

# La Importancia del Software para Administración de Pruebas

la mayoría de las organizaciones que realizan pruebas automatizadas desde hace tiempo, probablemente esté viendo cómo aumenta la cantidad de lenguajes. Gracias a las formas de abstracción más especializadas de los lenguajes de programación actuales de más alto nivel, este problema no va a desaparecer pronto.

La evolución está arraigada profundamente en la historia de los lenguajes de programación y el afán por alcanzar niveles más altos de abstracción, y comienza con el primer lenguaje de programación de alto nivel, FORTRAN. Desarrollado en 1953 por John Backus, FORTRAN abordó la necesidad de un mayor nivel de abstracción para los procesos de máquinas construidos en torno a la forma en que los seres humanos comunican de forma natural sus ideas: a través del lenguaje.

“Atacamos los desafíos exclusivos de cada prueba con el mejor lenguaje para cada tarea en cuestión. Mediante el uso de software comercial estándar de administración de pruebas para que actúe como un gran unificador durante todo el ciclo de vida del producto, aumentamos considerablemente nuestra productividad de ingeniería y minimizamos el tiempo de comercialización”.

— Simon Wiedemer, Jefe de Arquitectura de Pruebas Automatizadas, Festo AG & Co. KG

Tras el éxito de FORTRAN, se desarrollaron otros lenguajes como C, Pascal, ATLAS y PAWS. Y con cada nuevo lenguaje llegaron niveles más potentes de abstracción, como la programación orientada a objetos. Algunos de estos nuevos modelos de cómputo se desarrollaron para tareas de programación de propósito general, pero algunos se crearon para una aplicación en particular. Por ejemplo, LabVIEW se desarrolló para aplicaciones de pruebas, medición y control, y Python para tareas de scripting rápido.

Estos niveles cada vez mayores de abstracción dan como resultado lenguajes más adecuados para tareas específicas. En la actualidad, los mejores administradores de pruebas diseñan sistemas de prueba que aprovechan la potencia de varios lenguajes y ahorran tiempo de desarrollo mediante el uso de software de administración de pruebas.

### Desarrollo Tradicional de los Sistemas de Prueba

Como el software es la columna vertebral de la automatización cuando se construye un sistema de prueba, muchas organizaciones prefieren estandarizarse con un solo lenguaje general, que se utiliza para todos los aspectos del diseño del sistema de pruebas, desde los ensayos de componentes individuales hasta la administración de pruebas generalizadas. El resultado final es el desarrollo de un enfoque homogéneo de software de pruebas. La principal ventaja es que todos los miembros de un equipo pueden trabajar en un mismo entorno estandarizado, lo que permite el intercambio fácil de bibliotecas y módulos de código dentro de todo el equipo. La capacitación para este enfoque también se simplifica en gran medida, porque el equipo aprende y trabaja en un único entorno.

Sin embargo, la estandarización con un único lenguaje presenta algunos inconvenientes. El uso de un solo lenguaje puede limitar a los recién contratados a un determinado conjunto de habilidades o forzar a los empleados recién contratados a aprender herramientas nuevas. Este tema del desarrollo de un lugar de trabajo calificado se exploró en las Perspectivas de Pruebas Automatizadas 2014. Una vez que los estudiantes se gradúan, suelen tener preferencia por uno o más lenguajes específicos y experiencia en algunos

de ellos. Además, cuando se incorporan nuevos gerentes, normalmente optan por implementar un lenguaje de su elección, lo que ocasiona problemas de adaptación a nivel organizacional. Esto puede ser un ejercicio costoso que con frecuencia requiere migración de código, revalidación de la base de código y capacitación adicional en el nuevo lenguaje.

Los mejores gerentes de pruebas deben buscar un nuevo enfoque para el desarrollo del sistema de prueba que genere un sistema heterogéneo a partir de varios lenguajes. Este tipo de enfoque permite que un equipo utilice varios lenguajes, para construir sistemas de pruebas más potentes. Por ejemplo, Python podría utilizarse para las pruebas de validación y verificación (V&V) de scripting basadas en código desarrollado por los ingenieros de investigación y desarrollo (I+D). En el mismo sistema, C# podría utilizarse para desarrollar una interfaz orientada a objetos para el hardware personalizado o las bibliotecas .NET existentes mientras que LabVIEW se comunica con el hardware y recopila datos. Como todos los lenguajes están diseñados para trabajar con aplicaciones específicas, utilizar cada uno para aprovechar sus puntos fuertes debería, en última instancia, ahorrar tiempo y dinero.

Aunque este enfoque es beneficioso, puede representar un nuevo desafío para el desarrollo de los sistemas de prueba: ahora es necesario que diferentes lenguajes trabajen y se comuniquen entre sí para formar un solo sistema. Para resolver esto, los ingenieros de pruebas tienen que comprender no sólo el entorno en el que se especializan sino también los demás, para interactuar adecuadamente con ellos.

### La Solución de Software

Los departamentos de pruebas están recurriendo a software comercial estándar de administración de

pruebas para actuar como una especie de piedra de Rosetta mediadora entre los diferentes lenguajes. Este software no sólo ofrece a los usuarios un entorno común en el que pueden trabajar con cualquier tipo de código de prueba, sino que también realiza tareas ejecutivas, como la secuenciación y la llamada de cada prueba, el manejo del registro de datos y la generación de reportes. Cada ingeniero puede centrarse luego en escribir la mejor prueba para cada componente del DUT sin tener que preocuparse acerca de cómo comunicarse con las otras partes del código. Y puesto que los ingenieros pueden utilizar los entornos que les resultan más cómodos, una organización se puede centrar en la contratación de ingenieros con habilidades exclusivas para sus aplicaciones,

Además, los gerentes de pruebas pueden aprovechar toda la potencia de un diseño heterogéneo al tiempo que se evitan los nuevos desafíos que introduce tal diseño. Esto incluye el uso de software de administración de pruebas para contar con un proceso de desarrollo más modular, que produce un sistema más fácil de mantener y actualizar porque cada componente se puede actualizar de forma individual sin afectar el resto del sistema de prueba.

En última instancia, los administradores de pruebas suelen incluir el respaldo de sus proveedores comerciales que aplican parches y actualizan continuamente el software de prueba, lo que compensa aún más el costo de mantenimiento y aumenta la sustentabilidad de estos sistemas. Gracias a estas ventajas, combinadas con los beneficios de un diseño heterogéneo del sistema de prueba, los mejores gerentes de pruebas están construyendo el futuro de las pruebas automatizadas.

### ENFOQUES HETEROGÉNEOS DE DESARROLLO DE PRUEBAS

