

La Validación del Software: Un Requisito Técnico de la Norma ISO/IEC 17025.

Introducción

La validez de la información de la medición es de interés para todos aquellos que la utilizan o son afectados por ésta. Las mediciones, particularmente las realizadas en el ensayo y la calibración, a menudo involucran procesos complejos, utilizan una gran variedad de software, hardware, personas y formatos de datos.

El uso de software apropiado le facilita a la organización disminuir los errores y bajar los costos, compartir los datos, utilizar conjuntamente diferentes componentes, guías y prácticas para proveer procesos eficaces y eficientes. El principio de mejora continua debe ser aplicado también al software.

Cuando el laboratorio utiliza computadoras o equipos automatizados para la adquisición, procesamiento, registro, generación de informes, almacenamiento o recuperación de datos de ensayo o calibración, el laboratorio debe asegurar los aspectos que hemos analizado anteriormente; ya que los mismos inciden en la obtención de resultados técnicamente válidos, exactos y confiables. Sin embargo, el conocimiento de la importancia de la validación del software como adecuado para el uso, de porqué debe hacerse, cuándo debe hacerse, y saber exactamente lo que necesita realizarse, parece ser insuficiente.

Muchas orientaciones relacionadas con los métodos de validación son ofrecidas en la literatura científica, pero la mayoría de las veces éstas no son utilizadas adecuadamente por los laboratorios, los cuales no poseen personal experto en la materia (ingeniería del software). En algunos laboratorios se ve el proceso de validación del software como algo que sólo puede hacerse de una forma externa al laboratorio, en colaboración con otros laboratorios u organizaciones. Por

consiguiente, no desarrollan protocolos internos de validación que son de gran ayuda en numerosas situaciones, además de ser técnicamente factibles, como es el caso de la validación de hojas de cálculo.

La validación del software es necesaria ya que proporciona un alto grado de confianza y seguridad en el software y en los resultados que se obtienen al aplicarlo.

Para los laboratorios de ensayo y calibración que se encuentran acreditados o en proceso de acreditación, según los requisitos establecidos en la Norma ISO/IEC 17025 (NVC 2534:2000) "Requisitos generales para la competencia de los laboratorios de ensayo y calibración", la validación del software desarrollado por el usuario es un requisito técnico que debe ser cumplido con el mismo rigor que el resto de los requisitos.

La calificación de "adecuado para el uso", aplicada a un software relacionado con los procesos de ensayo o calibración, es una garantía de que la influencia del software sobre el resultado de la medición está bajo control, al realizar el análisis del proceso de medición como sistema, único enfoque que evalúa integralmente el proceso de medición.

Una vez más se ratifica, tomando como objeto de estudio la validación del software, que medir es un arte y que para ser buenos "maestros" en el arte de medir, hay que delimitar objetivamente todos los factores que influyen en el resultado de la medición, indudablemente, la utilización de software validado es uno de ellos.

Principios generales a utilizar en la validación

1. Especificación de los requisitos.

Una especificación documentada de los requisitos del software proporciona

la base para la validación y la verificación. El proceso de validación del software no puede completarse sin un establecimiento de las especificaciones de los requisitos.

2. Prevención de defectos.

Las necesidades del aseguramiento de la calidad del software fijan su atención en la prevención de defectos en el proceso de desarrollo del software y no en probar la calidad del código del software después que se escribe. La prueba del software está muy limitada en su capacidad de detectar todos los defectos latentes en el código del software. La complejidad de la mayoría del software impide que sea probado exhaustivamente. La prueba del software es una actividad necesaria.

3. Tiempo y esfuerzo.

La validación del software requiere tiempo y esfuerzo. La preparación para la validación debe comenzar con anticipación; es decir, durante la planificación del diseño y desarrollo y el diseño de la entrada de datos. La conclusión final que muestra que el software se encuentra validado debe estar basada en la evidencia recolectada a partir de los esfuerzos planificados dirigidos a lo largo del ciclo de vida del software.

4. Ciclo de vida del software.

La validación del software tiene lugar dentro del ambiente del ciclo de vida establecido del software. El ciclo de vida del software contiene las tareas de ingeniería de software y la documentación necesaria para soportar la validación del software. Además, el ciclo de vida del software contiene las tareas específicas de verificación y validación que son apropiadas para el uso previsto del software. En la presente nota técnica

no recomendamos un modelo particular de ciclo de vida (modelo lineal secuencial, modelo de construcción de prototipos, modelo de desarrollo rápido de aplicaciones, entre otros), sólo establecemos que se deben seleccionar los modelos más apropiados a utilizar en el proyecto de desarrollo del software. Varios modelos del ciclo de vida del software son definidos en la ingeniería del software.

El ciclo de vida puede ser seguido completamente o tener variaciones en su desarrollo, debido a las propias características y naturaleza del software que se desea desarrollar. Siguiendo las etapas del ciclo de vida se asegura la calidad y se proveen evidencias de validación.

5. Planificación.

El proceso de validación del software se define y controla a través de un plan. El plan de validación del software define lo que será logrado a través del proceso de validación del software. Los planes de validación del software son una herramienta importante del Sistema de Gestión de la Calidad de la organización.

Los planes de validación del software especifican aspectos tales como el alcance, el método de validación, los recursos, el cronograma, los tipos y la magnitud de las actividades que se deben llevar a cabo para la validación.

6. Procedimientos.

El proceso de validación del software se realiza a través del uso de procedimientos documentados. Estos procedimientos establecen "cómo", "quién" y "cuándo" se llevará a cabo la validación del software. Los procedimientos deben identificar las acciones específicas o la sucesión de acciones que deben tomarse para

completar las actividades individuales de validación.

7. Validación del software después de un cambio.

Debido a la complejidad del software, un cambio aparentemente pequeño puede tener un impacto significativo en todo el sistema. Cuando se hace cualquier cambio al software (incluyendo pequeños cambios), el estado de validación del software necesita ser restablecido. Siempre que el software sea cambiado, un análisis de validación debe dirigirse no solamente para la validación del cambio individual, sino también para determinar la magnitud e impacto de ese cambio en la totalidad del software.

8. Alcance de la validación.

El alcance de la validación debe estar basado en la complejidad del software y en los riesgos de seguridad; no en el tamaño de la organización o la restricción de los recursos. La selección de las actividades y tareas a llevar a cabo durante la validación deben corresponderse con la complejidad del diseño del software y el riesgo asociado con la utilización del software para el uso previsto. La documentación de la validación debe ser suficiente para demostrar que todos los planes y procedimientos de validación del software se han completado con éxito.

9. Independencia de la validación.

Las actividades de validación deben ser conducidas cumpliendo el precepto básico de gestión de la calidad sobre la "independencia de la revisión". La auto-validación es sumamente difícil. Cuando sea posible, siempre es mejor llevar a cabo el proceso de validación de manera independiente, sobre todo para las aplicaciones que poseen un

alto riesgo. Algunas organizaciones optan por la contratación de una tercera parte independiente, pero esta solución no siempre es factible. Otro enfoque es asignar la validación a miembros del personal de la propia organización que no están involucrados en el desarrollo del software o en su aplicación, pero que tienen suficiente conocimiento para evaluar el proyecto y conducir el proceso de validación. En estos casos las organizaciones más pequeñas necesitan ser creativas en la disposición y asignación de las tareas para mantener en todo momento la independencia.

10. Flexibilidad y responsabilidad.

La aplicación específica de estos principios de validación del software puede ser muy diferente de una aplicación a otra. El fabricante o desarrollador del software tiene flexibilidad a la hora de escoger como aplicar estos principios de validación, pero es el responsable de demostrar que el software se ha validado. Las etapas del proceso de validación descritas aquí pueden ser simplificadas dependiendo de la naturaleza del software y de su uso previsto.

Actividades del ciclo de vida del software

No es recomendable el uso de un modelo específico del ciclo de vida del software. Los diseñadores del software deben establecer un modelo del ciclo de vida del software que sea apropiado a su producto y organización. En la figura 1 se muestra un ciclo de vida básico. El modelo del ciclo de vida que se seleccione debe cubrir desde la creación del software hasta el retiro del uso (desinstalación) del mismo.

A continuación se relacionan las actividades que incluyen un modelo típico del ciclo de vida del software:

- a. Planificación de la calidad.
- b. Definición de los requisitos del sistema.
- c. Especificación detallada de los requisitos del software.
- d. Especificación del diseño del software.
- e. Implementación (construcción o codificación).
- f. Prueba.
- g. Instalación.
- h. Operación y apoyo.
- i. Mantenimiento.
- j. Desinstalación.

El ensayo, la verificación, y otras tareas de apoyo a la validación del software se llevan a cabo durante cada una de las actividades del ciclo de vida. El modelo del ciclo de vida organiza estas actividades de desarrollo del software en varias formas y suministra un marco para dar seguimiento y controlar el proyecto de desarrollo de software.



Consideraciones para las hojas de cálculo

Existen requisitos que deben ser considerados por el laboratorio para el desarrollo de las hojas de cálculo, los cuales son requeridos para hacer efectivo el trabajo de validación del software. Estos requisitos incluyen:

- a. Utilizar las características de seguridad y protección del software que se utiliza, por ejemplo:
 - Limitar el acceso a cada hoja de cálculo individual;
 - Limitar los cambios en todo un libro;
 - Utilizar claves de protección para hojas de cálculo individuales y libros de trabajo para limitar el acceso, exigiendo una contraseña (password) para abrirlo o guardarlo, o puede recomendarse que se abra como un libro de sólo lectura;
 - Proteger las celdas dentro de cada hoja de cálculo.
- b. Instrucciones de trabajo documentadas para el uso de cada libro de trabajo u hoja de cálculo, situadas preferiblemente en el propio libro de trabajo u hoja de cálculo, en forma digital;
- c. Juego de datos patrón para verificar el correcto funcionamiento de la hoja de cálculo cada vez que es utilizada;
- d. Establecer en un lugar visible de la hoja de cálculo o del libro de trabajo (preferiblemente en la primera hoja) la información relacionada con el nombre y la versión, nombre de la aplicación para la cual fue validada, autor, etc. Toda la

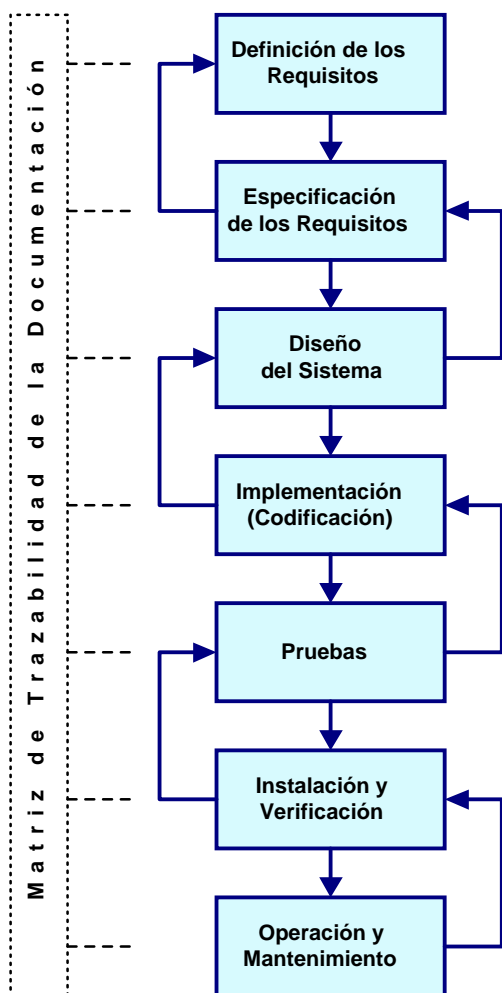


Figura 1.
Ciclo de vida básico del software.

información señalada anteriormente debe aparecer en el informe producido por la hoja de cálculo;

- e. Interfaz usuario consistente, para ello:
- Las hojas de cálculos y los libros de trabajo deben estar estructurados, utilizando los formatos más adecuados, según sea el caso. Las áreas correspondientes a datos de entrada, procesamiento de datos, resultados de los cálculos y datos constantes que son utilizados en los cálculos, deben estar bien delimitadas;
 - Deben utilizarse los recursos de validación que el software posee (definir entradas válidas de una celda, configurar restricciones y mensajes, auditorías de elementos no válidos);
 - Deben utilizarse títulos significativos;
 - Deben ocultarse las filas, columnas, hojas de cálculo y datos que no necesitan ser visibles al operador;
 - Las barras de herramientas que no son utilizadas deben deshabilitarse, al igual que las líneas de división de cada hoja de cálculo;
 - Las opciones del menú deben disminuirse al mínimo requerido para operar la hoja de cálculo satisfactoriamente.

Documentación de la validación

La documentación de la validación es un requisito muy importante y necesario en el Sistema de Gestión de la Calidad del laboratorio. Durante todo el ciclo de vida del software deben quedar evidencias objetivas del progreso y cumplimiento de cada etapa. Nótese en la figura 1 la presencia de la matriz de trazabilidad de la documentación.

La documentación de la validación del software debe incluir los siguientes aspectos:

- a. Los requisitos definidos por el usuario.
- b. El plan de desarrollo del software.

- c. El protocolo de validación utilizado.
- d. El criterio de aceptación.
- e. Las pruebas y sus resultados.
- f. Las conclusiones sobre si el software se ajusta o no al uso previsto.
- g. El manual del usuario del software.

Referencias

1. U.S. Department Of Health and Human Services. Food and Drug Administration. Center for Devices and Radiological Health. Center for Biologics Evaluation and Research. General Principles of Software Validation. Final Guidance for Industry and FDA Staff. 2002.
2. NPL. Software Support for Metrology. Best Practice Guide No. 3: Guidance on Developing Software for Metrology. 2001.
3. Gregory D. Gogates. Software Validation in Accredited Laboratories, A Practical Guide. 2001.
4. NPL Report CMSC 04/00. Guidelines to help users select and use software for their metrology applications. 2000.
5. NPL Report CISE 25/99. A methodology for testing spreadsheets and other packages used in metrology. 1999.
6. EA/GA(98)95. Guidelines for the use of computers and computer systems in accredited laboratories. 1998.
7. Roger S. Pressman. Ingeniería del software: Un enfoque práctico. 4^{ta} Edición. 1998.

Preguntas y comentarios

Dirigirlas por correo electrónico a la siguiente dirección:

laboratorio@lysconsultores.com



NOTAS TÉCNICAS PUBLICADAS

- NT 001/03 Organización de Comparaciones Interlaboratorio para Laboratorios de Calibración.**
<http://www.lysconsultores.com/nt001.htm>
- NT 002/03 Trazabilidad de las Mediciones.**
<http://www.lysconsultores.com/nt002.htm>
- NT 003/03 La Acreditación de Laboratorios un Reconocimiento de Competencia Técnica.**
<http://www.lysconsultores.com/nt003.htm>
- NT 004/03 Validación de Métodos de Ensayo.**
<http://www.lysconsultores.com/nt004.htm>
- NT 005/03 Verificación y Calibración: Mitos y Realidades.**
<http://www.lysconsultores.com/nt005.htm>

Temas de las próximas notas técnicas...

Errores de los Procesos de Medición.

El Sistema de Gestión de las Mediciones (ISO 10012:2003).

Cálculo de la Incertidumbre de la Medición en el Laboratorio de Ensayo.

Utilización de Gráficos de Control en el Laboratorio.

Lineamientos para la Utilización de Computadoras y Sistemas Computarizados en el Laboratorio.

¿Cuáles temas le gustaría que analicemos en las próximas NOTAS TÉCNICAS?



Contáctenos



Calle Urdaneta, N° 2A, 1^{er} Piso, entre Avenida Aragua y Calle Simón Rodríguez
(a 100 m del C.C. Maracay Plaza) - Maracay, Estado Aragua
Master-Telefax: (0243) 235 86 86 Telf: (0243) 236 38 89
formacion@lysconsultores.com
www.lysconsultores.com