

Estandarización de los procesos asociados al desarrollo de proyectos informáticos: un caso de estudio

Samuel Sepúlveda Cuevas y Ania Cravero Leal

Universidad de la Frontera, Departamento de Ingeniería de Sistemas,
Centro de Estudios en Ingeniería de Software (CEIS), Temuco,
Chile

{samuel.sepulveda, ania.cravero}@ceisufro.cl

Resumen Durante los últimos años la industria del software ha experimentado un gran crecimiento, junto a una mayor exigencia de los clientes y usuarios respecto de las prestaciones y calidad de sus productos. Esto ha generado necesidades crecientes de las empresas por incorporar estándares de calidad en sus procesos. La contribución de este trabajo es presentar una propuesta que ayude a estandarizar los procesos asociados al desarrollo de los proyectos informáticos pertenecientes a una empresa de servicios, a partir del uso de estándares y buenas prácticas de la industria del software. La propuesta analiza, clasifica y discrimina los tipos de proyectos, requerimientos y recursos necesarios para los proyectos que enfrenta la empresa. Los resultados de la evaluación inicial de esta propuesta muestran que ha tenido efectos positivos sobre la calidad del producto final y satisfacción de sus clientes internos.

Palabras clave. Proyectos Informáticos; Mejora de Procesos; Calidad del Software; Investigación-Acción.

Standardization of the Processes Associated with the Development of Informatics Projects: a Case Study

Abstract. In recent years the software industry has experienced tremendous growth, with a greater demand from customers and users regarding the performance and quality of their products. This has led to growing business needs to incorporate quality standards in its processes. The contribution of this paper is to present a proposal to help standardize processes associated with the development of IT projects belonging to a service company, from the use of standards and best practices in the software industry. The proposal analyzes, classifies and discriminates by project types, requirements and resources needed for the kind of projects that faces the company. The initial results show positive effects on

product quality and satisfaction of its internal customers.

Keywords. IT Projects, Process Improvement, Software Quality, Action Research.

1. Introducción

A partir de la denominada Crisis del Software [12], la calidad se ha instalado como un objetivo relevante en la comunidad de la Ingeniería del Software (IS), cuyas tendencias fueron importadas de los sistemas de calidad productivos y tuvieron su paralelo en la industria del software [26]. Las premisas en las que se basa la madurez de procesos del software es que la calidad de éste, está fuertemente influenciada por la calidad de los procesos de desarrollo y mantenimiento usados para construirlo, donde la Calidad del Software se traslada como tendencia de un foco en el producto a un foco en el proceso [33]. De esta manera surgen propuestas como ISO/IEC12207 [22], ISO9000-3 [21], IEEE1074 [18], IEEE730 [17] y CMMI (Software Engineering Institute [43], entre otros.

No es hasta principios del siglo XX que el concepto de calidad se enmarca en un contexto industrial e ingenieril, cuya evolución hasta el día de hoy, ha permitido observar, estudiar y proponer mejoras en los productos desarrollados y los procesos asociados [28].

Actualmente la industria del software presenta un gran crecimiento y debe asumir las altas exigencias de los clientes y usuarios respecto de las prestaciones y calidad de sus productos [37]. Para cumplir con lo anterior, la IS se plantea como la disciplina encargada de todos los

aspectos relacionados con la producción de software, incluyendo la gestión y aseguramiento de calidad, mediante propuestas que apuntan hacia la forma de gestionar los proyectos [44], prueba de ello son los estándares mencionados.

Considerando la importancia de los conceptos antes señalados es que se presenta una propuesta llevada a cabo en la subgerencia de sistemas de una empresa de la ciudad de Temuco, región de La Araucanía, Chile. Dicha empresa pertenece a un consorcio de empresas de servicios sanitarios, con presencia en 3 regiones del país y que cuenta con más de 336,000 clientes, con una participación de aproximadamente un 9% del mercado.

El principal objetivo de este trabajo es presentar una propuesta que ayude a estandarizar los procesos asociados al desarrollo de los proyectos informáticos gestionados por la subgerencia de sistemas de la empresa antes mencionada. La motivación que guía este trabajo se origina en el proceso sostenido de certificación que la empresa ha llevado a cabo desde el año 2008 a la fecha, en el cual la subgerencia de sistemas busca ordenar sus procesos y otorgar un servicio de calidad a sus clientes tanto internos como externos. La idea de realizar esta propuesta obedece a la diversidad respecto del conocimiento y uso de modelos para procesos de software por parte de los profesionales de dicha subgerencia. Lo anterior se hizo con el fin de aprovechar los conocimientos previos de los profesionales que forman parte de la subgerencia de sistemas y lograr así una sinergia que facilitara la activa participación de los involucrados. El artículo puede resultar de interés para gerentes y profesionales tanto del área de desarrollo y gestión de proyectos informáticos, como de tecnologías de información en general.

El trabajo se enmarca en la Investigación-Acción (IA), metodología en la que quienes realizan la práctica trabajan en conjunto con los investigadores, para probar y refinar los principios, herramientas, técnicas y metodologías que están siendo desarrolladas para resolver ciertos problemas del "mundo real" [29]. Si bien nuestra propuesta se basa en IA presentando cuatro etapas, considera también elementos específicos de la IS, como el ciclo de vida secuencial [44] y de la gestión de calidad, como

el ciclo de Deming [7]. Lo anterior difiere con lo mencionado por [2, 25], cuyos trabajos establecen que la IA comprende tres y cinco etapas respectivamente.

La estructura del artículo presenta en la segunda sección antecedentes relacionados con la calidad y mejora del proceso de software. La sección tres presenta un conjunto de trabajos relacionados con la propuesta diseñada. La sección cuatro muestra el diseño de la propuesta, a continuación, en la sección cinco se presenta el caso de estudio y los resultados iniciales. Finalmente, en la sección seis se resaltan las principales conclusiones y consideraciones futuras de la propuesta.

2. Antecedentes sobre calidad y mejora de procesos del software

A continuación se presentan una serie de antecedentes que dan cuenta de la calidad y mejora de procesos del software, las ventajas que acarrea el implementar estrategias de gestión de la calidad dentro de las organizaciones (aumento de competitividad, reducción de costos, fidelización de clientes, etc.), que han sido documentadas en textos clásicos como [7, 24, 41].

2.1. La calidad del software y su aseguramiento

Es posible observar cómo la industria del software ha incorporado conceptos propios del control y gestión de calidad, en no más de 40 años, a diferencia de otras que en casi 200 años han alcanzado su madurez [26].

Si bien los conceptos ya mencionados están ligados a procesos y productos de la manufactura, es interesante ver cómo las exigencias de calidad se trasladan a industrias centradas en los servicios como la del software. Así en el contexto de la Ingeniería de Software, el concepto de calidad puede entenderse como el grado en el que el software posee una combinación deseada de atributos como la confiabilidad o interoperabilidad, entre otros [16]. La calidad del software no es directamente comparable con la calidad en la manufactura,

presentando al menos 3 problemas propios de su contexto: *atender requerimientos que no se establecen directamente en la especificación del sistema, existencia de ciertas características de calidad complejas de especificar en forma no ambigua y dificultades en desarrollar un sistema que satisfaga las expectativas del usuario* [44].

El asegurar la calidad de un producto de software implica la realización de actividades que permitan asegurarla durante el desarrollo del producto, el cumplimiento de procedimientos y estándares definidos, garantizando así que éste cumple con las especificaciones definidas [22]. El concepto Aseguramiento de Calidad del Software (ACS) se usó por primera vez en 1974 dentro de la especificación de software MIL-S-52779 del gobierno estadounidense, planteando exigencias relacionadas con la administración del desarrollo del software. Tanto expertos en Calidad como en IS [7, 19, 44], establecen la relación existente entre la calidad asociada al proceso y su impacto en la calidad del producto. Si bien esto deriva de procesos para la manufactura, es posible adaptarla para procesos de software, no sin discrepancias entre los expertos [1, 15, 44].

2.2. Modelos asociados a la mejora del proceso de desarrollo de software

Para la Mejora del Proceso de Desarrollo de Software (MPS) se han propuesto diversas iniciativas, las cuales reportan casos de éxito o recomendaciones que enfatizan en aspectos como: (1) el uso de diferentes marcos de proceso, estándares o buenas prácticas de la industria y (2) métodos formales usados para la gestión y el desarrollo de proyectos de software [3, 5, 6, 14, 34]. La propuesta que aquí se presenta comparte los aspectos fundamentales con estas iniciativas relacionadas con la estandarización y control de los procesos asociados al desarrollo de proyectos informáticos, pero que al ser una adaptación basada en algunos de los estándares de la industria como IEEE730 y ISO/IEC12207 [17, 22], obviamente difiere en algunos aspectos cuyo detalle respecto de los instrumentos desarrollados y las adaptaciones realizadas se muestra en la sección 3.

Si bien el ACS se ha centrado en el producto, de un tiempo a esta parte los investigadores se han enfocado en el proceso asociado a la IS [46]. Durante la última década se ha visto una proliferación de propuestas que reportan resultados de iniciativas de mejoras, así [9] determina seis factores críticos de tipo organizacional que impactan directamente en una MPS exitosa, por otro lado [4] identifican los aspectos clave para la gestión de procesos de software. También es posible identificar barreras críticas para su éxito, entre ellas: *grupo humano poco experimentado y no tener una metodología clara para implementarla* [30]. Otra perspectiva que apunta a obtener experiencias exitosas al implantar MPS, plantea factores que permitirían acelerar el cambio de cultura y mantener los esfuerzos realizados [13]. Se ha mostrado cuantitativamente cómo se mejora de forma significativa el desempeño en la organización al alcanzar y mantener altos niveles de calidad en su proceso de software [10].

A continuación se presenta una breve descripción de los principales modelos y normas asociadas a la gestión y mejora del proceso de desarrollo de software, usados para el diseño de la propuesta:

(1) IEEE730-2002 [17], permite estandarizar la preparación y contenido de los planes de ACS, proporcionando los requisitos mínimos aceptables. En 1979 el IEEE emite la norma P-730 que hacía énfasis en la documentación y uso de revisiones formales en el desarrollo del software [28]. En su versión actual, éste define una lista de 16 secciones para el documento, las cuales consideran las distintas actividades que se deben llevar a cabo con el fin de definir y documentar el ACS del producto. Entre dichas secciones se encuentran: determinación de objetivos, documentación del proceso, definición de normas, prácticas, convenciones y métricas, gestión de riesgos, capacitación, planes de prueba y la presentación de informes de problemas y medidas correctivas, entre otras.

(2) ISO/IEC12207 es un marco de referencia desarrollado por ISO/IEC, que cubre todo el proceso de desarrollo de software, que incluye desde la conceptualización del producto hasta la entrega del mismo, así como procesos para la adquisición y suministros de productos y servicios

de software [22, 39]. Además ésta sugiere seguir la norma ISO 9126 para garantizar la calidad del producto y la ISO 9001 para el aseguramiento de la calidad [42].

Otros modelos y normas, que si bien se consideraron en un inicio para el diseño de la propuesta, pero que finalmente no se usaron son:

(1) *CMMI*, modelo desarrollado para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software [40]. Se basa en el concepto de cinco niveles que indican la madurez de los procesos alcanzados por la organización. Éste puede aplicarse en dos visiones: *Staged* y *Continuous* [33],

(2) *ISO9000-3*, diseñada para el desarrollo y la evaluación de software [21]. Podría utilizarse para regular el desarrollo de software, evaluando los procesos asociados [8]. El software debe ser validado como un todo en condiciones que sean similares al entorno de aplicación [45] y

(3) *ISO/IEC15504*, diseñada para la evaluación de procesos en donde se establecen los requisitos mínimos de entradas, salidas, insumos, productos, recursos y actividades [20]. Se requiere una estrategia para la recolección de datos y calificaciones de la capacidad del proceso basado en pruebas objetivas [38].

Si bien se reconoce la robustez y eficacia de dichos modelos y normas, estos no se consideraron finalmente en la propuesta debido fundamentalmente a razones de tiempo y costos asociados a su implementación, por lo que la gerencia general desestimó su implementación.

3. Trabajos relacionados

A continuación se presenta un conjunto de trabajos que presentan el diseño, aplicación y resultados de propuestas enfocadas en la realidad de las empresas de desarrollo de software latinoamericanas.

Se realizó un estudio a 114 empresas mexicanas de desarrollo de software, determinando que éstas tienen un conocimiento reducido de los modelos para evaluar la calidad del producto y procesos de desarrollo de software [11]. Por otro lado, se ha evaluado la posibilidad que pequeñas empresas de software se

introduzcan en procesos de mejora y puedan desarrollarlos de manera exitosa [36]. La realidad particular de seis países latinoamericanos, entre ellos Chile, es abordado en [23], donde los datos recopilados indican que si bien la situación de la Industria de Software es buena, está por debajo de países como Brasil y México en cuanto a la implementación de modelos de calidad.

Por su parte el proyecto Competisoft [32], provee un marco de referencia para la mejora y certificación de procesos de software, desarrollado especialmente para pequeñas empresas latinoamericanas. A su vez éste se basa en el modelo MoProSoft [31], que propone la definición de una nueva estructura de procesos y elementos de documentación, que ha sido aplicado con éxito en cuatro empresas mexicanas de desarrollo de software.

La propuesta Light MECPDS, presenta un modelo ligero de evaluación de calidad de procesos de desarrollo de software para motivar a las empresas colombianas a mejorar sus procesos y alcanzar niveles de madurez que garanticen su competitividad internacional [35]. Otra experiencia es el proyecto MOSCA [27], el cual persigue evaluar la calidad del software integrando los modelos de calidad del producto y de calidad del proceso, y que ha sido probado con éxito en dos empresas venezolanas de desarrollo de software.

La propuesta de estandarización que se presenta comparte el diagnóstico en cuanto a la necesidad e importancia de introducir procesos de mejora en la Industria del Software, pero también reconocemos que el alcance de ésta es más bien limitado y que de probar su éxito inicial en la empresa, se requiere pensar en un marco de trabajo más amplio como los ya presentados.

Se comparte con el estándar IEEE730 la definición de actividades que se deben llevar a cabo con el fin de definir y documentar el ACS del producto y por otro lado con la norma ISO/IEC12207 se siguen las etapas definidas para el desarrollo de un producto de software. La selección de dichos estándares se determinó a partir no sólo de factores técnicos como las etapas y productos definidos por dichos estándares, que se ajustan a lo realizado actualmente por la organización. Otro factor considerado dice relación con aspectos

organizacionales, pues si bien se considera que las demás propuestas mostradas son muy completas, el énfasis de la gerencia general estaba puesto en no incurrir en nuevas capacitaciones, tanto por razones de tiempo como por los costos que esto implica. Además, se consideró el conocimiento previo de ambas normas por parte de los profesionales de esta unidad de la empresa, con lo cual se buscó aprovechar los conocimientos previos y lograr así una sinergia y activa participación de los involucrados, tratando así de aumentar la factibilidad operacional de la propuesta.

4. Diseño de la propuesta de estandarización de proyectos informáticos

Si bien se ha mostrado la existencia de estándares y normas para la mejora de procesos de proyectos informáticos, actualmente la subgerencia de sistemas no cuenta con una metodología que estandarice su forma de gestionar sus proyectos entre los diferentes jefes de proyecto (JP) y sus equipos de trabajo.

El diseño que se presenta implicó realizar dos ciclos de IA, con una duración total de aproximadamente 4 meses. En el primer ciclo se definieron las principales actividades y tareas a desarrollar, observándose en general un alto nivel de acuerdo con los conceptos revisados y las actividades consideradas. Luego en el segundo ciclo se detallaron cada una de estas tareas y actividades, siendo este ciclo más complejo que el anterior, pues la necesidad de alcanzar acuerdos sobre definiciones de tareas específicas implicó considerar los diversos puntos de vista y necesidades implicadas. El equipo de trabajo que considera la propuesta incluye principalmente a los profesionales que se desempeñan como JP, a los cuales se agrega la presencia externa del investigador.

También se deben considerar las aportaciones hechas por los analistas y desarrolladores de la subgerencia de sistemas durante las reuniones de trabajo. A continuación se detallan las etapas, actividades e instrumentos que se definieron para la propuesta que intenta solucionar esta problemática.

4.1. Situación previa a la propuesta

La subgerencia de sistemas, ubicada en la ciudad de Temuco, es la encargada de entregar a las empresas del consorcio diferentes servicios tecnológicos, entre ellos la adquisición y desarrollo de sistemas informáticos. Estas necesidades son recopiladas a comienzo de cada año, pasando a formar parte del plan informático anual. También se considera que durante el transcurso del año pueden generarse requerimientos no contemplados en el plan, los cuales son priorizados y asignados a los JP en relación a su carga de trabajo y presupuesto.

Dicha subgerencia no tiene estandarizadas las etapas que debe cumplir cada proyecto. No obstante lo anterior, se hace uso de herramientas y procedimientos de apoyo que cada JP usa discrecionalmente, fruto de lo cual los productos de software que se adquieren o desarrollan son de calidades disímiles y no existe sinergia en el área.

Lo anterior repercute en que no existe una estandarización de los sistemas corporativos, tanto en forma como en fondo. Esto produce ineficiencias en el proceso, ya que las especificaciones entregadas a los proveedores dependen del JP que las realice. Se observa también que la evaluación anual de los JP respecto del desempeño alcanzado se realiza en base a una medida más bien de tipo cualitativo y no cuantitativo, pues no existen métricas para evaluar el cumplimiento de los objetivos trazados como prestadores de servicios dentro de la organización.

A partir de la situación existente en la subgerencia de sistemas y el uso de la IA, es que la propuesta consta de las siguientes etapas: (1) *analizar*, (2) *construir*, (3) *implantar* y (4) *probar*. Se muestran en la Tabla 1 las actividades a desarrollar para cada una de ellas. El detalle de cada etapa y sus actividades se explican en la siguiente sección.

4.2. Etapa de análisis

En esta etapa se realizó una revisión bibliográfica para conocer el estado del arte respecto de la calidad, del ACS y la MPS, entre otros temas relevantes. A partir de ésta se

identificaron los principios básicos a considerar dentro de la propuesta.

Tras esto, se decidió que los estándares que se usarían como base para la propuesta serían: IEEE730 e ISO/IEC12207 [17, 22]. La decisión se basó fundamentalmente en el conocimiento de los JP y desarrolladores de las normas ISO implementadas en otras unidades de la empresa y por otro lado que el modelo de base usado para el desarrollo de los proyectos es el ciclo de vida secuencial o de cascada, el cual es ampliamente conocido por los desarrolladores.

Luego se define el objeto de estudio, en nuestro caso el proceso asociado al desarrollo de proyectos informáticos, para continuar con un levantamiento de los procedimientos y procesos de éste, su organización y la situación actual, con la problemática que se desea resolver. La justificación y pertinencia de la propuesta se entrega a la gerencia a través de un anteproyecto y es discutida posteriormente en reuniones de trabajo, basando la argumentación en el estado del arte recabado y la caracterización de la situación actual de la organización. Esta etapa se da por finalizada con la corrección y actualización

de la propuesta y posterior aprobación formal por parte de la gerencia.

4.3. Etapa de construcción

Una vez aprobada la propuesta, se planifican y realizan las reuniones con los integrantes del equipo de trabajo, cuyo objetivo es consensuar los conceptos básicos para elaborar una metodología de apoyo al desarrollo de los proyectos informáticos (tipificación y etapas de los proyectos informáticos, productos de cada etapa, estándares de cada producto y la definición de los hitos de verificación). Luego se lleva a cabo un análisis comparativo con los estándares y normas de la industria revisadas en la etapa anterior, efectuándose los ajustes que se consideren necesarios.

4.3.1. Definir tipos de proyectos informáticos

Esta primera etapa de la construcción de la propuesta consiste en realizar una tipificación de los proyectos informáticos que realiza la

Tabla 1. Etapas y actividades definidas para la propuesta

Etapas	Actividades
Analizar	Revisión bibliográfica. Identificar los principios de calidad. Definir objeto de estudio. Identificar la organización. Analizar situación actual de estudio. Justificación y pertinencia del trabajo.
Construir	Definir tipos de proyectos informáticos. Definir etapas de un proyecto informático. Definir los productos de cada etapa. Definir estándares para cada producto. Definir relación: tipo proyecto–producto-etapa. Definir hitos de verificación. Definir objetivos del aseguramiento de calidad del software.
Implantar	Evaluar impacto de las acciones a implementar. Implantar propuesta dentro de los equipos de trabajo. Corregir posibles desviaciones.
Probar	Implantar propuesta en la organización. Realizar ajustes si fueran necesarios.

subgerencia de sistemas; luego se procede a determinar las etapas que debiera cumplir cada desarrollo de proyecto y posteriormente los productos generados en cada una de las etapas. Finalmente, se conjugan todos los elementos anteriores a través de lo que se define como la matriz tipo-etapa-producto.

Esta tipificación de proyectos se logra analizando los planes anuales de la subgerencia de sistemas de los años 2006, 2007 y 2008; la información obtenida fue consensuada con los JP, quienes son los responsables de ejecutar dichos planes. La tipificación arroja como resultado cuatro tipos de proyectos: *Soluciones de mercado empaquetadas* (T1), *Soluciones de mercado con desarrollo* (T2), *Desarrollo Interno-Externo* (T3) y *Desarrollo Interno* (T4). Además de los tipos obtenidos, se identifican dos atributos para caracterizar cada proyecto, que ayudan a comprender el alcance de cada uno de ellos: *Envergadura* y *Recursos*, los cuales indican el tamaño o esfuerzo requerido para su desarrollo o adquisición y la fuente de recursos humanos usados para su desarrollo. Todo lo anterior se resume en la Tabla 2.

4.3.2. Definir etapas de un proyecto informático

La definición de las etapas de los proyectos informáticos de la subgerencia de sistemas se realiza adaptando el modelo del ciclo de vida en cascada [37] y a partir de esto se obtienen las etapas de: *Formalización* (F), *Análisis* (A), *Diseño* (D), *Compra* (Cp), *Construcción* (Co), *Pruebas*

(P), *Implantación* (I), *Aceptación* (Ac) y *Explotación* (Ex). Las etapas identificadas para los proyectos y las actividades/objetivos que se deben realizar se detallan en la Tabla 3, donde las etapas y actividades definidas están acorde con lo mencionado en [22].

4.3.3. Definir los productos de cada etapa

Esta actividad consiste en definir los productos o “*entregables*” de cada una de las etapas definidas anteriormente. Estos productos finalmente constituirán los elementos sobre los cuales se evaluarán los estándares de calidad de los proyectos informáticos de la subgerencia de sistemas. Los productos se definen en base a lo señalado en el IEEE Std 730-2002 [17] y al ciclo de vida caracterizado en ISO/IEC 12207 [22], adaptándolos a las necesidades locales, considerando las sugerencias y aportes de los JP del área, las cuales se recabaron mediante: reuniones grupales, entrevistas y sesiones de *brainstorming*, cuyo resultado se muestra en la Tabla 4.

Dicha adaptación considera aspectos tales como: (1) *necesidad de definir al interior de la subgerencia de sistemas los roles y tareas asociadas al área de calidad*, (2) *se omitió la definición explícita de un plan de gestión de la configuración, así como el uso de estándares de la industria para la documentación de los procesos*, (3) *se definieron estándares propios y convenciones a nivel interno para la gestión documental*.

Tabla 2. Tipos de proyectos y sus características

Tipo	Descripción	Envergadura	Recursos
(T1)	Sistema de mercado. No es posible incorporar desarrollos para soportar funcionalidad específica. Ej.: Fin700, Sonda.	Alto	Externos
(T2)	Sistema de mercado. Es posible incorporar desarrollos para soportar funcionalidad específica de la empresa. Ej.: INFOR, Datastream.	Medio-Alto	Externos
(T3)	Sistema a medida; análisis se realiza en forma interna y la construcción se cotiza a un proveedor externo.	Medio-Bajo	Internos
(T4)	Sistema a medida; todo el ciclo de vida del proyecto se realiza en forma Interna.	Medio-Bajo	Internos

Tabla 3. Etapas de un proyecto y sus objetivos/actividades

Etapa	Descripción	Objetivos / Actividades
(F)	Establecer la planificación formal del proyecto, alcance, equipo, plazos, etc.	Formalización de necesidad. Planificación general, plazos. Definición equipo de trabajo. Confeción de la ficha FAI.
(A)	Establecer lo que el sistema debe hacer, restricciones de funcionamiento y sus procesos de desarrollo.	Requisitos del sistema Asegurar que requisitos son alcanzables. Planificación detallada. Definición marco de proyecto Definición del ambiente TI.
(D)	Tomar las decisiones con respecto a la organización lógica del software.	Formalizar soluciones TI para el SI. Asignar RR materiales c/u de las funciones. Diseño general. Ajustar especificaciones del producto. Modelo de datos. Diseñar casos de prueba.
(Cp)	Generar la Licitación/Cotización, adjudicándola, y cerrando el proceso con la contratación o compra del producto.	Ficha FAC. Cotización / licitación. Adjudicación.
(Co)	Implementar un programa basado en especificación de diseño.	Implementar producto. Planificación detallada. Asignación de recursos. Programación.
(P)	Comprobar si el software satisface las especificaciones.	Pruebas de software. Corrección errores Documentación pruebas.
(I)	Migrar herramienta desde ambiente de desarrollo al de explotación.	Migrar ambiente explotación. Parametrización. Capacitación. Entrega de manuales.
(Ac)	Solicitante aprueba formalmente software.	Aceptación formal solicitante. Inicio de producción.
(Ex)	Software entra en producción.	Paso a sistemas. Soporte normal.

4.3.4. Definir estándares por cada producto

En esta actividad se establece un “*documento patrón*” para cada producto definido. Dicho patrón involucra asignar un nombre codificado con el cual se reconoce el estándar interno que cada producto debe cumplir en cada etapa. Este nombre codificado se hizo siguiendo la definición interna y usando correlativos numéricos que ya tenía en uso la organización según el manual de procesos definidos en su sistema de gestión de calidad acorde a la norma ISO9001.

4.3.5. Definir relación: tipo proyecto– producto-etapa

Esta actividad consiste en unir los resultados de las etapas anteriores conformando la Matriz Tipo-Etapa-Producto, de forma tal que es posible unir las definiciones anteriores a cada tipo de proyecto. Los resultados obtenidos pueden verse en la Tabla 5, donde se relacionan las 9 etapas definidas para cada uno de los 4 tipos de proyectos identificados, estableciendo qué productos debieran considerarse en cada caso,

Tabla 4. Etapas, productos y estándares definidos

Etapa	Producto	Estándar
(F)	Solicitud de herramienta informática.	REG-SIS-001-FOR
	Ficha proyecto informático.	REG-SIS-002-FOR
	Carta Gantt de actividades.	REG-SIS-003-FOR
	Ficha Proyecto Informático (FPI).	REG-SIS-004-FOR
	Acta de reuniones.	REG-MQS-000-007
(A)	Especificación requisitos de software	REG-SIS-010-ANA
	Acta de reuniones	REG-MQS-000-007
(D)	Modelo del sistema	REG-SIS-030-DIS
	Plan de pruebas de software.	REG-SIS-031-DIS
	Acta de reuniones.	REG-MQS-000-007
(Cp)	Ficha FAC.	REG-SIS-040-COM
	Ficha proveedor de software.	REG-SIS-041-COM
(Co)	Diccionario de datos	REG-SIS-050-CON
	Especificación de packages.	REG-SIS-051-CON
	Acta de reuniones.	REG-MQS-000-007
(P)	Registro de pruebas de software.	REG-SIS-060-PRU
	Acta de reuniones.	REG-MQS-000-007
(I)	Manual de usuario.	REG-SIS-070-IMP
	Manual de sistema.	REG-SIS-071-IMP
	Acta de reuniones.	REG-MQS-000-007
(Ac)	Acta recepción de software.	REG-SIS-080-ACE
(Ex)	Acta recepción de sistemas	REG-SIS-090-EXP
	Acta de reuniones	REG-MQS-000-007

de forma tal que pueda seguirse la pauta definida y hacer los chequeos correspondientes a medida que cada proyecto avanza en su desarrollo.

4.3.6. Definir hitos de verificación

La verificación de cada producto respecto a lo definido en la metodología se realiza al finalizar cada etapa del proyecto, acorde se defina en la carta Gantt de cada uno de estos. El producto de esta actividad debe ser un documento del tipo acta de reunión (REG-MQS-000-007 acorde al estándar definido), el cual debe contener como mínimo: *fecha de la revisión, proyecto, etapa revisada, resultados de la verificación (por producto), principales observaciones, responsable de la verificación.*

4.3.7. Definir objetivos del ACS

Una vez definida la metodología es necesario preguntarse ¿Quién realizará la función de asegurar la calidad del software? La respuesta natural es *el área de ACS*, dada la importancia estratégica que tiene y que actualmente ésta no existe, se estableció que debiera ser un equipo independiente de quienes desarrollen los sistemas y con dependencia directa del subgerente de sistemas. Su misión debe ser el establecer un conjunto de actividades planificadas y sistemáticas, necesarias para aportar confianza respecto de que los productos desarrollados satisfacen los requerimientos del cliente, en base a los estándares de calidad definidos. En particular las funciones debieran

Tabla 5. Definición de la matriz tipo-etapa-producto

Etapa	Productos	Tipos de Proyecto			
		(1)	(2)	(3)	(4)
(F)	Solicitud de herramienta informática.	s	s	s	s
	Ficha de proyecto informático.	s	s	s	s
	Carta Gantt de actividades.	s	s	s	s
	Ficha FAI.	s	s/n	s/n	n
	Acta de reuniones.	s	s	s	s
(A)	Especificación requerimientos de software	s	s	s	s
	Acta de reuniones	s	s	s	s
(D)	Modelo del sistema	s	s	s	s
	Plan de pruebas.	s	s	s	s/n
	Acta de reuniones.	s/n	s/n	s/n	s/n
(Cp)	Ficha FAC.	s	s/n	s/n	n
	Ficha proveedor de software.	s	s	s	n
(Co)	Diccionario de datos	s	s	n	n
	Especificación de packages.	s	s	s	s
	Acta de reuniones.	s/n	s/n	s/n	s/n
(P)	Registro de pruebas de software.	s	s	s	s
	Acta de reuniones.	s/n	s/n	s/n	s/n
(I)	Manual de usuario.	s	s	s	s
	Manual de sistema.	s	s	s/n	n
	Acta de reuniones.	s/n	s/n	s/n	s/n
(Ac)	Acta recepción de software.	s	s	s	s
(Ex)	Acta recepción de sistemas	s	s	s/n	n
	Acta de reuniones	s/n	s/n	s/n	s/n

ser: (1) Definir, revisar y actualizar los estándares de calidad internos, (2) Asegurar niveles de calidad adecuados de los sistemas, (3) Identificar posibles desviaciones en los estándares aplicados, (4) Disminuir el impacto del paso a producción de sistemas, (5) Estandarizar la metodología de trabajo para el desarrollo de proyectos informáticos y (6) Elaborar planes de Aseguramiento de Calidad efectivos para cada proyecto, en los que se establezcan claramente los hitos de revisión y auditoría.

Queremos enfatizar que la conformación, características operativas y otros temas referentes al área de ACS están fuera de los objetivos de este trabajo.

4.4. Etapa de implantación

Una vez definidos los aspectos básicos de la propuesta se procede a su implantación entre todos los miembros del equipo de trabajo. La implantación se materializa mediante la realización de una serie de reuniones de inducción y capacitación, distinguiendo fundamentalmente dos grupos de trabajo: JP y personal de características técnicas sin una participación activa durante la definición de la metodología. Con los JP se discutieron diferentes aspectos de la misma con el fin de evaluar el impacto de las acciones a implementar y corregir posibles desviaciones.

4.5. Etapa de prueba

Una vez que la metodología se ha implantado dentro del equipo de trabajo, corresponde implantarla en la organización. Para esto se comunica de manera formal a la gerencia, se prepara un informe y presentación donde se reportan las principales características y beneficios que esta propuesta pretende alcanzar. En paralelo se comunica a los demás departamentos de la subgerencia de sistemas y clientes internos pertenecientes a las empresas del consorcio. Finalmente se recogen las últimas observaciones y se realizan los ajustes considerados como necesarios.

En la siguiente sección se resumen los principales resultados obtenidos de aplicar la propuesta en un proyecto de software que fue tomado como plan piloto, tras lo cual se realizó una revisión inicial del impacto generado por la propuesta. Dicha evaluación se realizó a partir de entrevistas y reuniones de trabajo con los JP.

5. Caso de estudio

Una vez definidas todas las etapas de la propuesta, se realizó una prueba piloto con la finalidad de medir su impacto y usarlo como caso de estudio. Se seleccionó un proyecto informático en ejecución y se le aplicó la propuesta, para así evaluar su uso por parte del equipo de trabajo y observar aspectos como: *tiempos de finalización de etapas intermedias de los proyectos, controles de agenda de actividades y presupuestos*, entre otros.

5.1. Caso de estudio: plataforma web de consultas de proveedores

El sistema seleccionado como caso de estudio fue la Plataforma Web de Consultas de Proveedores, orientada a consultas del estado de pago de los documentos emitidos por los proveedores y canal oficial para atender consultas, requerimientos, reclamos y solicitudes. De acuerdo con lo mostrado en la Tabla 2, este proyecto se tipificó según sus características como del tipo T3. Siendo el primer pilotaje de la propuesta, se trabajó con un proyecto

desarrollado con recursos internos y de una envergadura mediana, pues se pueden apreciar aspectos de la gestión del proyecto y además la relación con agentes externos al equipo de trabajo, siendo a la vez un proyecto no crítico para la empresa.

A partir de lo definido en la Tabla 5 se establecieron los productos que se deben entregar para cada una de las etapas del proyecto. Entre los productos más representativos por etapa se definieron: *etapa 1 - ficha de proyecto, etapa 2 - especificación de requerimientos, etapa 3 - acta de reunión, etapa 4 - ficha de proveedor, etapa 5 - especificación de packages, etc.*

5.2. Resultados y observaciones iniciales

Dados los antecedentes de la propuesta diseñada y de la prueba realizada, se observó que los JP no tienen el hábito de formalizar el cierre de ciertos procesos o etapas, por lo que se carece de evidencias tangibles del término de cada etapa. Asimismo, a partir de las reuniones y entrevistas sostenidas con los JP, se observó que la propuesta es bien evaluada no obstante el incremento de trabajo administrativo que ello conlleva, en particular por el registro de actividades que comúnmente no eran consideradas para efectos de la gestión y desarrollo del proyecto. Además se pudo observar cómo la propuesta contribuyó a internalizar en los integrantes de la subgerencia de sistemas, el concepto de calidad como una actividad transversal al desarrollo de los proyectos y no como una actividad aislada que ocurre después de codificar y recepcionar los proyectos.

Se corroboró que algunos de los problemas históricamente surgidos, durante el desarrollo o implantación de los proyectos, no ocurrieron o su detección oportuna permitió que su impacto fuese menor. Entre estos problemas históricos se destacan la *falta de finalización formal de las etapas, falta de formatos únicos para la entrega de resultados o reportes, retrasos en las entregas y presupuestos sobrepasados*. Además el control del avance real del proyecto resultó mucho más certero ya que los miembros del equipo de trabajo presentaron una visión muy optimista al

informar sus avances. De lo anterior, para los dos primeros problemas mencionados, podría atribuirse la mejora observada al hecho de contar con estándares de documentación, por otro lado respecto de los retrasos y presupuestos se han comparado los datos de proyectos similares desarrollados previa a la definición, obteniéndose mejoras cercanas a un 30% en cuanto a tiempos de entrega del producto y una disminución del 20% respecto del manejo de los presupuestos estimados versus los reales.

El proyecto tomó cuatro meses para su desarrollo, con lo cual se puede afirmar que en un período de ocho meses se ha diseñado, probado y analizado los resultados del caso de estudio de la propuesta de estandarización de proyectos aquí presentada. Lo anterior considera los tres meses de su diseño y un mes para el análisis de los resultados.

Finalmente cabe destacar que una vez se efectúen más pruebas sobre su uso y se hagan las correcciones que se consideren necesarias, la propuesta sería agregada al sistema de gestión de calidad existente en la empresa, con el nombre de "*Metodología de Proyectos TIC*", por lo que quedaría así oficializada para todo el consorcio empresarial.

5.3. Lecciones aprendidas

Respecto de la propuesta diseñada y su posterior prueba, se puede establecer que algunas de las prácticas e instrumentos definidos en la propuesta, como por ejemplo las actas de reunión se utilizaban previamente, pero no con el nivel de formalidad y asociando su presencia a un tipo particular de proyecto.

Se pudo observar que con la aplicación de la propuesta, el rol del JP se orienta más a gestionar y supervisar el desarrollo del producto, delegando así tareas propias del análisis y diseño de la solución en los especialistas del equipo de desarrollo. Como efecto de lo anterior, es posible para el JP detectar posibles problemas o desviaciones respecto de la planificación en fases tempranas, a diferencia de lo sucedido previamente, pues se encontraba mucho más inmerso en el desarrollo del producto que en la gestión del proyecto.

Se debe destacar también la disposición de los profesionales que participaron en el desarrollo de la propuesta, lo cual se atribuye inicialmente en que vieron en ésta un beneficio, manifestando una impresión positiva hacia la misma.

5.4. Debilidades y amenazas

No obstante que los resultados iniciales observados son promisorios, estamos conscientes que existen amenazas a la validez de los resultados obtenidos y es importante destacar tres elementos a la hora de analizarlos en forma más crítica: (1) falta aún realizar más pruebas y pilotajes con proyectos reales, que incluyan al menos un caso de estudio para cada tipo de proyecto identificado, pues en estos varían las envergaduras, recursos y esfuerzos requeridos para su implementación, (2) la carencia de datos cuantitativos en los proyectos ya desarrollados, lo cual implica no tener una fuente de datos cuantitativa para hacer comparaciones y (3) la prueba de la propuesta sólo fue ejecutada por un equipo de proyecto y parte de los participantes estuvieron involucrados en el diseño de la misma.

6. Conclusiones y consideraciones futuras

A partir de los antecedentes presentados y haciendo uso de los principios básicos de la gestión de proyectos informáticos, el ACS y la IA se ha desarrollado y probado una propuesta de estandarización de los procesos asociados al desarrollo de los proyectos informáticos para el consorcio de empresas a las cuales presta servicios la subgerencia de sistemas, proporcionando de esta manera una metodología estándar para esta importante actividad.

Los estándares de la industria usados como referencia permitieron definir las bases de la propuesta y así determinar los productos para cada etapa, dependiendo de las características de cada proyecto de software, contribuyendo así a homologar las funciones del personal abocado a estas tareas.

Los resultados iniciales al aplicar la propuesta muestran avances significativos en la mejora de

los procesos asociados al desarrollo de los proyectos informáticos y qué ha sido muy valorada por los JP. Ejemplos de lo anterior serían: (1) contar con evidencias tangibles del término de cada etapa durante el proyecto y (2) registrar actividades que comúnmente no eran consideradas para efectos de la gestión y desarrollo del proyecto, entre otras. La propuesta desarrollada permite identificar y resolver las desviaciones que pudieran producirse, durante cada una de las etapas del desarrollo del proyecto informático y no al final del ciclo como ocurría anteriormente.

Como trabajo futuro queda pendiente el seguir haciendo pruebas que permitan evaluar los resultados y las correcciones que requiera la propuesta, como pueden ser aspectos asociados a proyectos de gran tamaño o con un alto nivel de incertidumbre inicial, para recién ahí implantarla e incorporarla de manera formal al sistema de gestión de calidad de la empresa. Además sería importante considerar el impacto que dicha propuesta podría sufrir si se incorporan nuevas metodologías para el desarrollo de software cómo las basadas en modelos ágiles. Queda pendiente la opción de considerar el uso y adaptación de marcos de referencia más amplios y desarrollados especialmente para Pymes de software como Competisoft o MoProSoft y evaluar sus resultados.

Agradecimientos

Este trabajo se ha realizado con el apoyo de la Dirección de Investigación, Vicerrectoría de Investigación y Postgrado de la Universidad de La Frontera, a través del Proyecto de Investigación DIUFRO D114-0065. Agradecemos también a la Dra. Cristina Cachero por sus indicaciones y correcciones realizadas a los borradores de este trabajo. Finalmente agradecer al sr. Andrés González, subgerente de sistemas de la empresa donde se realizó el estudio.

Referencias

1. **Basili, V.R. (1992).** The Experimental Paradigm in Software Engineering. *International Workshop on Experimental Software Engineering Issues: Critical Assessment and Future Directions*, Dagstuhl Castle, Germany, 3–12.
2. **Baskerville, R.L. (1999).** Investigating Information Systems with Action Research. *Communications of Association for Information Systems*, 2(3es), Article no. 4.
3. **Beecham, S., Hall, T., & Rainer, A. (2003).** Software Process Improvement Problems in Twelve Software Companies: An Empirical Analysis. *Empirical Software Engineering*, 8(1), 7–42.
4. **Berander, P. & Wohlin, C. (2003).** Identification of key factors in software process management - A Case Study. *International Symposium on Empirical Software Engineering (ISESE 2003)*, Rome, Italy, 316–325.
5. **Boehm, B.W. (1984).** Software Engineering Economics. *IEEE Transactions on Software Engineering*, SE-10(1), 4–21.
6. **Misra, S.C., Kumar, V., & Kumar, U. (2010).** Challenges in Transitioning Plan-Driven Software Development Projects to Agile. *Software Quality Professional*, 12(3), 20–32.
7. **Deming, W.E. (1989).** *Calidad, Productividad y Competitividad: la salida de la crisis*. Madrid, España: Díaz de Santos.
8. **Dong-ping, T. & Jun-shan, C. (2010).** Exploring the Impact of Software Quality Management on Informatization. *2010 International Conference on E-Business and E-Government (ICEE)*, Guangzhou, China, 2700–2703.
9. **Dybä, T. (2005).** An Empirical Investigation of the Key Factors for Success in Software Process Improvement. *IEEE Transactions on Software Engineering*, 31(5), 410–424.
10. **Galín, D. & Avrahami, M. (2007).** Benefits of a Higher Quality Level of the Software Process: Two Organizations Compared. *Software Quality Professional*, 9(4), 27–35.
11. **Gutierrez, E., Gutiérrez, A.F., Pérez, A., & Márquez, L.F. (2009).** Acerca de la implementación de los modelos de calidad en la construcción de software en México. *Tecnura*, 13(25), 116–127.
12. **Glass, R.L. (1994).** The Software-Research Crisis. *IEEE Software*, 11(6), 42–47.
13. **Hadden, R. (1998).** Simple Ways to Succeed at Software Process Improvement. *Software Quality Professional*, 1(1), 38–43.
14. **Humphrey, W.S., Kitson, D., & Kasse, T.C. (1989).** The state of software engineering practice. *11th International Conference on Software Engineering*, Pittsburgh, Pennsylvania, 277–288.

15. **Humphrey, W.S. (1988).** Characterizing the Software Process: a maturity framework. *IEEE Software*, 5(2), 73–79.
16. IEEE Standard for a Software Quality Metrics Methodology. Std. 1061-1998.
17. IEEE Standard for Software Quality Assurance Plans. IEEE Std 730-2002.
18. IEEE Standard for Developing Software Life Cycle Processes. IEEE Std 1074-1997.
19. Software Engineering - Product quality - Part: 1 Quality model. ISO/IEC 9126-1, 2001.
20. Information Technology, Process Assessment. ISO/IEC 15504. Parts 1-5
21. Software engineering - Guidelines for the application of ISO 9001:2000 to computer software. ISO/IEC 90003:2004 .
22. Systems and Software Engineering - Software Life Cycle Processes. ISO/IEC 12207: 2008.
23. **Jenkins, M. (2007).** *Comparación de las Iniciativas Latinoamericanas para mejorar la Industria del Software*. Escuela de Ciencias de la Computación e Informática, San Pedro, Costa Rica.
24. **Juran, J.M. (1990).** *Juran Y El Liderazgo Para La Calidad: un manual para ejecutivos*. Madrid, España: Ediciones Díaz de Santos.
25. **Kock, N. (2004).** The three threats of action research: a discussion of methodological antidotes in the context of an information systems study. *Decision Support Systems*, 37(2), 265–286.
26. **Marciniak, J.J. (1994).** Software Engineering, a Historical Perspective. *Encyclopedia of Software Engineering* (1176–1182), New York: John Wiley.
27. **Mendoza, L.E., Pérez, M.A., & Grimán, A.C. (2005).** Prototipo de Modelo Sistemico de Calidad (MOSCA) del software. *Revista Computación y Sistemas*, 8(3), 196–217.
28. **Minguet, J.M. & Hernández, J.F. (2003).** *La Calidad del Software y su Medida*. Madrid, España: Editorial Centro de Estudios Ramón Areces.
29. **Moody, D.L., Shanks, G.G., & Darke, P. (1998).** Improving the Quality of Entity Relationship Models-Experience in Research and Practice. *Conceptual Modeling (ER'98). Lecture Notes in Computer Science*, 1507, 255–276.
30. **Niazi, M. (2009).** Software Process Improvement Implementation: Avoiding Critical Barriers. *CrossTalk, The Journal of Defense Software Engineering*, 22(1), 24–27.
31. **Oktaba, H. & Vázquez, A. (2006).** MoProSoft: A Software Process Model for Small Enterprises. *First International Research Workshop for Process Improvement in Small Settings*, 93–101.
32. **Oktaba, H., García, F., Piattini, M., Ruiz, F., Pino, F.J., & Alquicira, C. (2007).** Software process improvement: The COMPETISOFT project. *Computer*, 40(10), 21–28.
33. **Paulk, M.C. (2009).** A History of the Capability Maturity Model for Software. *Software Quality Professional*, 12(1), 5–19.
34. **Paulk, M.C. (2010).** The Impact of Process Discipline on Personal Software Quality and Productivity. *Software Quality Professional*, 12(2), 15–19.
35. **Pino, F.J., García, F., Ruiz, F., & Piattini, M. (2006).** Adaptación de las normas ISO/IEC 12207: 2002 e ISO/IEC 15504: 2003 para la evaluación de la madurez de procesos software en países en desarrollo. *IEEE Latin America Transactions*, 4(2), 85–92.
36. **Pino, F.J., García, F., & Piattini, M. (2009).** Key processes to start software process improvement in small companies. *24th Annual ACM Symposium on Applied Computing*, Honolulu, Hawaii, 509–516.
37. **Pressman, R.S. (2010).** *Software Engineering: A Practitioner's Approach (7th ed.)*. New York: McGraw-Hill Higher Education.
38. **Rout, T.P., Emam, K.E., Fusani, M., Goldenson, D., & Jung, H.W. (2006).** SPICE in retrospect: Developing a standard for process assessment. *Journal of System s and Software*, 80(9), 1483–1493.
39. **Satpathy, M., Harrison, R., Snook, C., & Butler, M. (2001).** A Generic Model for Assessing Process Quality. *New Approaches in Software Measurement. Lecture Notes in Computer Science*, 2006, 94–110.
40. **CMMI Product Team (2010).** *CMMI for Acquisition, Version 1.3* (CMU/SEI-2010-TR-032), Pittsburgh, PA: Software Engineering Institute.
41. **Shewhart, W.A. (1980).** *Economic Control of Quality of Manufactured Product*. Milwaukee, Wis. : American Society for Quality Control.
42. **Singh, R. (1996).** International Standard ISO/IEC 12207 Software Life Cycle Processes. *Software Process– Improvement and Practice*, 2(1), 35–50.
43. **CMMI Product Team. (2006).** *CMMI for Development, Version 1.2*. (CMU/SEI-2006-TR-008), Pittsburgh, PA: Software Engineering Institute.
44. **Sommerville, I. (2011).** *Software engineering (9th ed)*. Boston: Pearson.

45. Wegener, J., Sthamer, H., Jones, B.F., & Eyres, D.E. (1997). Testing real-time systems using genetic algorithms. *Software Quality Journal*, 6(2), 127–135.
46. Yu, E.S.K. & Mylopoulos, J. (1994). Understanding "Why" in Software Process Modelling, Analysis, and Design. *16th International Conference on Software Engineering*, Sorrento, Italy, 159–168.



Samuel Sepúlveda Cuevas es licenciado en Ciencias de la Ingeniería (1998) e Ingeniero Civil Industrial m. Informática (1999), por la Universidad de La Frontera, Temuco, Chile. Obtuvo su máster en Dirección y Gestión de Sistemas de Información y TIC, por la Universitat Oberta de Catalunya, España en 2006. Actualmente postula al grado de Doctor en Aplicaciones de la Informática por la Universidad de Alicante, España. Se desempeña como profesor en el Dpto. de Ingeniería de Sistemas y como investigador en el Centro de Estudios en Ingeniería de Software, Universidad de La Frontera. Sus intereses de investigación están en el área de Ingeniería de Software, Ingeniería de

Requerimientos y Modelado de Líneas de Productos de Software.



Ania Cravero Leal es licenciada en Ciencias de la Ingeniería (1996) e Ingeniera Civil Industrial m. Informática (1997), por la Universidad de La Frontera, Temuco, Chile. Obtuvo su máster en Tecnologías de la Información, por la Universidad Politécnica de Madrid, España (2006). Ha obtenido en 2010 su Doctorado en Cs. de la Computación y Sistemas Informática por la Atlantic International University, EE.UU. Se desempeña como profesora en el Dpto. de Ingeniería de Sistemas y como investigadora en el Centro de Estudios en Ingeniería de Software, Universidad de La Frontera. Sus intereses de investigación están en el área de Modelado Bases de Datos, Ingeniería de Requerimientos para Almacenes de Datos y Alineamiento de Objetivos.

Artículo recibido el 30/06/2011, aceptado el 06/03/2014.