*Serie Científica de la Universidad de las Ciencias Informáticas*
Vol. 13, No. 1, Mes Enero, 2020
ISSN: 2306-2495 | RNPS: 2343
Pág. 104-112
http://publicaciones.uci.cu

Tipo de artículo: Artículo de revisión
Temática: Ingeniería y gestión de software
Recibido: 20/09/19 | Aceptado: 04/01/2020 | Publicado: 06/01/2020

# Elementos para selección de Herramientas para la automatización de Pruebas no funcionales

# *Elements for the selection of tools for the automation of non-functional tests*

**Noel H. Hidalgo Reyes [1*], Dariel Costa Báez [2], Aymara Marin Diaz[3], Yaimí Trujillo Casañola [4]**

[1*] Dirección de Calidad de Software. Universidad de Ciencias Informáticas, Carretera a San Antonio de los Baños, Km 2 ½, Torrens, Boyeros. La Habana. Cuba. nhhidalgo@estudiantes.uci.cu
[2] Dirección de Calidad de Software. Universidad de Ciencias Informáticas, Carretera a San Antonio de los Baños, Km 2 ½, Torrens, Boyeros. La Habana. Cuba. dcosta@uci.cu
[3] Dirección de Calidad de Software. Universidad de Ciencias Informáticas, Carretera a San Antonio de los Baños, Km 2 ½, Torrens, Boyeros. La Habana. Cuba. amarin@uci.cu
[4] Dirección de Calidad de Software. Universidad de Ciencias Informáticas, Carretera a San Antonio de los Baños, Km 2 ½, Torrens, Boyeros. La Habana. Cuba. yaimi@uci.cu

[*] Autor para correspondencia: nhhidalgo@estudiantes.uci.cu

**Resumen**

El desarrollo de software debe estar enfocado en la imperiosa realidad de garantizar calidad tanto en todo su proceso como en sus resultados (producto), para lograr a través de éste, las ventajas competitivas en un entorno globalizado. Estar consciente de esta realidad en las instituciones desarrolladoras de software, se traducirá en dar formalidad a la disciplina de pruebas como apoyo a este proceso. Sin embargo, esta formalidad esperada en la disciplina y el uso de herramientas automatizadas es escasa en muchas instituciones en el campo. El presente trabajo se enfoca en el estudio de las pruebas automatizadas, algunos aspectos para el proceso de selección de las herramientas para dichas pruebas, siendo más específicos en las pruebas no funcionales, dado que el enfoque en esta área se ve positivamente alterado, pues los tiempos de los procesos de control de calidad serán significativamente más cortos y los resultados más precisos, por ende, se podrán invertir mejor sus recursos y fortalecer actividades como análisis de resultados, mejoramiento de procesos, productos y servicios.

*Serie Científica de la Universidad de las Ciencias Informáticas*
Vol. 13, No. 1, Mes Enero, 2020
ISSN: 2306-2495 | RNPS: 2343
Pág. 104-112
http://publicaciones.uci.cu

**Palabras clave:** calidad, herramientas automatizadas, pruebas no funcionales, software.

***Abstract***

The development of software must be focused on the imperative reality of guaranteeing quality both in its entire process and in its results (product), in order to achieve through this, the competitive advantages in a globalized environment. Being aware of this reality in the software development institutions, will result in giving formality to the testing discipline in support of this process. However, this formality expected in the discipline and the use of automated tools is scarce in many institutions in the field. The present work focuses on the study of automated tests, some aspects for the selection process of the tools for these tests, being more specific in the non-functional tests, given that the focus in this area is positively altered, since times of quality control processes will be significantly shorter and the results more precise, therefore, you can better invest your resources and strengthen activities such as results analysis, process improvement, products and services.

***Keywords:*** quality, automated tools, non-functional tests, software.

## Introducción

La tecnología alrededor del mundo avanza cada vez más rápido, por lo que se hace preciso reinventarse y contar con el manejo de procesos y herramientas que permitan estar a la vanguardia. El desarrollo de software en instituciones especializadas para tal fin debe estar enfocado en la imperiosa realidad de garantizar calidad tanto en todo su proceso como en sus resultados (producto) para así traducirse en una estrategia primordial para lograr a través de éste, las ventajas competitivas en un entorno altamente globalizado. La exigencia e interés creciente por parte del mercado de productos software de alta calidad, es un indicador de la percepción de esta como un elemento imprescindible para su comercialización. Propiciar calidad al software es una necesidad prioritaria para las organizaciones que lo desarrollan, ya sea para uso interno o para implementaciones externas en clientes.

Las actividades relacionadas con el aseguramiento de la calidad juegan un papel importante en la industria de software debido a su participación activa a través de todo el ciclo productivo. El control de la calidad de los productos o servicios desde etapas tempranas de los proyectos, permite minimizar los costos asociados con esfuerzo y tiempo, y que el producto resultante sea más confiable, lo que implica para el negocio un aumento de la calidad y satisfacción de los clientes.

*Serie Científica de la Universidad de las Ciencias Informáticas*
Vol. 13, No. 1, Mes Enero, 2020
ISSN: 2306-2495 | RNPS: 2343
Pág. 104-112
http://publicaciones.uci.cu

Las instituciones desarrolladoras de software, para garantizar calidad en sus productos deben realizar pruebas del software construido. Esto se logra a través de hacer énfasis en la disciplina de Pruebas dentro del proceso de desarrollo de software, práctica que generalmente se traduce en invertir entre el 30% y 50% del costo total del software en pruebas. Bajo esta premisa, las pruebas tienen una gran importancia y representan un gran volumen de actividades ya que se sugiere que en un proyecto de desarrollo de software al menos el 75% de las pruebas sean automatizadas y el resto manual; esto evidencia que la estrategia de automatizar pruebas es una decisión primordial. La automatización de las pruebas se logra a través del uso de herramientas de pruebas automatizadas (Sommerville, 2007).

En cuanto a las pruebas no funcionales, algunos de los tipos que se manejan en la fase de aseguramiento de calidad, como las de rendimiento, seguridad, carga, estrés y volumen, son difíciles y algunas veces imposible de simular manualmente, y en el caso que se hagan de esta manera, los resultados obtenidos no son lo suficientemente confiables.

La automatización de estas pruebas permite replicar errores sin cambiar las condiciones y almacena un registro exacto de los tiempos de respuesta, transacciones por segundo, código defectuoso, componentes de aplicación no encontrados, entre otros (Kumar, 2008). Adicionalmente, minimiza la cantidad de recursos necesarios para la ejecución de pruebas, los tiempos de reproceso y de ejecución de tal forma que los cronogramas de los proyectos no se ven afectados y se obtiene un producto con mayor calidad.

## Materiales y métodos

### Pruebas de Software

La fase de pruebas es una tarea que consume muchos recursos. Para ejecutar esta fase en la práctica las organizaciones que desarrollan software asignan un grupo de evaluadores, los cuales si realizan este proceso de forma manual pueden tardar varios meses e incluso años. Este enfoque consume mucho tiempo y conlleva altos costos. Para lograr una cobertura adecuada en la evaluación del sistema en cuestión, es necesario: seleccionar los datos de prueba, las variables del entorno de evaluación, determinar el número de pruebas y el tiempo asignado para este proceso. En la ejecución de la cobertura de evaluación, algunos autores desarrollan modelos de predecibilidad apoyados de métricas de software (Basili, 1996). Para optimizar los recursos que son empleados en la ejecución de miles o millones de pruebas es adecuado utilizar instrumentos de evaluación (Ragab, 2010). Se han realizado algunas propuestas para evaluar sistemas en Internet, entre estas propuestas esta (Davila-Nicanor, 2005), en donde se

*Serie Científica de la Universidad de las Ciencias Informáticas*
Vol. 13, No. 1, Mes Enero, 2020
ISSN: 2306-2495 | RNPS: 2343
Pág. 104-112
http://publicaciones.uci.cu

desarrolló una herramienta la cual automatiza la ejecución de las pruebas, reduciendo el tiempo proyectado para la ejecución de 5000 pruebas de 4 años a tan solo 6 horas.

En la actualidad la automatización de pruebas generalmente no se enfoca en la construcción de herramientas de automatización, sino en sacar el mayor provecho a las herramientas existentes y generar estrategias con base en ellas.

La fase de pruebas del ciclo de vida del Software es definida como" un proceso o una serie de procesos diseñados para asegurar que el código cumple para lo que está diseñado y no haga nada de forma involuntario" (Myers y Sandler, 2004). Para lograr realizar la verificación del código contra su diseño (y, por tanto, contra los requisitos del cliente), se toman datos de ejemplo de entrada, y se plantean salidas esperadas de acuerdo a los requerimientos. Para realizar una fase de pruebas correctamente, se recomienda responder las siguientes preguntas, antes inclusive de contratar al equipo de pruebas (Black, 2009):

¿Que podría probar?

¿Qué debería probar?

¿Qué puedo probar?

En lo que debería probar, debería priorizarse aquellas áreas en las cuales no se tiene cobertura de pruebas. Por tanto, al ejecutar la fase de Pruebas, es mejor evitar la creencia popular de evaluar distintos tipos de entrada a las cuales se generarán distintos tipos de salida, alcanzando una permutación entre entradas lo suficientemente completa para garantizar la calidad de las salidas del Software. Crear casos de prueba para todas las posibles entradas y salidas es impráctico, y requeriría de muchos recursos humanos para ser económicamente factible (Myers y Sandler, 2004).

Un enfoque más racional estaría enfocado en probar el software con el fin de buscar errores durante su ejecución. La anterior afirmación podría ir en contravía con la creencia común de que las pruebas se ejecutan para verificar que un programa cumple con los requisitos declarados por el cliente, lo cual en parte es cierto, pero para que el proceso de pruebas agregue valor al proceso es importante que demuestre que el software bajo prueba (o Software Under Test SUT) posee cualidades de confiabilidad, y para lograr demostrar ello es necesario detectar y remover errores (Myers y Sandler, 2004**).**

En cuanto a lo que se debería probar, el enfoque está en identificar aquellos errores o bugs que afecten la experiencia del cliente respecto a la calidad del producto. Los enfoques de pruebas deberían enfocarse a encontrar los defectos críticos que limitan la habilidad de los usuarios de llevar a cabo su trabajo usando el software desarrollado (Black, 2009). Respecto a lo que se puede probar, es revisar el recurso humano, financiero y de tiempo con el que se cuenta para realizar pruebas, y verificar que tanto de lo que "se debería probar" se puede realizar con los recursos disponibles.

*Serie Científica de la Universidad de las Ciencias Informáticas*
Vol. 13, No. 1, Mes Enero, 2020
ISSN: 2306-2495 | RNPS: 2343
Pág. 104-112
http://publicaciones.uci.cu

# Resultados y discusiones

## Pruebas de Software Automatizadas

La automatización de pruebas puede ser definida como el uso de software para controlar la ejecución de pruebas, la comparación de las salidas actuales con salidas predecidas, la configuración de las precondiciones, y otras funciones de control y reportes (Amaricai y Constantinescu, 2014). De ahí que las pruebas automatizadas generan como resultado scripts que a su vez generan reportes sobre la efectividad de las pruebas ejecutadas sobre el Software Bajo Prueba.

El uso de la automatización de pruebas ayuda en gran medida al personal de Calidad. Lo anterior dado que por lo general las organizaciones asumen que cuando un software pasa a la fase de pruebas, en dicha fase serán detectados todos los bugs posibles, lo cual en la práctica es imposible, dado la ausencia de los recursos necesarios para hacerlo. Por tanto, con el uso de pruebas automatizadas los probadores podrán dedicar más tiempo a cubrir otros aspectos del sistema, y delegar todo el tema de seguridad, carga, estrés, volumen y evaluación a las herramientas de prueba automatizadas. De igual manera, el uso de herramientas de prueba automatizadas mejora la calidad de las pruebas, originado por la mínima intervención humana que requieren durante su ejecución.

**¿Por qué automatizar?**

Al hablar de pruebas automáticas nos enfocamos en cómo mejorar nuestra forma de trabajar, no sólo se trata de realizar pruebas no funcionales, sino también funcionales, de tal forma que nuestros procesos dependan cada vez menos de las personas, lo que permite enfocar el recurso especializado más en proyectos estratégicos y menos en procesos operativos.

La automatización de las pruebas no funcionales reduce significativamente el esfuerzo dedicado a la detección de errores en productos que se encuentran en continuo mantenimiento. La automatización de las pruebas debe ser considerada un proyecto en sí mismo con objetivos definidos.

**¿Qué se debe automatizar?**

Las principales etapas que comprenden el proceso de pruebas de software son: Planeación y Control, Análisis y Diseño, Implementación y ejecución, Evaluación de los criterios de finalización y realizar reportes y Realizar actividades de cierre (Thomas Müller (chair) / Debra Friedenberg, 2011). La definición de qué procesos se automatizarán en la fase de ejecución de pruebas de un proyecto se debe realizar de forma temprana, especialmente en la etapa de “Análisis y Diseño” y basados en la arquitectura de la aplicación. Para definir esto, se deben tener en cuenta, entre otros, los siguientes criterios (Elfriede Dustin, 1999): procesos o funcionalidades que sean repetitivos, procesos que requieran pruebas con distintos datos, procesos con alto consumo de recursos, funcionalidades

*Serie Científica de la Universidad de las Ciencias Informáticas*
Vol. 13, No. 1, Mes Enero, 2020
ISSN: 2306-2495 | RNPS: 2343
Pág. 104-112
http://publicaciones.uci.cu

complejas, aplicaciones que deben correr en múltiples sistemas operativos, motores de bases de datos o navegadores y procesos críticos dentro de una aplicación.

## Automatización de Pruebas no Funcionales

Debido a la necesidad de obtener resultados con mayor exactitud de procesos de pruebas no funcionales como las de rendimiento, seguridad, carga, estrés, volumen y otras, que son difíciles de implementar y algunas veces imposibles de simular manualmente, se hace imprescindible la utilización de herramientas automatizadas. Muchos expertos y por experiencia en el campo, se ha comprobado que aún ejecutándolas manualmente, los resultados obtenidos no son suficientemente confiables, adicional a esto, si no hay automatización de pruebas, se torna complejo detectar si hay fallas de tipo no funcional dentro de una aplicación, es muy posible entonces que: si estas fallas existen, sean detectadas en producción, lo cual genera altos costos, no sólo por la corrección del error en una etapa muy avanzada de los proyectos, sino también por la mala imagen generada ante los clientes y posterior a esto, su insatisfacción.

## Mejores Prácticas en la Automatización de Pruebas

Con el fin de garantizar el éxito en los proyectos de automatización de pruebas, el ISTQB (International Software Testing Qualifications Board) ha propuesto una serie de buenas prácticas en el marco de la certificación en Automatización de Pruebas. Dichas prácticas son enfocadas como criterios de éxito, además de una lista de lo que se debe y no se debe hacer en este tipo de proyectos. Como buenas prácticas se propone (International Software Testing Qualifications Board ISTQB, 2016)**:**

1. Tener establecida una Arquitectura de Automatización de Pruebas (TAA - Test Automation Architecture), teniendo en cuenta que la automatización de pruebas es un proyecto de Software, y debe ser tratado como tal.

2. El Software Bajo Prueba (SUT *en sus siglas en inglés*) debe ser desarrollado para soportar pruebas automáticas, en dónde el SUT debe desacoplar la interacción con la interfaz gráfica de los datos y su apariencia visual. Esto se puede traducir en la creación de mecanismos para identificar de manera única cada elemento visual presente en el Software Bajo Prueba.

3. Identificar las partes del Software Bajo Prueba que son sensibles de ser probadas, e iniciar el proyecto por dichas partes. Realizar esta acción permitirá mostrar resultados rápidos en la ejecución del proyecto de automatización de pruebas, mostrando la facilidad de implementación de los scripts de prueba.

4. Establecer una estrategia de Automatización de Pruebas. Según el Instituto Internacional para Pruebas de Software (International Software Test Institute), una estrategia de pruebas debería:

- Las secciones del software que requieren auntomatización de pruebas.
- Cómo la tarea se llevará a cabo.

*Serie Científica de la Universidad de las Ciencias Informáticas*
Vol. 13, No. 1, Mes Enero, 2020
ISSN: 2306-2495 | RNPS: 2343
Pág. 104-112
http://publicaciones.uci.cu

- Cómo y dónde los scripts de pruebas tendrán mantenimiento.
- Cómo el proyecto se beneficiará de esta actividad.
- Cuanto será el ahorro en costos.

5. Seleccionar o Crear un Framework de Automatización de Pruebas. Al respecto de este punto, ISTQB recomienda que el Framework que se escoja en conjunto con la arqutectura de automatización de pruebas debe cumplir con los siguientes requisitos:

- Tener facilidad para generar reportes que muestren la calidad del Software Bajo Prueba.
- Permitir identificar la causa de la falla en la ejecución de las pruebas, ya sea que se trate del Software Bajo Prueba o de fallas en los scripts de pruebas.
- Tener definido claramente el entorno de pruebas requerido para la correcta ejecución del Framework.
- Facilitar la documentación de los casos de pruebas.
- Facilitar la trazabilidad de los pasos de los casos de pruebas.
- Permitir un fácil mantenimiento de los scripts.

**Aspectos para la selección de Herramientas para la Automatización de Pruebas no funcionales**

Existen múltiples herramientas para automatizar pruebas no funcionales por lo que es engorroso el trabajo de selección de estas para su uso, se debe investigar y llegar a categorizarlas en cuanto a sus principales características y variables como el tipo de licenciamiento, tipo de aplicación, tipos de pruebas soportadas y restricciones, asimismo se debe realizar un mapeo respecto a las necesidades de la institución, basadas puntualmente en la optimización de su proceso de pruebas.

Se debe definir el propósito para lo cual se requiere y el alcance que las mismas tendrán en el proceso de desarrollo de software. Se debe tener en cuenta además otros factores como por ejemplo: elegir un proyecto adecuado y utilizarlo como proyecto piloto para comenzar; también se puede evaluar a través de un cuestionario algunos aspectos subjetivos como son la motivación del equipo de pruebas, es necesario que se vean las pruebas automatizadas como una ayuda en su trabajo y no como una carga extra, es imprescindible contar con gente dispuesta a cambiar la forma de trabajo; y capacitar al personal que va interactuar con este tipo de pruebas.

Un punto importante a tener en cuenta para las pruebas no funcionales, es el monitoreo de los recursos físicos durante la ejecución de las pruebas, por ejemplo: el uso del procesador, memoria, disco duro y la red permiten evaluar las capacidades de infraestructura que se tienen de acuerdo a las aplicaciones y definir si se cumple o no con el requisito no funcional especificado.

*Serie Científica de la Universidad de las Ciencias Informáticas*
Vol. 13, No. 1, Mes Enero, 2020
ISSN: 2306-2495 | RNPS: 2343
Pág. 104-112
http://publicaciones.uci.cu

## Conclusiones

Existen programas de capacitación sobre pruebas automáticas y herramientas de automatización, que tienden a ser altamente costosos y de difícil acceso, por lo tanto, si se requiere involucrar la automatización de pruebas en los procesos de control de calidad de software de forma autónoma y empírica, es importante tener en cuenta los siguientes aspectos

- Idealmente se deben tener claros los objetivos del negocio y el alcance del proyecto, antes de iniciar una investigación herramientas para implementar y ejecutar las pruebas automáticas.
- Es fundamental contar con el aval y el apoyo de la alta gerencia al iniciar un proyecto de mejoramiento de esta envergadura.
- Se debe definir un equipo de investigación o automatización y disponer de proyectos pequeños que sirvan como piloto para probar las herramientas, metodologías, artefactos, entre otros y generar conclusiones.
- Se debe tener un acompañamiento constante del área de arquitectura y desarrollo y contar con el apoyo adecuado para realizar la correcta configuración de los ambientes requeridos para las pruebas.
- Es importante involucrar la automatización de pruebas no funcionales en el mayor número posible de proyectos en una institución, con el objetivo de crear una cultura de calidad y optimizar recursos.
- Para el análisis de los datos es necesario contar con bases estadísticas y conocimiento sobre contadores de rendimiento.
- Se debe ajustar el proceso de aseguramiento de calidad de software e incluir detalladamente la fase de pruebas automáticas.

## Referencias

**[1]**Amaricai, S. & Constantinescu, R. (2014). Designing a software test automation framework. Informatica Economica, 18(1), 152-161.

**[2]**Basili, V. R. (1996). A Validation of Object-Oriented Design Metrics As Quality Indicators. IEEE Trans. Softw. Eng., 22(10), 751--761. doi:10.1109/32.544352

**[3]**Black, R. (2009). Managing the testing process: practical tools and techniques for managing hardware and software testing (3rd). Wiley Publishing.

**[4]**Davila-Nicanor, L. a.-A. (2005). Reliability evaluation of Web-based software applications. In Computer Science, 2005. ENC 2005. Sixth Mexican International Conference on (pp. 106-112). doi:10.1109/ENC.2005.36

**[5]** Elfriede Dustin, J.R, John Paul. (1999). Automated software testing.

**[6]** International Software Testing Qualifications Board - ISTQB. (2016, enero). Advanced level specialist syllabus in test automation – engineering. Version draft beta, 13 Jan 2016.

**[7]**Kumar, N. S. (2008). Software Testing with Visual Studio Team System 2008.

**[8]**Myers, G. J. & Sandler, C. (2004). The art of software testing. John Wiley & Sons

**[9]**Ragab, S. a. (2010). Object oriented design metrics and tools a survey. In Informatics and Systems (INFOS), 2010 The 7th International Conference on (pp. 1-7).

**[10]**Sommerville, I. (2007). Software Engineering. Pearson Education.

**[11]** Thomas Müller (chair) / Debra Friedenberg, a. t. I. W. F. L. (2011). Certified Tester Foundation Level Syllabus (Vol. Versión 2011, pp. 78): International Software Testing Qualifications Board