

Una Visión de la Aplicación del Software en Metrología

Yenny Maritza Peña Puerto (ypena@cdtdegas.com)
César Augusto Almeida Ortega (calmeida@cdtdegas.com)
Corporación Centro de Desarrollo Tecnológico del Gas

Todas las teorías son legítimas y ninguna tiene importancia. Lo que importa es lo que se hace con ellas. Jorge Luis Borges (1899-1986) Escritor argentino

Conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico, orientados a producir bienes y servicios de utilidad económica, social, y política.

En esta sección, cada semestre, expertos nacionales y/o extranjeros, ofrecerán artículos técnicos que buscan sensibilizar a nuestros lectores, acercándolos con conocimiento, a la aplicación de la metrología en las diferentes actividades de nuestra sociedad.

RESUMEN.

En la actualidad, el uso de los sistemas automatizados es un rasgo característico de los procesos de medición y calibración de instrumentos. Dentro de estos sistemas existen diferentes aspectos que deben ser tenidos en cuenta para el aseguramiento de la cadena de medida, el análisis de datos y la generación de resultados. En el presente artículo se tratan los principales aspectos enfocados a la especificación, implementación y validación de los procesos de adquisición, transmisión, análisis y almacenamiento de datos en este tipo de sistemas con el fin garantizar la calidad las mediciones.

1. INTRODUCCION

Con el crecimiento de la electrónica, las telecomunicaciones y las denominadas tecnologías de la información - TI, el software toma cada vez más importancia dentro de los diferentes campos de la ciencia, y la metrología no es la excepción. Dentro de esta ciencia el software forma parte importante de cada uno de los puntos de cadena de medida, el análisis de datos y la generación de resultados. Dependiendo de la necesidad y la etapa de uso dentro del proceso de medición, estos software se pueden categorizar en: tipo COTS (Comercial Off-The-Shelf), suministrados por un proveedor o fabricante; tipo MOTS (Modified Off-The-Self), suministrados por un proveedor o fabricante con posibilidad de personalizaciones según las necesidades; y tipo CUSTOM (personalizado), desarrollados a la medida de necesidades específicas. Estos últimos pueden ser desarrollados por un tercero o por miembros de la misma institución que lo requiere.

En términos básicos, un sistema de medición o calibración automatizado puede dividirse en tres etapas, como se observa en la Figura 1. Cada una de estas etapas posee una función dentro del proceso de medición, por lo tanto, existen diferentes aspectos a considerar en la especificación, el desarrollo, la documentación y la validación del software asociado a las mismas. El no considerar ó ejecutar de forma adecuada estos aspectos, puede llevar diferentes problemas, desde la dificultad en el uso de los mismos, hasta la generación de resultados erróneos. También es importante mencionar que al igual que existen diferencias en los aspectos de desarrollo, existen diferencias en los procesos de pruebas y validación del software, de acuerdo con la función que éste cumple dentro de la metrología y la entidad en la cual es utilizado.

Considerando el panorama planteado, en este artículo se pretende realizar una descripción general de los procesos en los cuales interviene el software y el porqué de su importancia. Lo anterior, en gran medida, desde la perspectiva de las experiencias y desarrollos llevados a cabo durante los últimos seis (6) años, a lo largo de los cuales se ha logrado asimilar que el software debe cumplir su ciclo de vida considerando el fin último previsto, que para el caso de las aplicaciones utilizadas en metrología, consiste en suministrar resultados técnicamente válidos. Se inicia con instrumentos de medida, pasando por plataformas de control y adquisición de datos, para llegar, por último, al procesamiento y análisis de datos. En cada uno de estos puntos de aplicación se presentan las consideraciones más relevantes de especificación, desarrollo y validación del software desde la perspectiva del aseguramiento de la calidad de los resultados de medición.

2. SOFTWARE ASOCIADO A INSTRUMENTOS DE MEDIDA

Al hacer referencia, en este artículo, al software de un instrumento de medida, se habla de un software desarrollado específicamente para la medición de variables de proceso por medio de un instrumento, y el manejo y/o configuración de los parámetros asociados al mismo. Sin embargo, de acuerdo al mensurando, al principio de operación, al tipo de elemento (primario o secundario), el software desarrollado para el instrumento varía de complejidad, implementando opciones que van más allá de la configuración de unidades, intervalos de medición, coeficientes de corrección, como por ejemplo herramientas de diagnóstico o salidas que proporcionan acciones de control.



Figura 1. Etapas de un Sistema de Medición o Calibración Automatizado.

2.1 Requerimientos de Software para Instrumentos de Medida

El diseño del software para un instrumento de medida tiene como fin permitir al operador o usuario acondicionar el instrumento para realizar la medición y ajustar la forma en la que éste presenta o transmite la información; por otra parte, dependiendo de la variable a medir y la capacidad del instrumento, en ocasiones es necesario acceder a información, que si bien está asociada de forma directa a la variable de proceso, involucra una gran cantidad de parámetros que alteran de forma significativa tanto el funcionamiento del instrumento, como las características de las mediciones que se obtienen de él. Por esta razón, el software debe desarrollarse para que determine, en primera instancia, el modelo del instrumento, la información acerca del principio de medición, las características y límites de operación, y de esta forma, desarrollarse para que le permita al usuario sólo las modificaciones o configuración de parámetros dentro de límites permisibles (es importante destacar que el software asociado a instrumentos de medida, generalmente sólo es desarrollado por fabricantes de instrumentos, a diferencia del caso de adquisición y análisis de datos). De igual forma, se deben crear niveles de acceso para impedir que personal no autorizado realice modificaciones a parámetros propios del instrumento, por ejemplo la modificación del span del instrumento, que repercute directamente sobre la señal de salida, la cual, generalmente corresponde al tradicional lazo de salida analógica de 4-20 mA y, en el caso de no realizar la misma modificación en el sistema que adquiere esta señal, generaría un error en la medición. Esto mismo aplica a la modificación de otros parámetros, como tiempo de respuesta o pre-procesamiento de la señal de salida (promediado, filtrado, etc.), entre otros.

En el caso que se implemente la generación de alarmas en el instrumento (por ejemplo alarmas por nivel de medición, bajo o alto) debe especificarse la acción que se realiza y la consecuencia que esto trae en la salida del instrumento (por ej.: interrupción de la emisión de pulsos o cambio de la variable de salida a un valor preestablecido fuera del intervalo de medición). Por último, es importante que se documente la totalidad de parámetros a los que se puede tener acceso con el software, cuáles de ellos pueden ser configurados, para qué niveles de seguridad y qué consecuencias trae su modificación.

¹ Acrónimo de "Supervisory Control And Data Acquisition".

2.2 Desarrollo de la Interfaz de Usuario

Debido a la importancia que tiene la correcta configuración de los parámetros propios del instrumento, es primordial que la interfaz de usuario se encuentre subdividida en pantallas o secciones que indiquen claramente el campo de acción de cada parámetro modificable; por ejemplo una pantalla o sección dedicada a la variable principal (valor mínimo, valor máximo, amortiguamiento de la señal), otra pantalla o sección dedicada a la configuración de alarmas o salidas digitales, y otra pantalla o sección para la configuración de los protocolos de comunicación.

Para el ingreso de parámetros es importante que en caso que el usuario intente fijar un valor fuera de lo permitido, el software especifique claramente la falla, al igual que cuando, por alguna razón, se interrumpa la medición o la emisión de la señal de salida. También es importante incluir en la interfaz un indicador de estado de proceso, que muestre al usuario el avance en la ejecución de alguna función, la lectura o escritura exitosa de parámetros.

3. SOFTWARE ASOCIADO A CONTROL Y ADQUISICIÓN DE DATOS

El sistema de control y adquisición de datos SCADA¹ de un proceso de medición y/o calibración es uno de los componentes claves para el éxito del mismo. Es importante aclarar que el SCADA no es únicamente software, sino que es un componente hardware-software, donde éste último es desarrollado e implementado sobre una plataforma (hardware) particular. Otro aspecto a resaltar es que, como su nombre lo indica, en muchos casos este sistema debe atender una doble responsabilidad: adquisición y control. Primero, la adquisición de los datos correspondiente a las diferentes variables que forman parte del modelo matemático y, como segunda responsabilidad, mantener control sobre una o varias variables del proceso, junto con una determinada secuencia de actividades (para el caso de los bancos de calibración). La ejecución de estas actividades depende en gran medida de la capacidad de procesamiento de datos, la cual varía con respecto a la tecnología de cada sistema SCADA, al igual que la configuración, disponibilidad y versatilidad de módulos y tarjetas de expansión e interfaces disponibles. Con este panorama, a continuación se presentan los aspectos más relevantes a considerar en el momento de seleccionar, implementar o adquirir un sistema de SCADA para un proceso asociado a metrología.

3.1 Requerimientos para la Plataforma de Adquisición de Datos

Parte de los requerimientos del SCADA están asociados a la instrumentación que se ha seleccionado para realizar la medición y/o calibración, debido a que esta instrumentación condiciona la cantidad de señales, los tipos (analógicos o digitales), las características eléctricas, las frecuencias de operación y/o tiempos de respuesta. Aunque estos requerimientos están más asociados al hardware (en particular a las tarjetas entrada/salida, las cuales pueden incluso afectar la clase de exactitud de instrumento de medida), guardan una relación con el software desarrollado dentro del SCADA. Dentro de los aspectos a considerar en este sentido, se resalta la prioridad que se debe dar a la adquisición de las señales de los instrumentos asociados a la o las variables principales dentro del sistema (por ej.: señal de salida del medidor de volumen en un sistema de transferencia de custodia de fluidos). Es en este punto donde comienzan a presentarse inconvenientes, en particular, cuando el sistema también debe realizar labores de control, las cuales generalmente se priorizan sobre la adquisición de datos, lo que puede generar retardos, falta de sincronía en la adquisición o errores sistemáticos cuando el tiempo hace parte del modelo matemático. Una práctica recomendada en estos casos es contar con doble SCADA o por lo menos doble controlador², uno dedicado sólo a la adquisición de datos y otro sólo al control de proceso.

Otro aspecto a considerar es la ejecución de cálculos dentro del SCADA; esto corresponde a una práctica común para el caso de sistemas a nivel de control de procesos industriales, pero en el caso de aplicaciones asociadas a metrología, esto puede llegar a ser inadecuado. Es necesario realizar un estudio de la capacidad de procesamiento con el que cuenta el SCADA, debido a que la implementación de los modelos matemáticos de procesos de medición, dentro de este tipo de plataformas, puede generar resultados imprecisos debido a redondeos, limitantes en los operadores matemáticos disponibles, entre otros, además de la dificultad para rastrear los valores de entrada de dichos modelos o la modificación de los mismos. Por otro lado, está el hecho que el ejecutar operaciones numéricas dentro de los controladores generalmente tiene altos costos en términos de ocupación del ciclo de máquina, lo cual, puede disminuir el tiempo de respues-

ta del sistema en la adquisición de las variables principales del modelo de medición, generando las dificultades ya mencionadas. Cuando se requiere la estimación de incertidumbre asociada al proceso de medición, estos problemas se aumentan, haciendo completamente inadecuado que los cálculos se ejecuten dentro del SCADA. Lo más apropiado es que los cálculos necesarios para obtener los resultados de medición y su incertidumbre asociada sean ejecutados por herramientas de análisis de datos desarrolladas para tomar los datos de entrada de los valores adquiridos por el SCADA, a través de archivos planos (algunas veces denominados archivos históricos), bases de datos o servidores OPC³.

Adicionalmente, también es necesario que el software del SCADA permita ejecutar en lazo cerrado (es decir, considerando toda la cadena de medida) labores de aseguramiento metroológico y el ajuste y/o corrección de errores sistemáticos cuando sea necesario. El tipo de corrección a implementar en el sistema varía de un sistema SCADA a otro; el menos exigente y generalmente más adecuado es el que se realiza con base en ecuaciones lineales. Sin embargo, correcciones por métodos más complejos (ecuaciones polinómicas, linealización por punto, corrección predictiva) se encuentran disponibles actualmente, y la conveniencia o no de su implementación está asociada a la relación costo - beneficio (tiempo de respuesta y uso del ciclo de máquina - nivel de corrección requerido), la cual depende del desempeño del sistema SCADA.

3.2 Desarrollo de la Interfaz de Usuario

La interfaz de usuario constituye un elemento fundamental en el desarrollo de la automatización de los procesos de medición y/o calibración de instrumentos. Su función principal es permitir la interacción del metrologo con el proceso, para lo cual la interfaz debe ser clara, intuitiva y concreta, evitando la sobrecarga de información o la ambigüedad de los parámetros que en ella se presentan. Del mismo modo se debe tener precaución en los tiempos de actualización de datos en pantalla, ya que, tiempos de actualización muy rápidos pueden generar alto tráfico de datos y afectar la velocidad de respuesta del sistema de forma innecesaria.

Otra de las funciones de la interfaz del SCADA es la configuración de algunos de parámetros de operación del sistema. El ajuste o la elección de parámetros específicos únicamente debe realizarse en un modo operativo concreto del sistema (comúnmente modo de configuración). Lo más

² *Dispositivos electrónicos que proveen la inteligencia requerida para la automatización.*

³ *Acrónimo de "Object Linking and Embedding for Process Control".*

adecuado es que algunos parámetros específicos deban estar protegidos (parámetros inalterables) y otros puedan ser accesibles para una persona autorizada (parámetros configurables). Al respecto de estos parámetros es importante que se cuente con una descripción detallada de su función, unidades de medida y criterios de selección o modificación, ya que una inadecuada configuración de los mismos puede ocasionar problemas de operación o errores en las mediciones, incluso en sistemas estándar como los electrocorrectores y computadores de flujo en los sistemas de transferencia de custodia de fluidos.

Una de las claves del buen desempeño de una interfaz de usuario es su documentación (generalmente llamada manual de usuario). En ésta se debe realizar una descripción detallada de los menús, diálogos, pantallas del software, las funciones y el significado de los datos empleados para la ejecución de algoritmos y/o cálculos. La fallas en la configuración y manejo de los SCADA, en muchos casos se debe a documentación incompleta o ambigua, para lo cual es importante considerar adecuados procesos de formación y evaluación de quienes operan el sistema dentro de la adquisición o desarrollo del mismo (aspecto muchas veces obviado por los proveedores de este tipo de sistemas).

3.3 Generación de Archivos y Datos Históricos

La información que se obtiene al automatizar un proceso de medición y calibración de instrumentos es usada en análisis posteriores para la generación de resultados; por tal razón, es necesario asegurar, que aunque los denominados archivos históricos se utilicen en otro lugar o en una fecha posterior, sean almacenados o transmitidos sin afectar la calidad de las mediciones. Como primera medida se debe seleccionar un dispositivo de almacenamiento con la capacidad suficiente para recopilar los datos generados en el proceso de medición y/o calibración y así evitar la pérdida de información, y que a su vez posea la permanencia suficiente como para garantizar que los datos no se corrompan en condiciones normales de almacenamiento.

En el diseño de la estructura de archivos históricos, es importante tener en cuenta que se debe facilitar la posibilidad de que el software, que ha de procesar o visualizar los valores de medida y datos complementarios, verifique información relevante del proceso y determine la autenticidad e integridad de los datos. Si se detecta una irregularidad, los datos se deben descartar o

marcar como inservibles. También es importante que estén correctamente documentadas, la estructura de estos archivos, las variables y la unidad de medida en las cuales son registradas, la frecuencia de registro y el separador decimal, entre otros.

3.4 Validación

El objetivo principal en el proceso de validación es comprobar el correcto funcionamiento de la plataforma de adquisición y determinar si el flujo de datos es adecuado y satisface los requerimientos planteados inicialmente para la aplicación. Se debe verificar que las funciones y/o algoritmos son apropiadas y que la lectura de señales, tanto análogas como digitales, no presenta pérdida o alteración de información. Adicionalmente, es necesario evaluar si la interfaz de usuario presenta la información de forma inequívoca y si las restricciones de acceso o seguridad se encuentran implementadas y en funcionamiento. Por último, si se poseen interfaces para establecer comunicación con otros dispositivos electrónicos, con el usuario o con partes del software distintas de aquellas críticas en cuanto a metrología, su funcionamiento no se debe ver influenciado más allá de lo tolerable. Para la ejecución de la validación se debe realizar un diseño de pruebas y análisis de respuesta bajo situaciones controladas que permita establecer la influencia de factores externos y la posibilidad de mitigar su efecto mediante la ejecución de acciones controladas por software.

Una práctica adecuada para lograr el proceso de validación es realizar procesos de medición y/o calibración a un elemento conocido y comparar los resultados obtenidos con el valor conocido previamente, lo cual, permite conocer el desempeño global del sistema. También es común simular las señales generadas por los diferentes instrumentos de medición y corroborar que los resultados obtenidos y la operación del sistema son los adecuados en diferentes puntos de operación.

4. SOFTWARE ASOCIADO A ANÁLISIS DE DATOS Y GENERACIÓN DE RESULTADOS

En este apartado se hace referencia a aquellas herramientas informáticas o productos software que permiten organizar y analizar los datos emitidos por otros software asociados a instrumentos de medida y plataformas de adquisición de datos, para luego generar la información resultante de todo el proceso. Independiente del tipo de software utilizado (comercial, modificable

o personalizado), la criticidad que conlleva la emisión de un mal resultado y la preocupación por ejecutar las actividades de manera eficiente y emitir resultados en un menor tiempo, han ido incrementando considerablemente. Por esta razón, actualmente han surgido iniciativas para proponer que la selección o desarrollo de un producto software, asociado a estas tareas, incluya, como mínimo, los procesos claves y esenciales del ciclo de vida del software: análisis de requerimientos, diseño, implementación, pruebas y operación/mantenimiento, o, en otras palabras, que incluya las mejoras prácticas de la Ingeniería del Software. Dichos procesos claves y esenciales son descritos a continuación, aunque vale la pena aclarar que estos no corresponden a todas las etapas del ciclo de vida del software que se siguen actualmente, ya que, los diferentes modelos de dicho ciclo han ido evolucionando según la naturaleza de los proyectos, los métodos y las herramientas a utilizar.

4.1 Especificación y Análisis de Requerimientos

Este proceso permite conocer el ámbito y flujo de la información en general; el objetivo principal en esta etapa es determinar la naturaleza de la herramienta informática o software, necesidades a cubrir, capacidades y funciones requeridas. Los requerimientos identificados se deben especificar de manera clara y precisa, sin lugar a ambigüedades, y cubriendo el alcance necesario, para así lograr los objetivos propuestos y obtener resultados válidos. Es necesario, además, evaluar y definir la criticidad y complejidad del software para así poder determinar su nivel de riesgo.

Para el caso de software tipo comercial, contar con una adecuada especificación de requerimientos permite realizar una correcta selección del software a adquirir. Para el caso de los tipos modificable y personalizado, la especificación de requerimientos constituirá la base para la definición del objetivo final y alcance del software.

La importancia de este proceso se centra en que la calidad del producto se basa en el cumplimiento de los requerimientos planteados y, por consiguiente, de las necesidades y expectativas de los implicados.

4.2 Diseño e Implementación

Este proceso constituye la definición de la arquitectura de la solución, acorde con los requerimientos planteados. Para aplicaciones usadas en metrología, como parte de este proceso, algunas de las actividades a realizar corresponden a:

- Documentación de los modelos matemáticos a utilizar. Ésta permite identificar los elementos necesarios para el desarrollo de cálculos u operaciones a implementar.
- Evaluar y definir la integridad y seguridad del software. Es necesario identificar claramente datos de entrada (ya sean ingresados manualmente o leídos de un archivo externo), información de salida, sistema de unidades, precisión en los cálculos. Teniendo en cuenta que el software puede ser alterado durante su uso, ya sea, accidental o intencionalmente, es importante prevenir esta situación mediante la utilización de técnicas que permitan mantener un control sobre el personal que opera el software, por ejemplo, bloqueo o inhabilitación de campos, validación de la información, etc.
- Definir estructura del software. El software puede contener gran cantidad de datos de entrada, resultados intermedios e información de salida, cada uno de estos grupos de datos debe ser mostrado, por lo cual, establecer regiones o utilizar colores facilita su identificación. Las interfaces se deben diseñar con el fin de que sean amigables y fáciles de usar. Por lo general, el software se complementa con una base de datos que permite almacenar y administrar los datos provenientes del proceso que se esté llevando a cabo, por lo cual, es necesario que su estructura, también, sea definida.
- Definir arquitectura hardware y software. Es necesario definir la plataforma de desarrollo del software, las posibles interacciones que tenga con otros software, qué tipo de equipo se necesita para que funcione adecuadamente, si corresponde a un software para estación de trabajo o para trabajo en red y las versiones para las cuales es compatible. Al no tener en cuenta estos aspectos es posible que se deba incurrir en costos adicionales por la adquisición de otros componentes (por ej.: equipos con mayor capacidad, software adicional) que el software requiera para funcionar correctamente y que no se tenían previstos.

Una vez definido el diseño se procede a su implementación, es decir, a traducirlo en el producto final esperado. Para este proceso y, particular-

PARTICULARMENTE, EN SOFTWARE PARA METROLOGÍA, PUEDE SER QUE UN DESARROLLADOR DE SOFTWARE NETO NO SEA SUFICIENTE, SE REQUIERE QUE EL EQUIPO DE DESARROLLO CUENTE CON LAS COMPETENCIAS TÉCNICAS QUE RESPALDEN EL CONOCIMIENTO NECESARIO PARA ENTENDER EL PROCESO Y ASÍ DEFINIR LAS OPCIONES ADECUADAS

mente, en software para metrología, puede ser que un desarrollador de software neto no sea suficiente, se requiere que el equipo de desarrollo cuente con las competencias técnicas que respalden el conocimiento necesario para entender el proceso y así definir las opciones adecuadas.

Para el caso en el que se haya adquirido un software comercial, es necesario que el proveedor suministre un completo manual en donde sea posible evidenciar las bases tomadas para el desarrollo de este proceso.

4.3 Validación y Pruebas

Este proceso permite determinar si el software desarrollado o adquirido satisface las necesidades de los implicados, por medio de una actividad de seguimiento de los requerimientos establecidos.

Incluye la definición de casos de prueba que permitan comprobar la correcta funcionalidad del software, tanto a nivel de su estructura interna (para el caso de productos desarrollados), como de su uso en un ambiente real de operación; también, para la implementación y resultados de los modelos matemáticos utilizados. Hasta el momento, al parecer, ningún software ha demostrado estar libre de errores, esto se puede deber a que es necesario realizar un número importante de pruebas que permitan cubrir el intervalo más amplio de posibilidades, y en algunos casos es inviable completar todas las pruebas, o imposible realizar todas las pruebas matemáticas, como es el caso específico de las aplicaciones para metrología.

Es posible considerar, en dos grupos, los aspectos mínimos a validar y probar, así:

- Evaluación de requerimientos: garantizar que el software adquirido o desarrollado corresponda a una solución acorde con las necesidades y expectativas definidas durante las fases iniciales del ciclo de vida del software.
- Evaluación técnica del software: comprobar correcto funcionamiento, lógica y modelos matemáticos implementados, parámetros para entrada de datos, conversiones de unidades, desarrollo de operaciones, resultados.

La aplicación de varias técnicas puede reducir el número de defectos en el software, pero entre más técnicas se apliquen más costoso se hace el desarrollo de éste. Es claro que el software a ser usado en ambientes críticos (que pueden llegar a afectar, por ejemplo, la vida o el medio ambien-

te) requerirá más esfuerzos que uno a ser usado en ambientes no críticos. Algunas de dichas técnicas son descritas en [1], [2] y [6].

Para el caso en el que se haya adquirido un software comercial, es necesario que el proveedor suministre las evidencias de las validaciones y pruebas realizadas en fábrica, y que se incluya dentro del proceso de compra la realización de pruebas en ambiente real de operación.

4.4 Operación y Mantenimiento

Comúnmente el software es entregado a los usuarios como una caja negra, sobre la cual es difícil determinar o identificar los procesos internos. Es por esto que un proceso ideal serial aquel en el cual los implicados tengan un acompañamiento completo, que incluya desde brindar orientaciones y recomendaciones para apoyar la actividad de especificación de requerimientos, mantener una constante comunicación que permita informar avances y recibir percepciones, realizar una adecuada formación de competencias en el personal que va a hacer uso del software, hasta brindar el soporte necesario en la etapa de adaptación. En este aspecto, la documentación que acompaña el software constituye un componente importante; se recomienda que esta incluya, como mínimo: una descripción detallada de la estructura del software, sus funcionalidades, instrucciones de operación, modelos matemáticos implementados, variables de entrada, variables de salida, procedimiento frente a defectos, recomendaciones generales y documentos de referencia.

Para las actividades de mantenimiento del software, es necesario que uno de los requerimientos especificados se haya centrado en la facilidad de realizar copias de seguridad de los datos. Las aplicaciones enfocadas en metrología manejan gran cantidad de datos y es necesario contar con su disponibilidad a la hora de requerir rastreabilidad de resultados. Las copias de seguridad pueden estar configuradas de manera automática dentro del software, o bien, se pueden configurar alarmas periódicas que adviertan, al personal que maneja el software, la necesidad de realizar una copia. Es importante contar, también, con copias de los medios originales de instalación del software (para efectos de reinstalación).

Por último, a la hora de adquirir un software se debe tener presente cuanto tiempo se espera que permanezca operativo pues, debido al constante cambio de la tecnología, la “vida útil” de un software puede estar alrededor de los cinco



Figura 2. Relación entre un Producto Software y el Ambiente en que será ejecutado.

años. Los cambios que surgen durante este periodo, en la mayoría de los casos, obligan a que se realicen mejoras, ajustes o, incluso, nuevos desarrollos.

5. CONCLUSIONES

La especificación, diseño, implementación y validación de software, en aplicaciones asociadas a metrología, presenta diferentes retos y particularidades que deben ser abordadas de forma adecuada con el fin de garantizar resultados confiables. En el presente artículo se realizó una breve presentación de los aspectos más relevantes asociados al proceso de consecución (ya sea a través de desarrollo o adquisición) de este tipo de software, desde la especificación hasta la validación del mismo. Se hizo un mayor énfasis en el software asociado a adquisición y análisis de datos, porque es allí donde se presentan las mayores particularidades con respecto a otro tipo de aplicaciones, como por ejemplo el control de procesos.

Dentro de los aspectos presentados, se destaca la necesidad de una clara especificación de requerimientos como base para la obtención de resultados satisfactorios en la adquisición o desarrollo de un software asociado a metrología. También se resalta la importancia de un proceso de validación robusto que permita garantizar tanto la operatividad como los resultados emitidos por el software, como imprescindible para garantizar la confiabilidad de las mediciones (objetivo primordial de la metrología a cualquier nivel de aplicación).

Complementando estos aspectos igualmente se requiere de adecuados procesos de documentación y formación a los metrologos que interactúan con dichos software, ya que es allí donde puede aparecer mayor cantidad de fallas y donde, en muchos casos, se presentan más debilidades tanto en desarrollos como en adquisiciones de software. Como conclusión final del tema

presentado, se puede afirmar que el no tener en cuenta los aspectos mencionados puede llevar a la generación de resultados erróneos o poco confiables, en conjunto con sobrecostos, reprocesos, y muchas otras dificultades indeseables en aplicaciones de metrología asociadas a procesos industriales, legales o científicos.

Por último se ratifica que en software para metrología, puede ser que un desarrollador de software neto no sea suficiente, y se requiere que el equipo de desarrollo cuente con las competencias técnicas que respalden el conocimiento necesario para entender el proceso y así definir las opciones adecuadas. Es necesario entonces, el trabajo sinérgico entre los Ingenieros de Software y los Ingenieros expertos en Metrología.

REFERENCIAS

- [1] EUROLAB, "Guidance for the Management of Computers and Software in Laboratories with Reference to ISO/IEC 17025:2005", Technical Report, 2006.
- [2] European Cooperation in Legal Metrology WELMEC, "Guía del Software (Directiva 2004/22/EC relativa a Instrumentos de Medida)", Guía 7.2, 2009.
- [3] M G Cox and P M Harris, "Guidelines To Help Users Select And Use Software For Their Metrology Applications", NPL Report, 2000.
- [4] N Greif and G Parkin, "An International Harmonised Measurement Software Guide: The Need And The Concept", XIX IMEKO World Congress, 2009.
- [5] Organización Internacional de Metrología Legal OIML, "Requisitos Generales para los Instrumentos de Medida controlados por Software", Documento 31, 2008.
- [6] R M Barker, P M Harris and G I Parkin, "Development and Testing of Spreadsheet Applications", NPL Report, 2006.
- [7] R S Pressman, Ingeniería del Software Un Enfoque Práctico, Sexta Edición Mc Graw Hill, 2006.
- [8] T J Eward, N J McCormick, K M Lawrence and M J Stevens, "Distributed Computing for Metrology Applications", NPL Report, 2007.