

ARTÍCULO ORIGINAL

Método para el análisis de modificaciones post-implementación en sistemas de gestión de software

*Method for Analysis
Post-Implementation Modifications
in Software Management Systems*

Leidy Ramos González

lramosg@uci.cu • <https://orcid.org/0000-0001-5369-3064>

Yoandi Díaz Ramos

ydramos@uci.cu • <https://orcid.org/0000-0001-7349-4652>

Nemury Silega Martínez

nsilega@uci.cu • <https://orcid.org/0000-0002-8436-5650>

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, CUBA

Recibido: 2021-07-31 • Aceptado: 2021-09-03

RESUMEN

El modelo empresarial actual demanda sistemas de gestión que controlen de forma sistemática las actividades y procesos de la empresa. Estos sistemas deben considerar los parámetros económicos, de productividad, la satisfacción de los usuarios y clientes, así como su mejora continua. Esta es una característica de las empresas más competitivas y eficientes. Los sistemas de gestión evolucionan en la fase posterior a la implementación debido a los cambiantes requisitos comerciales y al entorno en que se desarrollan. La incorrecta atención a los cambios post-implementación afecta la calidad de la información que se gestiona e incide negativamente en el rendimiento de una empresa, en la satisfacción de los usuarios y en la calidad del sistema. Para abordar el problema descrito en el presente trabajo se elaboró un método para el análisis de las modificaciones post-implementación en los sistemas de gestión con el objetivo de disminuir el impacto negativo provocado.

PALABRAS CLAVE: Análisis; Cambios; Método; Modificaciones; Post-implementación.

ABSTRACT

The current business model demands management systems that systematically control the company activities and processes. These systems must take into account economic parameters, productivity, user and customer satisfaction as well as their continuous improvement. This is a characteristic of the most competitive and efficient companies. Management systems evolve in the post-implementation phase due to changing business requirements and the environment in which they are developed. Incorrect attention to post-implementation changes affects the quality of the information that is managed and negatively affects the performance of a company, user satisfaction and the quality of the system. In order to address the problem described in this paper, a method was developed for the analysis of post-implementation modifications in management systems with the aim of reducing the negative impact caused.

KEYWORDS: *Analysis; Changes; Method; Modification; Post-Implementation.*

INTRODUCCIÓN

La gestión empresarial ha evolucionado en la medida que el hombre avanza en la consecución de nuevas tecnologías y relaciones para el mejoramiento de los productos y servicios, en la satisfacción de un mercado cada día más creciente y complejo. Las empresas más competitivas y eficientes, suelen implementar sistemas de gestión que permitan que sus productos o servicios tengan elementos cualitativos, emitan confianza y favorezcan la decisión de los clientes (Ramos González *et al.*, 2018).

Un sistema de gestión empresarial es un conjunto de programas de computación, que brinda a la gerencia la organización necesaria para la toma de decisiones y la comunicación con los usuarios y otros sistemas. Es reconocido por un conjunto de elementos que mutuamente se relacionan o que interactúan entre sí, posibilitando el cumplimiento de los objetivos de la empresa. También son conocidos como sistemas de *software* empaquetados que respaldan la mayoría de las operaciones de las organizaciones (Parhizkar & Comuzzi, 2017).

Los sistemas de gestión empresarial en la actualidad son muy exitosos, flexibles y pueden ajustarse a diversos entornos empresariales, sin embargo, esta flexibilidad conduce a una mayor complejidad. La mayoría de ellos han pasado al menos un ciclo de implementación y pueden encontrarse en la etapa posterior a la implementación donde se enmarca fundamentalmente su evolución. Los sistemas inevitablemente cambian en la fase posterior a la implementación para alinear sus funcionalidades con los cambiantes procesos de negocio y requisitos. Los cambios posteriores a la implementación pueden ser impulsados por diferentes razones, por

ejemplo, una nueva política de gestión de la empresa, por factores externos, como proveedores de tecnología u organismos reguladores (Heredero *et al.*, 2019).

Los cambios post-implementación se centra principalmente en Factores Críticos de Éxito (FCE) (Cacha Ramos *et al.*, 2021; Ramos González *et al.*, 2018; Parhizkar & Comuzzi, 2017). La calidad del sistema y los datos que en él se encuentran, se consideran FCE de la fase posterior a la (Hassan y Holt, 2004; Soffer *et al.*, 2003) implementación. La atención incorrecta a las modificaciones post-implementación, puede afectar la calidad de la información que se gestiona y puede incidir negativamente en el rendimiento de la organización, en la satisfacción de los usuarios, en la calidad del sistema, de la información y del servicio. Una mala gestión del cambio puede ocasionar el rechazo del *software* implementado provocando el fracaso del proyecto.

En este trabajo se presenta un método para el análisis de las modificaciones post-implementación en sistemas de gestión de *software*. Además, se expone su validación utilizando varios métodos.

MATERIALES Y MÉTODOS

Para alcanzar resultados satisfactorios en la investigación se realizó una revisión sistémica a la bibliografía con el objetivo de identificar cómo diferentes investigadores abordan el tema en cuestión. Mediante este método, se identifica, evalúa y combina la evidencia de los estudios de investigación; siendo este un medio para evaluar e interpretar la información relevante disponible asociada a una investigación.

La solución contiene elementos comunes de trabajos precedentes sobre la temática y que han tenido éxito en la solución del problema que abordan. Para ello se utilizó el motor de búsqueda *Google Académico* que es reconocido como líder en la búsqueda de materiales científicos. Con este buscador se obtienen materiales de importantes fuentes de datos, entre las que se encuentran *IEEE*, *Scopus* y *Springer* que son reconocidas como referencias en investigaciones científicas.

Los trabajos definidos por varios autores (Ravasan & Mansouri, 2014; Alsulami *et al.*, 2013; Moalagh & Ravasan, 2013) tienen como premisa fundamental los beneficios en la fase posterior a la implementación y tienen un alcance centrado en la organización. En ellos se definen indicadores importantes como: la calidad del sistema, la calidad de la información, calidad del servicio, el uso, la satisfacción del usuario, el impacto individual y organizacional, éxito empresarial, el beneficio neto, la calidad de los proveedores externos y la alineación estratégica. Estos indicadores se tuvieron presente en la elaboración del método propuesto.

Varios investigadores (Cacha Ramos *et al.*, 2021; Heredero *et al.*, 2019; Lozano & Lisbel, 2017; Yu, 2005) reconocen los Factores Críticos del Éxito (FCE) durante la etapa posterior a la implementación como el factor clave para lograr el éxito organizacional, reafirmando que el sistema exitosamente instalado no es el final, sino que funciona y mejora continuamente a

lo largo del tiempo y en toda la organización. Los FCE formaron parte esencial de la solución como buenas prácticas en la etapa post-implementación.

Otros autores (Parhizkar & Comuzzi, 2017; Palinkas *et al.*, 2015) tienen como premisa el alcance del sistema, proponen métodos y herramientas para apoyar el análisis de impacto de los cambios posteriores a la implementación. Estos permiten la efectividad en la post-implementación del sistema teniendo como base las expectativas previas a la implementación. Estas investigaciones fueron relevantes para la elaboración de la solución propuesta.

Se elaboró un método que está integrado por un conjunto de componentes compuesto por el diseño del metamodelo de dependencias, el diseño del mecanismo de análisis de impacto y el diseño de métricas de evaluación de impacto.

El análisis de impacto de los sistemas de *software* se enfoca en el impacto negativo de los cambios propuestos en el código fuente existente, sin embargo, el presente trabajo se enfoca además en el impacto negativo de los cambios propuestos a nivel operacional, es decir, en las entidades conceptuales que definen un sistema y los procesos de negocios actualmente en ejecución. Se definió como primer momento el análisis del impacto operacional del cambio propuesto, y como segundo momento el análisis del impacto de la implementación del cambio en el código fuente existente.

Los cambios propuestos para un sistema, no necesariamente, implican la modificación del código fuente. Sin embargo, pueden implementarse a través de la configuración de las opciones existentes o mediante la funcionalidad proporcionada por un tercero externo.

RESULTADOS Y DISCUSIÓN

Se elaboró un método integrado por un conjunto de componentes que permiten mapear las dependencias entre las entidades en tiempo de diseño y en tiempo de ejecución de un sistema y detectar dependencias relevantes entre ellas para una determinada solicitud de cambio. Sobre la base de las dependencias identificadas, el marco ayuda al usuario a analizar el impacto de un cambio propuesto a través de un mecanismo de análisis de impacto. El alcance y la profundidad de un cambio propuesto también se pueden analizar a través de un conjunto de métricas de evaluación de impacto de modificación posterior a la implementación.

El conjunto de componentes se divide en:

- Diseño del metamodelo de dependencias: este metamodelo permite definir dependencias entre las entidades de un sistema, tanto en los niveles de estructura estática en tiempo de diseño como en las operaciones en tiempo de ejecución.
- Diseño del mecanismo de análisis de impacto: dada una modificación de un cambio posterior a la implementación, este mecanismo permite identificar los componentes del sistema afectados por el cambio y ayudar al usuario a terminar de manera segura las instancias de proceso actualmente en ejecución afectadas por el cambio.
- Diseño de las métricas de evaluación de impacto: estas métricas cuantifican el impacto de un cambio propuesto en función de los resultados del análisis de impacto.

1. DISEÑO DEL METAMODELO DE DEPENDENCIAS

El metamodelo de dependencias propuesto entre entidades que constituyen un sistema de gestión se muestra en la figura 1.

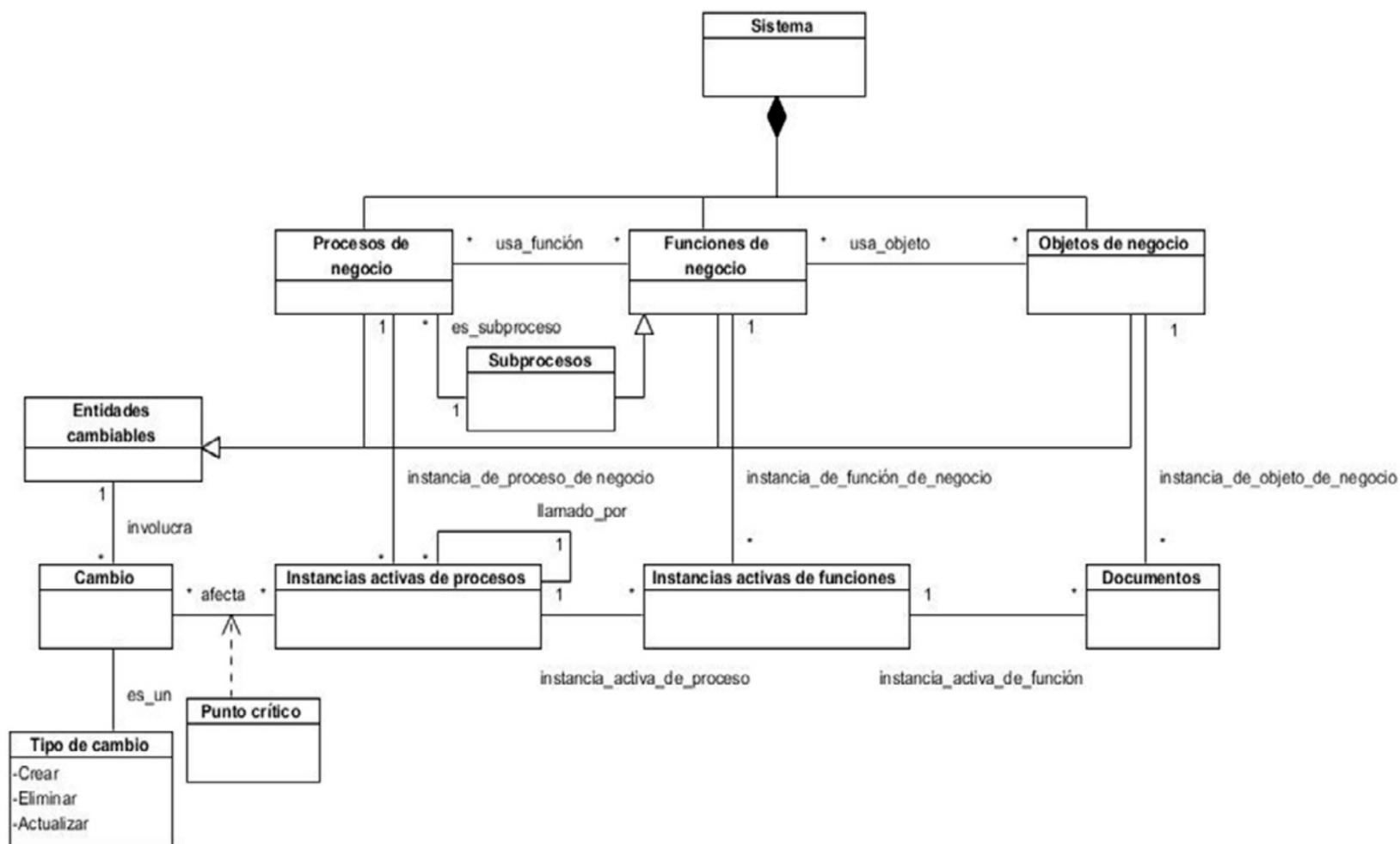


Figura 1. Metamodelo de dependencias de un sistema de gestión. Fuente: Elaboración propia.

El metamodelo integra perspectivas comunes de diferentes modelos propuestos en la literatura. La parte de tiempo de diseño del metamodelo se deduce de considerar una arquitectura tradicional de tres niveles: interfaz de usuario, aplicación y capas de datos. La capa de aplicación en el medio se descompone en procesos de negocios.

En lo que respecta al tiempo de ejecución, las instancias activas de procesos representan la ejecución individual de los procesos de negocios. Las instancias activas de procesos crean instancias de funciones individuales, que pueden requerir la creación de instancias de objetos de datos específicos o documentos, para su ejecución.

Los cambios posteriores a la implementación se refieren a cambios para alinear el sistema en ejecución con los cambiantes requisitos comerciales. Estos cambios generalmente se refieren a la estructura de diseño estático de un sistema. Sobre la base de este supuesto, el conjunto de posibles modificaciones posteriores a la implementación se deriva de la combinación de las operaciones de creación, actualización y eliminación con los elementos en tiempo de diseño del metamodelo, es decir, entidades cambiables.

Una modificación propuesta puede afectar una o más instancias de un proceso en ejecución. Para cada caso, es necesario identificar un punto crítico de ejecución con respecto a un cambio propuesto. Este punto crítico es un punto particular en la ejecución del proceso, más allá del cual la ejecución de una instancia de proceso puede terminarse de manera segura después de la

implementación del cambio propuesto. Se proporciona una caracterización más precisa de este punto crítico en la siguiente sección al discutir los mecanismos de análisis de impacto.

La tabla 1 presenta una representación teórica de los conjuntos del metamodelo de la figura 1, que se utilizan para respaldar la presentación del mecanismo de análisis de impacto y la formalización de las métricas. Las funciones se representan como actividades del Modelo y Notación de Procesos de Negocio (BPMN, por sus siglas en inglés), mientras que los objetos se representan como objetos de datos BPMN.

Tabla 1. Representación teórica de los conjuntos del metamodelo. Fuente: elaboración propia.

Representación teórica del metamodelo		
Elementos	Notación	Predicados
Sistema (SYS)	$SYS = [PRO; FUN; OBJ; PIN; FIN; DOC]$	-
Procesos de negocios	$PRO = \{pro_p\} p = 1; \dots; P$	-
Funciones de negocios	$FUN = \{fun_f\} f = 1; \dots; F$	$pro_fun \subseteq PRO \times FUN$
Objetos de negocio	$OBJ = \{obj_o\} o = 1; \dots; O$	$fun_obj \subseteq FUN \times OBJ$
Instancias activas de un proceso (pro_p)	$PIN_p = \{pin_{p,i}\} i = 1; \dots; I$ $PIN = \bigcup_{p=1}^P PIN_p$	$pro_p \subseteq PRO \times PIN$
Instancias activas de una función (fun_f)	$FIN_f = \{fin_{f,j}\} j = 1; \dots; J$ $FIN = \bigcup_{f=1}^F FIN_f$	$fun_f \subseteq PIN \times FIN$
Instancias de documentos activos de un objeto (obj_o)	$DOC_o = \{doc_{o,k}\} k = 1; \dots; K$ $DOC = \bigcup_{o=1}^O DOC_o$	$obj_o \subseteq FIN \times OBJ$
Conjunto de entidades cambiables	$CHE = PRO \cup FUN \cup OBJ = \{che\}$	-
Cambio post-implementación	$ch_c = ch_{visjo}; ch_{nuevo}; actualizar$ $ch_c = ch_{visjo}; eliminar$ $ch_c = ch_{nuevo}; crear$	-
Conjunto de cambios post-implementación	$CH = \{ch_c\} c = 1; \dots; C$	$Involucra \subseteq CH \times CHE$ $Afecta \subseteq CH \times PIN$ $Involucra(ch_c, che), che \in CHE$ $Afecta(ch_c, pin), pin \in PIN$
Punto crítico (cp) de la instancia de un proceso ($pin_{p,i}$)	$cp: CH \times PIN \mapsto FIN$	-
Impacto de un cambio (ch_c) del sistema (en tiempo de diseño (DTI) y en tiempo de ejecución (RTI))	$IMP(ch_c, sys) = [DTI, RTI]$ $DTI(ch_c) = \{pin \in PIN \mid Afecta(ch_c, pin)\}$ $RTI(ch_c) = \{che \in CHE \mid Involucra(ch_c, che)\}$	

2. DISEÑO DEL MECANISMO DE ANÁLISIS DE IMPACTO

Dado un sistema y un cambio propuesto, el análisis del impacto es el problema de determinar qué predicados se involucran y se afectan evaluándolos a verdadero, es decir, para definir el impacto del tiempo de diseño y el del tiempo de ejecución, además para respaldar la terminación segura de las instancias del proceso afectado por el cambio.

La investigación se centra en la actualización de los cambios posteriores a la implementación, que abarca también la creación, eliminación y actualización de tipos de cambios.

La figura 2 muestra la vista del proceso de análisis de impacto de un cambio utilizando la notación BPMN. Las fases que se pueden automatizar completamente se modelan como tareas de servicio BPMN (ícono de una herramienta en la esquina superior izquierda de la tarea), mientras que las fases que pueden requerir intervención humana se modelan como tareas de usuario BPMN (ícono de una persona en la esquina superior izquierda).

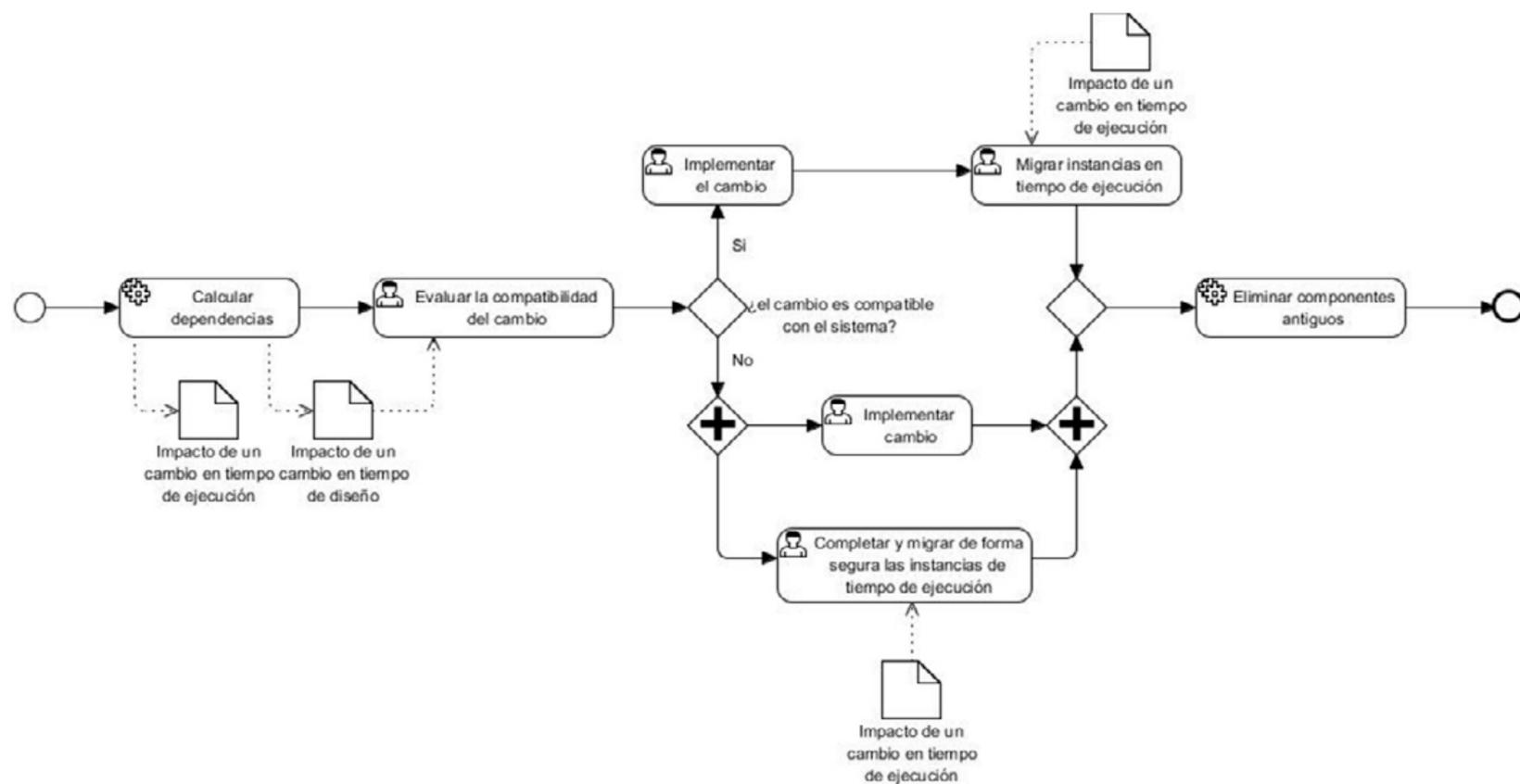


Figura 2. Mecanismo del análisis del impacto de un cambio. Fuente: elaboración propia.

Fase *Calcular dependencias*

En esta fase, las dependencias en el tiempo de diseño y en el tiempo de ejecución del cambio propuesto, se calculan en función del modelo de dependencia. Las dependencias de tiempo de diseño, son aquellas donde el cambio involucra un objeto de datos y todas las funciones tanto de entrada como de salida que se ven afectadas por el cambio. También forman parte todos los procesos que usan las funciones afectadas y, recursivamente, todos los procesos que usan los procesos afectados como subprocessos en algún punto y que se ven afectados por el cambio. En otras palabras, identificar el impacto en el tiempo de diseño implica navegar por las relaciones entre las entidades del metamodelo de la figura 1. En particular, las relaciones del subprocesso permiten tener en cuenta las dependencias transitivas entre los procesos que utilizan las funciones afectadas.

Las dependencias en tiempo de ejecución, comprenden las instancias de los procesos que se están ejecutando actualmente, es decir, iniciadas pero aún no terminadas.

Fase *Evaluar la compatibilidad en tiempo de diseño*

No todos los cambios son críticos para un sistema, de hecho, algunos cambios pueden ser perfectamente compatibles con las entidades afectadas por el cambio y, por lo tanto, con el sistema. Una actualización compatible de un objeto puede agregar un nuevo campo para capturar

más detalles sobre ese objeto. El nuevo objeto creado puede reemplazar la versión anterior, ya que los procesos y las funciones que la usan pueden simplemente ignorar el nuevo campo. En este caso se dice que la nueva versión de una entidad es compatible con la versión antigua.

La evaluación de la compatibilidad entre los objetos de datos, las funciones o los servicios y los procesos de negocios implica verificar la similitud sintáctica y semántica entre dos versiones de la misma entidad. La similitud sintáctica se refiere a la similitud entre las especificaciones de diferentes entidades, por ejemplo, la concordancia entre las firmas de dos funciones o la concordancia sintáctica entre las etiquetas de actividad en los modelos de un proceso. La similitud semántica se refiere a la similitud del significado de dos entidades, por ejemplo, dos funciones de negocios son semánticamente similares si implementan la misma función entre los parámetros de entrada y salida, o dos procesos son semánticamente similares si producen los mismos resultados para el cliente.

Sobre la base de si un cambio propuesto es compatible con el sistema existente, esta investigación propone dos rutas diferentes (ver figura 2). El caso trivial, es el de la compatibilidad de la versión antigua de una entidad con la nueva versión. Este caso implica la implementación del cambio y la migración de las instancias en ejecución afectadas a la nueva versión de la entidad, después de lo cual se puede eliminar la versión anterior. El otro caso es el de la no compatibilidad, para el cual la versión anterior de una entidad se puede eliminar solo después de que las instancias de proceso afectadas por el cambio se hayan llevado a un estado seguro con respecto al cambio propuesto.

Fase *Implementar cambio (caso de compatibilidad y no compatibilidad)*

Esta tarea implica que la nueva entidad en un cambio propuesto se implementa en el sistema. Se configura una nueva versión del objeto o se diseña e implementa un nuevo proceso. En la práctica, la implementación de una nueva entidad en un sistema puede variar desde hacer cambiar algunos parámetros en la configuración del sistema, lo que se puede hacer muy rápidamente, hasta el desarrollo de un nuevo código o la personalización de la base de código existente del sistema.

Fase *Migrar instancias en tiempo de ejecución (caso de compatibilidad)*

Si una nueva entidad es totalmente compatible con el sistema, o al menos es semántica y sintácticamente similar a la versión antigua, entonces se puede decir que es capaz de sustituir perfectamente a la entidad existente sin un impacto específico. La migración de una entidad en tiempo de diseño, por ejemplo, un proceso de negocio o un objeto de negocio a una nueva entidad significan asegurarse de que una entidad utiliza la nueva entidad en cualquier instancia futura.

Fase *Completar y migrar de forma segura las instancias en tiempo de ejecución (caso de no compatibilidad)*

La no compatibilidad de un cambio con el sistema existente implica que las instancias actualmente en ejecución afectadas por un cambio no pueden terminar su ejecución, debido a que

no pueden usar una nueva versión de una entidad. En este caso, para cada instancia en ejecución afectada, se define un punto crítico (*cp*) de ejecución con respecto al cambio propuesto. Este punto crítico se define como la última actividad en una instancia de proceso que utiliza la entidad involucrada en el cambio. Una instancia de proceso afectada por el cambio es segura si su ejecución está actualmente más allá del punto crítico determinado por el cambio, mientras que no es seguro de lo contrario. El propósito de esta fase es identificar las instancias críticas del proceso y llevarlas a una ejecución completa segura, esto se logra asegurando que la entidad no se elimine hasta que todas las instancias del proceso crítico hayan pasado su respectivo punto crítico.

Fase *Eliminar componentes antiguos (casos de compatibilidad y no compatibilidad)*

Finalmente, una vez que el cambio se ha implementado y todas las instancias en ejecución afectadas se migran a nuevas versiones, las entidades antiguas se pueden eliminar del sistema. Esta tarea puede variar algunos parámetros en la configuración del sistema y puede que afecte la base de código del sistema.

3. DISEÑO DE LAS MÉTRICAS DE EVALUACIÓN DE IMPACTO

Las decisiones sobre la implementación del cambio son tomadas generalmente por el comité de control de cambios y por los jefes de proyecto, que pueden o no tener un conocimiento específico sobre la estructura interna y el panorama del tiempo de ejecución del sistema. Por tanto, la información obtenida del análisis de impacto se debe sintetizar en un conjunto de métricas que le sean de utilidad para los que toman las decisiones.

Esta investigación propone un conjunto de métricas para cuantificar el impacto de un cambio propuesto. A pesar de que estas métricas han sido evaluadas cualitativamente por expertos, no es propósito de esta investigación proponer escalas de medición validadas para estas métricas, se aspira para trabajos futuros.

Se proponen tres niveles de impacto de un cambio: entidad, fase y sistema.

A nivel de entidad, el impacto de un cambio se divide en el porcentaje de procesos, funciones, objetos, instancias de procesos, instancias de funciones o documentos que pueden verse afectados por el cambio. Para ello se proponen las métricas de la figura 3.

Utilizando la ponderación aditiva simple (Fernández Pérez, 2018) se calcula el impacto de un cambio. Se tienen en cuenta las *n* variables

$$IMP^{PRO}(ch) = \frac{\{pro \in PRO \mid involucra(ch,pro)\}}{|PRO|} \quad (1)$$

$$IMP^{FUN}(ch) = \frac{\{fun \in FUN \mid involucra(ch,fun)\}}{|FUN|} \quad (2)$$

$$IMP^{OBJ}(ch) = \frac{\{obj \in OBJ \mid involucra(ch,obj)\}}{|OBJ|} \quad (3)$$

$$IMP^{PIN}(ch) = \frac{\{pin \in PIN \mid afecta(ch,pin)\}}{|PIN|} \quad (4)$$

$$IMP^{FIN}(ch) = \frac{\{fin \in FIN \mid afecta(ch,fin)\}}{|FIN|} \quad (5)$$

$$IMP^{DOC}(ch) = \frac{\{doc \in DOC \mid afecta(ch,doc)\}}{|DOC|} \quad (6)$$

Figura 3. Métricas de evaluación de impacto de cambio. Fuente: elaboración propia.

Nota: las variables que se utilizan en la figura 3 se explican de forma detallada en la Tabla 1.

consideradas más importantes por parte de los expertos. Para obtener esa valoración, cada una de estas variables es ponderada en relación a su importancia relativa con respecto al resto de las variables aplicando el método suma ponderada. Se le asigna un peso a cada variable con un valor del uno al cinco teniendo en cuenta la importancia de las variables, este peso aparece fijo para cada una de ellas.

$$\omega_n = \frac{c_n}{\sum_{i=1}^k c_i} \quad (7)$$

Dónde:

ω_n : Peso normalizado de cada variable,

C_n : Peso asignado a la variable y

k : Total de variables

El primer nivel de agregación está asociado a la dimensión de fase, es decir, entre el impacto en tiempo de diseño $IMP^{DT}(ch)$ y tiempo de ejecución $IMP^{RT}(ch)$. Los pesos ω_l y ω_m capturan la importancia relativa de las entidades para una fase determinada. Los pesos de ω_l son de las entidades asociadas al impacto en tiempo de diseño y los pesos de ω_m son asociados al impacto en tiempo de ejecución, utilizando la fórmula:

$$IMP^{DT}(ch) = \sum_{l \in \{PRO, FUN, OBJ\}} \omega_l * IMP^l(ch) \quad (8)$$

$$IMP^{RT}(ch) = \sum_{m \in \{PIN, FIN, DOC\}} \omega_m * IMP^m(ch) \quad (9)$$

Las métricas de evaluación de impacto pueden agregarse aún más a nivel del sistema. Los pesos y capturan la importancia relativa de un impacto de cambio en las entidades de tiempo de diseño y tiempo de ejecución en un sistema respectivamente. El impacto global de un cambio propuesto en un sistema se puede expresar como:

$$IMP^{SYS}(ch) = \omega_{DT} * IMP^{DT}(ch) + \omega_{RT} * IMP^{RT}(ch) \quad (10)$$

El primer caso se aplica para los escenarios de cambio estándar, que pueden ser relevantes para la mayoría de las organizaciones que utilizan un sistema, mientras que el último método es preferible cuando la percepción del impacto del cambio se debe a cambios específicos de la organización, industria, tipo de procesos o productos. Luego de obtenido en impacto, el resultado se interpreta utilizando los siguientes umbrales definidos para este tipo de escenario:

- Bajo si, $0 < IMP(ch) \leq 0.4$
- Medio si, $0.4 < IMP(ch) \leq 0.7$
- Alto si, $0.7 < IMP(ch) \leq 1$

Los valores de los pesos y los umbrales que caracterizan las métricas de evaluación de impacto pueden ser determinados y validados por expertos o por los responsables de la toma de decisiones, sin embargo, no es propósito de esta investigación proponer escalas de medición validadas para estas métricas, se aspira para trabajos futuros.

VALIDACIÓN DEL MÉTODO

Los métodos de validación para investigaciones científicas en la rama de la informática, se enfocan en demostrar la relevancia del problema que resuelve y el rigor con que fueron construidos (Kataev *et al.*, 2016), para garantizar la incorporación de una contribución científica al sustento de la propuesta (García Rodríguez, 2018; Niño Benitez, 2018). Una vez demostrada la valía de la investigación, es importante valorar la satisfacción de los clientes para con la propuesta (Silega Martínez *et al.*, 2015). En la validación de la presente investigación los métodos empleados también fueron utilizados en tesis doctorales y de maestría relacionadas con el objeto de la investigación y la calidad de *software*.

Uno de los métodos empleados fue el Caso de Estudio aplicado al proyecto Desarrollo del Sistema de Gestión de la Empresa de Seguros Nacionales del Centro de Informatización de Entidades. Se decidió aplicar un caso de estudio para valorar el efecto de la implementación del método elaborado. Como resultado se obtuvo el impacto negativo provocado por los cambios en tiempo de ejecución antes y después de aplicado el método elaborado.

Los resultados obtenidos antes de aplicar el método propuesto expone que la solución desde el punto de vista arquitectónico no fue compleja sin embargo requiere de abundantes cambios en la aplicación y en las funcionalidades del sistema. La solución tuvo en cuenta el impacto en tiempo de diseño mientras que en el impacto en tiempo de ejecución se afectaron el 68 % de las instancias que se estaban ejecutando en ese momento. La información de las instancias que se encontraban en ejecución no pudo ser salvada exceptuando la que se encontraba en el último evento de la ejecución. La reacción de los clientes y usuarios no fue favorable. La solución de la solicitud de cambio fue ejecutada en un tiempo estimado de ocho horas con una complejidad alta y fue realizada por un desarrollador.

Luego de aplicado el método elaborado el impacto obtenido en tiempo de diseño es de 25.1 % y en tiempo de ejecución el 27.6 % del total de instancias que se estaban ejecutando en ese momento considerándose bajo el impacto negativo provocado por el cambio para ambos tiempos. La información no sufrió pérdidas tanto la que se encontraba en el sistema como la que se estaba en ejecución en el momento del cambio. La reacción de los clientes y usuarios fue favorable. La solución de la solicitud de cambio fue ejecutada en un tiempo estimado de dos horas y 30 minutos con una complejidad alta y fue realizada por un desarrollador.

Además, fueron aplicados el método grupo focal con el propósito de que surjan criterios, actitudes, reacciones y experiencias entre los participantes respecto al método desarrollado y la técnica *Iadov* para verificar la satisfacción de los usuarios finales.

Para aplicar el método grupo focal se conformaron tres equipos de trabajo con personas que cuentan con experiencia en la temática y fueron convocados en dos momentos. En el primer encuentro se presentó el método desarrollado como solución al problema planteado donde cada grupo emitió su criterio proponiendo elementos válidos que se tuvieron en cuenta

para las recomendaciones y el desarrollo futuro de la investigación. En un segundo momento se convocó a los tres grupos unidos con el objetivo de llegar a un consenso final sobre la validez, utilidad y novedad de la solución propuesta. Como resultado final de la aplicación de este método se obtuvieron las siguientes conclusiones:

- El conjunto de componentes elaborado permite que al finalizar su ejecución se disminuya el impacto negativo de la gestión de los cambios post-implementación.
- La solución desarrollada permite calcular impacto del cambio para una entidad, para una fase y para todo el sistema.
- Impide la pérdida de información de las instancias que se encuentran en ejecución en el momento del cambio.
- Es novedosa y clara la propuesta de solución.

Con la aplicación de la técnica de Iadov se reafirman la validez y beneficios de aplicar la solución propuesta. Destacándose lo siguiente:

- Evita la pérdida de información del sistema en ejecución.
- Disminución del rechazo al cambio por parte de clientes y usuarios finales.
- Disminución del impacto negativo provocado al cambio por parte de clientes y usuarios finales.
- Ahorro de tiempo y recursos para arreglar determinado cambio.

La valoración satisfactoria por los clientes y los resultados alcanzados permite concluir que la solución propuesta valora el impacto negativo de los cambios y tiene en cuenta las dos etapas (durante el desarrollo de *software* y cuando el *software* se encuentra en total ejecución). El método disminuye el impacto negativo provocado por las modificaciones post-implementación y demuestra su utilidad y contribución a la solución del problema.

CONCLUSIONES

- El desarrollo de la investigación, permitió identificar soluciones relacionadas con las modificaciones post-implementación. Se identificó que la mayoría de las soluciones estudiadas abordan la post-implementación centrada en la organización siendo menor la cantidad que definen acciones enfocadas al sistema.
- El método desarrollado para el análisis de las modificaciones post-implementación en sistemas de gestión de *software* permite calcular el impacto negativo provocado por un cambio y si es factible o no ejecutarlo.
- El conjunto de componentes elaborados permite mapear las dependencias entre las entidades en tiempo de diseño y en tiempo de ejecución de una entidad, fase y sistema. Además, posibilita detectar dependencias relevantes entre las entidades para una determinada solicitud de cambio y utiliza un grupo de métricas para la evaluación del impacto negativo.
- El método fue validado utilizando como eje principal de validación un caso de estudio para probar el efecto de la implementación en cuanto a las variables tiempo e impacto.

REFERENCIAS

- Alsulami, M., Rahim, M., & Scheepers, H. (2013). Development of a model to understand how consultants manage conflicts during ERP post-implementation change process: A dialectic perspective. 24th Australasian Conference on Information Systems, 1-11.
- Cacha Ramos, J. F., De la Cruz Vera, C. K., De la Torre Toutin, S., & Pareja Castañeda, A. J. (2021). Factores críticos de éxito para la implementación de la hipoteca inversa en Lima Metropolitana y Callao. <https://repositorio.esan.edu.pe///handle/20.500.12640/2377>
- Fernández Pérez, Y. (2018). Modelo computacional para la evaluación y selección de productos de *software*. <https://repositorio.uci.cu/jspui/handle/123456789/7871>
- García Rodríguez, A. M. (2018). Modelo de Recomendación de Escenarios al iniciar la Mejora de Procesos de *Software*. <https://repositorio.uci.cu/jspui/handle/123456789/7867>
- Herederero, C. de P., Agius, J. J. L. H., Romero, S. M.-R., & Salgado, S. M. (2019). Organización y transformación de los sistemas de información en la empresa. ESIC.
- Kataev, M., Bulysheva, L., Emelyanenko, A., & Bi, Z. (2016). Enterprise Diagnostics for Evaluation of Enterprise Business Processes. *Journal of Industrial Integration and Management*, 01(02), 1650008. <https://doi.org/10.1142/S2424862216500081>
- Lozano, C., & Lisbel, J. (2017). Factores en la fase de post-implementacion que influyen en los logros de los beneficios esperados en sistemas ERP. Repositorio de Tesis - UNMSM. <https://cybertesis.unmsm.edu.pe/handle/20.500.12672/6411>
- Moalagh, M., & Ravasan, A. Z. (2013). Developing a practical framework for assessing ERP post-implementation success using fuzzy analytic network process. *International Journal of Production Research*, 51(4), 1236-1257. <https://doi.org/10.1080/00207543.2012.698318>
- Niño Benitez, Y. (2018). Guía para la gestión del requisito no funcional seguridad en el desarrollo de aplicaciones web. <https://repositorio.uci.cu/jspui/handle/123456789/7916>
- Palinkas, L. A., Horwitz, S. M., Green, C. A., Wisdom, J. P., Duan, N., & Hoagwood, K. (2015). Purposeful Sampling for Qualitative Data Collection and Analysis in Mixed Method Implementation Research. *Administration and Policy in Mental Health and Mental Health Services Research*, 42(5), 533-544. <https://doi.org/10.1007/s10488-013-0528-y>
- Parhizkar, M., & Comuzzi, M. (2017). Impact analysis of ERP post-implementation modifications: Design, tool support and evaluation. *Computers in Industry*, 84, 25-38. <https://doi.org/10.1016/j.compind.2016.11.003>
- Ramos González, L., Silega Martínez, N., & Díaz Ramos, Y. (2018). Revisión sobre el análisis de modificaciones post-implementación en sistemas de gestión. *Revista Cubana de Ciencias Informáticas*, 12, 236-251.
- Ravasan, A. Z., & Mansouri, T. (2014). A FCM-Based Dynamic Modeling of ERP Implementation Critical Failure Factors. *International Journal of Enterprise Information Systems (IJEIS)*, 10(1), 32-52. <https://doi.org/10.4018/ijeis.2014010103>

Silega Martínez, N., Febles Rodríguez, J. P., & Noguera García, M. (2015). Método para la Transformación Automatizada de Modelos de Procesos de Negocio a Modelos de Componentes para Sistemas de Gestión Empresarial. <https://repositorio.uci.cu/jspui/handle/ident/8584>

Yu, C. (2005). Causes influencing the effectiveness of the post-implementation ERP system. *Industrial Management & Data Systems*, 105(1), 115-132. <https://doi.org/10.1108/02635570510575225>

Copyright © 2021 Ramos-Gonzalez, L., Diaz-Ramos, Y., Silega-Martinez, N.



Este obra está bajo una licencia de Creative Commons Atribución-No Comercial 4.0 Internacional