5 cosas que los ingenieros de test deberían saber sobre software

Por Elijah Kerry

NATIONAL INSTRUMENTS
www.ni.com/spain

Elijah Kerry es Product Manager de LabVIEW de National Instruments que se ocupa de las aplicaciones críticas de gran tamaño y de las prácticas de ingeniería de software. Es Ingeniero en Informática graduado por la Universidad de Missouri, Columbia.

Para obtener más información sobre estas y otras mejores prácticas para el desarrollo de grandes aplicaciones con LabVIEW, visite ni.com/largeapps.

Los modernos sistemas de test dependen de las soluciones basadas en software para satisfacer las necesidades de los dispositivos bajo test (DUTs) complejos y también para satisfacer los exigentes plazos de entrega. Como resultado, muchas de las mismas mejores prácticas y herramientas de desarrollo que son aspectos fundamentales de la ingeniería de software han llegado a ser igualmente importantes en el desarrollo de las aplicaciones de test.
En el desarrollo de software para un sistema de tests, hay que tener en cuenta las siguientes cinco prácticas básicas para estar seguro de que se está ofreciendo una aplicación fiable y de alta calidad en el tiempo acordado.

1. Si no está utilizando un control del código fuente, está jugando con fuego.

El control de código fuente es una herramienta fundamental para cualquier persona que desarrolle software - sin importar si se trabaja en un equipo de cientos de personas o de una sola. Sin él, las tareas simples, tales como el intercambio de código o la gestión de versiones diferentes pueden resultar difíciles y plantear riesgos que ocasionan demoras y conducen a la pérdida del trabajo. Los proveedores de software ofrecen numerosas soluciones que van desde Microsoft Team Foundation Server para Perforce a herramientas gratuitas de código abierto, tales como Subversion, cualquiera de las cuales se pueden utilizar con el código gráfico desarrollado con el software NI LabVIEW.

Una de las ventajas de tener un sistema de control del código fuente es que se puede realizar el seguimiento, gestionar y revisar los cambios de la aplicación a lo largo del tiempo a través de una combinación de operaciones de diferencias y fusiones. Gracias al Sistema de Desarrollo Profesional de LabVIEW, se pueden integrar las operaciones de diferenciación y fusión gráficas con los clientes de control del código fuente. Una vez configurada, una operación de diferenciación o fusión en el control del código fuente hace que se muestre automáticamente un cuadro de diálogo en LabVIEW que sirve de guía a través de los cambios e identifica cuando y que se ha modificado.

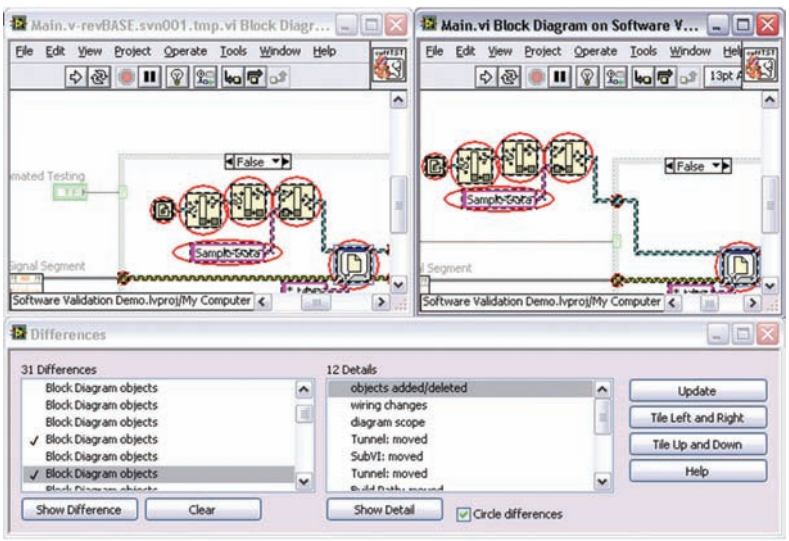


Figura 1. La diferenciación gráfica se puede invocar automáticamente desde los clientes de control del código fuente utilizando la interfaz de línea de comandos que se proporciona con el Sistema de Desarrollo Profesional de LabVIEW.

2. Desarrollar sin requisitos es solo crear prototipos.

Los prototipos son una parte importante del proceso de desarrollo, ya que normalmente, se utilizan para demostrar una idea simple o para probar la viabilidad de un concepto para una nueva tecnología. Sin embargo, al software que se utiliza en los prototipos se le dota de poca planificación o poco cuidado en la arquitectura, por lo que es inadecuado para una aplicación final. Es importante distinguir entre la de creación de prototipos y las fases de desarrollo del ciclo de vida del software - un indicador clave es si existen requisitos para el software que incluyan las especificaciones generales de la arquitectura y un plan de test.

Los documentos de los requisitos constituyen una forma efectiva de alinear las expectativas de los clientes con los desarrolladores, coordinar los equipos grandes, documentar el estado de un proyecto y garantizar que el código se test a fondo. Entre las herramientas comunes de almacenamiento y gestión de estos documentos se incluyen Microsoft Word, Microsoft Excel, Adobe Acrobat, Telelogic DOORS y RequisitePro. NI proporciona la capacidad de automatizar la integración entre estas interfaces y los productos de software cómo NI LabVIEW y TestStand de NI para realizar un seguimiento automático de la cobertura de los requisitos y de la generación de informes para el análisis de la trazabilidad y de la cobertura previa.

3. Es posible medir la capacidad de test y la calidad del código.

El análisis estático del código se refiere a cualquier herramienta o método que tiene criterios preestablecidos mediante los cuales se puede comparar el código fuente para ver si cumple con las normas de esti-

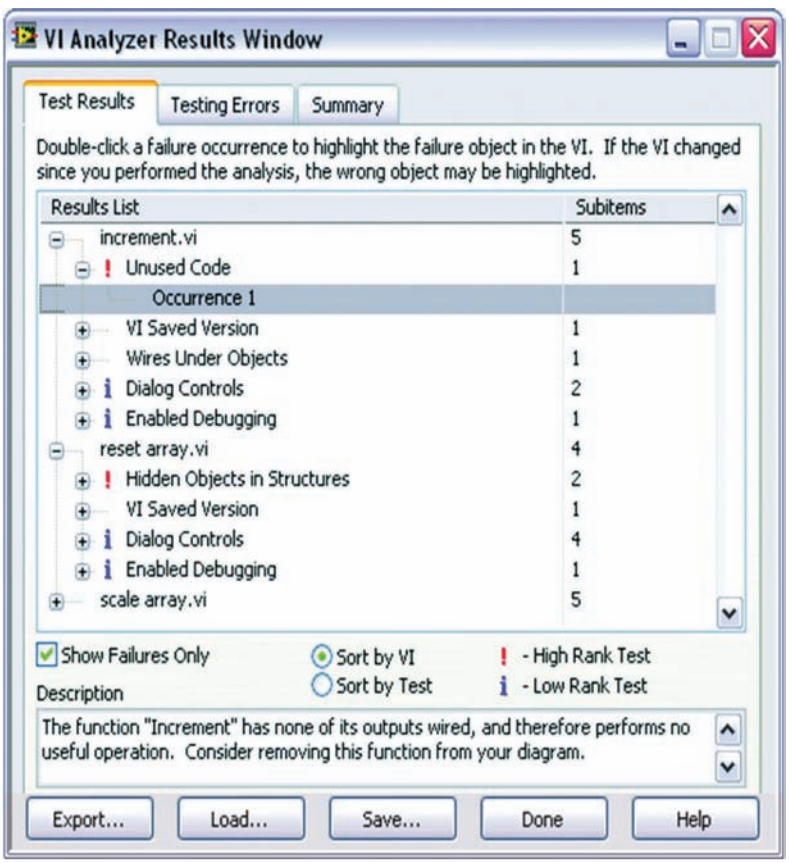
lo, organización y técnicas. Además, el análisis de código estático puede ayudar a demostrar que el código está mal escrito e identificar las áreas problemáticas. También se pueden utilizar métricas de la complejidad de código, tales como la modularidad y la complejidad ciclomática (Cyclomatic Complexity) para determinar el tamaño y la capacidad de test de un proyecto. Esto es útil cuando el código es heredado y se requiere la corrección de errores o la agregación de funciones.

Para realizar el seguimiento del progreso y encontrar los problemas en etapas tempranas, se combina la presentación de informes regulares de las métricas del análisis del código con frecuentes revisiones por parte de otros homólogos Se puede automatizar el análisis del código estático del código de LabVIEW con el Módulo LabVIEW VI Analyzer Toolkit, que ofrece la posibilidad de personalizar más de 80 tests, incluyendo el análisis de los resultados, la complejidad, la documentación e incluso la corrección ortográfica. Hay también disponible un asistente para la creación de nuevas tests con LabVIEW VI Scripting.

4. Puede pensar que su código funciona, pero hay que demostrarlo.

Se sabe que algo está roto cuando no funciona. Pero es aún más difícil probar a otra persona o a una autoridad externa que algo funciona correctamente. La audiencia para este tipo de demostración varía, pero podría ser un cliente, un grupo de control de calidad o incluso una agencia de regulación gubernamental.

El test y la depuración de un software es una parte inseparable del proceso de desarrollo, pero se pueden utilizar herramientas automatizadas, como LabVIEW Unit Test Framework Toolkit, para hacer frente al reto de los tests de software complejos. La automatización de este proceso reduce la cantidad de tiempo que se invierte en la realización de los tests y hace que estas sean lo más exhaustivas posibles. Esto no sólo ayuda a asegurar que el software que se produce sea de la más alta calidad posible, sino que también se ahorra dinero al descubrir problemas en etapas tempranas y al reducir el tiempo de test.



La validación y test funcional del código es una parte fundamental del proceso de ingeniería de software y una práctica estándar para cualquiera que tenga que probar que el código funciona. Demostrar que un software funciona es más complejo que demostrar que la aplicación se ejecuta, puesto que se necesita la validación del funcionamiento correcto. Esto requiere documentación y resultados de tests que demuestren que la aplicación se comporta de la forma en que fue diseñada.

5. La reutilización no es un mito, pero requiere planificación.

La creciente complejidad de los sistemas de tests está convergiendo con ciclos de lanzamiento más breves para muchos dispositivos bajo test (DUTs), lo cual ha provocado una gran necesidad de librerías de reutilización. Reutilizar significa que el hardware y el software puedan abarcar varios sistemas de test y ser fácilmente adaptados para las iteraciones en un nuevo DUT. También significa que equipos de desarrollo independientes puedan utilizar controladores y API ya existentes para maximizar la eficiencia y reducir aún más la fase de programación del ciclo de vida. Sin embargo, muchos programadores tienen a menudo dificultades para incorporar las prácticas de reutilización del código con éxito - generalmente debido a una mala planificación y a la incapacidad para adaptarse a las nuevas necesidades y presentar fácilmente estos cambios a un gran número de programadores y aplicaciones.

Un ejemplo ampliamente utilizado de librerías de reutilización es la colección de más de 8.000 controladores de instrumentos de NI disponible en ni.com/idnet. El éxito de estas librerías reutilizables de instrumentos se basa en las APIs claramente definidas para la comunicación de instrumentos y la encapsulación de la funcionalidad de bajo nivel en las librerías privadas. Sin embargo, el VI Package Manager ofrece una solución más sofisticada para la difusión de librerías de reutilización en toda la empresa y la gestión de las versiones que se utilizan en diferentes proyectos.

Figura 2. VI Analyzer proporciona una herramienta de diálogo y presentación de informes interactiva para examinar la calidad y la capacidad de test de VIs.