

**Contribuciones del Instituto Nacional  
de Investigaciones Nucleares al avance de la Ciencia  
y la Tecnología en México**

Edición conmemorativa 2010

# Garantía de calidad del software en el ININ

34

Elvira Gaytán Gallardo  
Departamento de Automatización e Instrumentación  
[elvira.gaytan@inin.gob.mx](mailto:elvira.gaytan@inin.gob.mx)

## 1. Introducción

La garantía de calidad del software es un tema muy importante de la ingeniería de software, que establece acciones planificadas y sistemáticas destinadas a mejorar la calidad del mismo y se traduce en el ahorro de costos de análisis, diseño, desarrollo y mantenimiento, y en general en un software robusto. La necesidad de desarrollar software basado en estándares que permiten al ININ ser proveedor de empresas que requieren software calificado, tales como la Central Nucleoeléctrica de Laguna Verde, llevó al desarrollo de un Plan de Garantía de Calidad de Software (PGCS), para la organización, control y supervisión de tareas en el desarrollo de proyectos de software, que permite garantizar la calidad y funcionamiento adecuado de los sistemas. El PGCS contribuye a eliminar los desarrollos de sistemas en forma aleatoria, usando una estrategia de resolución de problemas de software de una manera organizada. Este plan fue revisado por las gerencias de Garantía de Calidad y de Ciencias Aplicadas, aprobado por la Gerencia de Garantía de Calidad y aceptado por la Dirección de Investigación Científica.

La ingeniería de software fue abordada por Fritz Bauer en la primera conferencia sobre desarrollo de software celebrada en Garmisch, Alemania, en octubre de 1968, patrocinada por el comité de ciencias de la OTAN (Organización del Tratado del Atlántico Norte). Las discusiones en torno a este tema se iniciaron en 1967 por científicos de diversas naciones, miembros de esta organización, con el objetivo de tomar posibles acciones internacionales en el campo de las ciencias de la computación. En otoño de 1967 el comité de ciencias de la OTAN estableció un grupo de estudio, al cual se le encomendaron los problemas relacionados con software. Más tarde este grupo de estudio recomendó realizar una conferencia en ingeniería de software, frase escogida deliberadamente y de una forma provocativa, implicando la necesidad de basar el desarrollo de software en fundamentos teóricos y disciplinas prácticas, como es tradicional en cualquier rama de la ingeniería. La conferencia se realizó en 1968 con la asistencia de más de 50 expertos en el área, discutiendo sobre cinco temas principales: relación del software con el hardware, diseño, producción, distribución y mantenimiento de software.

En esa época, el crecimiento de la demanda de sistemas de computación cada vez más complejos, provocó lo que se llamó “crisis del software”, entre los años 1965 y 1985, aproximadamente. Muchos proyectos importantes superaban con creces los presupuestos y fechas estimados y algunos de ellos eran tan críticos (sistemas de control de aeropuertos, equipos para medicina, entre otros) que sus implicaciones iban más allá de las pérdidas millonarias que causaban. Desde entonces y hasta la fecha han surgido he-

ramientas, metodologías y tecnologías que presentan soluciones a problemas de planificación, costos y garantía de la calidad de software.

En 1977, la NASA (National Aeronautics and Space Administration), el Departamento de Ciencias Computacionales de la Universidad de Maryland y la Corporación de Ciencias Computacionales, crearon el Laboratorio de Ingeniería de Software. La finalidad de este laboratorio fue investigar la efectividad de las tecnologías, metodologías, herramientas y modelos de proceso de la ingeniería de software, en el desarrollo de aplicaciones críticas, para identificar y aplicar en éstas prácticas de desarrollo de software con éxito.

## 2. Antecedentes en el ININ

En el año 1995, en el Laboratorio de Física de Plasmas de la Gerencia de Ciencias Básicas, se iniciaron estudios en ingeniería de software, originados por la necesidad de comprender las etapas de desarrollo de software y su relación con el hardware. Estos estudios siguen avanzando conforme se desarrolla software aplicado a la instrumentación y desde el año 1999 colaborando con el Departamento de Automatización e Instrumentación [1-18].

En el año 2004, en el Departamento de Automatización e Instrumentación se desarrolló el Plan de Garantía de Calidad de Software [19] y los procedimientos que lo complementan [20-24], para obtener la calificación del ININ en el desarrollo de software para la Central Nucleoeléctrica de Laguna Verde (CLV). Se tuvo como antecedente el apoyo proporcionado al Departamento del Acelerador con la SOT ININ-PE-029/03 de CLV “Primera Etapa del Proyecto de Estudios de Vientos de la Central Nucleoeléctrica Laguna Verde en lo que corresponde al Servicio de Software”. Se colaboró también en la elaboración de la Especificación de Requerimientos de Software [25] y del documento Administración de la Configuración de Software [26].

El PGCS fue elaborado a solicitud de la Gerencia de Garantía de Calidad del ININ, debido a la necesidad del Departamento de Calificación y Verificación de la Calidad de Equipos y de la Gerencia de Sistemas, de realizar un sistema para la CLV. Uno de los requisitos para la realización de este sistema fue que el ININ contara con un plan de garantía de calidad de software que permitiera al ININ estar calificado para el desarrollo de software de esta empresa. Esta tarea fue encomendada al Departamento de Automatización e Instrumentación debido a la experiencia mostrada en el área, ante auditoría realizada por la CLV. Para esta fecha, los temas relacionados con el plan de garantía de calidad de software ya habían sido objeto de estudio en la tesis de doctorado “Modelo de Proceso de Software para Programación Gráfica” [27]. En el año 2005 el PGCS y sus procedimientos fueron utilizados para desarrollo de software en el servicio ININ-GCE-008/05 “Actualización e Incorporación de la LECA a la BDI de Ingeniería de la CLV” para la Central Laguna Verde (CLV). El servicio fue desarrollado por el Departamento de Calificación y Verificación de la Calidad de Equipo y por la Gerencia de Sistemas.

En el año 2006 el PGCS fue actualizado y, con el fin de que fuese aplicado en el ININ, se impartió el curso Garantía de Calidad de Software [28], asistiendo personal de la Gerencia de Garantía de Calidad y de los departamentos de Sistemas Electrónicos, Calificación y Verificación de la Calidad de Equipos, Sistemas Nucleares y Automatización e Instrumentación. En este año, CLV solicitó el servicio ININ-GSC-022/06, Contrato: CFE-GCN-5-002-05 “Sistema de Monitoreo Meteorológico (SIMM)”. El sistema fue desarrollado por el Departamento de Automatización e Instrumentación, aplicando el PGCS y sus procedimientos, creando los siguientes documentos:

- Especificación de Requerimientos de Software (ERS) [29],
- Diseño del Software [30], Desarrollo del Software [31],
- Programa de Verificación [32],
- Programa de Pruebas [33],
- Verificación de Especificación de Requerimientos de Software ERS [34],
- Administración de la Configuración de Software ACS y Entrega del Producto de Software EPS,
- Manual del Usuario [35],
- Instrucciones de prueba [36, 37].

Las figuras 1, 2, 3, 4 y 5 presentan: la hoja de firmas de la Especificación de Requerimientos de Software, el programa de verificación, el programa de pruebas y las portadas de un registro de verificación de software y de una instrucción de prueba, respectivamente. Para la administración de configuración de software se verificó que todos los elementos de configuración existieran y estuviesen actualizados a la última versión. La aplicación del PGCS permitió atender satisfactoriamente la auditoría de CLV realizada al sistema.

INSTITUTO NACIONAL DE INVESTIGACIONES NUCLEARES  
GERENCIA DE CIENCIAS APLICADAS



Hoja de Firmas

ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

SISTEMA DE MONITOREO METEOROLÓGICO SIMM

Versión: 2006

Preparado por:	M. en C. Francisco Javier Bucio Valdovinos	Fecha:	06/09/2006
Revisado por:	Ing. Tonatihu Rivero Gutiérrez	Fecha:	4/10/06
Verificado por:	Dra. Elvira Gaytán Cellerco	Fecha:	7/sep/06
Aprobado por:	Dr. Javier C. Palacios Hernández	Fecha:	07-22-06
Acceptado por (CLV):	LCA. Arturo Ramírez García	Fecha:	06-09-2006

2006/09/06

GARANTIA DE CALIDAD  
DOCUMENTO VERIFICADO Y  
LIBERADO POR: V.GC-AU-2006  
CLAVE 4537  
FIRMA: [Firma] FECHA: 12/09/2006

Figura 1. Hoja de firmas de la especificación de requerimientos de software.







 <b>I.N.I.N</b>	<b>Instituto Nacional de Investigaciones Nucleares</b> <b>REGISTRO DE VERIFICACIÓN DE SOFTWARE</b> <b>Plan de Garantía de Calidad de Software</b>	FP.AU-02/1/07
<b>(1) Proyecto o Servicio:</b> Actualización del Sistema de Monitoreo Meteorológico de CLV (1-MMS.1P64)		<b>Registro:</b> R.AU/CLV-1/06
<b>(2) Elemento ó sistema que se verifica:</b> <input checked="" type="checkbox"/> ERS <input type="checkbox"/> DDS <input type="checkbox"/> ACS <input type="checkbox"/> EPS		<b>(3) Número:</b> 1 <b>Fecha:</b> 14/07/06 <b>Total de Hojas Anexas:</b> 0
<b>(4) Identificación de partes a verificar</b> 1. Solicitud del desarrollo del software 2. Obtención de los requerimientos del software. 3. Elaboración del documento ERS. El documento ERS debe contener: 3.1 Descripción general de los requerimientos 3.2 Requerimientos Específicos. <div style="text-align: right; margin-top: 10px;">                     Anexe hoja si es necesario. No. de Hojas anexas: <u>  0  </u> </div>		
<b>(5) Para el caso de Pruebas del Software:</b> Instrucción Aplicada: _____		
<b>(6) Observaciones</b> 1. Se cuenta con la solicitud del desarrollo del software por parte de la Central Nucleo-eléctrica Laguna Verde (ININ - GSC-022/06). 2. Los requerimientos del sistema se concretaron en las reuniones sostenidas con personal de la Central Nucleo-eléctrica Laguna Verde, Participantes de ININ: Tonalíuh Rivero, Raúl Ramírez, Francisco Javier Buclo y Paulino Rojas. Por parte de CLV: Arturo Ramírez García, Rodrigo Mancilla y Alejandro Romero. Durante los días 6, 6 y 7 de julio de 2006. 3. El Documento ERS cumple los puntos especificados en el procedimiento P.AU-10 revisión vigente. Se requiere completar los anexos con la información faltante de parte CLV.		
<b>(7) Resultados</b> <div style="display: flex; justify-content: space-around; align-items: center;"> <input checked="" type="checkbox"/> Conforme                 <input type="checkbox"/> No Conforme             </div>		
<b>(8) Disposición</b> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <input type="checkbox"/> Rechazo  <input type="checkbox"/> Reparación                 </div> <div style="text-align: center;"> <input type="checkbox"/> Reelaboración  <input checked="" type="checkbox"/> Aceptado con Condiciones  <input type="checkbox"/> Aceptado sin Modificaciones                 </div> </div>		
<b>(9)</b> Realizado por: Dra. Elvira Gaytán Gallardo.		Firma: 
Aceptado por Responsable del PSI: Ing. Tonalíuh Rivero Gutiérrez		Firma: 

Figura 4. Verificación de ERS del Sistema de Monitoreo Meteorológico

 <b>ININ</b>	<b>AREA:</b> DEPARTAMENTO DE AUTOMATIZACIÓN E INSTRUMENTACIÓN		
	<b>INSTRUCCIÓN:</b> PRUEBAS AL MÓDULO COMUNICACIÓN CON DATALOGGER, DEL SISTEMA DE MONITOREO METEOROLÓGICO DE CLV (MMS.1P64)		
Nº: LAU/CLV-01	REV.: 0	FECHA DE EMISIÓN: OCTUBRE/2006	HOJA: 1 DE: 5

ÍNDICE	PÁGINA
1. OBJETIVO.....	2
2. REFERENCIAS .....	2
3. DESARROLLO .....	2
3.1. PREPARACIÓN DE LAS PRUEBAS.....	2
3.2. REALIZACIÓN DE LAS PRUEBAS.....	3
3.3. REGISTROS DE LAS PRUEBAS.....	3
4. ANEXOS.....	3

<b>GARANTIA DE CALIDAD</b> DOCUMENTO VERIFICADO Y LIBERADO POR: <i>V.GC-AU.211</i> CLAVE <i>4538</i> FIRMA <i>[Signature]</i>	FECHA: <i>31/OCT/2006</i>
---	------------------------------

<b>PREPARADO POR:</b> DRA. ELVIRA GAYTÁN GALLARDO <i>[Signature]</i>	<b>FECHA:</b> Octubre/2006
<b>REVISADO POR:</b> ING. TONATIUH RIVERO GUTIÉRREZ <i>[Signature]</i>	<b>FECHA:</b> Octubre /2006
<b>APROBADO POR:</b> DR. JAVIER PALACIOS HERNÁNDEZ <i>[Signature]</i>	<b>FECHA:</b> Octubre /2006

Figura 5. Pruebas a un módulo del Sistema de Monitoreo Meteorológico.

En el año 2007, el Departamento de Automatización e Instrumentación desarrolló el análisis de la norma mexicana NMX-I-059-NYCE-2005 [38], a través de MoProsoft (borrador de la norma), del Manual de Calidad y del PGCS y se identificó la compatibilidad del Sistema de Gestión de Calidad del ININ y del PGCS con esta norma [39].

En 2008, la central nucleoelectrónica de Laguna Verde, a través del Departamento de Educación Continua del ININ, solicitó al Departamento de Automatización e Instrumentación el estudio de su programa de aseguramiento PAG-14-2007 [40] y la preparación de un curso para su personal, con base en este programa. Este curso fue redactado bajo lineamientos de CLV y vendido en su totalidad, cediendo el ININ los derechos de autor e impartiendo a profesionales de CLV en noviembre de 2008 y febrero de 2009.

En 2008 fue sometida al OIEA para su evaluación la idea de proyecto ARCAL (Acuerdo Regional de Cooperación para la Promoción de la Ciencia y Tecnología Nucleares en América Latina y el Caribe) "Validación de software aplicado a la instrumentación nuclear". Incorporando además ideas presentadas por Perú y Venezuela y con el apoyo de oficiales del OIEA, México fue el responsable del diseño final del proyecto, el cual se tituló "Actualización de conocimientos, introducción de nuevas técnicas y mejora de la calidad de las actividades de instrumentación nuclear", ARCAL XCIX, iniciando en el año 2009 con la participación de 16 países de la región. México se convirtió en líder de este proyecto, del cual, hasta el momento, se han realizado en nuestro país la reunión inicial de actividades, así como una reunión de expertos de Argentina, Brasil, Chile y México, con la finalidad de crear una metodología de validación de software para su aplicación en los países participantes en el mismo, así como un laboratorio de ingeniería de software (figuras 6 y 7)



Figura 6. Primera reunión de coordinadores del proyecto ARCAL XCIX.



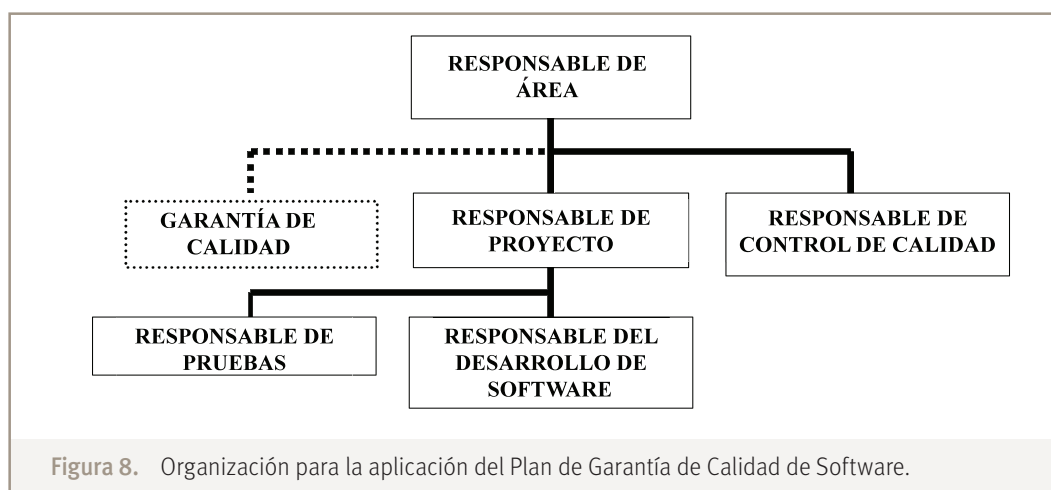
Figura 7. Reunión para desarrollo de una metodología de validación de software, con la participación de Brasil, Argentina, Chile y México.



### 3. Plan de garantía de calidad de software

El PGCS y los procedimientos que lo integran se desarrollaron tomando como referencia estándares del Institute of Electrical and Electronics Engineers (IEEE), guías regulatorias de la Comisión Reguladora Nuclear de los Estados Unidos y los libros de ingeniería de software de Pressman [41] y Sommerville [42].

El plan de Garantía de Calidad de Software se aplica en el desarrollo de productos de software y firmware (combinación de hardware y software, en el cual el software reside en unidades de almacenamiento) desarrollados en el ININ. Este plan toma como referencia el estándar IEEE Std. 730-1998 [43] para desarrollo de software de calidad, controlando y documentando las etapas del ciclo de vida del software y se apoya en cinco procedimientos: Especificación de Requerimientos de Software (ERS) [20], Diseño y Desarrollo de Software (DDS) [21], Verificación y Validación del Software (VVS) [22], Documentación para el Usuario [23], y Administración de la Configuración del Software (ACS) [24]. La figura 8 presenta la organización para la aplicación del PGCS.



El diagrama de la organización para la aplicación del PGCS presenta la distribución de las responsabilidades de los participantes en un proyecto de software. El *responsable de área* se encarga de definir con el cliente las condiciones del contrato y requerimientos preliminares, asegurar que los productos se realizan aplicando el PGCS y sus procedimientos, y preparar y firmar el acta de entrega del producto. El *responsable de control de calidad* debe proporcionar entrenamiento a los responsables del desarrollo de software y de pruebas para la aplicación del PGCS, asegurar que todos los requerimientos se encuentran contenidos en la especificación de requerimientos de software, elaborar o supervisar el programa de verificaciones y validaciones, y ejecutarlo. El *responsable del proyecto* se encarga de interpretar el contrato y definir la especificación de requerimientos, informar al personal técnico los requerimientos, aplicar el PGCS y sus procedimientos, mantener comunicación con el cliente, preparar o revisar los documentos que resulten del proyecto, y asegurar que todos los requerimientos se encuentran contenidos en la especificación. El *responsable del desarrollo de software* debe aplicar el PGCS en el proyecto, diseñar y desarrollar el software y generar los documentos del proyecto. El *responsable de pruebas* realiza el programa de verificaciones y validación. *Garantía de Calidad* es responsable de verificar el cumplimiento del PGCS y auditar periódicamente su eficacia.

#### 3.1 Especificación de Requerimientos de Software (ERS)

La Especificación de Requerimientos de Software (ERS) toma como referencia el estándar IEEE 830-1998 [44] y presenta una descripción detallada de los requerimientos a cumplir por el producto de software. La ERS contiene los requerimientos esenciales del software y sus interfaces externas, siendo la base para el

desarrollo o adquisición del software. Debe contener una explicación detallada de las necesidades que habrán de resolverse mediante un sistema, así como el alcance, variables y/o procesos de entrada y salida, interfaces, interacciones con otros componentes, restricciones de uso, propiedades funcionales y no funcionales, etc. Estos requerimientos pueden ser establecidos en el contrato que se elabora entre el cliente y el Instituto. La ERS debe ser documentada y los requerimientos deben ser verificados y validados.

### 3.2 Diseño y Desarrollo de Software (DDS)

El diseño de software toma como referencia el estándar IEEE 1016-1998 [45] y muestra la manera en que el software es estructurado para satisfacer los requerimientos identificados en la ERS, mediante la proyección de los requerimientos en la descripción de la estructura del software, componentes, interfaces y datos necesarios para la etapa de implementación. El diseño de software es una representación o modelo del sistema de software que será creado y debe proporcionar información precisa, lo cual facilita la planeación, análisis e implementación. Es necesario describir cómo el diseño cumple con los requerimientos de software. Para el desarrollo del software se debe describir la información del autor y del sistema, el código fuente, las herramientas de desarrollo, directorios que contienen el código fuente y el procedimiento para hacer los programas ejecutables. El desarrollo de software se documenta como un anexo al diseño de software, cuando éste es requerido.

### 3.3 Verificación y Validación del Software (VVS)

Se realiza en cada una de las etapas del Ciclo de Vida del Software (CVS), incluyendo su documentación. La VVS toma como referencia el estándar IEEE 1012-1998 [46] y describe los pasos para su ejecución. Es necesario generar un registro de verificación y validación del software, el cual debe contener al menos los siguientes puntos: nombre del sistema o módulo que será verificado y validado, datos de las personas implicadas en la verificación y validación, identificación de partes a verificar y validar, criterios de aceptación, proceso de verificación y validación, resultados de la verificación, discrepancias, acciones correctivas y conclusiones.

#### 3.3.1 Plan de Verificación y Validación de Software (PVVS)

Los procesos de verificación y validación de software proporcionan los requerimientos para una evaluación objetiva de productos de software a través del ciclo de vida, comprobando si los requerimientos de software y del sistema son correctos, completos, exactos, consistentes y comprobables. Comprueban además si los productos de cada fase de desarrollo cumplen los requerimientos o condiciones impuestas en la fase previa y si el sistema o componente final cumple con los requerimientos especificados.

El PVVS establece todas las pruebas a realizar en cada etapa del ciclo de vida, con el fin de garantizar que el producto de cada etapa contenga los requerimientos necesarios para la siguiente, por lo que cada una de ellas generará un reporte de verificación y validación como producto resultante. El PVVS identifica las actividades que deben ser verificadas y validadas a través del ciclo de vida del software, descritas a continuación.

#### a) Administración

La administración se ejecuta durante todos los procesos y actividades del ciclo de vida del software. Es en esta actividad donde se genera el Plan de Verificación y Validación del Software (PVVS). Durante el desarrollo de esta actividad se realiza una revisión continua del proceso de verificación y validación y se preparan los programas de desarrollo (calendarización) y seguimiento. La administración coordina además los resultados del proceso de verificación y validación con otros resultados, como aseguramiento de calidad, administración de configuración, revisiones y auditorías; identifica los requerimientos de software que son afectados por cambios y planea las tareas de verificación y validación de los cambios. La administración del proceso de verificación y validación realiza también mediciones del proyecto.

**b) Adquisición**

El proceso de adquisición se inicia con la definición de necesidades para adquirir un sistema, producto o servicio de software. Continúa con la preparación de una solicitud, selección de proveedores y la administración del proceso de adquisición a través de la aceptación del sistema, producto o servicio de software.

**c) Suministro**

Este proceso se inicia con una propuesta del proveedor, como respuesta a la solicitud de un sistema, producto o servicio de software, y continúa con la determinación de procedimientos y recursos necesarios para la administración del proyecto, desarrollo y ejecución de planes del proyecto, con tiempos de ejecución.

**d) Desarrollo**

El proceso de desarrollo contiene las actividades y tareas a desarrollar, análisis de requerimientos, diseño, codificación, integración, pruebas, instalación y aceptación de los productos de software. Todas las actividades deben ser verificadas y validadas.

**e) Operación**

Es el uso del software por el usuario final en el ambiente de operación. La verificación y validación se realizan mediante pruebas operacionales y del sistema con la finalidad de evaluar nuevas restricciones, proponer cambios y evaluar los procedimientos de operación.

**f) Mantenimiento**

El proceso de mantenimiento se inicia cuando se requieren modificaciones del código y documentación asociada, causada por un problema o necesidad de mejorar el sistema. La verificación y validación de las tareas de mantenimiento cubren modificaciones correctivas, adaptativas y de perfeccionamiento, migración y retiro del software.

### 3.3.2 Etapas del ciclo de vida del software que deben ser verificadas y validadas

**a) Especificación de requerimientos**

Se verifican y validan los requerimientos de software y su seguimiento, y se generan los planes de prueba de verificación y validación del sistema y de su aceptación.

**b) Diseño de software**

Se verifican y validan las vistas de diseño del software, se generan y verifican los planes de prueba de componentes y de integración, y se realizan el diseño y verificación de pruebas.

**c) Desarrollo de software**

Se evalúa el código fuente y su documentación, se generan y verifican los procedimientos y casos de prueba, se ejecutan y verifican las pruebas de componentes.

**d) Pruebas**

Esta actividad incluye todas las pruebas del software, de integración y de calidad del sistema. Se genera y verifica el procedimiento de pruebas de aceptación, y se ejecutan y verifican las pruebas de integración del sistema y de aceptación. El objetivo de la verificación y validación de las pruebas es asegurar que se cumple con los requerimientos especificados, de software y del sistema.

#### e) **Instalación y entrega del sistema**

Consiste en la instalación del producto de software en el ambiente de trabajo y la revisión y aceptación del comprador. Se realiza una auditoría de la configuración de instalación y se genera el “Reporte Final de Verificación y Validación” del software. El PVVS debe incluir un programa de pruebas de aceptación que contenga los criterios de aceptación contra los que se va a medir el resultado, los cuales resultan de la especificación de requerimientos.

#### 3.3.3 **Reporte de Verificación y Validación del Software (RVVS)**

El RVVS integra los resultados de la ejecución del PVVS, los cuales servirán para determinar si el software está listo para su uso. El RVVS deberá incluir los registros de las revisiones, pruebas de aceptación y el resumen de anomalías y resoluciones.

### 3.4 **Documentación para el usuario**

Los requisitos que se deben cumplir para la elaboración de manuales del usuario se encuentran en el procedimiento P.AU-12 y toman como referencia el estándar IEEE Std. 1063-2001 [47]. Los puntos que el manual debe contener son: título del manual del usuario, índice, introducción, definiciones, información de uso del manual, Instrucciones de instalación, operación del software y hardware, información de los comandos del software y referencias. El manual debe incluir la explicación del uso del software, uso de los elementos de las interfaces gráficas, tales como botones, iconos, controles, indicadores y el uso especial de teclas. El propósito y operación de cada función deben ser descritos, incluyendo las interfaces con otras funciones, así como las precauciones y limitaciones del sistema. Todos los mensajes de error deben ser incluidos en el documento con una identificación del problema, caso probable y las acciones correctivas que el usuario debe aplicar. El manual debe contener la descripción de todos los problemas conocidos con suficiente detalle para que el usuario pueda hacer que el sistema se recupere por sí mismo y reporte claramente el problema al personal de soporte técnico. El manual debe contener la descripción de los problemas que pueden ocurrir al usar el software y la forma de restablecer las operaciones.

### 3.5 **Administración de la Configuración del Software (ACS)**

Para la Administración de la Configuración del Software se aplica el procedimiento P.AU-06, que toma como referencia el estándar IEEE 828-1998 [48]. La ACS es un conjunto de actividades que se aplican al iniciar un proyecto de software y termina con el fin del ciclo de vida del software. Este proceso permite conocer el estado de cada uno de los elementos de la configuración del software, garantizando que no se realizan cambios incontrolados, y que todos los participantes del desarrollo y los que utilizan el sistema, disponen de la versión adecuada de los productos del proyecto. Los elementos de configuración del software son los componentes de hardware y software, o ambos, que conforman un producto y que se tratan como entidades simples en el proceso de administración de la configuración. La ACS permite conocer el estado de cada uno de los elementos de la configuración del software, garantizando que no se realizan cambios incontrolados y que todos los participantes del desarrollo y los que utilizan el sistema, disponen de la versión adecuada de los productos del proyecto.

#### 3.5.1 **Plan de Administración de la Configuración del Software (PACS)**

El PACS establece el mecanismo de identificación y control de los componentes de un sistema de software, incluyendo su documentación, con el fin de mantener vigente la última versión de los mismos.

## 4. El plan de garantía de calidad de software y su compatibilidad con la norma mexicana NMX-I-059-NYCE-2005 para la industria del software

En el año 2005, el *Diario Oficial de la Federación* anunció la publicación de la Norma Mexicana *Tecnología de la información –Software– Modelos de procesos y evaluación para desarrollo y mantenimiento de software*, NMX-I-059-NYCE-2005, en cuatro partes: Definición de conceptos y productos, Requisitos de procesos, Guía de implantación de procesos y Directrices para la evaluación de procesos.

El documento que sirvió de base a la norma mexicana para la industria de desarrollo y mantenimiento de software, se denominó “*Modelo de Procesos para la Industria de Software (MoProSoft)*” y fue desarrollado, a solicitud de la Secretaría de Economía, por la Facultad de Ciencias de la Universidad Nacional Autónoma de México [49]. El modelo está dirigido a empresas o áreas dedicadas al desarrollo y/o mantenimiento del software y puede ser usado por empresas que no cuentan con procesos establecidos. Las organizaciones que tienen procesos definidos pueden usarlo como punto de referencia para identificar los puntos que faltan por cubrir. Debido a que el ININ ya cuenta con procesos establecidos, se realizó un estudio de MoProSoft, con el fin de identificar la compatibilidad que el PGCS tiene con la norma NMX -I-059-NYCE-2005.

### 4.1 Metodología

La metodología empleada para determinar la compatibilidad del PGCS con la norma consistió en:

- Estudio y análisis del documento MoProSoft.
- Estudio y análisis del manual de calidad [50].
- Revisión del PGCS [19].

#### 4.1.1 Estudio y análisis del documento MoProsoft

MoProSoft pretende proporcionar a las industrias de software en México, que en su mayoría son pequeñas y medianas empresas (PyMES), un modelo basado en el estándar ISO/IEC 15504:2003. Las principales características de MoProSoft son: fácil de entender, fácil de aplicar, no costoso en su aplicación y puede ser la base para alcanzar evaluaciones exitosas con otros modelos como CMMI [51] del Instituto de Ingeniería de Software de la Universidad Carnegie Mellon, en Estados Unidos. MoProSoft está enfocado en nueve procesos y considera las tres categorías básicas de la estructura de una organización que son: La alta dirección, gerencia y operación.

- La categoría alta dirección contiene el proceso de gestión de negocio y proporciona los elementos para el funcionamiento de los procesos de la categoría de gerencia y se retroalimenta con la información generada por ellos.
- La categoría de gerencia está integrada por la gestión de procesos, proyectos y recursos y proporciona los elementos para el funcionamiento de los procesos de la categoría de operación; recibe y evalúa la información generada por éstos y comunica los resultados a la categoría de alta dirección.
- La categoría de operación está integrada por los procesos de administración de proyectos específicos y de desarrollo y mantenimiento de software. Esta categoría realiza las actividades de acuerdo a los elementos proporcionados por la categoría de gerencia y entrega a ésta la información y productos generados.

MoProSoft basa su estrategia de mejora en que la organización debe implantar los procesos definidos por el modelo, los cuales evolucionarán con base en las sugerencias de mejora, alcanzando los objetivos del plan estratégico de la organización con metas cada vez más ambiciosas.

#### 4.1.2 Manual de Calidad

El Instituto Nacional de Investigaciones Nucleares, a través del Manual de Calidad [50] y sus procedimientos, establece, documenta, implementa, mantiene y mejora la eficacia de los procesos del Sistema de Gestión de Calidad.

El Manual de Calidad establece las medidas necesarias para la aplicación del Sistema de Gestión de Calidad, basado en las normas ISO 9001:2000 y NMX-CC-9001-IMNC-2000.

#### 4.1.3 Plan de Garantía de Calidad del Software

El PGCS ha sido explicado en el inciso 3.

#### 4.1.4 Discusión

De los estudios realizados al Modelo de Procesos para la Industria del Software, al Manual de Calidad y al Plan de Garantía de Calidad de Software, se obtiene la información de la tabla 1, donde se indica la compatibilidad que existe entre la norma NMX-I-059-NYCE-2005 con el Sistema de Gestión de Calidad (Manual de Calidad) y el PGCS.

Los nueve procesos considerados por la norma NMX-I-059-NYCE-2005, obtenidos con base en el Modelo de Procesos para la Industria del Software, son cubiertos en el Instituto Nacional de Investigaciones Nucleares a través del Manual de Calidad y del Plan de Garantía de Calidad de Software y son descritos a continuación:

**a) Gestión de Negocios**

Establece la razón de ser de la organización, sus objetivos y las condiciones para lograrlos, para lo cual es necesario considerar las necesidades de los clientes, así como evaluar los resultados para poder proponer cambios que permitan la mejora continua. Habilita a la organización para responder a un ambiente de cambio y a sus miembros para trabajar en función de los objetivos logrados.

**b) Gestión de Procesos**

Establece los procesos de la organización en función de los procesos requeridos e identificados en el plan estratégico. Define, planifica e implanta actividades de mejora en los mismos.

**c) Gestión de Proyectos**

Asegura que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización.

**d) Gestión de Recursos**

Consigue y dota a la organización de los recursos humanos, infraestructura, ambiente de trabajo y proveedores, así como crea y mantiene la base de conocimiento de la organización.

**e) Recursos Humanos y Ambiente de Trabajo**

Proporciona los recursos humanos adecuados para cumplir las responsabilidades asignadas a los roles dentro de la organización, así como la evaluación del ambiente de trabajo.

**f) Bienes, Servicios e Infraestructura**

Proporciona proveedores de bienes, servicios e infraestructura que satisfagan los requisitos de adquisición de los procesos y proyectos.

**g) Conocimiento de la Organización**

Mantener disponible y administrar la base de conocimiento que contiene la información y los productos generados por la organización.

**h) Administración de Proyectos Específicos**

Establece y lleva a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.

**i) Desarrollo y Mantenimiento de Software**

Realiza de manera sistemática las actividades de análisis, diseño, construcción, integración y pruebas de productos de software, nuevos o modificados, cumpliendo con los requerimientos especificados.

Es importante resaltar que los puntos correspondientes al Manual de Calidad y al PGCS no son descritos en el documento debido a que el Manual de Calidad puede ser consultado en la página del ININ, mientras que los documentos del Plan de Garantía de Calidad de Software y Procedimientos están resguardados por la Gerencia de Garantía de Calidad del ININ. El lector interesado en el tema puede consultarlos.

**Tabla 1. Compatibilidad de la Norma NMX- I-059-NYCE-2005, con el Manual de Calidad del ININ y el Plan de Garantía de Calidad de Software**

MoProSoft	Manual de Calidad	PGCS
Gestión de Negocios	Punto 5.1. Compromiso de la Dirección	
Gestión de Procesos	Punto 5.5.2. Representante de la Dirección	
Gestión de Proyectos	Punto 5.6. Revisión por la Dirección.	
Gestión de Recursos	Punto 6. Gestión de Recursos.	
Recursos Humanos y Ambiente de Trabajo	Punto 6.2. Recursos Humanos	
Bienes, Servicios e Infraestructura	Punto 6.3. Infraestructura	
Conocimiento de la Organización	Se realiza a través del Sistema Integral de Información Administrativa	
Administración de Proyectos Específicos	Punto 7. Realización del Producto, Punto 7.1. Planificación de la realización del producto	El PGCS y procedimientos contribuyen a la planificación y realización de productos de software
Desarrollo y Mantenimiento de Software	Punto 7.3 Diseño y Desarrollo	Se aplican procedimientos para Análisis, Diseño, Desarrollo, Configuración, Verificación, Validación y Documentación del Software.

## 5. Resultados

Contar con un Plan de Garantía de Calidad de Software ha permitido la participación del ININ en el desarrollo de sistemas que requieren de software calificado, tales como la central nucleoelectrica de Laguna Verde. De acuerdo al modelo de madurez de capacidades (CMM) del Instituto de Ingeniería de software de la Universidad Carnegie Mellon [51], cuyo objetivo es mejorar la calidad de los procesos de desarrollo, administración y mantenimiento de software, al aplicar el PGCS y sus procedimientos, el ININ se encuentra en un nivel de madurez 2. Este nivel de madurez se define como un proceso disciplinado donde se pueden repetir tareas en proyectos de software y se aplica la administración del proyecto para seguimiento de coste, planificación y funcionalidad.

Desde 1995 se ha venido utilizando la ingeniería de software en el desarrollo de sistemas que involucran hardware y software, conforme se ha avanzado en la investigación en el ININ, lo cual ha sido reportado en las referencias. Esto ha facilitado la elaboración de documentos que plasman la resolución de problemas de software de una manera organizada.

Con el análisis de MoProsoft, del Manual de Calidad y del PGCS se ha logrado identificar la compatibilidad del Sistema de Gestión de Calidad del ININ con la norma NMX-I-059-NYCE-2005. El desarrollo del Plan de Garantía de Calidad de Software ha permitido al ININ ser proveedor de empresas que requieren software calificado [52] y aplicar este plan en el desarrollo de software para automatización de procesos o sistemas. Otro resultado importante es la participación en eventos nacionales e internacionales, donde se resalta la importancia de la ingeniería de software en el ámbito nuclear y otros.

## 6. Perspectiva de la garantía de calidad de software

A medida que aumenta la calidad en el desarrollo de software, basado en estándares y procedimientos, se incrementa la posibilidad de calificar en el desarrollo de sistemas hacia el exterior del Instituto. A través del análisis y estudio de los adelantos en el área de ingeniería de software, refiriéndose principalmente a la Garantía de Calidad de Software, se ha logrado avanzar de forma consistente en el ININ, obteniendo conocimiento de la situación actual con respecto a la norma NMX-I-059-NYCE-2005. Esto permite contar con bases que pueden ser usadas cuando se requiera desarrollar software certificado por esta norma mexicana. Los avances obtenidos hasta el momento permiten al ININ:

- Desarrollar e impartir cursos de Garantía de Calidad de Software.
- Aplicar el PGCS en sistemas desarrollados en el ININ para automatización de equipos y/o de procesos.
- Verificar el cumplimiento del PGCS en el desarrollo de sistemas realizados para la CLV.
- Ser consultores en esta área.
- Contribuir a nivel internacional, a través de proyectos como el ARCAL XCIX, en Centro América, el Caribe y América del Sur, aplicando los conocimientos adquiridos en las investigaciones realizadas, con la finalidad de crear una metodología de validación de software para su aplicación en los 16 países miembros del proyecto.

## Agradecimientos

Se hace patente el agradecimiento al Departamento de Calificación y Verificación de la Calidad de Equipos, y a las Gerencias de Sistemas y de Garantía de Calidad, el haber propiciado el desarrollo del Plan de Garantía de Calidad de Software, permitiendo que los estudios realizados en ingeniería de software fuesen aplicados en la resolución de problemas reales. Al ingeniero Tonatiuh Rivero Gutiérrez por su apoyo en la revisión de los procedimientos; al maestro en ciencias Alejandro Arenas Ochoa por la revisión del PGCS; al ingeniero Marcelino Ambriz Jaimes por la aprobación del PGCS; al doctor Javier Palacios Hernández por la revisión del PGCS y aprobación de los procedimientos, y al doctor Luis Carlos Longoria Gándara por la aceptación del PGCS. Al grupo editor del Modelo de Procesos para la Industria del Software de la Facultad de Ciencias de la UNAM, por haber puesto el modelo MOPROSOFT a disposición del público, a través de Internet. Finalmente, al Organismo Internacional de Energía Atómica por el apoyo brindado a través del proyecto ARCAL XCIX.



## Bibliografía

1. *Report on a Conference Sponsored by the NATO Science Committee*, Garmisch, Germany, 1968.
2. *Software Engineering Laboratory Series, SEL-82-906, Annotated Bibliography of Software Engineering Laboratory Literature*, Florida, EUA, National Aeronautics and Space Administration, 1990.

## Referencias

1. Celis González OC, Gaytán Gallardo E., Graficación de las señales generadas por el Tokamak Novillo, IV Congreso Internacional de Investigación en Ciencias Computacionales CIICC97. Septiembre de 1997, Durango, Dgo. Instituto Tecnológico de Durango, 71-79, 1997.
2. Gaytán Gallardo E. Sistema de programación gráfica para un digitalizador de señales (2262), 1er. Simposio Internacional de Computación. México, D. F. Noviembre de 1997, Instituto Politécnico Nacional. 493-499, 1997.
3. Gaytán Gallardo E. Sistema de Programación gráfica para el digitalizador 8212A, VII Congreso Científico y Tecnológico ININ-SUTIN, Ocoyoacac, Edo. de Méx. Diciembre de 1998, Instituto Nacional de Investigaciones Nucleares, 143-147, 1997.
4. Gaytán Gallardo E, Ramírez Jiménez FJ. Instrumentación Virtual para control de sistemas GPIB y tarjetas de adquisición de datos. IX Seminario de Especialidades Tecnológicas SETEC98, ININ-IMP-IIE. Ocoyoacac, Edo. de Méx. Julio de 1998. Instituto Nacional de Investigaciones Nucleares, 1998.
5. Gaytán Gallardo E, Ortega C. Sistema de procesamiento de datos para espectroscopía en el Tokamak Novillo, VIII Congreso Científico y Tecnológico ININ-SUTIN, Ocoyoacac, Edo. de Méx. Diciembre de 1998. Instituto Nacional de Investigaciones Nucleares, 101-106, 1998.
6. Gaytán Gallardo E. Estudio de Modelos de Procesos de Software, VIII Congreso Científico y Tecnológico ININ-SUTIN, Ocoyoacac, Edo. de Méx. Diciembre de 1998. Instituto Nacional de Investigaciones Nucleares, 147-152, 1998.
7. Gaytán Gallardo E. Instrumentación Virtual en la Adquisición de datos a través del bus GPIB, 1er. Taller Internacional de Instrumentación Virtual y Sistemas Digitales, Querétaro, Qro. Agosto de 1999. Centro de Investigación en Computación, Instituto Politécnico Nacional, 10-15, 1999.
8. Gaytán Gallardo E. Metodología para mejorar la calidad en la integración de sistemas con instrumentación virtual, X Congreso Científico y Tecnológico ININ-SUTIN, Ocoyoacac, Edo. de Méx. Diciembre de 2000. Instituto Nacional de Investigaciones Nucleares, 39-22, 2000.
9. Gaytán Gallardo E, Cruz de León CH. Automatización de una Liofilizadora, 3er. Simposio Inter-Tecnológico de Computación e Informática, Puebla, Pue. Octubre de 2002. Instituto Tecnológico de Puebla, 357-364, 2002.
10. Gaytán Gallardo E, Ortiz Hernández J., Segovia de los Ríos JA, Knowledge Representation of acquisition and Control System with Graphical Programming using UML Notation. *WSEAS Transactions on Systems* **3**, 1712-1717, 2004.
11. Gaytán Gallardo E, Ramírez Jiménez FJ, Ortiz Hernández J, Segovia de los Ríos JA, Longoria Gándara LC, Requirements Analysis for Graphical Programming and a Practical Experience in a System for the Measurement of X-Rays. *WSEAS Transactions on Systems* **3**, 1718-1725, 2004.
12. Gaytán Gallardo E, Ramírez Jiménez FJ, Application of Software Engineering in the Development of Instrumentation for Medical Physics, Guadalajara, Jal. Marzo 2004, American Institute of Physics. Conference Proceedings of the eighth Mexican Symposium on Medical Physics, **724**, 150-155, 2004.

13. Gaytán Gallardo E, Rivero Gutiérrez T, Palacios Hernández J, Ortiz Hernández J, Segovia de los Ríos JA. Is the Software Engineering essential in the Nuclear Sciences? Nuclear Science Symposium, Roma, Italia, October 2004, Nuclear Science Symposium Conference Record IEEE, 3, 1877-1880, 2005.
14. Gaytán Gallardo E., Desales Galeana G. *Refurbishing of a Freeze Drying Machine, used in Nuclear Medicine for Radiopharmaceuticals Production*. Guanajuato, Gto. Marzo, 2006, American Institute of Physics. Conference Proceedings of the Ninth Mexican Symposium on Medical Physics, 2006; **854**, 101-103, 2007.
15. Gaytán Gallardo E, Tovar Muñoz VM, Cruz Estrada P, Vergara Martínez FJ, Rivero Gutiérrez T. *Software System for the Calibration on X-ray Measuring Instruments*. Guanajuato, Gto. March, 2006, American Institute of Physics. Conference Proceedings of the Ninth Mexican Symposium on Medical Physics, **854**, Issue 1, 154-156, 2007.
16. Gaytán Gallardo E, Software Requirements Analysis for Nuclear Experiments, Nuclear Science Symposium, San Diego, EUA, October 2007, Nuclear Science Symposium Conference Record, IEEE, **3**, 978-981, 2008.
17. Gaytán Gallardo E, Rivero Gutiérrez T, Quality Assurance for Validation of Software Applied to Nuclear Instruments Validation Procedures of Software Applied in Nuclear Instruments. Vienna, Austria, 2006, IAEA-TECDOC-**1565**, 13-21, 2007.
18. Gaytán Gallardo E, Ortiz Hernández J, Quality Assurance for Software Development Applied to Electronic Instruments, Cuernavaca, Mor. Septiembre de 2007, Electronics, Robotics and Automotive Mechanics Conference Record, IEEE Computer Society, 712-717, 2007.
19. Gaytán Gallardo E, Rivero Gutiérrez T. Plan de Garantía de Calidad de Software, PL.GC-2. Rev. o, ININ, Julio de 2004. Revisión de Palacios Hernández J y Arenas Ochoa A; aprobación de Ambriz Jaimes M; aceptación de Longoria Gándara L. C.
20. Gaytán Gallardo E. Especificación de requerimientos de Software, PAU-10. Rev. o, ININ, Julio de 2004.
21. Gaytán Gallardo E. Diseño y Desarrollo de Software, PAU-11. Rev. o, ININ, julio de 2004.
22. Gaytán Gallardo E., Verificación y Validación del Software, PAU-07. Rev. o, ININ, agosto de 2004.
23. Gaytán Gallardo E. Documentación del software, PAU-12. Rev. o, ININ, agosto de 2004.
24. Gaytán Gallardo E. Administración de la Configuración del Software, PAU-06. Rev. o, ININ, julio de 2004.
25. Gaytán Gallardo E, Rivero Gutiérrez T, Marroquín González JL. Especificación de requerimientos de Software, primera etapa del proyecto de estudios de vientos de la Central Nucleoeléctrica Laguna Verde en lo que corresponde al servicio de software. IT.AU-03-09 Rev. 1. 2003.
26. Marroquín González J. L. Gaytán Gallardo E., Administración de la Configuración del Software, primera etapa del proyecto de estudios de vientos de la Central Nucleoeléctrica Laguna Verde en lo que corresponde al servicio de software. IT.AU-03-11, Rev. o. 2003.
27. Gaytán Gallardo E, *Modelo de Proceso de Software para Programación Gráfica*, Tesis de Doctorado, Metepec, Edo. de México. Instituto Tecnológico de Toluca, 2005.
28. Gaytán Gallardo E, *Manual del Curso Garantía de Calidad de Software*, ININ, 2006.
29. Bucio Valdovinos, FJ. IT.AU-0608. Rev. o. Sistema de Monitoreo Meteorológico SIMM, Especificación de Requerimientos de Software (ERS).
30. Bucio Valdovinos FJ. IT.AU-0611. Rev. o. Diseño del Software para el Sistema de Monitoreo Meteorológico de CLV SIMM, 2006.
31. Bucio Valdovinos, FJ. IT.AU-0616. Rev. o. Sistema de Monitoreo Meteorológico SIMM, Desarrollo del Software, 2006.
32. Gaytán Gallardo E, Programa de Verificación, Sistema de Monitoreo Meteorológico PR.V.AU/CLV-1/06. Rev. o, 2006.

33. Gaytán Gallardo E, Programa de Pruebas, Sistema de Monitoreo Meteorológico PR.PB.AU/CLV-1/06, Rev. 0, 2006.
34. Gaytán Gallardo E, Registro de Verificación de Software de la Especificación de Requerimientos de Software (ERS), del Sistema de Monitoreo Meteorológico R.AU/CLV-1/06, 2006.
35. Bucio Valdovinos, FJ, MU.AU-0601. Sistema de Monitoreo Meteorológico SIMM, Manual del Usuario. Rev. 0, 2006.
36. Gaytán Gallardo E, Instrucción: Pruebas al Módulo: Comunicación con datalogger del Sistema de Monitoreo Meteorológico I.AU/CLV-01, Rev. 0, 2006.
37. Gaytán Gallardo E, Instrucción: Pruebas al Sistema de Monitoreo Meteorológico I.AU/CLV-09, Rev. 0, 2006.
38. NMX-I-059-NYCE-2005, *Tecnología de la información. Software. Modelos de procesos y evaluación para desarrollo y mantenimiento de software*. México, 2005.
39. Gaytán Gallardo E. El Plan de Garantía de Calidad de Software y su Compatibilidad con la Norma NMX-I-059-NYCE-2005 para la Industria del Software. XVII Congreso Científico y Tecnológico ININ-SUTIN. Ocoyoacac, Edo. de México, Diciembre de 2007, Instituto Nacional de Investigaciones Nucleares, 47-51, 2007.
40. Carballo Chávez G, Ortega Wright E, de Santiago Naranjo ML, Lozano Gómez R, Aguirre C. A, Cárdenas Jáuregui JB, Carrera Morales JA, Vega Lara VR, Muñoz Huerta J, Álvarez Lara JL, Vivanco González G. Hernández de la Garza R. *Programa de Aseguramiento de Calidad de Software de la GCN*. PAG-14-2007, Gerencia de Centrales Nucleoeléctricas, México. 2007.
41. Pressman Roger S, *Software Engineering, a Practitioner's Approach, Fifth Edition*, EUA, McGraw-Hill. 2001.
42. Sommerville Ian. *Software Engineering, Seventh edition*, EUA, Addison Wesley. 2004.
43. IEEE Std. 730-1998, *Software Quality Assurance Plans*. Nueva York, EUA, 1998.
44. IEEE Std. 830-1998, *Software Requirements Specifications*. Nueva York, EUA, 1998.
45. IEEE Std. 1016-1998, *Recommended Practice for Software Design Descriptions*. Nueva York, EUA, 1998.
46. IEEE Std. 1012-1998, *Software Verification and Validation Plan*. Nueva York, EUA, 1998.
47. IEEE Std. 1063-2001, *Software User Documentation*. Nueva York, EUA, 2001.
48. IEEE Std. 828-1998, *Software Configuration Management Plans*. Nueva York, EUA, 1998.
49. Oktaba H, Alquicira Esquivel C, Su Ramos A, et al, *Modelo de Procesos para la Industria de Software (MoProSoft)*, versión 1.3, UNAM. México, 2005.
50. Pérez Gómez FP, Arenas Ochoa A, Olivares Orozco LA, Ambriz Jaimes M, Ortiz Magaña JR, *Manual de Calidad M.CAL-1*, Rev. 4, ININ. 2007
51. <http://www.sei.cmu.edu/cmmi/general/>
52. Gaytán Gallardo E, Rivero Gutiérrez T, Palacios Hernández J, 2005, *El Plan de Garantía de Calidad de Software permite al ININ ser calificado como Proveedor de Software de Laguna Verde*, Contacto Nuclear **39**, ININ. México, 2005.