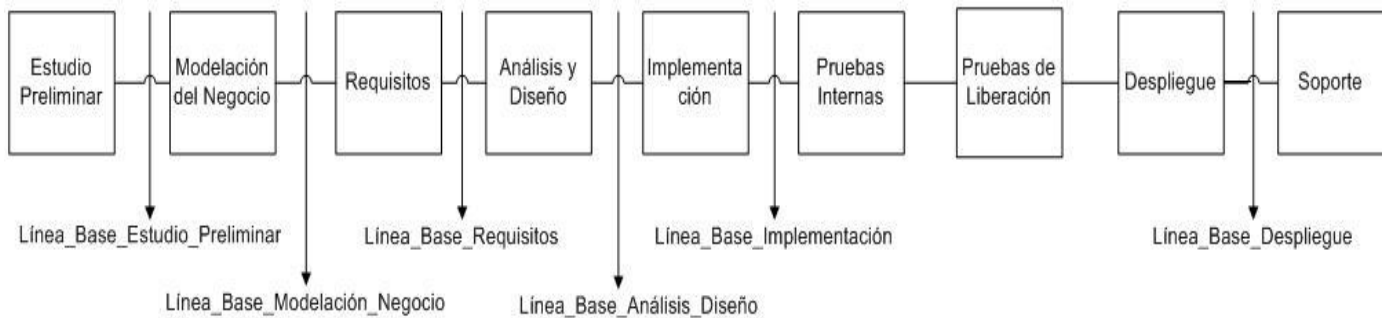


Figura 1: Líneas Base

1.2.8 Análisis de impacto

Un impacto es el efecto o impresión de una cosa en otra; puede pensarse como las consecuencias de un cambio. El análisis de impacto se usa para determinar el alcance de las demandas de cambio como una base para el planeamiento exacto de recursos y horarios, y para confirmar la justificación de beneficios/costos.

El análisis de impacto de los cambios estima lo que se afectará en el software y la documentación relacionada si un cambio propuesto es realizado, se define como el proceso de evaluar los efectos en otros componentes del sistema que resulta del cambio propuesto, determina el alcance y la complejidad del mismo.

1.3 Roles asociados al Comité de Control de Cambios (CCC)

El Comité de Control de Cambios está compuesto por un grupo de personas que desempeñan un papel importante dentro del área de procesos de Administración de la Configuración y se definen por defecto para esta área aunque pueden incluirse otros según las necesidades del proyecto.

Los integrantes del comité tienen las siguientes responsabilidades:

- Autorizar la creación de líneas base.
- Asignar responsables de resolver las no conformidades.
- Analizar el impacto de las Solicitudes de Cambios.
- Tomar una decisión acerca de las Solicitudes de Cambios: aprobarlas, rechazarlas o posponerlas.

- Asignar responsables a la implementación de los Cambios aprobados.

Las Solicitudes de Cambio son evaluadas entre todos los miembros de este comité en la Reunión de Revisión desarrollada para tales fines, acordándose el modo en que se deberá proceder.

Los miembros permanentes del CCC son:

- Jefe de proyecto
- Arquitecto de software
- Administrador de configuración
- Administrador de calidad

1.4 Requisitos

Según el modelo CMMI, un requisito es: [7]

- Una condición o capacidad necesitada (requerida) por el usuario para resolver un problema o alcanzar un objetivo.
- Una condición o capacidad que debe ser cumplida o poseída por un producto o por un componente de producto para satisfacer un contrato, estándar, especificación, u otro documento impuesto formalmente.
- Una representación documentada de una condición o capacidad tal y como en los incisos anteriores.

1.4.1 Clasificación de los requisitos

CMMI clasifica los requisitos como se muestra a continuación: [7]

- **Requisitos del Cliente**

Resultado de la obtención, consolidación y resolución de conflictos entre las necesidades, expectativas, limitantes e interfaces de los involucrados relevantes del producto, presentados de forma que sea aceptable por el cliente.

- **Requisitos del Producto**

Refinamiento de los requisitos del cliente al lenguaje de los desarrolladores, haciendo los requisitos implícitos en requisitos derivados explícitos. El desarrollador utiliza los requisitos del producto para guiar el diseño y construir el producto.

➤ **Requisitos de Componentes de Productos**

Una especificación completa del componente del producto incluyendo la forma, función, desempeño y cualquier otro requisito.

➤ **Requisitos No-técnicos**

Provisiones contractuales, compromiso, condiciones, y términos que afectan cómo los productos o servicios serán adquiridos.

➤ **Requisitos Técnicos**

Propiedades (atributos) de productos y servicios, que serán adquiridos o desarrollados.

1.4.2. Trazabilidad de requisitos.

La trazabilidad de requisitos se define como la habilidad para describir y seguir la vida de un requisito en ambos sentidos, hacia sus orígenes o hacia su implementación, a través de todas las especificaciones generadas durante el proceso de desarrollo de software. Esta es clave para conseguir una exitosa gestión de los mismos ya que permite conocer qué elementos se ven afectados cuando ocurre un cambio en algún otro que tenga relación con el primero, o sea, cuando algún requisito o cualquier otro elemento traceable es modificado, todas las relaciones asociadas a ese requisito se convierten en "sospechosas", en ese caso se deben revisar los cambios y determinar si los elementos asociados deben ser cambiados también.

1.4.2.1 Trazabilidad bidireccional

CMMI define la trazabilidad bidireccional como:

Trazabilidad bidireccional: Es una asociación entre dos o más entidades lógicas, definida y congruente en cualquier dirección (es decir, desde y hacia la entidad).[8]

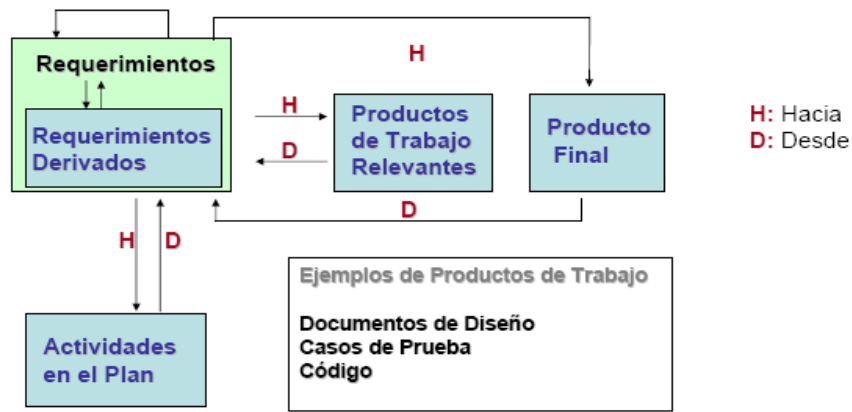


Figura 2: Trazabilidad bidireccional en CMMI

1.4.3 Matrices de trazabilidad

Para facilitar el trabajo de determinar las relaciones entre los requisitos y el seguimiento de los mismos, se utilizan matrices de trazabilidad; cada una de estas identifica los requisitos relacionados con uno o más aspectos del sistema o su entorno. Estas matrices se generan de forma automática con herramientas que se mencionarán más adelante.

1.4.4 Guía de trazabilidad

Como parte del programa de mejoras que se lleva a cabo en la UCI, guiado por el modelo CMMI, se desarrolló una guía de trazabilidad. Esta define los siguientes elementos a trazar y sus ubicaciones: [9]

Elemento a trazar	Ubicación	Descripción
Objetivos.	Proyecto Técnico (PT)	Principales necesidades de los involucrados y usuarios.
Reglas del Negocio	Documento Reglas del Negocio.(RN)	Términos necesarios para comprender el negocio.
Requisitos	Especificación de Requisitos de Software	Documento de descripción de los requisitos funcionales y no funcionales del sistema

Casos de Uso del sistema (CUS)	Especificación de Casos de Uso	Documento de descripción de los casos de uso, sus escenarios y actores.
Entidades del Negocio	Diccionario de Datos (DD)	Documento que describe las entidades del negocio.
Reportes	Salidas del Sistema (SS)	Documento donde se describen los reportes del sistema y sus elementos principales de interfaz
Diagramas de Clases.(DC)	Modelo de Análisis (si el proyecto realiza clases de análisis) Modelo de Diseño (MD)	Documento donde se describen la relación entre las clases del sistema y una descripción de cada una.
Entidades de la Base de Datos (tablas)	Diagrama de Entidad- Relación. (DER) en la Herramienta	Diagrama donde se describe la relación entre las entidades de la base de datos (Tablas).
Diagrama de Interacción. (DI)	Herramienta	
Diagrama de Despliegue. (DD)	Modelo de Despliegue	Documento donde se incluye el Diagrama de Despliegue y la descripción de cada uno de los elementos del diagrama
Casos de prueba. (CP)	Diseño de Casos de Prueba	Documento que describe los escenarios del caso de uso a probar y los pasos para obtener el resultado que se espera.

Clases de Implementación	Código Fuente	
--------------------------	---------------	--

Tabla 1: Relación entre los ECS y sus ubicaciones.

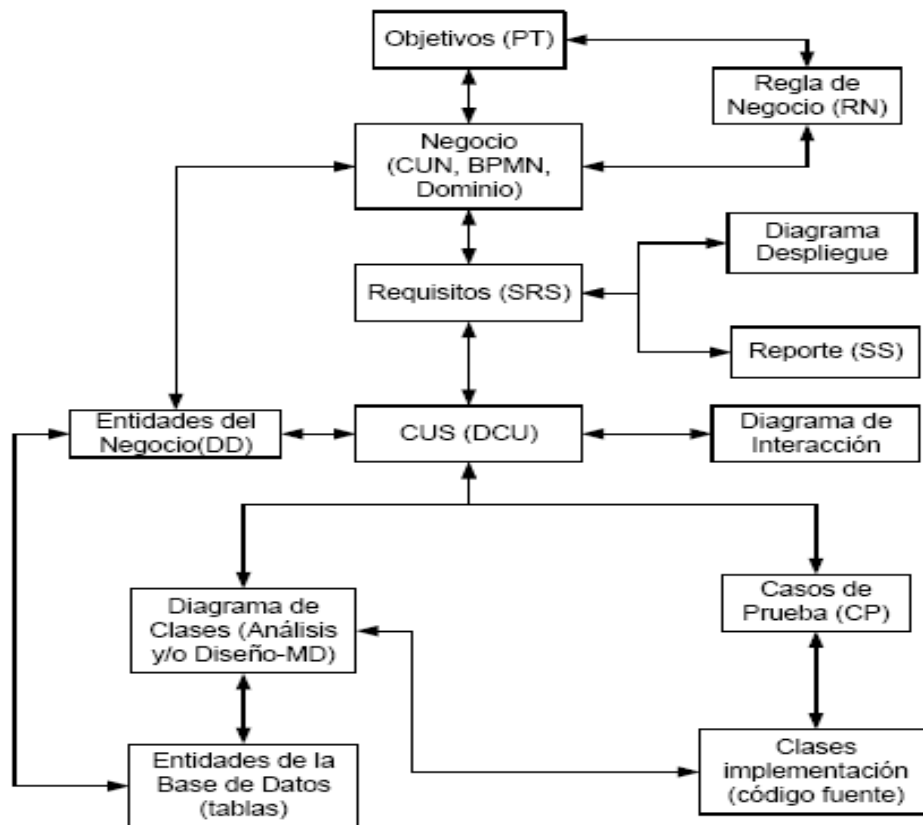


Figura 3: Relación entre los elementos a trazar en el proyecto

1.4.5 Información de trazabilidad

Esta es la información clave que se utiliza para determinar la magnitud del cambio ya que indica las relaciones potenciales entre los ECS y coleccionando estas es posible identificar los caminos de propagación e impacto dentro de un sistema.

1.5. Herramientas para la Administración de Requisitos

Estas herramientas surgen con el objetivo de apoyar cada una de las distintas etapas del ciclo de vida de

desarrollo. A continuación se hace una valoración de algunas herramientas para apoyar la trazabilidad de requisitos, estas están actualmente aceptadas en la UCI y son las que se utilizan en los proyectos como parte del programa de mejoras que se lleva a cabo. Cabe aclarar que para aplicar el método de evaluación de impacto que se propone se pueden utilizar cualquiera de estas herramientas.

1.5.1 OSRMT (Open Source Requirement Management Tool)

Es una herramienta Open Source para administración de requisitos desarrollada en Java que permite llevar el ciclo completo de vida de desarrollo de software y la gestión de varios tipos de requisitos (requisitos, casos de uso, casos de prueba, etc.); garantiza la trazabilidad entre los mismos y los visualiza de forma jerárquica, es una herramienta multiplataforma.

Tiene como desventaja que su documentación es pobre. Como ha sido desarrollada por una sola persona, no cuenta con un soporte empresarial y no es posible generar automáticamente un documento de requisitos para entregar al cliente.

The screenshot shows a window titled "Columns: Traceability". It contains two dropdown menus: "Trace From" set to "Requirement" and "Trace To" set to "Requirement". The "Trace Type" dropdown is set to "Traceability Matrix". There is an "Apply" button. Below these controls is a table with the following data:

-->	Requirement ...	Requirement Six	Requirement ...	Requiremene... new one	Req Four	Requirement ...	Req Twos	2222
Requirement Eight			X		X			
Requirement Six								
Requirement Eleven								
Requiremenet Ten				X				
new one								
Req Four								
Requirement Nine								
Req Twos								
2222			X			X		
Req Five		X			X			
1111								

Figura 4: Matriz de trazabilidad en OSRMT

1.5.2 Rational RequisitePro

Es una herramienta que ofrece Rational Software para tener un mayor control sobre los requisitos planteados por el usuario y todos aquellos requisitos técnicos o nuevos requisitos de usuario que surjan durante el ciclo de vida del proyecto.

Con esta los requisitos se encuentran documentados bajo un esquema organizado de documentos; estos esquemas, cumplen completamente con los estándares requeridos por IEEE (Instituto de Ingenieros Electricistas y Electrónicos), ISO (Organización Internacional para la Estandarización), SEI (del inglés Software Engineering Institute), CMMI y por el Rational Unified Process.

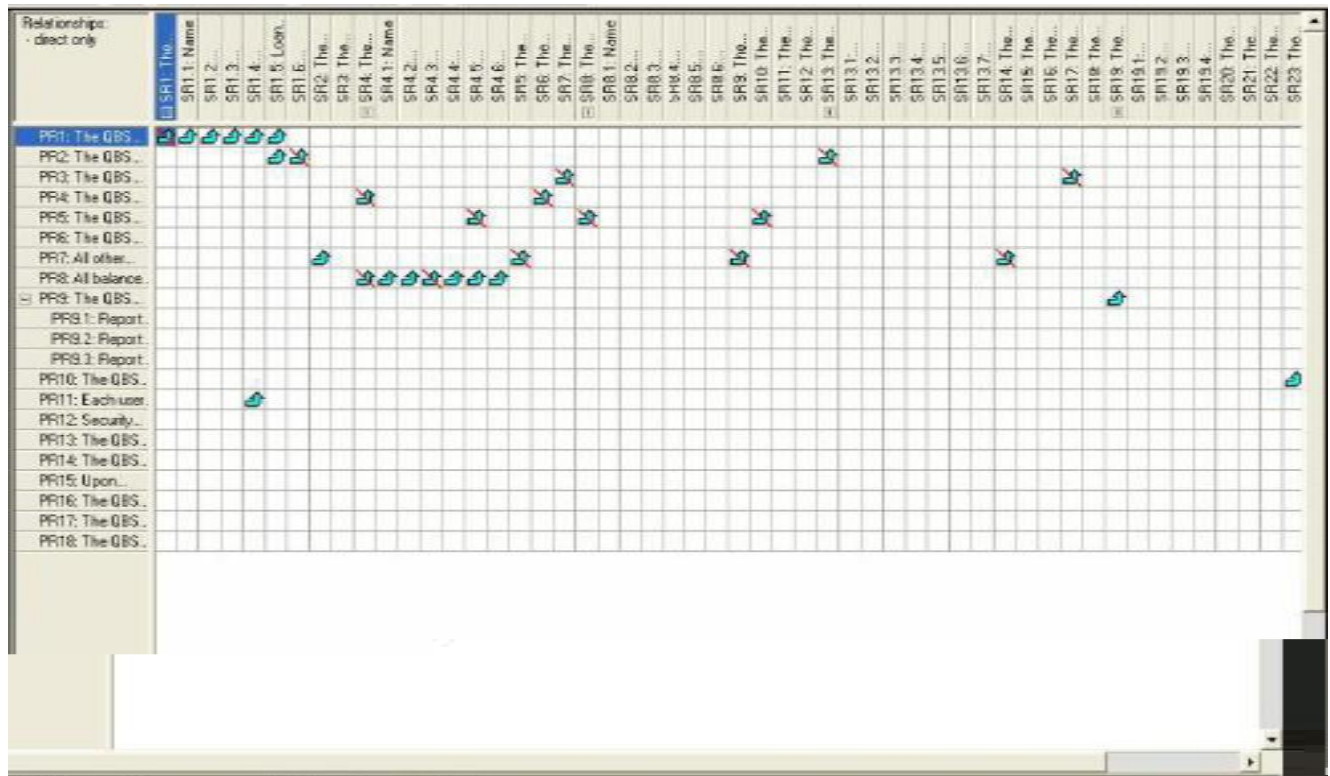


Figura 5: Matriz de trazabilidad en Rational RequisitePro

1.5.3 Enterprise Architect

Es una de las herramientas más completas ya que soporta diagramas UML, genera documentos y reportes personalizados, además trabaja con frameworks de distintos lenguajes de desarrollo como Java y .Net.

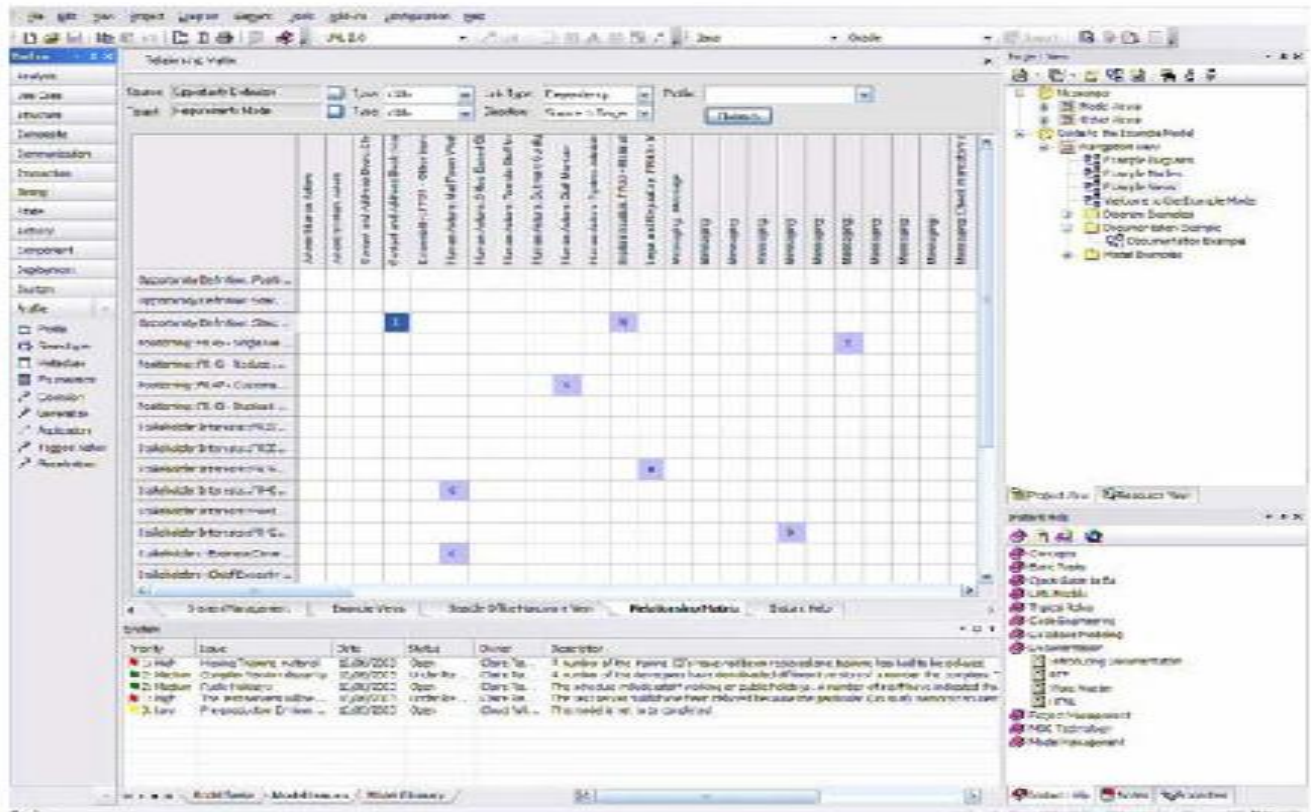


Figura 6: Matriz de trazabilidad con el Enterprise Architect.

1.6 Métodos de estimación

Existen varios métodos de estimación de esfuerzo y costo, que se utilizan para medir el tamaño del software, o para estimar los costos del proyecto, algunos de estos métodos son:

➤ COCOMO II

COCOMO II es un modelo que permite estimar el costo, esfuerzo y tiempo cuando se planifica una nueva actividad de desarrollo de software. Está asociado a los ciclos de vida modernos. El modelo original COCOMO ha tenido mucho éxito pero no puede emplearse con las prácticas de desarrollo de software más recientes tan bien como con las prácticas tradicionales. Apunta hacia los proyectos de software de los 90 y de la primera década del 2000, y continuará evolucionando durante los próximos años. [18]

COCOMO II es la aplicación de ecuaciones matemáticas sobre los puntos de función sin ajustar o la cantidad de líneas de código estimados para un proyecto. Está orientado a la magnitud del producto final,

estas ecuaciones son ponderadas por una serie de factores de costo que influyen en el cálculo del esfuerzo que se necesita para llevar a cabo un proyecto de software.

La estimación por COCOMO II , resulta muy útil para estimar un proyecto en forma global, cuando se tiene un conjunto de Casos de Uso bastante amplio (aproximadamente 50) y con escaso nivel de detalle. [27]

➤ **Análisis por puntos de función(FPA)**

Es un método utilizado en ingeniería de software para medir el tamaño del software. Esta métrica se define como una métrica funcional, dado que se enfoca a la funcionalidad que el software proporciona al usuario.

Se basa principalmente en la identificación de los componentes del sistema informático partiendo de transacciones y grupos de datos lógicos que son relevantes para el usuario en su negocio. A cada uno de estos componentes se le asigna un número de puntos por función, basándose en el tipo de componente y su complejidad; la sumatoria de esto da los puntos de función sin ajustar. El ajuste es un paso final, basándose en las características generales de todo el sistema informático que se está contando. [17]

➤ **Puntos de caso de uso**

El método de Puntos de Casos de Uso es un método de estimación y cálculo de tamaño del software basado en cuentas hechas sobre los casos de uso para un sistema de software. [19]

Surgió como una mejora al método de puntos de función pero basando las estimaciones en el modelo de casos de uso, producto del análisis de requisitos.

Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de pesos a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. [26]

Para la realización del método de análisis de impacto de los cambios que se propone como parte de la investigación se estudiaron los métodos de estimación planteados anteriormente y se llegó a la conclusión de que se podía efectuar una analogía con el método de estimación Puntos de caso de uso, este es una mejora del método Análisis por punto de función (FPA), trabaja con los mismos aspectos de este e incorpora nuevos factores que lo hacen mucho más completo como por ejemplo el Factor Ambiente. No se tomaron elementos del método COCOMO II ni del método Análisis por punto de función (FPA) por sus desventajas. COCOMO II está basado en estimaciones y parámetros que pueden ser vistos de distinta manera por los analistas que lo utilicen, además puede ser un poco complicado a la hora de aplicarlo ya

que cuenta con una gran cantidad de fórmulas, además no es recomendable para proyectos pequeños. El método Análisis por puntos de función no es muy preciso cuando se trata de proyectos pequeños y en ocasiones resulta poco confiable, además en proyectos que se desarrollan con presupuestos ajustados necesita una dedicación adicional por parte del personal. La estimación a partir de Puntos de Función ajustados es difícil de realizar si no se cuenta con una base histórica de proyectos que provea los coeficientes de conversión.

El método Puntos de casos de uso calcula el tamaño del software basándose en Casos de Uso y Actores, el método de análisis de impacto de los cambios que se propone calcula el tamaño del cambio siguiendo esta misma lógica y con la cantidad de ECS afectados, de este método se tomaron algunos elementos como el cálculo del Factor de Peso, la fórmula para calcular los ECS desajustados y ajustados y el cálculo del esfuerzo, además se le añadieron nuevas actividades como el cálculo del costo, se analizaron los riesgos en la toma de decisiones y se le asignó una complejidad a cada uno de los ECS teniendo en cuenta sus atributos más importantes, según esta complejidad se les dio un Factor el cual se utiliza en el cálculo del Factor de Peso.

1.7 Análisis de impacto como proceso

En el proceso de análisis de impacto se llevan a cabo fases individuales, estas son:

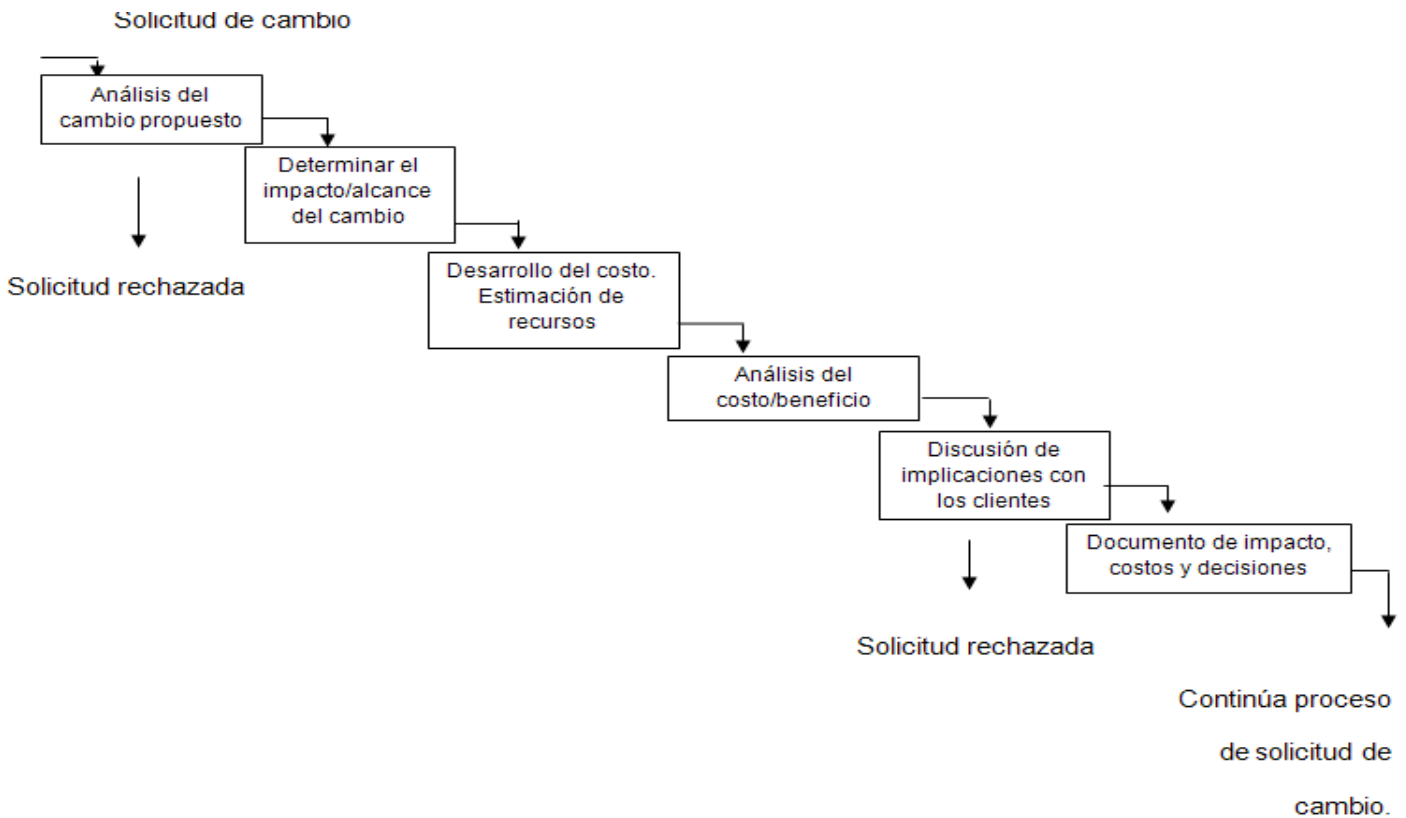


Figura 7: Proceso de análisis de impacto

Estas fases se describen a continuación con más detalles:

- **Análisis del cambio propuesto:** El desarrollador debe asegurarse de que se entiende el cambio propuesto y los componentes del sistema afectados. Se debe verificar el cambio para su validez, con vistas a entender de forma profunda los objetivos del cambio propuesto.
- **Determinar el impacto/alcance del cambio:** Esta fase involucra el uso de una técnica de análisis para identificar el impacto en los artefactos de todas las fases del ciclo de vida de desarrollo.
- **Desarrollo del costo y estimaciones de los recursos:** Durante esta fase se aplican métodos de estimación como modelos de costo algorítmico o el análisis estático que se usan para intentar calcular el costo potencial del cambio.
- **Análisis del costo/beneficio:** Durante esta fase se hace una comparación entre el peso de los costos estimados del cambio contra los beneficios esperados para determinar si el cambio sería

eficaz. El dato usado para este análisis se extrae de la información en la forma de petición de cambio.

- **Discusión de las implicaciones con los clientes:** Una vez que se han identificado los costos relativos y beneficios involucrados, la decisión para rechazar o proceder con el cambio puede tomarse. Esto debe hacerse después de la consulta con la organización del cliente por medio de una discusión informal o vía una tabla de la revisión formal.
- **El documento de impacto, costos y decisiones:** Se debe registrar el efecto potencial y el costo probable del cambio en todos los artefactos de desarrollo calculado en las fases anteriores. Esto puede incluir artefactos que relacionan al software, hardware, datos, documentación y factores humanos (la ayuda en línea, los dispositivos de entrada etc.). Tal documentación debe hacerse aun cuando se ha tomado la decisión de rechazar el cambio debido a que esta información puede ser útil para evaluar los cambios futuros.

Cuando se va a desarrollar un método para evaluar el impacto de los cambios, se pueden tener en cuenta estas fases para tenerlas como una guía. El método que se va a desarrollar para dar cumplimiento a los objetivos propuestos tiene en cuenta estas fases para tener un trabajo organizado y obtener mejores resultados.

1.8 Clasificación de los métodos de análisis de impacto

Se han propuesto varios métodos que intentan analizar el impacto del cambio en las diferentes fases de desarrollo del sistema. Estos pueden ser clasificados en los tipos principales siguientes [10]:

- **Los métodos semánticos:** El análisis de gráficos, tablas, el código etc. para extraer la información con respecto a las dependencias entre los componentes del sistema.
- **Los métodos heurísticos:** El uso de reglas basadas en métodos de Inteligencia Artificial para predecir el impacto potencial de un cambio propuesto.
- **Los métodos estocásticos:** El uso de probabilidades de propagación de impacto para determinar el probable efecto de un cambio.
- **Los métodos exhaustivos:** Técnicas de prerregistro de trazabilidad basadas en la fuerza bruta.
- **Los métodos híbridos:** Métodos que consisten en una combinación de los cuatro métodos

anteriores.

El objetivo de todos estos métodos es intentar extraer, guardar y analizar las relaciones de trazabilidad de la representación de un sistema. Las relaciones que existen entre los componentes de un sistema son a menudo implícitas en la representación del sistema y es el trabajo del análisis de impacto ayudar a hacer estas relaciones explícitas para facilitar el análisis. [17]

1.9. Métodos de análisis de impacto de los cambios

Los métodos de análisis de impacto han sido desarrollados para identificar las consecuencias de realizar cambios a los artefactos del sistema y para apoyar la decisión de realizarlos. A continuación, exponemos algunos de estos métodos.

1.9.1 Método de análisis de los cambios en software orientado a objetos

Este método propone cinco algoritmos que trabajan juntos para analizar el impacto que los cambios propuestos pueden tener en el sistema. Se puede incluir dentro de los métodos semánticos, ya que basa su funcionamiento en el análisis del código.

Estos algoritmos verifican cada clase que ha sido propuesta para ser cambiada, esto es llamado cambio de clase, luego verifica las clases relacionadas con la que se desea cambiar para ver si alguna se ve afectada. En la figura 8 se muestran las relaciones entre los algoritmos. [12]

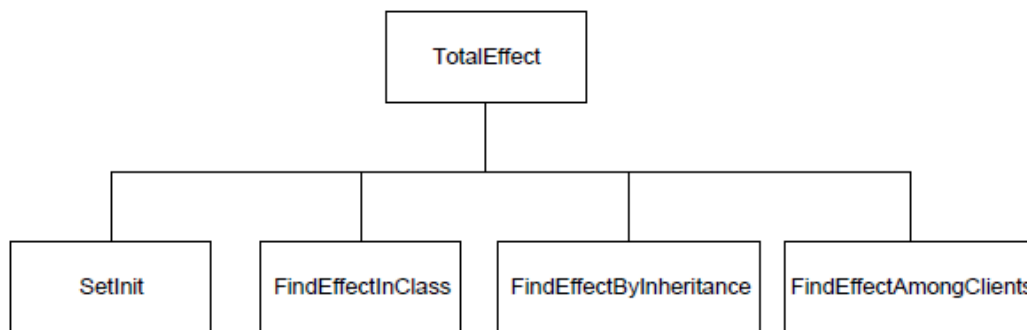


Figura 8: Relaciones entre los algoritmos de análisis de impacto de los cambios en el software orientado a objetos

Cuando los ingenieros desean realizar un cambio propuesto, es necesario especificar qué partes del sistema va a cambiar por la especificación de un conjunto de criterios. Después de que estos se han especificado, los algoritmos calculan el impacto del cambio para cada criterio, luego encuentran todas las funciones que pueden ser impactadas. De acuerdo con los criterios especificados, los algoritmos primero calculan el impacto que el cambio puede tener dentro de las clases, después calculan todos los miembros afectados en ella, y examinan las relaciones entre los objetos del sistema.

De acuerdo con las características de la herencia y la encapsulación, los algoritmos calculan los efectos del cambio siguiendo diferentes tipos de relaciones en el sistema, estos continúan hasta que no hallan nuevas clases que se vean afectadas con este.

Este método no es óptimo para el objetivo planteado porque solo está orientado a los elementos del software, y no tiene en cuenta otros que también son importantes, por lo cual no se puede hacer un análisis del impacto del cambio en todos los ECS que se ven afectados.

1.9.2 Método diseño relacional

Este es un método inicial basado en un conjunto estructurado de conocimientos de arquitectura de decisión que fue desarrollado para utilizar las decisiones de diseño y predecir el impacto del cambio en un nivel de la arquitectura [13]. Este se amplió mediante la exploración de las dependencias entre las decisiones sobre el cambio de arquitectura y los elementos arquitectónicos, y luego utiliza el cálculo de probabilidades para cuantificar estas dependencias.

Este método es aplicable durante y después del desarrollo del diseño de la arquitectura. Cuando se toma la decisión de realizar el cambio, sirve de apoyo para el análisis de las causas subyacentes del mismo en diferentes elementos de la arquitectura, incluyendo los componentes de diseño, modelos de datos y componentes de implementación. Se centra en los dominios del sistema en el que la identificación de los cambios y el proceso de evaluación son generalmente iniciados en etapas posteriores al diseño de la arquitectura [14].

Es desarrollado y aplicado en el ámbito de los tradicionales sistemas de software donde el diseño de la arquitectura y la modelación de procesos son normalmente precedidos de los requisitos del proceso de negocio. [15]

Al igual que el método anterior este es muy específico, solo se enfoca en los elementos de la arquitectura

del software y no realiza un análisis de todos los ECS, por lo que no es el más apropiado para darle solución al problema que se plantea.

1.9.3 Procedimiento para la Toma de decisiones en la Gestión de Cambios

Este es un método elaborado por dos estudiantes para su tesis de diplomado en el año 2008, en la figura se muestra los pasos para llevar a cabo el mismo. [23]

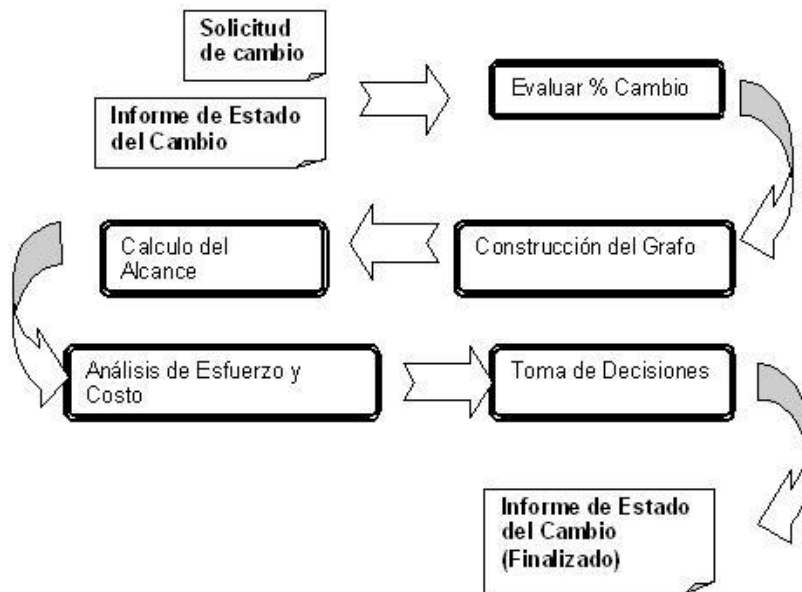


Figura 9: Procedimiento para la Toma de decisiones en la Gestión de Cambios

Este es un híbrido, es decir, una combinación de otros métodos. Después de hacer un análisis se llegó a la conclusión de que no se ajusta a las necesidades actuales de los proyectos que se encuentran dentro del programa de mejora que se lleva a cabo en la UCI por diversas razones, una de ellas es que solamente está enfocado a los ECS que propone RUP y no tiene en cuenta los que son resultado de la implementación de los 7 procesos del nivel 2 de CMMI, que se deben tener cuenta en todos los proyectos que trabajan dentro del programa de mejora de la universidad. Esto no quiere decir que no se le puedan agregar otros ECS, sin embargo, la complejidad técnica que se les asigna a cada uno de ellos se deja a criterio de expertos, no se explica ninguna forma objetiva para obtener estos valores, lo que trae como

consecuencia que si en un futuro se quieren agregar otros ECS se debe realizar un estudio para ver cuál sería el peso del nuevo ECS, esto sería mucho más fácil si se tuviera una forma de calcularlo. Lo mismo ocurre con el peso que se le asigna a las relaciones entre los ECS, en el método se definen un grupo de relaciones y un peso asignado a cada una de ellas, pero no se explica de qué forma se obtuvieron esos valores. Por estas razones no se escogió este procedimiento como base para cumplir el objetivo de la presente investigación.

1.10 Ventajas del análisis de impacto

El análisis de impacto puede ayudar a determinar el alcance y la magnitud de un cambio propuesto y de esta forma desarrollar las estimaciones con respecto al esfuerzo del personal, los recursos requeridos y el costo potencial de los cambios, también ayuda en la comunicación de la complejidad de un cambio solicitado a los clientes o usuarios terminales que lo propusieron y facilita la toma de decisiones para decidir si es factible o no realizarlo; esto es útil al intentar demostrar que el cambio puede no merecer el costo requerido.

Finalmente, este tipo de análisis reduce la cantidad de mantenimiento correctivo que resulta de introducir un cambio, previniendo los impactos grandes y las consecuencias de llevarlo a cabo.

1.11 Conclusiones

El estudio de los principales conceptos relacionados con el análisis de impacto de los cambios muestra la importancia que tiene dentro del proceso de desarrollo de software realizar una buena gestión de los mismos. Los cambios deben ser tratados de manera formal y consciente, no se deben realizar a la ligera, se debe analizar, aprobar, desarrollar, implementar y documentar. Debido a la gran cantidad de información y la complejidad de las relaciones que se establecen entre los ECS, rastrear y evaluar el efecto de un cambio puede ser costoso ya sea por el tiempo que consume o por los errores que produce. La proposición de un método que defina las actividades, personas y tareas involucradas en este proceso contribuiría a una mejor práctica de la Gestión de Configuración y por tanto un mayor acercamiento a los estándares de calidad.

Capítulo 2: Propuesta de Solución

2.1 Introducción

Los cambios son inevitables en el desarrollo del software. Los requisitos cambian y evolucionan a medida que van formulándose. Estos cambios afectan las funcionalidades del sistema y las metas comerciales de la organización para la que el software se desarrolla, por estas razones es importante que sean tratados cuidadosamente.

En este capítulo se presenta en detalles un método para evaluar el impacto de los cambios a los requisitos del producto de software. Para el desarrollo del mismo es necesario saber identificar los ECS y conocer cada uno de estos; estos conceptos son tratados en el presente capítulo.

2.2 Identificación de los Elementos de Configuración de Software

La identificación de los ECS que se realice en el proyecto es una actividad muy importante, esta se define como:

Identificación de los Elementos de Configuración de Software: consiste en identificar y asignar nombres significativos y consistentes a todos y cada uno de los elementos que forman parte del producto de software, en cada fase de su desarrollo, es decir, a cada uno de los ECS. [20]

2.3. Lista de los Elementos de Configuración de Software

Los elementos de configuración que se listan a continuación son resultado del programa de mejora que siguen los proyectos y que se deben manejar por defecto. Estos son los ECS que se van a tener en cuenta al aplicar el método que se va a proponer. Estos se encuentran divididos por proceso al que pertenecen. [21]

No	Nombre	Descripción	Responsable(s)
Administración de Requisitos (REQM)			
	0103_CritPvReq	Criterios para definir proveedores válidos de requisitos.	Analista(s)
	0104_DD	Diccionario de Datos	Analista(s)
	0106_RgInconsistencias	0106_Registro de Revisiones de Inconsistencias.xls	Analista(s)
	0107_SS	Salidas del sistema	Analista(s)
	0108_EvalCU	Evaluación de Casos de Uso.	Analista(s)
	0109_MNCU	Modelo de negocio con CU	Analista(s)
	0110_MTPv	Matriz de disponibilidad de proveedores	Analista(s) Jefe de Proyecto Planificador
	0111_InfDiagnóstico	Informe Diagnóstico (Opcional)	Arquitecto de información
	0112_MDom	Modelo de Dominio	Analista(s)
	0113_ERS	Especificación de Requisitos de Software	Analista(s)
	0114_ECU	Especificación de casos de uso	Analista(s)
	0115_MPN-BPM	Modelo de Procesos de	Analista(s)

		Negocio con BPM	
0116_RNeg		Reglas de Negocio	Analista(s)
0118_CRC		Criterios para validar requisitos del cliente.	Analista(s)
0121_MDI		Modelo de Diseño	Analistas
0122_CPb		Casos de Prueba	Analistas
0123_MDE		Modelo de Despliegue	Arquitecto de Software
0124_LevInfo		Levantamiento de Información	Arquitecto de Información
0125_OrgzacionInfo		Organización de la Información	Arquitecto de Información
0126_DisenArq		Diseño de la Arquitectura	Arquitecto de Información
<i>[Definido por el Estándar de codificación del Proyecto]</i>		Código fuente (elaborado por los programadores del proyecto)	Desarrollador
<i>[Definido por el Estándar de codificación del Proyecto]</i>		Archivos de configuración (de las herramientas de autoconstrucción)	Desarrollador
<i>[Definido por el Estándar de codificación del Proyecto]</i>		Archivos de contenido (archivos de internacionalización)	Desarrollador Especialista funcional
<i>[Definido por el Estándar de codificación del Proyecto]</i>		Base de Datos (scripts, configuración)	Diseñador de la Base de Datos

	<i>[Definido por el Estándar de codificación del Proyecto]</i>	Componentes externos (marcos de trabajo que se modifiquen)	Desarrollador
	DER (opcional)	Diagrama Entidad Relación	Analistas
	DInt (opcional)	Diagramas de interacción	Analistas
	5107_GlsTermino	Glosario de términos	Analista(s)
Planeación del Proyecto (PP)			
	0204_ActaTerminación	Acta de Terminación	Jefe de Proyecto
	0205_PlanPrb	Plan de Pruebas	Jefe de Proyecto Planificador Administrador de la calidad
	0206_PlanDSw	Plan de Desarrollo de Software	Jefe de Proyecto Planificador
	0208_PryTec	Proyecto Técnico	Jefe de Proyecto
	0315_PlanMed	Plan de Mediciones	Jefe de Proyecto Planificador Administrador de la calidad
Monitoreo y Control del Proyecto (PMC)			
	0224_PlanRgMonitoreo	Planes y Registro de Monitoreo	Jefe de Proyecto Planificador
Administración de Acuerdos con Proveedores (SAM)			

0231_CritSelecProv	Criterios de Selección de Proveedores	Jefe de Proyecto
0232_ListaAdq	Lista de tipos de adquisición	Jefe de Proyecto
0233_RpEstadoProv	Reporte de estado del proveedor	Jefe de Proyecto
0239_AcuerdoProv	Acuerdo con el Proveedor	Jefe de Proyecto
0240_PlanTransferencia	Plan de transferencia	Jefe de Proyecto

Tabla 2: Elementos de Configuración de Software.

En CMMI no solo se tienen en cuenta los ECS que cambian sino también aquellos que aportan información y sirven como base para el desarrollo posterior del proyecto. En la guía que se mostró anteriormente, se definen una serie de ECS, de estos existen algunos que no cambian pero que si forman parte de la línea base, por lo que se deben mantener, estos son:

Registro de revisiones de inconsistencias

Matriz de disponibilidad de proveedores

Informe diagnóstico

Levantamiento de información

Organización de la información

Archivos

Componentes externos

Acta de terminación

Criterios de selección de proveedores

Lista de tipos de adquisición

2.4 Artefacto que se genera durante el proceso de Gestión de Cambios.

Solicitud de Cambio: este artefacto surge a partir de que cualquier involucrado en el proyecto esté interesado en que se realice un cambio, para notificar formalmente esta necesidad deberá completar el formulario, este se emplea para documentar y seguir los defectos, las solicitudes de alguna mejora o cualquier otra solicitud de cambio sobre el producto.

La Solicitud de Cambio debe contemplar información suficiente que permita determinar, por qué es necesario el cambio, qué es lo que debe ser cambiado, quién lo solicita y una breve descripción del problema que permita recomendar una solución. Esta puede ser de Mejora o de Error, es de Mejora cuando se desea agregar un nuevo requisito o modificar uno ya existente con nuevas funcionalidades no acordadas inicialmente y de Error cuando se detectan errores en algún elemento de la Línea Base después de haber sido liberada.

2.5. Propuesta del método de evaluación del impacto de los cambios a los requisitos del producto de software.

A continuación se propone el método de evaluación del impacto de los cambios a los requisitos del producto de software, en epígrafes posteriores se hace un análisis detallado de cada actividad por la que está compuesto y lo que se debe realizar en cada una de ellas.

1. Nombre del método

Método de evaluación del impacto de los cambios a los requisitos del producto de software

2. Objetivo

Ofrecer a los equipos de desarrollo de los proyectos de software de la UCI un método que evalúe el impacto de los cambios propuestos en los requisitos del producto de software.

3. Entradas

Solicitud del Cambio.

Matriz de trazabilidad.

4. Salidas

Se obtiene como salida del método la Solicitud del Cambio, de forma actualizada con el resultado del análisis de impacto, teniendo en cuenta los siguientes parámetros que se encuentran dentro de la propia

Solicitud de Cambio: los ECS que se vieron afectados con el cambio, el esfuerzo y el costo de realizarlo, la cantidad de requisitos afectados, la cantidad de personas implicadas y el tiempo de desarrollo necesario para implementar el cambio. Ver los anexos 1 y 2.

5. Responsables de aplicar el método

Miembros del Comité de Control de Cambio: El Comité de Control de Cambios tiene la responsabilidad de analizar, revisar, aceptar o rechazar las Solicitudes de Cambio realizadas, las cuales son evaluadas entre todos los miembros del mismo en la Reunión de Revisión desarrollada para tales fines, acordándose finalmente el modo en que se deberá proceder.

6. Desarrollo del Procedimiento

- **Identificar la fuente del cambio.**
- **Identificar los elementos relacionados con el cambio.**
- **Construir la tabla de los elementos afectados con el cambio.**
- **Calcular el factor de peso para cada tipo de elemento.**
- **Calcular los ECS sin ajustar.**
- **Calcular los ECS ajustados.**
- **Calcular el Esfuerzo y el Costo.**
- **Toma de Decisiones.**

2.5.1 Identificar la fuente del cambio

En esta actividad se identifica el elemento al que se le va a realizar algún cambio. La mayoría de las veces el cliente solicita un cambio a algún requisito, ya que es la parte del sistema de las que tiene mayor conocimiento.

2.5.2 Identificar los elementos relacionados con el cambio

Luego de identificar el ECS que se desea cambiar, se buscan todos los que están relacionados con este. Para esto se pueden utilizar cualquiera de las 3 herramientas (OSRMT, Rational RequisitePro, Enterprise Architect) que se mostraron en el capítulo 1, con estas se muestra un árbol de trazabilidad con todos los elementos que se relacionan con el que se cambiará.

2.5.3 Construir la tabla de los elementos afectados con el cambio

Luego de obtener el árbol con todos los ECS que se relacionan con el que se va a cambiar, se construye una tabla con los afectados, ya que puede existir alguno que aunque se relacione no sufra ningún cambio.

La tabla se conformará como sigue:

No	Elementos Afectados
1	Elem. X
2	Elem. Y
3	Elem. Z

Tabla 3: Tabla de elementos afectados.

La sección de elementos afectados se debe llenar con el nombre del ECS.

2.5.4 Calcular el Factor de Peso (FP) para cada tipo de elemento

Para realizar esta actividad se tiene en cuenta cada uno de los ECS que se definen en la guía de trazabilidad. Los valores de FP que se obtienen en este paso del método se utilizan para calcular en el siguiente paso los ECS sin ajustar.

La fórmula para calcular el Factor de Peso (FP) es:

$$FP_{ECS_i} = \sum (CantEC_i * Factor)$$

CantEC_i : Cantidad de elementos del mismo tipo.

El FP de cada uno de los ECS se obtiene multiplicando la cantidad de elementos del mismo tipo (simple, medio y complejo) por el valor que tenga su Factor correspondiente y luego se realiza la sumatoria de todos los valores obtenidos.

Para obtener cada uno de los Factores con que se va a trabajar se realizó una entrevista a un grupo conformado por 15 expertos, con un promedio de 4 años de experiencia en proyectos productivos, que desempeñan los roles de Jefe de proyecto, Analista, Arquitecto, Administrador de Configuración, Especialista en Medición y además se incluyeron profesores de Ingeniería de Software. El formato de la

encuesta aplicada se muestra en el Anexo 3.

A continuación se muestran las tablas con las que se determina la complejidad y el valor del Factor correspondiente de cada uno de los ECS.

Casos de prueba:

Tipo	Descripción	Factor
Simple	3 escenarios o menos	1
Medio	4 a 7 escenarios	2
Complejo	8 o más escenarios	3

Caso de Uso del Negocio:

Tipo	Descripción	Factor
Simple	3 transacciones o menos	5
Medio	4 a 7 transacciones	10
Complejo	8 o más transacciones	15

Caso de Uso del Sistema:

Tipo	Descripción	Factor
Simple	3 transacciones o menos	5
Medio	4 a 7 transacciones	10
Complejo	8 o más transacciones	15

Clases de Implementación:

Tipo	Descripción	Factor
Simple	4 métodos o menos	3
Medio	5 a 11 métodos	6
Complejo	12 o más métodos	9

Cronograma:

Tipo	Descripción	Factor
Simple	15 tareas o menos	3
Medio	16 a 42 tareas	6
Complejo	43 o más tareas	9

Diagrama de Clases:

Tipo	Descripción	Factor
Simple	8 clases o menos	5
Medio	9 a 16 clases	10
Complejo	17 o más clases	15

Diagrama de Despliegue:

Tipo	Descripción	Factor
Simple	4 nodos o menos	1
Medio	5 a 9 nodos	2
Complejo	10 o más nodos	3

Diagrama de Interacción:

Tipo	Descripción	Factor
Simple	3 clases o menos	3
Medio	4 a 7 clases	6
Complejo	8 o más clases	9

Diagrama de Procesos con BPMN:

Tipo	Descripción	Factor
Simple	6 actividades o menos	5
Medio	7 a 16 actividades	10
Complejo	17 o más actividades	15

Entidades del negocio:

Tipo	Descripción	Factor
Simple	4 atributos o menos	3
Medio	5 a 9 atributos	6
Complejo	10 o más atributos	9

Modelo de Dominio:

Tipo	Descripción	Factor
Simple	9 clases o menos	3
Medio	10 a 18 clases	6
Complejo	19 o más clases	9

Proveedor de Requisitos:

Tipo	Descripción	Factor
Simple	x	1

Proveedor de Servicios o Productos:

Tipo	Descripción	Factor
Simple	x	1

Proyecto Técnico:

Tipo	Descripción	Factor
Simple	x	1

Reglas del Negocio:

Tipo	Descripción	Factor
Simple	x	1

Reporte (Documento de las salidas del sistema):

Tipo	Descripción	Factor
Simple	x	1

Requisitos:

Tipo	Descripción	Factor
Simple	2 CU o menos	5
Medio	3 a 5 CU	10
Complejo	6 o más CU	15

Tablas (Base de Datos):

Tipo	Descripción	Factor
Simple	5 columnas o menos	3
Medio	6 a 10 columnas	6
Complejo	11 o más columnas	9

Las reglas del negocio, los reportes de las salidas del sistema, los proveedores de producto, servicios y de requisitos y además el proyecto técnico siempre van a ser simples y su factor 1 porque los cambios que estos sufren no son complejos de implementar.

2.5.5 Calcular los ECS sin ajustar

Luego de calcular el FP para cada elemento se procede a calcular el valor de los ECS sin ajustar. Estos se calculan con el objetivo de obtener una aproximación del tamaño del cambio, pero esta estimación no depende solamente del tamaño por lo que más adelante es necesario ajustarlo utilizando otros factores externos.

Para esto se utiliza la siguiente fórmula:

$$ECS_SA = \sum FP_{ECS\ i}$$

$FP_{ECS\ i}$: Factor de peso de cada ECS.

Para obtener el valor de los ECS_SA se suman todos los valores que se obtuvieron en la actividad anterior.

2.5.6 Calcular los ECS ajustados

El valor que se obtuvo anteriormente no es un valor exacto por lo que es necesario ajustarlo con otros factores externos que influyen tales como el Factor de Complejidad Técnica (FCT) y el Factor Ambiente(FA). Otros métodos menos utilizados trabajan con otros factores externos como por ejemplo:[28]

Factor del Valor Agregado:

Factor	Descripción	Complejidad	P*V
Ayuda Integrada	La aplicación debe contar con una ayuda integrada al sistema.	0	0
Mantener una interfaz familiar al negocio.	El cliente de la aplicación solicita que la interfaz o el diseño gráfico del sistema cuente con características particulares o explícitas.	0	0
Aplicación multidioma.	La aplicación debe ser desarrollada en varios idiomas.	0	0

Factor del Cliente:

No	Factor	Descripción	Peso	Valoración	P*V
FC 1	Existe sistema anterior.	¿El proyecto se ha automatizado en otro momento?	0.5	(Si,No)	
FC 2	Existen resultados de informatización.	¿Existen algunas funcionalidades o paquetes funcionales informatizados?	1	(Si,No)	
FC 3	Existe Dirección de Informatización o Informática.	¿Se cuenta en la organización con personal calificado en los conocimientos informáticos y/o con una Dirección o grupo dedicado a la Informatización?	1	(Si,No)	
FC 4	Existe infraestructura tecnológica en la organización.	La infraestructura tecnológica representa la integración de un conjunto de elementos de hardware(servidores, puestos de trabajo, redes ,enlaces de telecomunicaciones, etc.) y servicios(soporte técnico, seguros, comunicaciones, etc.) que en conjunto dan soporte a las aplicaciones (sistemas	1	(Si,No)	

		informáticos) de una empresa u organización.			
FC 5	Existe base legal en la organización.	¿La organización cuenta con una base fundamentada jurídicamente y apoyada en la ley?	1	(Si,No)	
FC 6	Es el primer proyecto con la organización.	¿Es la primera vez que se realiza un proyecto con la organización?	0.5	(Si,No)	
FC 7	Existe una estructura clara en la organización.	¿Los procesos y áreas funcionales están claramente definidos?	1	(Si,No)	
FC 8	Existe un especialista para atender el proyecto.	¿En la organización se cuenta con un especialista en disposición de atender el proyecto?	1	(Si,No)	

Estos otros factores se estudiaron pero se llegó a la conclusión de que no se ajustaban al objetivo de la investigación porque tienen en cuenta factores propios del software que no influyen en los cambios por ejemplo que la plataforma sea multidioma o que tenga ayuda integrada, además otros factores que tienen que ver con el cliente en sí, como por ejemplo, que exista base legal en la organización, que sea la primera vez que se realiza un proyecto, entre otros. El método de evaluación de impacto trabaja con los cambios que puedan ocurrir en los requisitos del software por lo que se debe mantener la estructura del sistema.

Los ECS Ajustados se obtienen mediante la siguiente fórmula:

$$ECS_AJ = ECS_SA * FCT * FA$$

FCT: Factor de complejidad técnica

FA: Factor ambiente

➤ **Factor de Complejidad Técnica (FCT)**

Para obtener el FCT se deben considerar cada uno de los elementos de la siguiente tabla y valorar su peso según al análisis realizado del cambio. El Valor Asignado lo introducen los integrantes del CCC luego

de reunirse y llegar a un acuerdo sobre el valor que puede tomar teniendo en cuenta el análisis de impacto en dependencia de como afecte este factor en los cambios a realizar, se le asigna un valor entre 0 y 5:

Factor	Descripción	Peso	Valor Asignado	Valoración
T1	Sistema distribuido.	2		Sí/No
T2	Objetivos de performance o tiempo de respuesta.	1		La velocidad puede ser limitada por el equipo provisto por el cliente.
T3	Procesamiento interno complejo.	1		Cálculos Complejos.
T4	El código debe ser reutilizable.	1		Se desea poder modificar las fuentes para añadir más funcionalidad al sistema.
T5	Facilidad de uso.	0.5		Normal.
T6	Portabilidad.	2		Se solicitó o no.
T7	Facilidad de cambio.	1		Se requiere un costo moderado de mantenimiento.
T8	Concurrencia.	1		Si se encuentra habilitada o no en el sistema.
T9	Incluye objetivos especiales de seguridad.	1		Si posee o no alta seguridad.
T10	Provee acceso directo a terceras partes.	1		Si/No.

Tabla 4: Peso de los factores de complejidad técnica.

Cada uno de estos puntos se debe evaluar según la siguiente escala:

Descripción	Valor
Irrelevante	De 0 a 2.
Medio	De 3 a 4.
Esencial	5

Tabla 5: Escala de los factores de complejidad técnica.

Las fórmulas para determinar el FCT son:

$$TFactor = \sum (ValorAsignado * Peso)$$

$$FCT = 0.6 + (0.01 * TFactor)$$

Para realizar este cálculo, se debe evaluar cada factor, asignándole un valor como se menciona anteriormente, después se multiplican y se suma cada producto para obtener el TFactor. Luego se debe seguir la segunda fórmula multiplicando el TFactor por 0.01 y sumar el resultado a 0.6, estos valores constantes son coeficientes de ajuste.

➤ **Factor De Ambiente (FA)**

Este factor corresponde a puntos que estén relacionados con el proyecto que va a evaluar su entorno de desarrollo y los conocimientos de su personal. Los elementos que están en color gris se tuvieron en cuenta al principio del proyecto por lo que no es necesario volverlos a calcular, por esa razón el valor asignado de cada uno será de 5.

El proyecto ya se debe encontrar en alguna fase del desarrollo por lo que ya se deben haber adquirido la experiencia y conocimiento necesario de su entorno y por defecto se propone un valor asignado de 5 para los proyectos que no hayan sufrido cambios muy grandes en su organización y/o estructura, en caso de los proyectos que sí hayan sufrido modificaciones de personal, estructura o tecnologías después de su inicio se les recomienda que sí revisen todos los factores además de los obligatorios a evaluar (Motivación, Estabilidad de los requisitos y Personal a tiempo parcial).

Los valores asignados de los factores de la tabla que están sin colorear si deben calcularse porque son factores que si pueden ser diferentes ante la aparición de algún cambio, a estos se les asigna un valor entre 0 y 5:

Factor	Descripción	Peso	Valor Asignado	Comentarios
E1	Familiaridad con el modelo de desarrollo utilizado.	1.5	5	El grupo está familiarizado con el modelo a utilizar.
E2	Experiencia en la aplicación.	0.5	5	No se tiene experiencia en aplicaciones similares.
E3	Experiencia en orientación a objetos.	1	5	El grupo tiene experiencia en programación orientada a objetos.
E4	Capacidad del analista líder.	0.5	5	Se seleccionó a uno de los integrantes del grupo para esta tarea.
E5	Motivación.	1	0-5	El grupo está motivado.
E6	Estabilidad de los requisitos	2	0-5	Se esperan cambios.
E7	Personal a tiempo parcial	-1	0-5	Todo el grupo a tiempo parcial.
E8	Dificultad del lenguaje de programación	-1	5	Se usará lenguaje Java.

Tabla 6: Peso de los factores de ambiente.

Los factores sin colorear se deben calificar con un valor de 0 a 5 según la tabla que se muestra en la figura 12. Las fórmulas para calcular el FA son:

$$AFactor = \sum (ValorAsignado * Peso)$$

$$FA = 1.4 + (-0.03 * AFactor)$$

Para obtener el AFactor se debe sumar todos los productos obtenidos al multiplicar el peso de cada punto por el valor asignado, después se multiplica por -0.03 y se le suma el 1.4, con esto se obtiene el FA; los valores constantes son coeficientes de ajuste.

2.5.7. Calcular el Esfuerzo y el Costo que implica el cambio

Es muy importante calcular el costo del cambio propuesto y además el esfuerzo de las personas

implicadas, ya que son dos factores de vital importancia en el desarrollo de los proyectos.

2.5.7.1 Análisis de Esfuerzo que implica el cambio

Para calcular el esfuerzo que implica el cambio se tiene en cuenta el valor de los ECS ajustados y las horas/hombre necesarias para construir el ECS, para obtener este valor el CCC se reúne y los encargados de desarrollar el ECS informan el valor del esfuerzo para realizar dicho elemento.

$$E = ECS_AJ * \sum H/H_{ECS\ i}$$

ECS_AJ: ECS ajustados

H/H_{ECS i}: Esfuerzo en horas/hombre necesario para realizar el ECS.

2.5.7.2 Análisis de Costo que implica el cambio

Para calcular el costo de realizar un cambio se utiliza la siguiente fórmula:

$$C = E * TH$$

E: Esfuerzo del cambio

TH: Tarifa honoraria

El costo de realizar un cambio se determina multiplicando el Esfuerzo que se calculó en la actividad anterior por una tarifa honoraria, esta tarifa es el salario del trabajador responsable de implementar el cambio.

2.6. Toma de decisiones respecto al cambio.

Por último, se procede a efectuar la toma de decisiones por parte del CCC. Para esto se realiza una reunión y en esta se tendrá en cuenta la siguiente guía para la toma de decisiones, esta contiene elementos que ayudan a dar una valoración sobre la factibilidad del cambio propuesto.

Los elementos a tener en cuenta son:

- **Cantidad de requisitos a cambiar.**
- **Esfuerzo del cambio.**
- **Costo del cambio.**
- **Riesgos que pueden surgir o afectarse con el cambio.**

2.6.1 Cantidad de requisitos a cambiar

Se debe realizar una proporcionalidad entre los requisitos que se cambian y el total de estos, y si es una cantidad muy grande se recomienda analizar la factibilidad del cambio, teniendo en cuenta los otros factores que influyen.

2.6.2 Esfuerzo del cambio

Se debe tener en cuenta el valor del esfuerzo que implica el cambio (este fue calculado anteriormente durante la ejecución del método) y la cantidad de personas implicadas, con esto se obtiene la cantidad de horas necesarias para implementarlo. Luego este valor se divide entre la cantidad de horas que se trabaja en una semana y de esta forma se obtiene el Tiempo de Desarrollo. Es muy importante que este tiempo sea menor que el pactado con el cliente en el contrato, si esto no ocurre se debe aconsejar rechazar el cambio o discutirlo con el cliente porque no es posible realizarlo en tiempo.

A continuación se muestra un ejemplo para ilustrar este paso:

Datos

Esfuerzo = 1000 H/H

Cantidad de hombres = 10 hombres

Se divide el esfuerzo de realizar el cambio entre la cantidad de hombres que lo implementan, esto da como resultado la cantidad de horas que se necesitan para realizarlo:

$$\text{TotalHoras} = 1000 / 10 = 100 \text{ H}$$

Luego se divide este total entre la cantidad de horas que se trabaja en una semana completa que es de 40 H, esto da como resultado el Tiempo de Desarrollo (TD):

$$\text{TD} = 100/40 = 2.5 \text{ semanas}$$

Con 10 hombres se puede realizar el cambio en 2 semanas y media.

2.6.3 Costo del cambio

Luego de calcular el costo que implica realizar el cambio se debe comparar el valor obtenido con el presupuesto que el proyecto dispone para respaldar los cambios y si el costo supera dicho presupuesto se

aconseja rechazar el cambio o discutir con los clientes si a pesar de todo desean realizarlo.

2.6.4 Riesgos que pueden surgir o afectarse con el cambio

Para hacer un análisis de riesgos se debe tener en cuenta el factor de criticidad de los riesgos que pueden surgir a causa del cambio, este se calcula como se muestra a continuación:

$$\text{Factor de criticidad} = \text{Impacto} * \text{Probabilidad}$$

El Impacto es el efecto que tiene el riesgo sobre los objetivos del proyecto, este es un valor numérico. Y la Probabilidad es la probabilidad de ocurrencia del riesgo, este también es un valor numérico. Este factor se evalúa en una matriz de Probabilidad de Impacto que establece rangos de baja, media y alta criticidad, como se muestra [22]:

Matriz de Probabilidad e Impacto					
Probabilidad	Amenazas				
0.90	0.05	0.09	0.18	0.36	0.72
0.70	0.04	0.07	0.14	0.28	0.56
0.50	0.03	0.05	0.10	0.20	0.40
0.30	0.02	0.03	0.06	0.12	0.24
0.10	0.01	0.01	0.02	0.04	0.08
Impacto	0.05	0.10	0.20	0.40	0.80

Figura 11: Matriz de probabilidad de impacto

Los rangos para establecer los colores son los siguientes:

Verde: 0.01 - 0.05

Amarillo: 0.06 - 0.14

Rojo: 0.18 - 0.72

Los valores que se encuentran dentro del rango de color verde son riesgos de baja criticidad que solo

requieren un seguimiento. El color amarillo son riesgos de media criticidad que requieren de un seguimiento y un plan de mitigación. Los de color rojo son de alta criticidad que requieren de un seguimiento, un plan de mitigación y un plan de contingencias.

Es de vital importancia conocer este factor de criticidad porque de esta forma se puede analizar si aumenta o disminuye la criticidad de algún riesgo y si este conlleva a que puedan o no surgir nuevos riesgos y además si se pueda agravar aún más la situación del proyecto.

Si el factor de criticidad de los nuevos riesgos que surgen a causa del cambio se encuentra en el rango de 0.06 a 0.14 se debe valorar la factibilidad de realizar el cambio, y si supera el 0.14 y además existe un número considerable de riesgos se recomienda no aceptar el cambio.

Si el factor de criticidad de los riesgos que ya se tienen en el proyecto pasa a ser de alta criticidad, también se recomienda no realizar el cambio.

Cuando el proyecto experimenta un evento de riesgo negativo, el costo del proyecto casi siempre aumenta, y se produce un retraso en el cronograma de trabajo del proyecto.

2.7 Conclusiones

En este capítulo se propuso un método para evaluar el impacto de los cambios en los requisitos del producto de software, este será utilizado en los proyectos productivos que se llevan a cabo en la UCI. Para el desarrollo del método se tuvieron en cuenta los ECS que se definen en el programa de mejora que se lleva a cabo en los proyectos productivos, las relaciones que se establecen entre ellos, así como su complejidad técnica. Se formuló el cálculo del alcance del cambio, el esfuerzo y el costo necesario para llevarlo a cabo y con estos valores el CCC podrá tomar una decisión con respecto al cambio, si es factible llevarlo a cabo o no.

Capítulo 3: Validación de Resultados

3.1 Introducción

Con el objetivo de verificar si se cumplieron los objetivos trazados con el desarrollo de este trabajo, se realiza la validación del método de evaluación de impacto de los cambios a los requisitos del producto de software. Para la aceptación de la propuesta se emplea la validación a través de un caso de estudio.

3.2. Aplicación del Procedimiento a un Caso de Estudio

Para comprender mejor el procedimiento que se presentó en el capítulo anterior se valida el mismo aplicándolo al caso de estudio que se presenta a continuación.

3.2.1 Caso de estudio

En el Proyecto de Desarrollo de Software X de la facultad Y de la UCI, que se encuentra dentro del programa de mejora, se desean hacer algunos cambios que han solicitado los clientes, el equipo de trabajo tiene que realizar un análisis para decidir si es posible realizarlos o no, y luego informarle a los clientes los resultados obtenidos.

El proyecto se encuentra a mediados de la fase de construcción y ya se han implementado muchas de las funcionalidades del sistema.

En el proceso de contratación con el cliente y durante la fase de inicio se confeccionó el Plan de Proyecto donde se pudo estimar el esfuerzo en horas–hombres que se necesita para la realización del trabajo y contiene además una estimación del costo que implica el desarrollo del producto.

La estimación realizada a inicios del proyecto arrojó como resultado que el esfuerzo que se necesitaba para la realización del mismo sería de 1500 horas-hombre.

El equipo de desarrollo está constituido por 13 trabajadores los cuales están distribuidos de la siguiente forma:

1 Jefe de Proyecto

2 Analistas

1 Arquitecto de software

5 Programadores

1 Integrador del Sistema

1 Responsables de las Pruebas

1 Administrador de la Configuración

1 Diseñador de Base de Datos

Cada uno de estos trabajadores cobra \$ 3.00 por hora de trabajo. El presupuesto destinado al inicio del proyecto para subsidiar los cambios que sean necesarios fue de \$ 700.

El proyecto cuenta con un CCC integrado por:

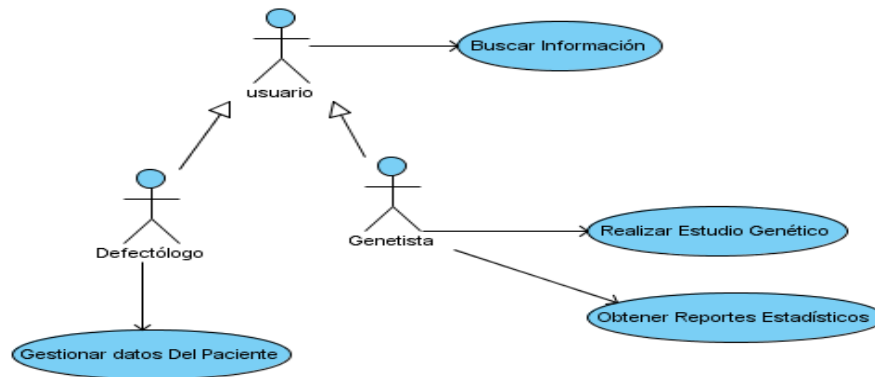
- Jefe de proyecto
- Analista
- Administrador de la Configuración
- Arquitecto de software

El cliente propone un cambio que considera necesario para el ajuste del sistema a la empresa, este consiste en realizar un cambio en la Interfaz de Usuario que corresponde al caso de uso Gestionar Datos del Paciente, el cliente desea que se le agreguen nuevas funcionalidades tales como: Dirección y Sexo del paciente.

El administrador de la configuración llena la Solicitud de Cambio. El ECS afectado directamente sería el caso de uso del sistema Gestionar Datos del Paciente, específicamente en su escenario Insertar Datos del Paciente.

A continuación se muestran algunos de los diagramas que se relacionan con el cambio:

- **Diagrama de caso de uso del sistema:** el elemento afectado con el cambio es solo el caso de uso Gestionar Datos del Paciente.



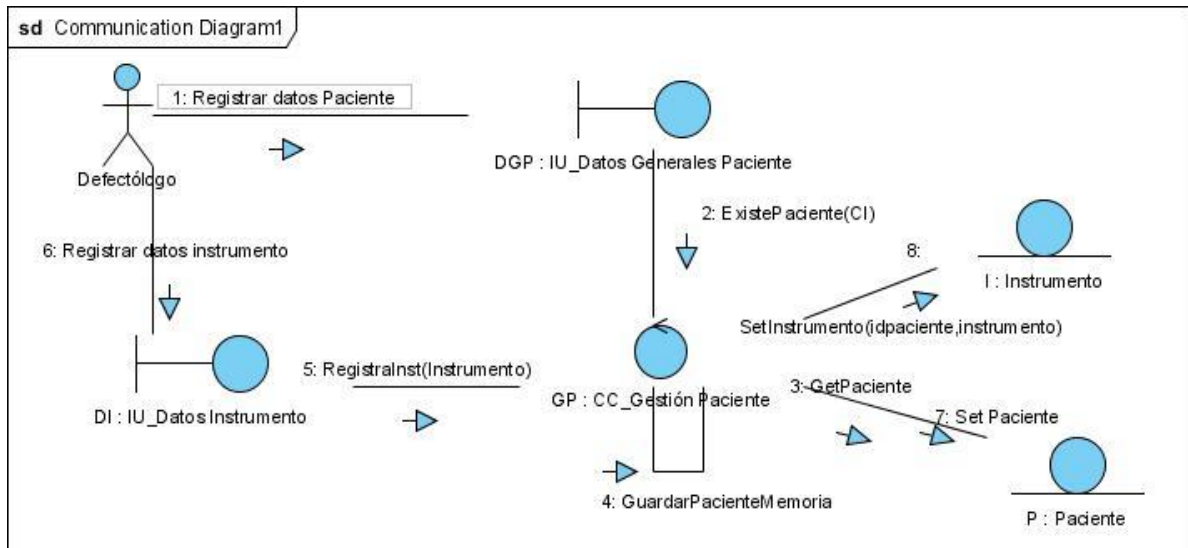
➤ Descripción del caso de uso que se va a cambiar y el diagrama de colaboración correspondiente:

Caso de Uso:	Gestionar Datos de Pacientes
Actores:	Defectólogo
Resumen:	El caso de uso se inicia cuando el defectólogo desea introducir un nuevo paciente al sistema o actualizar los datos de uno existente.
Precondiciones:	
Flujo Normal de Eventos	
	El sistema muestra un listado con todos los pacientes y las opciones de Insertar y modificar los datos de un paciente.
El defectólogo selecciona un elemento de la lista: En caso de:	
Insertar datos de paciente. Sección Insertar datos de paciente.	
Modificar datos de paciente. Sección	

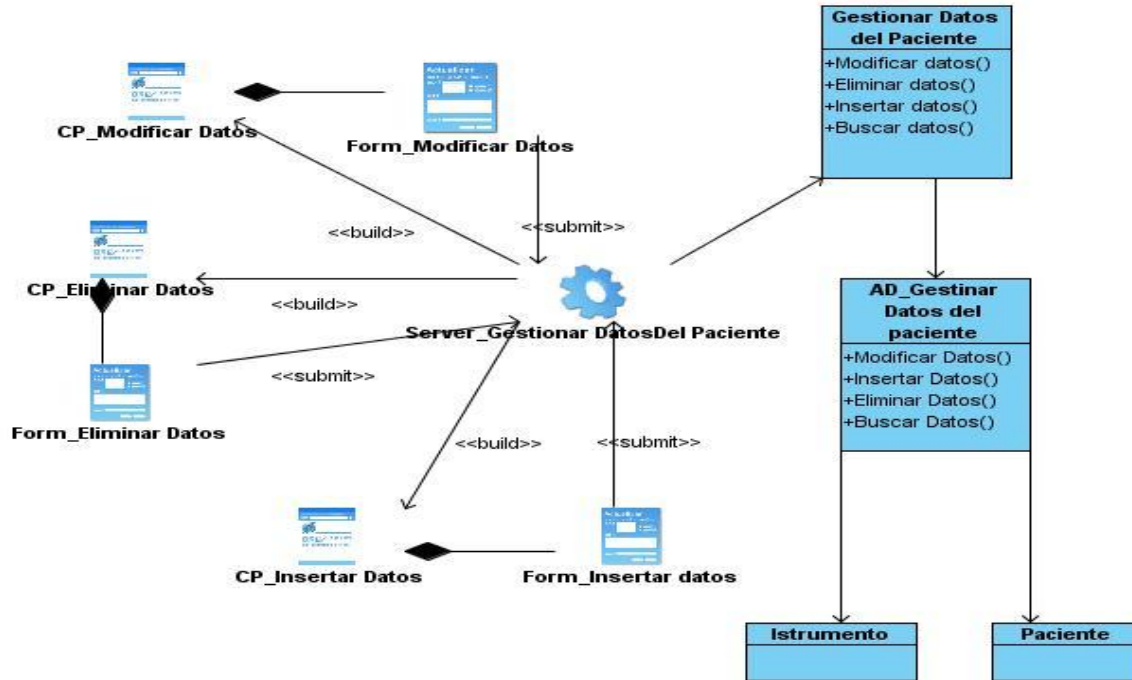
Modificar datos de paciente.	
Sección Insertar Datos del Paciente	
Acción del Actor	Respuesta del Sistema
1. El defectólogo solicita registrar datos del paciente.	2. El sistema muestra un formulario con los campos a completar del paciente.
El defectólogo introduce los datos del cliente y selecciona la opción aceptar.	El sistema valida los datos de entrada
	El sistema verifica que el paciente no este registrado.
	El sistema guarda en memoria los datos del paciente y muestra un formulario con los campos a completar del instrumento.
El defectólogo introduce los datos del instrumento y selecciona la opción aceptar.	El sistema valida los datos de entrada.
	El sistema registra los datos generales del paciente y los datos del instrumento. Muestra el mensaje "Datos del paciente registrados satisfactoriamente"
Flujos Alternos	
	4.1 El sistema muestra el mensaje "Datos de entrada no válidos" y pasa a la acción 2 del flujo normal d los eventos.
	5.1 El sistema muestra el mensaje "El paciente ya existe en el sistema" y pasa a

	la acción 2 del flujo normal de los eventos.
	8.1 El sistema muestra el mensaje "Datos de entrada no válidos" y pasa a la acción 5 del flujo normal de los eventos.
Sección Modificar Datos de discapacitado	
Acción del Actor	Respuesta del Sistema
	1. El sistema busca los datos del discapacitado seleccionado y los muestra en un formulario
2. El defectólogo modifica los datos deseados y selecciona la opción aceptar	3. El sistema valida los datos de entrada y actualiza los datos del discapacitado.
Poscondiciones	Quedan registrados los datos del paciente.

➤ Diagrama de colaboración:



➤ Diagrama de clases del diseño:



3.2.2 Aplicación del método

➤ Identificar la fuente del cambio.

El ECS afectado con el cambio es el Caso de Uso del Sistema Gestionar Datos del Paciente, específicamente en su escenario Insertar Datos del Paciente, a este se le van a insertar dos nuevos campos: Dirección y Sexo, aparte de los que ya se habían acordado que tenía que llevar la aplicación.

➤ Identificar los elementos relacionados con el cambio.

Los ECS que se encuentran relacionados con el cambio son:

- Diagrama de Clase del Diseño
- Entidades de la Base de Datos
- Diagrama de Interacción
- Clases de Implementación

- Entidades del Negocio
- Casos de Prueba.

Para mostrar cómo se encuentran relacionados estos elementos se utiliza la herramienta OSRMT y queda como se muestra a continuación:

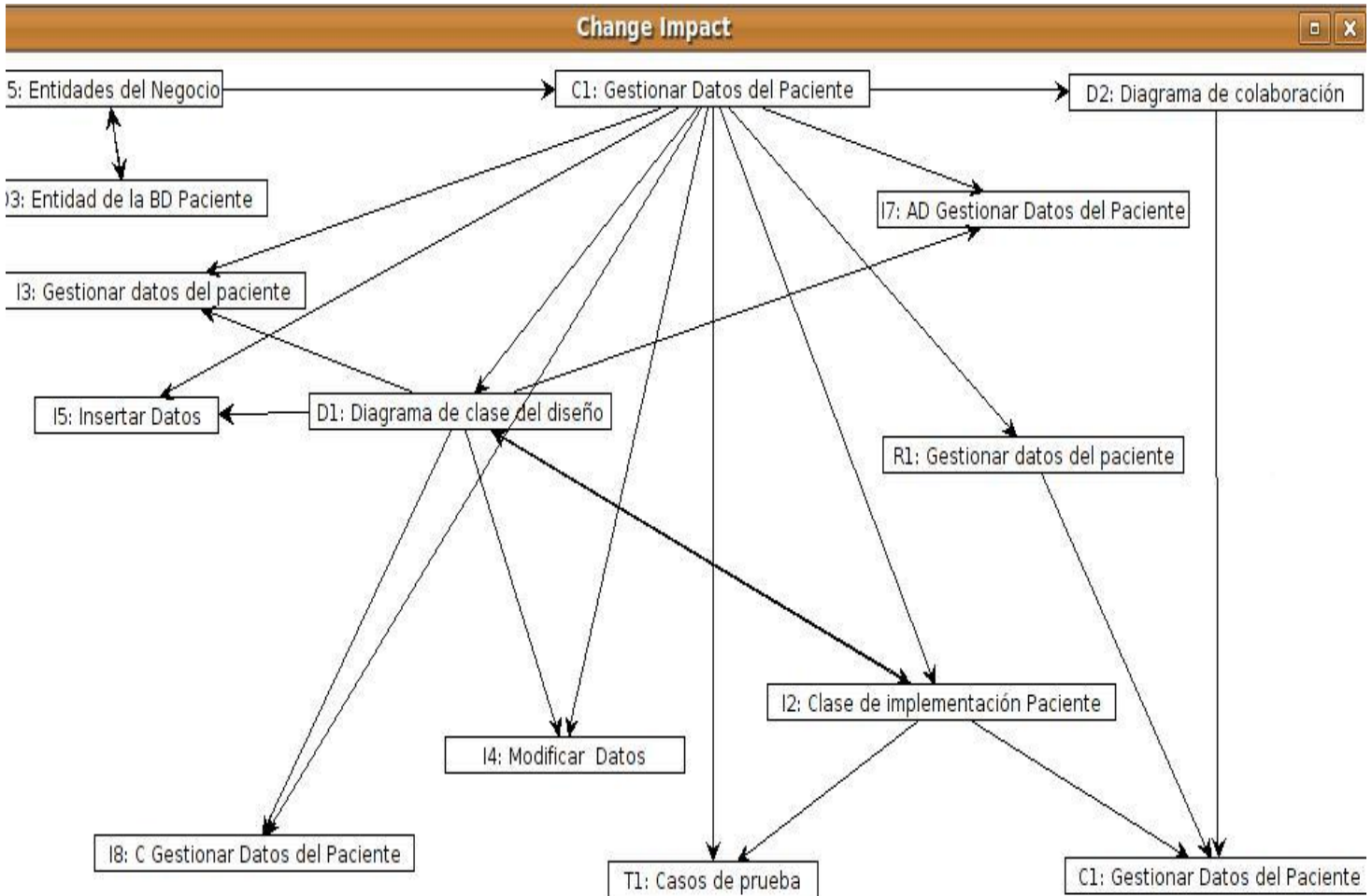


Figura 12: Relación entre los ECS afectados como se muestra en la OSRMT.

Se debe aclarar que aquí también se pueden utilizar las herramientas Enterprise Architect o Rational RequisitePro, según la conveniencia del desarrollador que está aplicando el método.

➤ **Construir la tabla de los elementos afectados con el cambio.**

De todos los elementos que se relacionan con el que se va a cambiar los que se ven afectados directamente son los siguientes:

No	Elementos Afectados
1	CUS_ Gestionar Datos del Paciente
2	Diagrama de Clase del Diseño
3	Diagrama de Colaboración
4	Clases de Implementación
5	Entidades de la Base de Datos
6	Cronograma de tareas

Tabla 7: Elementos afectados con el cambio.

➤ **Calcular el factor de peso (FP) para cada tipo de elemento.**

Para aplicar este paso del método se construye una tabla como la que se muestra a continuación.

Nombre del ECS	Cantidad de elementos	Parámetro	Complejidad	Factor
Diagrama de Clase del Diseño	1 Diagrama	8 clases	Simple	5
Diagrama de Colaboración	1 Diagrama	5 clases	Medio	6
Clases de Implementación	C_Paciente	7 métodos	Medio	6
	C_ Gestionar Datos del Paciente	15 métodos	Complejo	9
	C_Insertar Datos	4 métodos	Simple	3
	C_Modificar Datos	4 métodos	Simple	3

	AD_ Gestionar Datos del Paciente	12 métodos	Complejo	9
	C_Controladora Gestionar Datos	8 métodos	Medio	6
Entidades de la Base de Datos	Paciente	10 tablas	Medio	6
Cronograma	1 Cronograma	6 tareas	Simple	3
CUS_ Gestionar Datos del Paciente	1 CUS	2 transacciones	Simple	5

Tabla 8: Tabla para organizar los datos con los que se calcula el Factor.

En esta se refleja el nombre del ECS afectado, la cantidad de elementos de ese tipo que sufren cambios, el parámetro por el cual se evaluó su complejidad, esta se refleja en la columna que sigue y por último el factor asociado a esa complejidad.

La fórmula para calcular el FP de cada uno de los ECS es la siguiente:

$$FP_{ECS_i} = \sum (CantEC_i * Factor)$$

Aplicando esta, los FP quedan como sigue:

Diagrama de Clase del Diseño:

$$FP_{DiagCD} = 1 * 5$$

$$FP_{DiagCD} = 5$$

Diagrama de Colaboración:

$$FP_{DiagColab} = 1 * 6$$

$$FP_{DiagColab} = 6$$

Clases de Implementación:

$$FP_{C_Implementen} = 2 * 3 + 2 * 6 + 2 * 9$$

$$FP_{C_Implementen} = 36$$

Entidades de la Base de Datos:

$$FP_{EntidadesBD} = 1 * 6$$

$$FP_{EntidadesBD} = 6$$

Cronograma:

$$FP_{Cronograma} = 6 * 3$$

$$FP_{Cronograma} = 18$$

CUS_ Gestionar Datos del Paciente:

$$FP_{CUS} = 2 * 5$$

$$FP_{CUS} = 10$$

➤ **Calcular los ECS sin ajustar.**

Con los datos anteriores se calcula el valor de los ECS sin ajustar como sigue:

$$ECS_SA = \sum FP_{ECS\ i}$$

$$ECS_SA = 5 + 6 + 36 + 6 + 18 + 10$$

$$ECS_SA = 81$$

➤ **Calcular los ECS ajustados.**

Para esta actividad se aplica la siguiente fórmula:

$$ECS_AJ = ECS_SA * FCT * FA$$

El valor de los ECS_SA se calculó anteriormente, ahora se procede a hallar el Factor de Complejidad Técnica (FCT).

Lo primero es darle un valor asignado a cada uno de los factores como se muestra en la tabla:

Factor	Descripción	Peso	Valor Asignado	Comentario
T1	Sistema distribuido.	2	5	No

T2	Objetivos de performance o tiempo de respuesta.	1	3	La velocidad puede ser limitada por el equipo provisto por el cliente.
T3	Procesamiento interno complejo.	1	0	Cálculos Complejos.
T4	El código debe ser reutilizable.	1	5	Se desea poder modificar las fuentes para añadir más funcionalidad al sistema.
T5	Facilidad de uso.	0.5	3	Normal
T6	Portabilidad.	2	5	Se solicitó.
T7	Facilidad de cambio.	1	5	Se requiere un costo moderado de mantenimiento.
T8	Concurrencia.	1	1	Habilitada en el sistema.
T9	Incluye objetivos especiales de seguridad.	1	5	Alta seguridad
T10	Provee acceso directo a terceras partes.	1	0	No posee

Tabla 9: Criterios para dar el Valor Asignado en el cálculo del FCT.

Luego de asignarle los números correspondientes al campo Valor Asignado se aplica la fórmula del FCT y se realiza el cálculo:

$$TFactor = \sum (\text{ValorAsignado} * \text{Peso})$$

$$FCT = 0.6 + (0.01 * TFactor)$$

$$TFactor = 2*5 + 1*3 + 1*0 + 1*5 + 0.5*3 + 2*5 + 1*5 + 1*1 + 1*5 + 1*0$$

$$TFactor = 40.5$$

$$FCT = 0.6 + (0.01 * 40.5)$$

FCT= 1.005

Luego se procede a calcular el Factor Ambiente (FA):

Primero se le da un valor a los factores que no aparecen sombreados:

Factor	Descripción	Peso	Valor Asignado	Comentarios
E1	Familiaridad con el modelo de desarrollo utilizado.	1.5	5	El grupo está familiarizado con el modelo a utilizar.
E2	Experiencia en la aplicación.	0.5	5	No se tiene experiencia en aplicaciones similares.
E3	Experiencia en orientación a objetos.	1	5	El grupo tiene experiencia en programación orientada a objetos.
E4	Capacidad del analista líder.	0.5	5	Se seleccionó a uno de los integrantes del grupo para esta tarea.
E5	Motivación.	1	4	El grupo está motivado.
E6	Estabilidad de los requisitos	2	5	Se esperan cambios.
E7	Personal a tiempo parcial	-1	2	Todo el grupo a tiempo parcial.
E8	Dificultad del lenguaje de programación	-1	5	Se usará lenguaje Java.

Tabla 10: Criterios para dar el Valor Asignado en el cálculo del FA

Con estos datos se calcula el FA según las fórmulas y se procede con el cálculo:

$$A_{Factor} = \sum (\text{ValorAsignado} * \text{Peso})$$

$$FA = 1.4 + (-0.03 * A_{Factor})$$

$$AFactor = 1.5*5 + 0.5*5 + 1*5 + 0.5*5 + 1*4 + 2*5 + (-1*2) + (-1*5)$$

$$AFactor = 24.5$$

$$FA = 1.4 + (-0.03 * 24.5) = 1.4 + (-0.74)$$

$$FA = 0.66$$

Finalmente, se sustituyen los valores calculados para obtener los ECS ajustados:

$$ECS_AJ = ECS_SA * FCT * FA$$

$$ECS_AJ = 81 * 1.005 * 0.66$$

$$ECS_AJ = 53.73$$

➤ **Calcular el Esfuerzo y el Costo que implica el cambio**

Esfuerzo del cambio:

En la siguiente tabla se muestran las horas que se dedica a desarrollar cada ECS que se afecta con el cambio:

Nombre del ECS	H/H
Diagrama de Clase del Diseño	30 min
Diagrama de Colaboración	45 min
Clases de Implementación	2 h=120 min
Entidades de la Base de Datos	5 min
Cronograma	10 min
CUS_ Gestionar Datos del Paciente	10 min
Total	220 min=3.6 h

Tabla 11: Esfuerzo de los ECS

Con estos datos se aplica la fórmula y se obtiene el Esfuerzo necesario para realizar el cambio:

$$E = ECS_AJ * \sum H/H_{ECS I}$$

$$E = 53.73 * (30+45+120+5+10+10) = 11820.6 \text{ min}$$

$$E = 197.01 \text{ H/H}$$

Costo del cambio:

Con el esfuerzo calculado anteriormente y la tarifa honoraria que es el salario del trabajador encargado del cambio se procede a calcular cuánto va a costar implementar el cambio propuesto como se muestra a continuación:

$$C = E * TH$$

$$C = 197.01 \text{ H/H} * \$3.00$$

$$C = \$ 591.03$$

➤ Toma de decisiones

Luego de haber analizado el cambio propuesto y haber aplicado todos los pasos del método se demostró que con dicho cambio solo se afecta un requisito, lo que posibilita que no sea demasiado complicado realizarlo. El costo no superó el presupuesto que el proyecto tenía disponible para subsidiar los cambios, por lo que se puede realizar el cambio.

En cuanto al esfuerzo se puede decir que con la intervención del jefe de proyecto, un analista, un programador y el diseñador de base de datos se pueden realizar en 1 semana y 2 días el cambio propuesto, este tiempo de desarrollo se encuentra dentro del tiempo pactado con el cliente. A continuación se ejemplifica como se calculó:

$$E = 197.01 \text{ H/H}$$

Cantidad de personas = 4 hombres

$$\text{TotalHoras} = 197.01 / 4$$

$$\text{TotalHoras} = 49.25 \text{ H}$$

TD = 49.25 /40

TD= 1.2 semanas

Con el cambio no surgieron nuevos riesgos en el proyecto y los que ya se encontraban no se agravaron con la implementación del mismo, o sea, su factor de criticidad mantuvo su valor.

Teniendo en cuenta todo lo antes planteado se puede llegar a la conclusión de que si se puede realizar el cambio sin que se afecte el desarrollo exitoso del proyecto.

3.3 Conclusiones

Luego de finalizar el capítulo y realizar la validación con resultados satisfactorios se demostró que el método propuesto es fácil de aplicar y no exige gran cantidad de personal para llevarlo a cabo; se puede terminar en poco tiempo lo cual permite que se le dé al cliente una respuesta rápida sobre la decisión que se toma respecto a su solicitud.

Conclusiones Generales

Con la culminación del presente trabajo se cumplieron las tareas y el objetivo fundamental de esta investigación, arribando a las siguientes conclusiones:

- Se realizó un estudio de los conceptos fundamentales de la Gestión de Configuración lo que permitió obtener una gran cantidad de información sobre los elementos a manejar en el desarrollo de este trabajo.
- Se estudiaron los métodos y técnicas que existen para el análisis de impacto de los cambios en los procesos de desarrollo de software y se valoró la utilización de alguno de ellos.
- Se estudiaron las matrices de trazabilidad para facilitar el trabajo con los requisitos y las herramientas de apoyo a la administración de requisitos.
- Se analizaron diferentes métodos de estimación para el análisis de costos y esfuerzo, viendo sus ventajas y desventajas lo cual aportó información valiosa para el desarrollo del método propuesto.
- Se realizó una propuesta de un método para el análisis de impacto de los cambios a los requisitos del software, el cual ayuda a determinar cuánto pueden afectar los cambios en las funcionalidades del sistema, y el impacto que puede tener en el costo y en el esfuerzo de las personas implicadas.
- Se validaron los resultados a través de un caso de estudio y se realizó una valoración objetiva de los resultados obtenidos.

Recomendaciones

Se recomienda:

- Perfeccionar el procedimiento propuesto a medida que surjan nuevas necesidades y proponer herramientas que ayuden a aplicarlo de forma óptima.
- Que en la toma de decisiones se establezcan límites para la cantidad de requisitos afectados y para el análisis del costo, que permitan dar un número específico para facilitar la decisión a tomar con respecto al cambio.
- Realizar una investigación para ver que otros factores se pueden ver afectados con los cambios además del Factor de Complejidad Técnica y el Factor Ambiente.
- Obtener históricos mediante la aplicación del método para valorar la acotación de los coeficientes de ajuste y los pesos de los factores utilizados.
- Profundizar el estudio del análisis de impacto en los programas docentes.
- Que se aplique el procedimiento propuesto en los proyectos productivos de la UCI.

Referencias Bibliográficas

- [1]. Pressman, Roger S., "Ingeniería de software, un enfoque práctico". Quinta edición, 1994. 614 págs.
- [2]. IEEE Std-729, "Glosario Estándar IEEE de Terminología de Ingeniería del Software", 1983.
- [3]. CMMI for Development, Versión 1.2 ,2006.573 págs.
- [4]. CMMI for Development, Versión 1.2 ,2006. 573 págs.
- [5]. Nazim, H. Madhavji, "Environment Evolution: The Prism Model of Changes". IEEE Transaction on Software Engineering, Volumen 18, No. 5,1992.
- [6]. CMMI for Development, Versión 1.2 ,2006. 573 págs.
- [7]. CMMI for Development, Versión 1.2 ,2006. 573 págs.
- [8]. CMMI for Development, Versión 1.2 ,2006. 573 págs.
- [9]. Planilla 5138_Guía de Trazabilidad., Expediente de proyecto de PM 2.0.
- [10]. Lock, Simon, "Requirement Level Change Management and Impact Analysis", 1998.
- [11]. Bohner, S. A. y Arnold, R. S., "An Introduction to Software Change Impact Analysis , Software Change Impact Analysis". IEEE Computer Society ,1996 .
- [12]. Michelle, L. Lee, "A Dissertation Submitted to the Graduate Faculty of George Mason University In Partial Fulfillment of The Requirements for the Degree of Doctor of Philosophy Information Technology".
- [13]. www.rational.com.ar
- [14]. "Using Bayesian belief networks for change impact analysis in architecture design " por Antony Tang, Ann Nicholson, Yan Jin y Jun Han .J. of Soft. Maint, vol. 80, 2007.
- [15]. Soffer, Pnina, "Scope Analysis: Identifying the Impact of Changes in Business Process Models," en Journal of Software Process: Improvement and Practices, vol. 10, no.1, 2005.
- [16]. Bach, James, "The Highs and Lows of Change Control ",1998.
- [17]. Durán, Sergio E., "Puntos por Función. Una métrica estándar para establecer el tamaño del software". Boletín de Política Informática Núm. 6, 2003.
- [18]. Moreno, Ana M., "Estimación de Proyectos Software".

- [19]. Badani, Sigifredo E., "Métricas de estimación de tamaño Puntos de Caso de Uso", 2002.
- [20]. De Antonio, Angélica., "La gestión de la configuración". Madrid: Universidad Politécnica de Madrid, 2001.
- [21]. Planilla 5602_Estándares de Configuración, Expediente de proyecto de PM 2.0.
- [22]. PMI, Project Management Body of Knowledge. PMI Communications, 2004.
- [23]. Quintana, Dayanis A., Cid Yusely, "Propuesta de un Procedimiento para la Toma de Decisiones en la Gestión de Cambios de los Proyectos Productivos de la Facultad 3.UCI.Ciudad Habana, 2008. 111 págs.
- [24]. Pressman, Roger S., "Ingeniería de software, un enfoque práctico". Quinta edición, 1994. 614 págs.
- [25]. Pressman, Roger S., "Ingeniería de software, un enfoque práctico". Quinta edición, 1994. 614 págs.
- [26]. Colectivo Autores Técnicas de Estimación , UCI, 2005
- [27]. Peralta, Mario, " Estimación Del Esfuerzo Basada En Casos De Uso", Buenos Aires, Argentina.
- [28]. Rodríguez, Dayami , Febles, Ailyn , Monagas, Miguel A. y Delgado, Ramsés, "Método de Estimación para los proyectos llave en mano de la Universidad de las Ciencias Informáticas", UCI.
- [29]. <http://calisoft.uci.cu>

