

Artículo original
CALIDAD

REVISIÓN, VERIFICACIÓN Y VALIDACIÓN EN UN PROCESO DE DESARROLLO DE SOFTWARE/
REVIEW, VERIFICATION AND VALIDATION IN A SOFTWARE DEVELOPMENT PROCESS

Javier Blanco-Llano^I, Aida Rodríguez-Hernández^{II}

^I Desoft S.A., División Territorial de Ciudad de la Habana. La Habana, Cuba.

E-mail: javier.blanco@hab.desoft.cu

^{II} Instituto Superior Politécnico José Antonio Echeverría, Cujae. Facultad de Ingeniería Industrial. La Habana, Cuba.

E-mail: aida@ind.cujae.edu.cu

Recibido: 21/09/2010

Aprobado: 03/01/2011

Resumen / Abstract

En el presente artículo se expone una interpretación, a través de una metodología en los proyectos de desarrollo de software, del requisito ISO para el diseño. Esto responde a la necesidad de brindar aplicaciones informáticas que respondan con calidad a los requisitos de los clientes, para poder cumplir con la misión asignada de brindar soluciones integrales eficaces en Tecnologías de la Información y las Comunicaciones (TIC) a la sociedad cubana. Dicha metodología, que responde a las etapas establecidas en el Ciclo Shewhart/Deming, clarifica qué hacer en el control de la calidad para aplicar el requisito ISO y está en fase de implementación en la empresa Desoft S.A.

This article offers an interpretation of the ISO requirements to design using a methodology in software development projects. This responds to the need of giving informatic solutions which may satisfy the client's requirements with quality. This let us accomplish the assigned mission of giving ICT efficient and integral solutions to Cuban society. This methodology, which follows the established steps on Shewhart/Deming Cycle, explains what to do regarding quality control to apply the ISO requirements and it is in an implementation phase in Desoft S.A. enterprise.

Palabras clave / Key words

Desarrollo de software, revisión, verificación, validación.

Software development, review, verification, validation.

I. INTRODUCCIÓN

En una empresa productora de software es de suma importancia la implementación de un Sistema de Gestión de la Calidad (SGC), que permita satisfacer de una forma más eficiente y eficaz las necesidades y los requisitos de los clientes. Existen diversos modelos propuestos para la evaluación de la conformidad del producto software como CMM (Modelo de Madurez de las Capacidades, por sus siglas en inglés), CMMI (integración de los modelos complementarios basados en CMM) y el modelo propuesto por ISO 9000, que consiste en un conjunto de requisitos a cumplir por la organización. Los dos modelos referenciados inicialmente se basan en evaluar los procesos a partir del cumplimiento de prácticas base.

El modelo ISO 9000 plantea requisitos para el diseño de un producto genérico, lo cual equivale a un proceso productivo de software, dado el carácter no repetitivo del producto obtenido.

En la división territorial de Ciudad de la Habana de la empresa Desoft, se consideró integrar la actividad existente del Control de Calidad con la aplicación de los requisitos 7.3.4, 7.3.5 y 7.3.6 de la NC-ISO 90003: 2006 [1] en el proceso de desarrollo del software para la revisión, la verificación y la validación. El objeto de la actividad integrada se interpreta como asegurar en las etapas definidas para el proceso de desarrollo de software las revisiones técnicas de los artefactos obtenidos en cada una de las etapas, las verificaciones de que los artefactos revisados satisfacen lo definido en las entradas para cada etapa y la validación, en demostrar que el producto es capaz de satisfacer los requisitos del cliente. Para lograr los objetivos propuestos los autores elaboraron un conjunto de procedimientos conformadores de la metodología que se expone en el presente artículo.

II. MATERIALES Y MÉTODOS

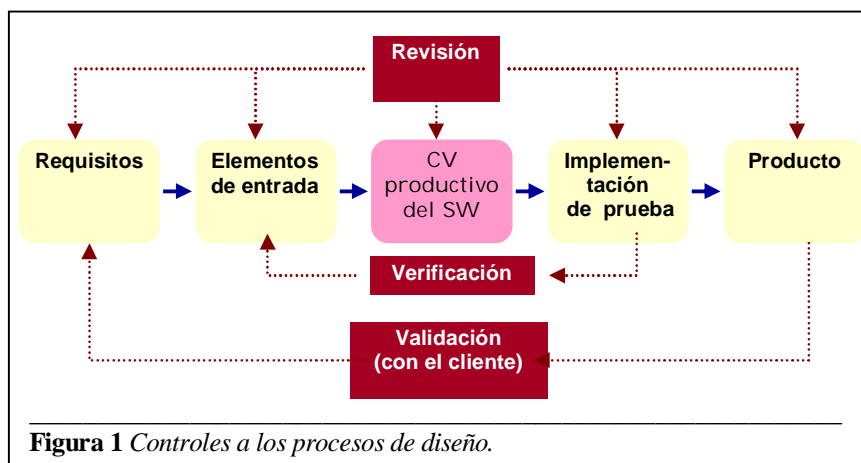
Como elementos conceptuales para definir las tres actividades y esclarecer sus diferencias se recurrió como normativa a las definiciones establecidas en las normas ISO. En el caso de las actividades relacionadas con la revisión, la norma NC-ISO 9000: 2005 [2] plantea que son aquellas emprendidas para asegurar la conveniencia, adecuación y eficacia del tema objeto de la revisión, para alcanzar unos objetivos establecidos. Se observa que en el diseño y desarrollo del software se distinguen para la revisión dos rangos de actividades: aquellas relacionadas con las revisiones de la gestión de proyecto y las llamadas revisiones técnicas, que se centran en el producto o servicio; lo cual se establece en la norma ISO/IEC 12207: 2003 [3].

Según la norma NC-ISO 9000: 2005 las actividades relacionadas con la verificación se refieren a la confirmación, mediante la aportación de evidencias objetivas, de que se han cumplido los requisitos especificados. En el diseño y desarrollo del software, la verificación concierne al proceso de examinar el resultado de una actividad conocida para determinar la conformidad con el requisito declarado para esa actividad, lo cual queda declarado en la norma ISO/IEC 12207: 2003.

Por su parte, las actividades de validación según la norma NC-ISO 9000: 2005 están relacionadas con la confirmación, mediante aportación de evidencia objetiva, de que se han cumplido los requisitos para una aplicación específica prevista. En el diseño y desarrollo del software se establece la validación como la confirmación mediante inspección y suministro de una evidencia objetiva de que los requisitos particulares para un uso específico planificado son satisfechos [3].

Para organizar la secuencia lógica de las actividades y facilitar la inserción de la metodología propuesta en un Sistema de Gestión de la Calidad (SGC) ISO 9000, se buscó observar analogías con el conocido Ciclo Shewhart/Deming.

La metodología que se propone busca cumplir con todos los aspectos antes citados, centrándose en el ciclo de vida productivo del software, para seguir los principios establecidos en los controles a los procesos de diseño [4], según se muestra en la Figura 1.



A continuación se explica cómo llevar a cabo cada uno de estos aspectos según los procedimientos elaborados.

1. Revisión en el desarrollo del software

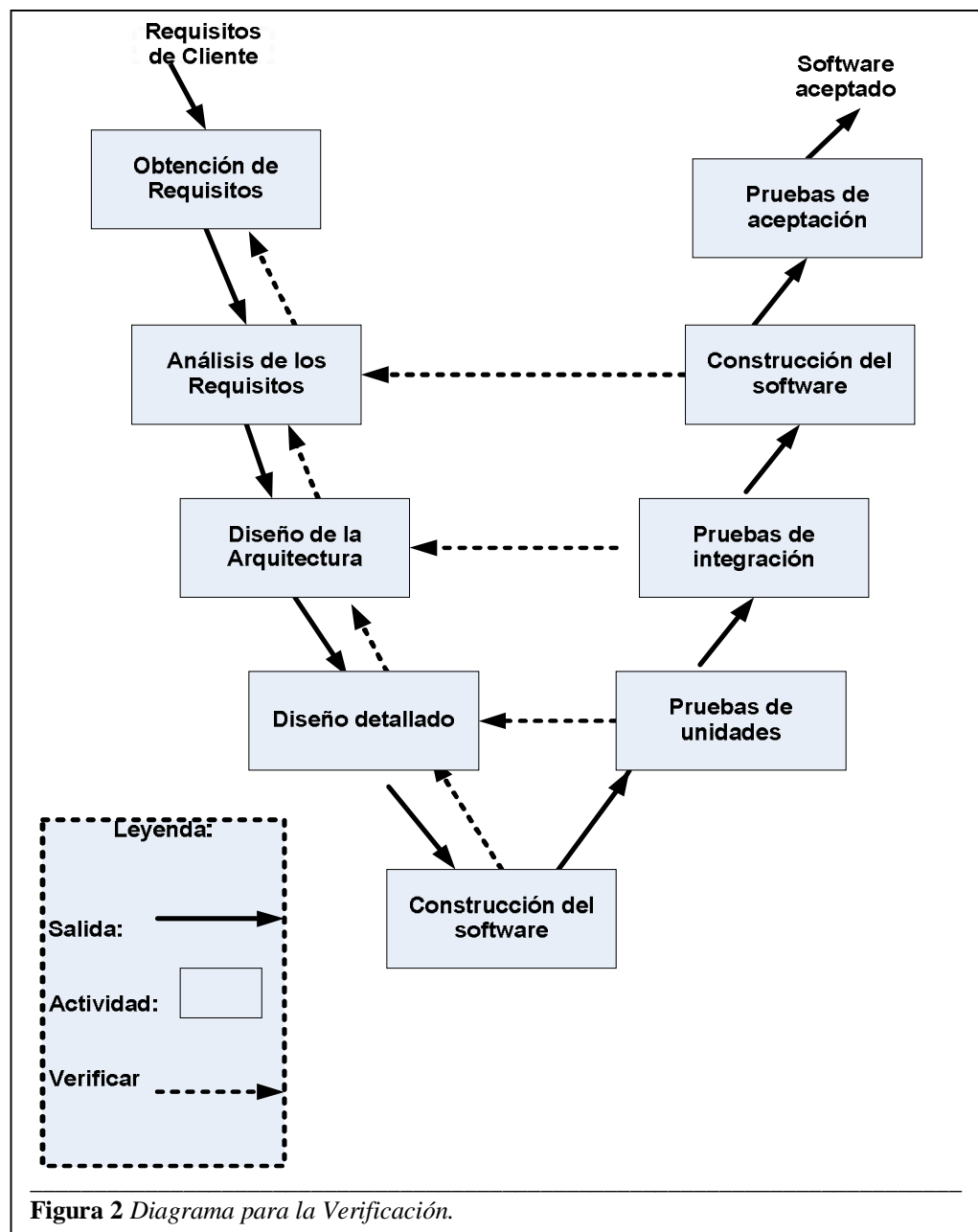
En la metodología propuesta, la actividad de la revisión se ejecuta sobre los artefactos de las etapas de un proyecto de desarrollo de software. Para ello se desarrollan las llamadas revisiones técnicas, que proporcionan evidencias objetivas de que los artefactos están completos y conformes con la documentación aplicada para el desarrollo de los proyectos de software en las salidas de las diferentes fases del ciclo productivo. En este caso, al revisar se busca la conformidad y la observancia con lo establecido en el modelo de proceso de desarrollo de software RUP “Rational Unified Process”, a través del lenguaje de modelado UML y otras normativas y especificaciones propias del software y que, además, los cambios realizados a éstos estén correctamente implementados y que afecten solamente a aquellas áreas identificadas por el proceso de la Gestión de la

Configuración. Esto implica que los artefactos estén listos para ser verificados para que puedan considerarse aptos para las próximas etapas (subprocesos), ya que constituyen los elementos de información necesarios para la continuación del proceso en las etapas iniciales. La revisión técnica incluye, además, la constatación de que el artefacto esté habilitado en las carpetas correspondientes a la etapa en cuestión en la biblioteca de producción, acorde a las fechas establecidas en el cronograma.

Las revisiones técnicas deben ser realizadas por los especialistas del área de Calidad o en su defecto por un especialista designado por el Jefe del Proyecto, el cual debe asumir, además, la responsabilidad de dejar constancia de los resultados de las revisiones en los registros habilitados al efecto.

2. Verificación en el desarrollo del software

En la metodología se definen las verificaciones como las actividades que se realizan para comprobar si las salidas en las diferentes etapas y/o subprocesos de desarrollo (código y artefactos) cumplen con las condiciones o los requerimientos impuestos sobre ellos en las entradas por las etapas previas. Se entiende por etapas de desarrollo en el ciclo de vida productivo del software: la gestión de requisitos, el análisis, diseño, codificación y pruebas. Esto se muestra en el esquema presentado en la Figura 2, el cual se estructura a partir de los principios establecidos para los modelos de procesos de desarrollo de productos software orientados a objetos [5].



Las verificaciones en la metodología propuesta constatan que el producto de cada etapa del proyecto es adecuado, completo, consistente y que está acorde a los requerimientos establecidos para el software en las entradas. Esto quiere decir que con las verificaciones se asegura que el producto software satisface los requisitos especificados para el diseño en el transcurso de todas las actividades realizadas durante el ciclo de vida del desarrollo, persiguiendo dos criterios fundamentales en cuanto al código [6]:

- El software debe realizar de forma correcta todas las funciones para las que ha sido concebido.
- El software no debe hacer ninguna función en sí misma o en combinación con otras, que pueda degradar el rendimiento de todo el sistema.

La verificación establece, en cuanto a las pruebas habilitadas al efecto para cumplir los criterios antes enunciados para el código, que debe haber trazabilidad entre los elementos del software y las especificaciones de requisitos, de modo que todos los requerimientos, incluidos nivel de acabado, rendimiento, entre otros; sean adecuadamente cubiertos por pruebas y que los resultados de éstas sean repetibles, aún cuando se hayan producido cambios.

En cuanto a los artefactos, los que constituyen las salidas de las fases iniciales del ciclo de vida de desarrollo de software: las verificaciones, se refieren en primer lugar a que éstos estén consistentes con los requerimientos impuestos en las entradas de las etapas (los que a su vez se materializan también a través de otros artefactos). Esto implica realizar un análisis cualitativo del resultado obtenido para comprobar que se logra lo esperado. También se debe verificar la trazabilidad entre los artefactos de salida con respecto a los de entrada (resultados versus requerimientos), invocando la gestión de la configuración.

La verificación supone, por lo tanto, un proceso de mejora continua, que debe usarse a lo largo de todo el ciclo de vida del producto, y que gracias a la gestión de configuración, permite monitorizar en cualquier momento el grado de seguimiento de los procedimientos establecidos para el desarrollo del producto que garantizan su calidad.

El punto crítico de la verificación [6] es el incremento considerable del costo de desarrollo. No obstante, si se considera todo el ciclo de vida del producto software, es decir, desde que se concibe hasta su abandono, una vez que ha quedado obsoleto; la verificación reduce el costo general del software. Un plan efectivo de verificación reduce en una proporción de 4 a 1 [6], el número de defectos en el sistema desplegado en producción y dado que el costo de reparación de éstos defectos es del orden de 20 a 100 veces [6] superior en operaciones de mantenimiento, que los incurridos en las etapas de diseño, se justifica perfectamente la inversión inicial en prevención mediante procedimientos de verificación.

Las verificaciones fueron concebidas, en cuanto a su ejecución, como realizadas por especialistas que no tengan una relación directa con el proyecto de desarrollo que se esté verificando como contrapartida. Este especialista debe tener la responsabilidad de dejar constancia de los resultados de las verificaciones en el registro habilitado al efecto.

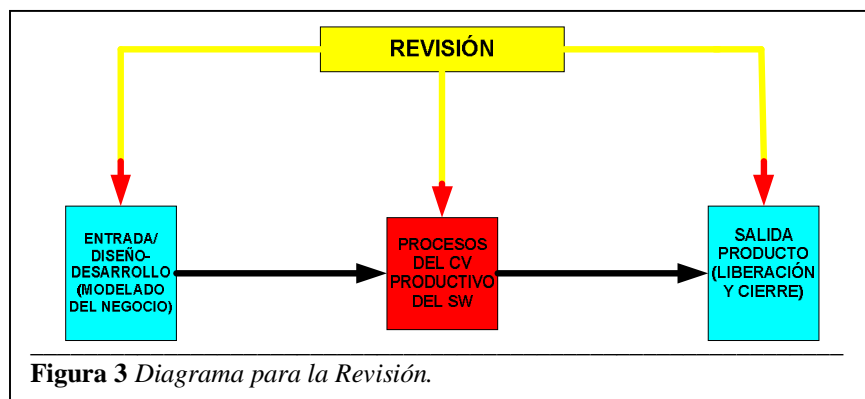
3. Validación en el desarrollo del software

La validación en la metodología propuesta se refiere a la comprobación al final del ciclo de vida del desarrollo, de que el producto creado satisface correctamente la especificación de requisitos del producto y las expectativas que han depositado en el mismo los clientes.

La validación se realiza para comprobar, por tanto, si el producto obtenido cumple los requisitos del cliente. Esta actividad debe realizarse en las áreas del ambiente de operación, lo que constituye las pruebas de aceptación, Pruebas Piloto, por un especialista que registra los resultados y con la participación del cliente.

4. Tareas a realizar por Calidad para el área de Desarrollo

El Especialista de Calidad que asume la ejecución de la metodología debe revisar los artefactos generados en cada una de las etapas declaradas por la documentación vigente en la empresa para la ingeniería del proceso de desarrollo de software, comparando las propiedades y el acabado alcanzado con los requisitos técnicos establecidos en el modelo de proceso de desarrollo de software RUP [5], a través del Lenguaje Unificado de Modelado UML, [7]. Esto se hace extensible a los artefactos propios de la Entrada y de la Salida de los procesos productivos según se muestra en la Figura 3.



REVISIÓN, VERIFICACIÓN Y VALIDACIÓN EN UN PROCESO DE DESARROLLO DE SOFTWARE

Los artefactos que se deben revisar en cada etapa son los que se muestran en la Tabla 1.

TABLA 1 Artefactos objeto de revisión en las fases del ciclo de vida productivo del software	
Etapas	Artefacto
Modelado del Negocio	Especificación de Modelado de Negocio Glosario de Términos Plan de Riesgos
Gestión de requisitos	Catálogo de requisitos Especificaciones de Casos de Uso Modelo de Casos de Uso Modelo de Casos de Prueba Métricas de la etapa
Análisis	Descripción de la arquitectura Realización de Casos de Uso Modelo de análisis Prototipo IGU Métricas de análisis
Diseño	Modelo de diseño Modelo de datos Especificación de Casos de Prueba Métricas de diseño
Codificación	Modelo de codificación Métricas de codificación
Pruebas	Lista de errores Métricas de Pruebas
Liberación	Manual de usuario Manual de instalación Aval de Liberación Adicciones y/o Correcciones a nueva versión
Cierre	Valoración Final del proyecto

Se deben revisar además los artefactos propios de la Gestión de Proyectos (Cronograma, Acta Entrega al Cliente, Acta de Aceptación del Cliente, Solicitud de Cambio y Gestión de Cambio) a través de todas las etapas del proyecto. A los efectos de los proyectos de desarrollo la etapa de Modelado se considera perteneciente al proceso de Gestión de Proyecto.

En cuanto a la verificación, el especialista de calidad debe verificar el cumplimiento satisfactorio de cada una de las etapas declaradas. Para ello se debe analizar el resultado obtenido en la etapa contra los requerimientos impuestos en las entradas de las mismas a través de los artefactos, para valorar si éstos se han alcanzado. En este caso se trata de realizar un análisis cualitativo de los resultados obtenidos en la etapa, que implica tener una visión general de los requerimientos del producto desde la etapa del modelado. Se toma como premisa que se debe verificar siempre luego de haber revisado, o sea, la verificación conlleva el análisis de los resultados obtenidos y plasmados en los artefactos correspondientes, los que no deben tener errores técnicos, ortográficos, etc.

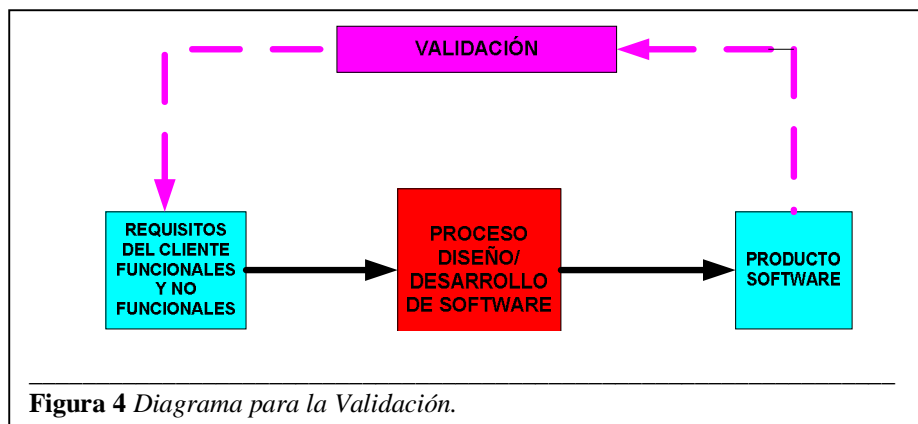
En la etapa de Prueba, el especialista de calidad verificará, ejecutando las pruebas establecidas y/o planificadas en los artefactos para las Especificaciones de los Casos de Prueba, realizando la recogida de las métricas establecidas en la metodología. Esta recogida de las métricas se debe realizar cada vez que se realice una verificación en cualquiera de las etapas de desarrollo del producto software, según se muestra en la Tabla 2.

En cada una de las etapas revisadas/verificadas se emitirá un acta de Aceptación por Calidad de no detectarse No Conformidades o de haber sido éstas corregidas y verificadas, una vez que hayan concluido las revisiones y las verificaciones correspondientes. Para la validación, el especialista de calidad ejecuta las pruebas piloto en presencia de los clientes. Estas pruebas buscan corroborar la conformidad del producto software con respecto a los requisitos funcionales y no funcionales, según se muestra en la Figura 4.

En el caso de detectarse una NO CONFORMIDAD, en la etapa de validación se ejecutarán las acciones definidas en el procedimiento de calidad correspondiente, para poder certificar la eficacia de las acciones correctivas tomadas. Esto se hace extensivo a las etapas de Revisión y de Verificación.

Solamente se admite la aceptación de una actividad de revisión/verificación/validación con observaciones en el caso de la revisión. Se refiere a errores ortográficos puntuales y/o técnicos de menor cuantía que se considere no tengan incidencia

TABLA 2 Métricas a registrar en los procesos de desarrollo en las diferentes actividades	
Actividad	Métricas
Gestión de Requisitos	<ul style="list-style-type: none"> ◊ Cantidad de Requisitos obtenidos. ◊ Cantidad de Casos de Uso obtenidos. ◊ Cantidad de Requisitos abarcados en los casos de uso. ◊ Cantidad de Horas/Hombre consumidas en la actividad. ◊ Cantidad de errores o no conformidades por cada entregable.
Análisis	<ul style="list-style-type: none"> ◊ Cantidad de Casos de Uso realizados ◊ Cantidad de Prototipos diseñados ◊ Cantidad de Clases de Análisis ◊ Cantidad de Horas/Hombre consumidas en la actividad. ◊ Cantidad de errores o no conformidades por cada entregable. ◊ Cantidad de no conformidades del cliente con el Modelo de Análisis.
Diseño	<ul style="list-style-type: none"> ◊ Cantidad de Clases ◊ Cantidad de Horas/Hombre consumidas en la actividad. ◊ Cantidad de errores o no conformidades por cada entregable. ◊ Cantidad de no conformidades del cliente con el Modelo de Diseño.
Codificación	<ul style="list-style-type: none"> ◊ Cantidad de Requisitos abarcados ◊ Cantidad de Casos de Uso implementados ◊ Cantidad de Elementos de Código (clases, métodos, etc.) ◊ Cantidad de Elementos de Código (clases, métodos, etc.) documentados ◊ Cantidad de Componentes reutilizables generados ◊ Cantidad de Horas/Hombre consumidas en la actividad.
Pruebas	<ul style="list-style-type: none"> ◊ Cantidad de Casos de Prueba diseñados ◊ Cantidad de Casos de Prueba ejecutados ◊ Cantidad de Casos de Prueba ejecutados con errores graves ◊ Cantidad de Casos de Prueba ejecutados con errores leves ◊ Cantidad de Casos de Prueba ejecutados con errores de cosmética ◊ Cantidad de Horas/Hombre consumidas en la actividad.
Pilotaje	<ul style="list-style-type: none"> ◊ Cantidad de errores graves detectados durante el pilotaje. ◊ Cantidad de errores leves detectados durante el pilotaje. ◊ Cantidad de errores de cosmética detectados durante el pilotaje. ◊ Cantidad de requisitos no funcionales no tenidos en cuenta en la etapa de Gestión de Requisitos. ◊ Cantidad de requisitos funcionales no tenidos en cuenta en la etapa de Gestión de Requisitos ◊ Cantidad de problemas detectados respecto a la documentación del sistema ◊ Tiempo promedio de aprendizaje por parte del usuario ◊ Cantidad de Horas/Hombre consumidas en la actividad.
Planificación	<ul style="list-style-type: none"> ◊ Cantidad de Horas/Hombre consumidas en la actividad.

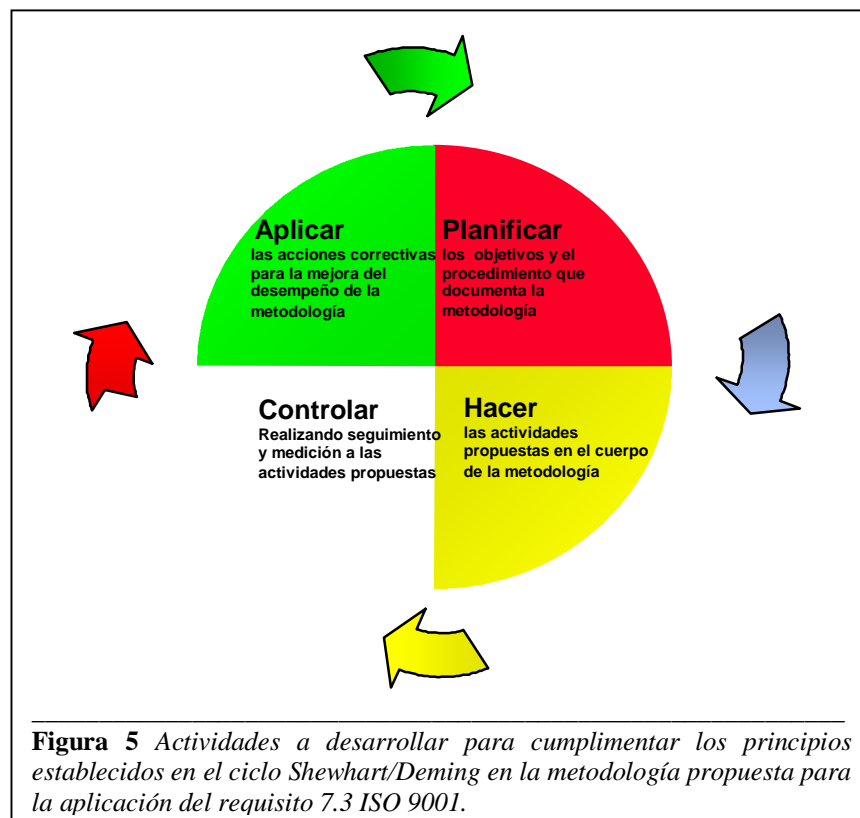


significativa en el desarrollo ulterior del proyecto. De todas formas se hace el señalamiento en el registro, lo que implica la corrección del error por parte de los técnicos, aunque no conlleve seguimiento para constatar la eficacia de la acción tomada. Para poder cumplir con lo establecido en el Ciclo Shewhart/Deming en cuanto a la planificación de las acciones en cualquier tipo de actividad que se pretenda organizar bajo los principios de un Sistema de Gestión de la Calidad, se establecieron en la documentación, procedimientos que soportan la metodología propuesta en unos formatos de registro diseñados para los planes de revisión y verificación. Estos registros deben contener las etapas del proyecto de software que son revisadas o verificadas, las tareas de revisión o verificación requeridas en cada etapa, así como los recursos, las responsabilidades y el cronograma asociado. Las tareas para la revisión serán fundamentalmente asociadas a las revisiones técnicas definidas en NC-ISO 90003: 2006, pero para la verificación serán los distintos tipos de prueba de sistema, unidad, etc.

Por su parte se establece en la documentación para la planificación de la validación un formato de registro que debe contener los requisitos que se validan, las tareas a ejecutar, así como los recursos, responsabilidades y cronograma para la validación.

En el marco de lo referente al control de la metodología propuesta, se ejecutarán acciones de seguimiento y control, con su respectivo registro de los resultados e informe, para la posterior toma de acciones correctivas para la mejora continua del desempeño de la metodología. Estas acciones serán llevadas a cabo por personal designado del área de Calidad y se pueden integrar como uno de los objetivos de un programa de auditorías.

Para una mejor comprensión de las acciones desarrolladas para organizar la implementación de la metodología, se presenta en la Figura 5 la secuencia de actividades para el cumplimiento de lo establecido en el ciclo Shewhart/Deming.



III. RESULTADOS Y DISCUSIÓN

Los resultados obtenidos al aplicar la metodología se evidencian de forma resumida en la Tabla 3, al realizar las revisiones/verificaciones a la documentación del ciclo de vida productivo del software.

Éstos permitieron evidenciar, de manera concisa y eficiente, diferentes hallazgos o sucesos que llevaban implícitos no conformidades con respecto a lo establecido para el proceso de desarrollo de software [5].

Los resultados también facilitaron la aplicación de mejoras en las diferentes etapas del ciclo de vida productivo del software permitiendo una mejor investigación de las causas para la toma de acciones correctivas al ganar en objetividad en cuanto a la definición de sucesos.

TABLA 3 Resultados del control	
Artefacto	Suceso
Catálogo de requisitos	-No declaración de los requisitos no funcionales, RNFs. -Existencia de requisitos funcionales redundantes.
Especificación de Caso de Uso	-Diagrama de Actividades que no se corresponde adecuadamente con el despliegue de las actividades del cliente. -Inexistencia del Diagrama de Actividades en la etapa de Gestión de Requisitos. -Mal diseño del caso de uso por no abarcar todas las funcionalidades definidas en el Catálogo. -No adecuación del caso de uso con la funcionalidad que describe.
Modelo de Caso de Prueba	-Existencia de requisitos funcionales sin caso de prueba asociado. -Inexistencia del artefacto en la etapa de la Gestión de Requisitos. -No inclusión de los flujos alternativos en el modelo.
Realización de Caso de Uso	-No declaración del artefacto para todos los casos de uso declarados en la etapa de la Gestión de Requisitos.
Modelo de Análisis	-Inexistencia de los Diagramas de Colaboración y no listado de los casos de uso que componen los paquetes. -Mala trazabilidad con la etapa de la Gestión de Requisitos por no declaración oportuna del caso de uso desde esta etapa.
Modelo de Datos	-No existencia o declaración del artefacto en la etapa de Diseño.
Modelo de Diseño	-Carencia de formato para definir la distribución del artefacto, lo que define trazabilidad.
Especificación de Caso de Prueba	-No existencia del artefacto en la etapa de Diseño. -Inexistencia de trazabilidad con el requisito involucrado.

Una de las mejoras resultantes fue la de planificar la capacitación técnica a los técnicos, según los diferentes roles, para elevar sus competencias en relación con las exigencias de la norma ISO relacionada con el Sistema de Gestión de la Calidad y el software [1], así como la inclusión de cursos de actualización en temas de Ingeniería Informática en el plan de capacitación de la empresa.


La metodología también influyó en organizar más efectivamente el proceso de Control de calidad, al llevar las acciones involucradas de forma planificada en tiempo y forma con relación a las diferentes etapas del ciclo de vida productivo. Esto permitió agilizar las acciones de revisión/verificación y la posibilidad de controlar más eficazmente la evolución de los diferentes proyectos de desarrollo, todo ello acorde con el proceso de la gestión de proyectos.

La experiencia obtenida confirma la necesidad de insertar la metodología propuesta como parte de un Sistema de Gestión de la Calidad para que los hallazgos se conviertan de manera efectiva en un mejoramiento progresivo de la calidad.

IV. CONCLUSIONES

1. En la metodología propuesta las revisiones técnicas se realizan sobre los artefactos en las diferentes etapas de un proyecto de desarrollo de software, para constatar la conformidad con la documentación establecida y la habilitación en las carpetas de producción
2. En dicha metodología las verificaciones buscan la constatación de que el producto de software de cada etapa de proyecto de desarrollo de software, ya sea artefacto o código, es adecuado, completo y consistente con los requerimientos establecidos en las entradas de cada etapa. En el caso del código, se verifica la trazabilidad entre sus elementos y las especificaciones de requisitos. En el caso de los artefactos, se realiza un análisis cualitativo para verificar el logro del resultado esperado.
3. En la metodología expuesta las validaciones se realizan a través de las llamadas “pruebas piloto” con el cliente, para corroborar la conformidad del producto con respecto a los requisitos funcionales y los requisitos no funcionales.
4. Las acciones para la mejora se realizan a través de un análisis de los resultados obtenidos al aplicar la metodología y una vez que se han efectuado las acciones correctoras que sean necesarias.
5. A partir de los resultados obtenidos se definieron acciones concretas de capacitación encaminadas a mejorar la calidad en el desarrollo del software.

V. RECOMENDACIONES

Se recomienda que las acciones previstas en la presente metodología se integren en el marco de un Sistema de Gestión de la Calidad que sustente su efectividad. 

VI. REFERENCIAS

1. OFICINA NACIONAL DE NORMALIZACIÓN (La Habana, Cuba). *NC-ISO/IEC 90003:06 Ingeniería del software- Directrices para la aplicación de la NC-ISO 9001:01 al software de computación*. La Habana, Cuba: 2006.
2. OFICINA NACIONAL DE NORMALIZACIÓN (La Habana, Cuba). *NC-ISO 9000:05 Sistema de gestión de la calidad. Fundamentos y Vocabulario*. 2da. Edición. La Habana, Cuba: 2005.
3. OFICINA NACIONAL DE NORMALIZACIÓN (La Habana, Cuba). *NC ISO/IEC 12207:03 Tecnología de la Información- Procesos del ciclo de vida del software*. La Habana, Cuba: 2003.
4. GONZÁLEZ, Aleida. "Módulo SGC ISO 9000:2000". En: *Maestría de Calidad Total* 2009).
5. JACOBSON, Ivar; BOOCH, Grady and RUMBAUGH, James. *El Proceso Unificado de Desarrollo de Software*. Madrid: Pearson Educación S.A., 2000.
6. "Calidad-Metodología, RUP, Procesos Ágiles, Aseguramiento". *Kynetia* [en línea]. 2007, [fecha de consulta: 07/07/2010]. Disponible en: <http://www.kynetia.es/especialización/calidad.html>
7. RUMBAUGH, J; JACOBSON, Ivar and BOOCH, Grady. *El Lenguaje Unificado de Modelado. Libro Introductorio*. Madrid: Pearson Educación S.A., 1999.