

PROPUESTA METODOLÓGICA PARA EL ASEGURAMIENTO DE CALIDAD DE PROYECTOS INFORMÁTICOS: CASO DE ESTUDIO EN EMPRESA DE SERVICIOS SANITARIOS.

Samuel Sepúlveda¹, Andrés González², Coral Calero³ y Cristina Cachero⁴.

¹Universidad de La Frontera, Departamento de Ingeniería de Sistemas.

² Subgerencia Corporativa de Sistemas, Grupo Aguas Nuevas.

³ Universidad Castilla-La Mancha, Escuela Superior de Informática.

⁴ Universidad de Alicante, Departamento de Lenguajes y Sistemas Informáticos.

E-mail: ssepulve@ufro.cl, andres.gonzalez@aguasnuevas.cl,
coral.calero@uclm.es, ccachero@dlsi.ua.es

Resumen. 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. Esto genera necesidades crecientes de las empresas por incorporar estándares de calidad en sus procesos. En la actualidad, todas las organizaciones que aspiran a la excelencia están comprometidas a elevar sus niveles de calidad, para lo cual es vital el compromiso permanente de todos y cada uno de los miembros de la organización. En el marco del esfuerzo conjunto por mejorar los niveles de calidad, el presente trabajo entrega un aporte para mejorar los procesos asociados al desarrollo de los proyectos informáticos pertenecientes a la empresa Aguas Nuevas S.A., a partir de una adaptación de IEEE 730-2002, analizando y discriminando los tipos de requerimientos establecidos para cada uno de los proyectos estudiados. El objetivo apunta fundamentalmente a desarrollar y estandarizar pautas para la gestión de dichos proyectos y así también contar con criterios únicos a la hora de evaluar los resultados obtenidos. Finalmente se puede concluir que el modelo propuesto, tuvo efectos positivos, mejorando el producto final y satisfaciendo en mejor forma a los clientes internos del consorcio Aguas Nuevas S.A.

1. Introducción.

Desde la denominada Crisis del Software [1], el tema de calidad se ha instalado como un objetivo relevante en la comunidad de la Ingeniería del Software (IS). Las tendencias de calidad en esta área fueron importadas de los Sistemas de Calidad productivos y tuvieron su paralelo en la Industria del Software [2], así es posible ver una analogía entre la etapa de pruebas de la IS con lo que puede denominarse ensayos no destructivos. Posteriormente la Calidad del Software se traslada, como tendencia, de un foco en el producto a un foco en el proceso. De esta manera surgen propuestas como ISO9000-3 [3], ISO/IEC12207 [4], CMMI [5], IEEE1058 [6], IEEE1074 [7] y IEEE730 [8], entre otros.

Por otro lado y vista desde una perspectiva histórica, la calidad ha estado directamente ligada con el desarrollo de la humanidad. Pero no es hasta principios del siglo XX [9], que surge este concepto dentro de un contexto

industrial y de ingeniería, cuya evolución hasta el día de hoy, ha permitido observar, estudiar y proponer cómo establecer mejoras en los productos desarrollados y procesos asociados a su desarrollo.

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 [10]. Para cumplir con lo anterior es que la IS se ha formulado cómo la disciplina encargada de todos los aspectos relacionados en la producción del software. Esto incluye la gestión y aseguramiento de la calidad, por medio de propuestas que apuntan hacia la forma de gestionar los proyectos informáticos y de software, como prueba de ello se tienen los estándares mencionados anteriormente.

Considerando la importancia de los conceptos anteriormente señalados es que se propuso el diseño de una metodología para el aseguramiento de la calidad de sus proyectos informáticos para la empresa Aguas Araucanía S. A., perteneciente al consorcio Aguas

Nuevas S.A., que cuenta con presencia en las regiones de Arica y Parinacota, Araucanía y Magallanes y Antártica Chilena, proveyendo de servicios sanitarios como agua potable, alcantarillado, tratamiento de aguas, etc., a sus más de 336.000 clientes.

Específicamente es en la empresa Aguas Araucanía S. A., donde se encuentra inserta físicamente la subgerencia corporativa de sistemas, responsable de proveer los servicios informáticos y de comunicaciones a todas las empresas del consorcio. Es aquí donde se impacta positivamente con la metodología diseñada, pues como objetivo central del trabajo se propuso el desarrollo de pautas estandarizadas para la gestión de proyectos informáticos y de paso también contar con criterios únicos a la hora de definir y posteriormente evaluar los resultados obtenidos en dichos proyectos.

La estructura del artículo presenta en la segunda sección una revisión resumida de conceptos relativos al Software como producto y su Calidad. En la sección 3 se muestra el caso de estudio y se discute sobre la experiencia revisada. Finalmente en la sección 4 se muestran las principales conclusiones, impactos y consideraciones de la metodología diseñada y su aplicación en el caso de estudio.

2. El Software y su Calidad.

2.1 Conceptos previos.

Hoy en día es común oír hablar respecto de la calidad en diversos ámbitos: empresa privada, organismos públicos y del estado, centros educacionales, etc. Lo anterior es debido fundamentalmente a que ésta juega un papel fundamental en la competitividad de las organizaciones, así como también considera que los consumidores de productos y servicios ya no se conforman con lo que les ofrecen, pues modifican y aumentan sus gustos y exigencias. Las ventajas de su implementación dentro de las organizaciones han sido ampliamente documentadas y entre éstas se tienen: ventajas competitivas, reducción de costos y pérdidas, fidelización de clientes, aumento de las utilidades, mejora de imagen ante sus accionistas, etc. [11-13].

Puede observarse a partir del trabajo realizado por [9], cómo se desarrolla y evoluciona el concepto calidad. Además es posible, a través de las comparaciones hechas por [2], observar cómo la industria del software

ha debido incorporar conceptos propios del control y gestión de calidad, en un período de no más de 40 años, a diferencia de otras industrias a las que ha tomado alrededor de 200 años alcanzar su madurez.

Teniendo claro que los conceptos mencionados anteriormente están muy ligados a procesos y productos provenientes de la manufactura, resulta muy interesante entonces plantearse cómo las exigencias de calidad se trasladan a industrias centradas en los servicios, donde un caso particular es la industria del software. En este contexto es que [14] establece que la calidad del software no es directamente comparable con la calidad de manufactura de productos, además identifica al menos tres problemas básicos que son propios del desarrollo de software: atender requerimientos que no se establecen directamente en la especificación del sistema, existencia de ciertas características de la calidad que son complejas de especificar en forma no ambigua y la dificultad en desarrollar un sistema que satisfaga las expectativas del usuario.

2.2 Consideraciones al implantar la Calidad.

Uno de los aspectos anteriormente mencionados guarda relación con el costo asociado a la calidad. De acuerdo con [9] estos costos, en un mediano y largo plazo, se revierten en ahorros y beneficios tangibles e intangibles. Obviamente implantar sistemas de calidad en una organización no implica hacerlo a costo cero, por lo cual deben considerarse los *costos de la no calidad* asociados, que según [10] contemplan: costos de prevención, de evaluación y de fallos. Estos mismos autores exponen experiencias que muestran una relación directa entre el aumento del costo de la no calidad y la tardanza en que una falla es detectada dentro de un proyecto de desarrollo de software. Así se entiende que si se aumenta la calidad del producto se disminuye el costo asociado a la no calidad del mismo, pero se da también que el costo del sistema de calidad aumenta, luego el costo total mínimo de la calidad se puede alcanzar en un punto que es bastante “*lejano a la perfección*” [9].

Otro concepto inherente a la calidad tiene relación con la realización de actividades que permitan asegurarla durante el desarrollo del software, el cumplimiento de procedimientos y estándares definidos, garantizando así que el producto cumple con las especificaciones

definidas [10, 14], esto es lo que se conoce como control de calidad. Estas y otras consideraciones son las que se presentan en el siguiente apartado, donde se discuten y exponen aspectos relevantes respecto del Aseguramiento de Calidad del Software (ACS) y la Mejora del Proceso de Software (MPS).

2.3 Asegurando la Calidad.

Según lo planteado por [9] hay evidencias que en 1974 se usa por primera vez el concepto de ACS en una especificación de software militar estadounidense, la MIL-S-52779, donde planteaba a los contratistas exigencias relacionadas no solo con el desarrollo del software sino también con su administración. Luego surgen variadas normas a partir de ésta, hasta que en 1979 el IEEE emite la norma P-730, haciendo especial énfasis en la documentación y uso de revisiones formales en el desarrollo del software.

Aún con todo lo mencionado, surge la pregunta ¿qué es el ACS? Para responder esta pregunta, se recurre nuevamente a autores clásicos de la IS como [10, 14]; ambos concuerdan en que debe incorporar y definir marcos de trabajo y estándares organizacionales, que permitan desarrollar software de calidad. Además se considera que producto de la capacidad que se tenga para garantizar esta calidad, se deriva la madurez de la ingeniería desarrollada.

Es posible observar cómo diversos autores, tanto expertos en calidad como del área de la IS [11, 14], establecen la relación existente entre la calidad asociada al proceso y cómo ésta impacta en la calidad del producto. Si bien esta mirada deriva directamente de procesos manufacturados, se sostiene que es posible adaptarla para los procesos de software, no sin discrepancias entre los expertos [14-16].

2.4 Acerca del Estándar IEEE Std 730-2002

El estándar IEEE Std 730-2002 (*una revisión del IEEE Std 730-1998*) [8], es una norma que permite estandarizar, proporcionando los requisitos mínimos aceptables, la preparación y contenido de los planes de ACS (*SQAP de sus siglas en inglés*). 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. También plantea la posibilidad de agregar secciones adicionales, según se requiera. Luego se especifican los detalles, para cada una de las secciones definidas. Un detalle interesante se encuentra en la bibliografía del IEEE Std 730-2002, donde pueden observarse los demás estándares IEEE en los cuales éste se basa.

2.5 Mejorando el Proceso.

Tal como lo plantea [17], históricamente el desarrollo de software ha estado centrado en el producto, pero de un tiempo a esta parte los investigadores se han enfocado en el proceso asociado a la IS. [17] propone además un modelo de proceso de software que se centra en las relaciones intencionales entre los actores involucrados. Es durante la última década que se ha visto una proliferación de modelos e iniciativas que apuntan a obtener iniciativas exitosas de MPS. Sin embargo y de acuerdo con [18], a pesar de la gran cantidad de anécdotas, experiencias y casos de estudio, no hay datos concluyentes respecto de los factores críticos de éxito para implementar una MPS. A partir de estos mismos datos, [19] identifica siete barreras críticas para el éxito de MPS, entre ellas: *grupo poco experimentado, no tener una metodología clara para implementar el PMS*. Además, proporciona directrices para evitar cada una de ellas. Es importante considerar que en [18], se plantea que uno de los factores que deben ser considerados son los organizacionales, tanto o más que los tecnológicos.

Otra perspectiva que apunta a obtener experiencias exitosas al implantar PMS es la que propone [20], donde plantea factores que permitirían acelerar el cambio de cultura y mantener los esfuerzos realizados. Entre los factores mencionados se encuentran: *No basta con el compromiso del liderazgo de la empresa, Entrega de incentivos y recompensas durante la implantación del PMS, Utilización de parámetros de referencia y fijación de objetivos para conseguir la aceptación*, entre otros. Por otro lado el trabajo realizado por [21], muestra cuantitativamente cómo se mejora el desempeño de forma significativa en la organización al alcanzar y mantener altos niveles de calidad en su proceso de software.

3. Diseño de la Propuesta Metodológica.

3.1 Consideraciones iniciales.

El trabajo realizado presenta características teórico-prácticas, cuyas principales actividades se detallan a continuación.

La metodología que se presenta en este trabajo, es una adaptación del ciclo de vida secuencial de la IS [10, 14] y del ciclo de Deming [11], fundamentada en que ambas han sido ampliamente probadas y aceptadas. Asimismo, esta última permite incorporar el concepto de mejora continua. La metodología se compone de cuatro etapas: (1) *analizar*, (2) *construir*, (3) *implantar* y (4) *probar*; el detalle de las etapas y actividades puede ser visto en la Tabla 1.

En términos generales, esta metodología consiste en realizar una revisión bibliográfica para conocer el estado del arte respecto de la calidad, del ACS y MPS entre otros temas relevantes. A partir de esta revisión se identifican los principios de la calidad.

Luego se define el objeto de estudio, 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.

A partir de la información anterior, se realizaron reuniones con representantes de la organización y el equipo de trabajo. El objetivo de las reuniones fue consensuar los conceptos básicos para elaborar una metodología de ACS, tales como tipificación de los proyectos informáticos, etapas de los proyectos, productos de cada etapa, estándares de cada producto y la definición de los hitos de verificación. Con la información obtenida se realizó un análisis comparativo con el estándar IEEE Std 730-2002 y se efectuaron los ajustes necesarios. Esta elección resultó de evaluar los distintos tipos de proyectos informáticos, existentes en el contexto local, además es el resultado de discriminar aquellos que son relevantes para la empresa analizada.

Posteriormente, la metodología de ACS propuesta se aplicó en proyectos de software que fueron tomados como planes piloto, tras lo cual se realizaron pruebas iniciales a la propuesta, junto a una primera revisión y evaluación del impacto generado por su uso, para, finalmente proponer algunas mejoras y trabajos futuros para potenciarla.

3.2 La situación actual.

La subgerencia de sistemas entrega al grupo Aguas Nuevas diferentes servicios, entre ellos la adquisición y desarrollo de sistemas informáticos. Estas necesidades informáticas son recopiladas a comienzo de cada año, y pasan 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 jefes de proyectos en relación a su carga y presupuesto anual. Esta 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 jefe de proyecto usa discrecionalmente, producto 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. Esta inexistencia de estándares produce algunas ineficiencias en el proceso, ya que las especificaciones entregadas a los proveedores dependen del jefe de proyectos que las realice. Otro factor importante que se observa es que la evaluación anual de los jefes de proyectos respecto del desempeño alcanzado se realiza en base a una medida más bien de

Tabla 1. Etapas y Actividades definidas para la metodología.

Etapas	Actividades
(1)	Revisión bibliográfica. Identificar los principios de calidad.
	Definir Objeto de estudio. Identificar la organización. Analizar situación actual de estudio. Justificación y Pertinencia del trabajo.
	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 SQA.
	Evaluar impacto de las acciones a implementar. Corregir posibles desviaciones.
(4)	Implantar metodología en la organización. Realizar ajustes si fueran necesarios.

características cualitativas y no cuantitativas, ya que no existen métricas para evaluar el cumplimiento de los objetivos trazados como prestadores de servicios dentro de la organización.

3.3 Tipificación de Proyectos.

La primera etapa del diseño consiste en realizar una tipificación de los proyectos informáticos que realiza la Subgerencia de Sistemas; una vez realizada esta tipificación, se procede a determinar las etapas que deberá 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 la matriz tipo – etapa – producto.

Esta tipificación de proyectos se logró analizando los planes anuales de sistemas de los años 2006, 2007 y 2008; la información obtenida fue consensuada con los jefes de proyectos, quienes son los responsables de ejecutar dicho planes. La tipificación arrojó como resultado las siguientes 4 categorías de proyectos: *Soluciones de mercado empaquetadas* (T1), *Soluciones de mercado con desarrollo* (T2), *Desarrollo Interno – Externo* (T3) y *Desarrollo Interno* (T4).

Además de las categorías obtenidas, se identificaron dos atributos de cada tipo de proyecto, que ayudan a comprender el alcance de cada uno de ellos: *Envergadura* (ENV) y *Recursos* (RR), los cuales indican el tamaño o esfuerzo requerido para su desarrollo o adquisición y la fuente de recursos usada para su desarrollo. Todo lo anterior se resume en la Tabla 2.

3.4 Etapas identificadas para los Proyectos.

La definición de las etapas de los proyectos informáticos de la Subgerencia de Sistemas se realizó adaptando el modelo del ciclo de vida [10], contrastando los planes anuales de sistemas de los años 2006, 2007 y 2008; la información obtenida fue consensuada con los jefes de proyectos. A partir de esto se obtuvieron las etapas de: *Formalización* (F), *Análisis* (A), *Diseño* (D), *Compra* (Cp), *Construcción* (Co), *Pruebas* (P), *Implantación* (I), *Aceptación* (Ac) y *Explotación* (Ex). El detalle de las etapas identificadas para los proyectos y las actividades/objetivos que se deben realizar se detallan en la Tabla 3.

Tabla 2. Tipos de proyectos y sus características.

Tipos	Descripción	(ENV)	(RR)
(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

3.5 Definición de Productos por Etapa.

Esta actividad consistió 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 definieron en base a lo señalado en estándar IEEE Std 730-2002 [8], adaptándola a la realidad y necesidades locales; asimismo se consideraron las sugerencias y aportes de los jefes de proyectos del área. El resultado se muestra en la Tabla 4, además en las Fig. 2 y 3, se observa el diseño de dos de estos instrumentos.

3.6 Definición de Matriz Tipo-Etapa-Producto.

La última actividad del diseño consiste en unir los resultados de las etapas anteriores conformando la Matriz Tipo-Etapa-Producto, de forma tal que se permite operativizar las definiciones anteriores, generando así una herramienta que resulta de gran utilidad para la definición del Plan de ACS para 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, de forma tal que pueda seguirse la pauta definida y hacer los chequeos correspondientes a medida que cada proyecto avanza en su desarrollo.

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

Etapa	Descripción	Objetivos / Actividades
(F)	Se establece la planificación formal del proyecto, alcance, equipo, plazos, etc.	- Formalización de necesidad. - Planificación general, plazos. - Definición equipo de trabajo. - Confección de la ficha FAI.
(A)	Establece 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)	Etapa en que se toman decisiones con respecto a la organización lógica del software	- Formalizar soluciones tecnológicas para c/u de las funciones del sistema. - Asignar RR materiales c/u de las funciones. - Diseño general. - Ajustar especificaciones del producto. - Modelo de datos. - Diseñar casos de prueba.
(Cp)	Se genera la Licitación/Cotización, adjudicándola, y cerrando el proceso con la contratación o compra del producto.	- Ficha FAC. - Cotización / licitación. - Adjudicación.
(Co)	Proceso de 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.

3.7 Hitos de Verificación Producto-Etapa.

La verificación de cada producto respecto al estándar de calidad definido, se realizará al finalizar cada etapa, de acuerdo a la carta Gantt establecida por el jefe de proyecto. El producto de esta revisión será un acta de reunión en la cual se especificará: *Fecha de revisión, Proyecto, Etapa a revisar, Resultados de la verificación (por producto), Observaciones y Responsable de la verificación.* El detalle asociado a la verificación Producto-Etapa puede verse en la Fig. 1.

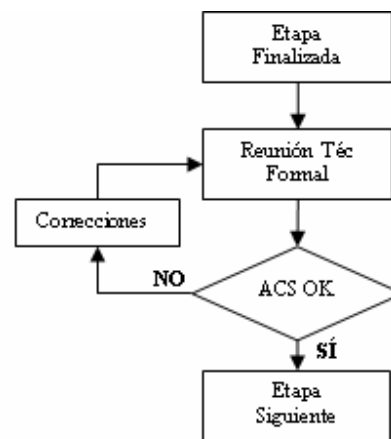


Fig. 1 Verificación Etapa-Producto.

3.8 Definición del Área de ACS.

Una vez definida la metodología es necesario y válido preguntarse ¿Quién realizará la función de ACS? la respuesta natural es “El área de ACS” y dada la importancia estratégica que tiene, debiera ser un equipo independiente de quienes desarrollen los sistemas y con dependencia directa del subgerente de sistemas. Ésta tendrá como misión 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 serán:

- Definir, revisar y actualizar los estándares de calidad.
- Asegurar niveles de calidad adecuados de los sistemas informáticos, es decir, asegurar que las necesidades de los clientes están siendo satisfechas en forma adecuada.

- Identificar posibles desviaciones en los estándares aplicados, así como comprobar que se han llevado a cabo las medidas preventivas o correctivas necesarias para eliminar estas desviaciones.
- Disminuir el impacto del paso a producción de sistemas.
- Estandarizar la metodología de trabajo para el desarrollo de proyectos informáticos.
- Elaborar “*Planes de Aseguramiento de Calidad del Software*” efectivos, para cada proyecto informático, en los que se establezcan claramente los hitos de revisión y auditoría.

4 Caso de Estudio y Resultados Iniciales.

Una vez definidas todas las etapas de la metodología de ACS, corresponde aplicarla y realizar pruebas piloto del modelo, en base a un caso real, con la finalidad de medir el impacto de la metodología.

Esta actividad consistió en seleccionar un proyecto informático en ejecución y aplicarle la metodología, para observar las desviaciones existentes.

El sistema seleccionado fue la Plataforma de Consultas de Proveedores, aplicación Web, orientada a consultas del estado de pago de los documentos emitidos por los proveedores del grupo Aguas Nuevas S. A., pretende convertirse además en el canal oficial de la empresa para atender consultas, requerimientos, reclamos y solicitudes.

Dados los antecedentes de la metodología diseñada y del pilotaje realizado, se observó que los Jefes de Proyecto 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, se observó que la metodología 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 aseguramiento de calidad. Además se pudo observar como la metodología de ACS 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 los proyectos.

Se corroboró que algunos de los problemas surgidos durante la implantación de los proyectos realizados sin una metodología formal, no hubiesen ocurrido o su impacto habría sido menor. Finalmente, el control del avance real de los proyectos resultó mucho más certero a partir de esta herramienta, ya que los jefes de proyectos presentaron una visión muy optimista al informar sus avances.

Tabla 4. Etapas, productos y estándares definidos.

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

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

REGISTRO	
FICHA DE PROYECTO INFORMÁTICO (FPI)	
Datos Generales del Proyecto	
Nombre	Canal de consulta de proveedores.
Ámbito	Subgerencia Corporativa de Finanzas.
Inicio - Término	01-05-2008 29-08-2008.
Recursos Asignados	\$ 1.000.000.-
Descripción	El proyecto "ventanilla única de proveedores" tiene por finalidad proveer información acerca del estado de pago de los documentos generados por los proveedores de Aguas Nuevas y sus filiales; proporcionar un canal de consulta entre el consorcio y sus proveedores y habilitar una herramienta que permita administrar, controlar y gestionar las consultas realizadas por los proveedores.
Objetivos	Implementar un canal formal de comunicación entre las compañías del grupo Aguas Nuevas y sus proveedores.
Datos del Mandante	
Nombre	Roberto Castro.
Gerencia	Subgerencia de Finanzas.
Email - Fono	(2) 5844611
Cargos y sus Responsabilidades Internos	
Jefe de Proyecto	Andrés González N.
Email - Fono	andres.gonzalez@aguasnuevas.cl (45)-207366
Usuarios Líderes	Patricio Carmona
Email - Fono	(2) 5844611

Fig. 2 Diseño de la Ficha de Proyecto, REG-SIS-002-FOR.

REGISTRO			
ACTA REUNIÓN			
MINUTA DE REUNIÓN			
Área	SubGerencia de Finanzas.		
Fecha / Hora	22 de Mayo del 2008 Hora inicio: 10:30 Hora termino: 14:15		
TEMAS TRATADOS			
1	Proyecto "Canal de Consulta de Proveedores".		
PARTICIPANTES			
1	Paula Sandoval Finanzas		
2	Roberto Castro Finanzas		
3	Patricio Carmona Finanzas		
4	Andrés González Sistemas		
COMPROMISOS		FECHA	RESPONSABLE
1	Definir mecanismo de autenticación de proveedores. Se recomienda autenticar mediante consulta de documento válido.	25-07-2008	Patricio Carmona
2	Definir estados de los documentos de pago (a mostrar a los proveedores)	25-07-2008	Patricio Carmona
3	Definir funcionalidad "Pronto Pago".	25-07-2008	Patricio Carmona
4	Definir tipos de consultas de proveedores (consultas, sugerencias, redamos, etc.)	25-07-2008	Patricio Carmona
5	Implementar funcionalidad "monitor de antigüedad de consultas", la que enviará correos cuando la consulta tenga una cierta cantidad de horas sin respuestas	N/A	Andrés González
6	Implementar funcionalidad "actualización de datos", consiste en permitir al proveedor cambiar datos, los que se actualizarán en fin700 en forma manual.	N/A	Andrés González

Fig. 3 Diseño del Acta de Reunión, REG-MQS-000-007.

5. Conclusiones

El presente trabajo identificó los principios que deben ser considerados para el ACS de la organización en estudio. En base a ellos se adaptó la metodología del ciclo de vida al desarrollo de proyectos informáticos del consorcio Aguas Nuevas S.A., proporcionando de esta manera una metodología estándar para esta importante actividad.

El estándar de ACS IEEE Std 730-2002 usado como referencia permitió definir las bases de la metodología adaptada y así definir productos para cada etapa del desarrollo de software, contribuyendo así a homologar las funciones del personal abocado a estas tareas.

La metodología desarrollada permite identificar y resolver las desviaciones que pudieran producirse respecto de lo esperado, al término de cada una de las etapas del desarrollo del proyecto informático y no al final del ciclo como ocurría anteriormente.

Cabe destacar que en estos momentos la metodología diseñada se está agregando al sistema de gestión de calidad ISO 9001 existente en la empresa, con el nombre de “Metodología de Proyectos TIC”, por lo que queda así oficializado para el Grupo Aguas Nuevas S.A. y por lo tanto el proceso es auditable tanto en forma interna como externa. Queda pendiente la implementación de la auditoría que se contempla por cada etapa, pero se espera que el próximo año se tengan los recursos disponibles para esta actividad.

Referencias

- [1] Glass, R.L., *The Software-Research Crisis*. IEEE Software, 1994. **11**(6): p. 42-47.
- [2] Marciniak, J.J., *Software Engineering a Historical Perspective*. Encyclopedia of Software Engineering, 1994: p. 1176-1182.
- [3] Kehoe, R. and A. Jarvis, *ISO 9000-3: a tool for software product and process improvement*. 1996.
- [4] ISO/IEC, *ISO/IEC 12207 Information technology -- Software life cycle processes*. 1995.
- [5] Software Engineering Institute (SEI), C.M.U., *CMMI for Development, Version 1.2*. 2006.
- [6] IEEE, *IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans*. 1998.
- [7] IEEE, *IEEE Std 1074, IEEE Standard for Developing Software Life Cycle Processes*. 2002.
- [8] IEEE, *730-2002, Standard for Software Quality Assurance Plans*. 2002: p. 0_1- 10.
- [9] Minguet, J.M. and J.F. Hernández, *La Calidad del Software y su Medida*. 2003, Madrid: Editorial Centro de Estudios Ramón Areces S.A. 261.
- [10] Pressman, R.S., *Ingeniería del Software, un enfoque práctico*. 5ta. ed, ed. M.H. Interamericana. 2001. 610.
- [11] Deming, W.E. and J.N. Medina, *Calidad, Productividad y Competitividad: la salida de la crisis*. 1989: Editorial Díaz de Santos, S.A. 391.
- [12] Juran, J.M., *Juran Y El Liderazgo Para La Calidad*. 1990: Editorial Díaz de Santos, S.A.
- [13] Shewhart, W.A., *Economic Control of Quality of Manufactured Product*. 1980. 501.
- [14] Sommerville, I., *Ingeniería del Software, 7ª edición*. 7ª ed. 2005: Pearson Addison Wesley. 697.
- [15] Humphrey, W.S., *Characterizing the Software Process: a maturity framework*. IEEE Software Management, 1988. **5**(2): p. 73-79.
- [16] Basili, V.R., *The Experimental Paradigm in Software Engineering*. Experimental Software Engineering Issues: Critical Assessment and Future Directives, 1992.
- [17] Yu, E. and J. Mylopoulos. *Understanding "Why" in Software Process Modelling, Analysis, and Design*. in 1994. *Proceedings. ICSE-16., 16th International Conference on Software Engineering*. 1994. Sorrento, Italy.
- [18] Dybå, T., *An Empirical Investigation of the Key Factors for Success in Software Process Improvement*. IEEE Transactions on Software Engineering, 2005. **31**(5): p. 410-421.
- [19] Niazi, M., *Software Process Improvement Implementation: Avoiding Critical Barriers*. CrossTalk, The Journal of Defense Software Engineering, 2009. **22**(1): p. 24-27.
- [20] Hadden, R., *Simple Ways to Succeed at Software Process Improvement*. Software Quality Professional, 1998. **1**(1): p. 38-43.
- [21] Galin, D. and M. Avrahami, *Benefits of a Higher Quality Level of the Software Process: Two Organizations Compared*. Software Quality Professional, 2007. **9**(4): p. 27-35.