

Nivel de madurez y grado de satisfacción de las herramientas libres para pruebas funcionales

Maturity and satisfaction of free tools for functional tests

Edgar Serna M.¹, Wilson Andrés Bedoya O.², Katherine López M.³

¹Facultad de Ingenierías, Instituto Tecnológico Metropolitano Medellín, Colombia
edgarserna@itm.edu.co

²TATA Consultancy Services Medellín, Colombia
w.bedoyaoquendo@tcs.com

³EDATEL Telecomunicaciones Medellín, Colombia
klopezm@edatel.com.co

Resumen — Este trabajo tiene como finalidad proporcionar información acerca de algunas herramientas libres para implementar pruebas funcionales, desde la perspectiva de su utilización organizacional y como una alternativa a otras existentes. Se seleccionaron nueve herramientas a las que se aplicaron principios de selección y valoración orientados a encontrar el nivel de madurez y el grado de satisfacción. El proceso de selección y las métricas aplicadas se describen en este estudio de revisión a la web.

Palabras clave — Pruebas funcionales, software libre, madurez, satisfacción.

Abstract — This work aims to provide information about some free tools to implement functional testing, from the perspective of organizational use as an alternative to other existing. We selected nine tools that were applied principles of selection and assessment oriented to find the level of maturity and degree of satisfaction. The selection process and the metrics applied are described in this literature review to the web.

Keywords — Functional tests, free software, maturity, satisfaction.

I. INTRODUCCIÓN

Las pruebas constituyen un área importante del ciclo de vida de los productos software y su objetivo es asegurar la calidad, fiabilidad y robustez de los mismos dentro del contexto o escenario en el que se prevé se utilización [1]. Es una tarea compleja que exige dedicación y trabajo, pero aunque se considera importante para asegurar la calidad del software, en la industria no siempre reciben la atención necesaria. Esto se debe entre otras causas a que otras actividades técnicas, como la documentación, los enfoques

existentes y otras que hacen parte de esta actividad, condicionan los procesos y se deben seleccionar y utilizar de acuerdo con los contextos en que se desarrolla e implementa el proyecto.

Para llevar a cabo esta tarea existen herramientas de apoyo enmarcadas en la filosofía del el software libre, pero aunque conocen de ellas, muchas organizaciones y profesionales, por diversas razones, no las tienen en cuenta para utilizarlas en los proyectos. Algunas de las causas es que se tiene la percepción de otras brindan mayor soporte en aspectos como garantía, acompañamiento y respaldo, aunque existen investigaciones convalidadas en las cuales se desmienten estos *mitos* [2].

En este trabajo se describen algunas herramientas libres disponibles para realizar pruebas al software y se analizan sus características. Además, se hace un análisis detallado de aquellas cuyo objetivo es aplicar pruebas funcionales orientadas a la web. Entre sus funciones se encuentra asegurar el correcto funcionamiento del sistema y verificar que cumpla con los requisitos específicos para los cuales se desarrolló. A estas pruebas también se les denomina de caja negra, y su objetivo es analizar los resultados que ofrece el sistema luego de aplicar una variación de datos de entrada.

Este documento tiene la siguiente estructura: la sección dos presenta la metodología aplicada para el proceso de búsqueda, la selección y la calificación a las herramientas del estudio; en el apartado tres se describen los resultados y el análisis realizados, y en la sección cuatro se presentan las conclusiones.

II. METODOLOGÍA

Para la revisión sistemática de la literatura se aplicó la metodología propuesta en [3] y [4], orientada a lograr la calidad de la información que se pretende utilizar en los análisis y relevancia al proceso de revisión, y para identificar el nivel de madurez de las herramientas y determinar el grado de satisfacción que generan, se basa en la propuesta de [5].

A. Revisión sistemática

Para recopilar la información referente a la investigación se siguieron los siguientes pasos:

- *Preguntas de investigación.* Las preguntas de investigación para el estudio fueron:
P1: ¿Qué nivel de madurez tienen las herramientas libres para realizar pruebas de Verificación?
P2: ¿Cuál es el grado de satisfacción que tienen los usuarios con estas herramientas?
- *Proceso de búsqueda.* En el contexto del software libre la información se actualiza constantemente, porque es un mundo dinámico y constantemente se anuncia nuevos desarrollos y nuevas versiones. Teniendo en cuenta esta característica, la búsqueda para esta investigación se realizó orientada a la web y mediante combinaciones de palabras como: *Testing tools, Verification, Software testing, test case, Black Box y Functional test*, combinados con *Free Software y Open Source*. Se encontraron diversas fuentes que eventualmente podrían satisfacer las características de la búsqueda, sin embargo, fue necesario evaluarlas para asegurar la veracidad y credibilidad de la información. El motor de búsqueda utilizado fue Google [6] a través de Google Academic, y se aplicó la bitácora de exploración propuesta en [7].
- *Criterios de inclusión y Exclusión.* Los sitios web revisados debían ser relevantes para las pruebas de software mediante herramientas libres. De acuerdo con [8] existen algunas técnicas para filtrar información de sitios web, entre las que destacan:
 - Conocer las claves de búsqueda para obtener los resultados más relacionados con la intención inicial.
 - Comprender las diferentes secciones del buscador.
 - Enjuiciar críticamente qué resultados identificar como veraces.

Siguiendo los indicadores que se describen en el Anexo 1 se realizó una exploración detallada a los sitios web hallados en el proceso de búsqueda, con el objetivo de obtener la

información necesaria para tomar decidir cuál o cuáles de ellos incluir en la muestra final.

Luego de este proceso se seleccionó el sitio <http://www.opensourcetesting.org/>, porque fue el cumplió la mayoría de los indicadores de la Tabla 1. De las 122 aplicaciones en este sitio se seleccionó un conjunto de herramientas libres orientadas a la realización de pruebas funcionales, del cual se hizo un filtro posterior. A esta muestra se aplicaron las fases de análisis propuestas en [5]:

Fase 1. Orientada a lograr un conocimiento global de cada una, mediante un proceso de identificación a los proyectos que las originaron.

Fase 2. Para identificar el nivel de respaldo a la herramienta.

Fase 3. Con el objetivo de determinar la estabilidad del producto y el grado de satisfacción que genera entre los usuarios.

Con base en la información obtenida en estas fases se calificaron las herramientas de acuerdo con los siguientes criterios:

N/A = No se encontró

M = Bajo

R = Regular

B = Adecuado

MB = Muy adecuado

Antigüedad. Se refiere a que si la herramienta es nueva puede que no tenga un nivel de madurez suficiente. Este análisis se hizo de acuerdo con los datos de la Tabla 1.

Tiempo	Calificación
Antes del 2000	MB
2001 al 2005	B
2006 al 2010	R
2011 al 2013	M

Tabla 1: Criterios para la antigüedad

Actualización. Se tuvo en cuenta la fecha de publicación de la última versión, de acuerdo con los criterios de la Tabla 2.

Tiempo	Calificación
2011 al 2013	MB
2006 al 2010	B
2001 al 2005	R
Antes del 2000	M

Tabla 2: Criterios para la actualización

Descargas. Desde la publicación de la primera versión. Los criterios se describen en la Tabla 3.

Descargas	Calificación
1001 en adelante	MB
501 a 1000	B
101 a 500	R

1 a 100

M

Tabla 3: Criterios para las descargas

La muestra final de herramientas quedó constituida por un conjunto de nueve, la cual se describe en las Tablas 4 y 5.

III. RESULTADOS

HERRAMIENTA	DESCRIPCIÓN	SITIO
XML Test Suite	Comprueba el contenido y la estructura de páginas HTML, que estén bien formadas y que los enlaces sean válidos. Las pruebas están escritas en XML.	http://xmltestsuite.sourceforge.net
XENU	Comprueba los enlaces rotos de los sitios Web. La verificación se hace en enlaces, imágenes, marcos, plugins, fondos, mapas locales de imágenes, hojas de estilo, scripts y applets java y muestra continuamente una lista actualizada de las direcciones URL que se puede ordenar por diferentes criterios.	http://home.snafu.de/tilman/xenulink.html
Link Evaluator	Extensión de Firefox diseñada para evaluar la disponibilidad de links de una página web determinada. El resultado de cada enlace evaluado es resaltado con un color basado en el éxito relativo del resultado: verde para un éxito completo, tonos de amarillo para un éxito parcial, y rojo para cuando falla.	http://evaluator.oclc.org/
Link checker	Comprueba los vínculos de los sitios web. Los resultados son arrojados en diferentes formatos: HTML, SQL, CSV, XML, en gráficos, con diferentes colores o en texto normal. Soporta protocolos HTTP/1.1, HTTPS, FTP, news, nntp, mailto y telnet.	http://wummel.github.com/linkchecker/
TestPlan	Prueba páginas web dinámicas a través de Firefox o Internet Explorer. Evalúa Correo electrónico y FTP, texto, HTML, CSV, XML y datos, posee un rastreo avanzado para la localización de errores durante la noche.	http://testplan.brainbrain.net/
Selenium IDE	Complemento de Firefox para las pruebas de grabación y reproducción como una serie de pasos ejecutados y con posibilidad de hacer debug y definir break points durante la reproducción de la prueba. Los casos pruebas se pueden grabar en: HTML, Ruby, Perl, C#, Junit, Groovy, TestNG, Python, PHP y RSpec.	http://seleniumhq.org/
JMeter	Diseñado para probar aplicaciones Web en cuanto a rendimiento de los recursos, tanto estáticos como dinámicos (archivos, Servlets, scripts de Perl, objetos Java, bases de datos y consultas, servidores FTP y mucho más). Se puede simular una carga pesada en un servidor de red o para analizar el rendimiento general bajo diferentes tipos de carga y hacer análisis gráfico de rendimiento.	http://jmeter.apache.org/
WebInject	Realiza pruebas automatizadas de aplicaciones web y servicios web. Prueba los componentes individuales del sistema que tengan interfaces HTTP, JSP, ASP, CGI, PHP, AJAX, Servlets, formularios HTML, XML / SOAP, REST Web Services, etc. permite ejecutar muchos casos de prueba y recopilar / reportar sus resultados. Ofrece visualización en tiempo real los resultados y también se puede utilizar para el seguimiento de los tiempos de respuesta del sistema.	http://webinject.org/
soapUI	Permite crear fácil y rápidamente y ejecutar automatizado regresión funcional, cumplimiento y pruebas de carga. En un entorno único ensayo, SoapUI ofrece cobertura de la prueba completa y soporta todos los protocolos estándares y tecnologías. Simplemente, no hay límites a lo que puede hacer con sus pruebas.	http://www.soapui.org/

Tabla 4: Herramientas de la muestra

Herramienta	Versión Inicial	Versión Actual	Licencia	Plataforma	Interfaz	Lenguaje
XML Test Suite	Versión. 1.0 04-11-2002	Versión 1.2 08-05-2003	LGPL	Windows Linux	Comand o	Java
XENU	Versión. 1.0a 10-08-1997	Versión 1.3.8 04-09-2010	Freeware	Windows	GUI	C++

Link Evaluator	Versión 0.01 Oct 2005	Versión 0.9.9.7 06-05-2009	Apache, versión	Windows Linux	GUI	HTML
LinkChecker	Versión 4.1 24-11-2004	Versión 4.81 16-10-2011	GPL	Windows Linux	GUI	Python
TestPlan	Versión 0.0r12 18-09-2008	Versión 1-0-r6 09-10-2010	GPL	Windows Linux	Comando	Java-Test
Selenium IDE	Versión 0.8.0 03-06-2006	Versión 1.10.0 20-12-2012	GPL	Windows Linux	GUI	Java, Perl, Csharp, Php, Python, Ruby, Javascript Java
Jmeter	Versión 1.0 15-12-1998	Versión 2.9 29-01-2013	Apache, versión	Windows Linux	GUI	Java
WebInject	Versión 0.90 19-02-2004	Versión 1.41 04-01-2006	GPL	Windows Linux	GUI	Perl
soapUI	Versión 1.5 06-04-2006	Versión 4.5.1 27-06-2012	LGPL	Windows Linux	GUI	Java

Tabla 5: Información básica de las herramientas

En la Tabla 6 se describe los hallazgos de la fase 2.

Herramienta	Guía de instalación	Manual de usuario	Preguntas frecuentes	Soporte Online			Código documentado
				Foro	Lista email	Blog	
XML Test Suite	Sí	Sí	No	No	No	No	Sí
XENU	Sí	Sí	Sí	Sí	Sí	Sí	No
Link Evaluator	Sí	Sí	Sí	No	No	No	No
LinkChecker	Sí	Sí	Sí	Sí	No	No	Sí
TestPlan	Sí	Sí	No	Sí	Sí	No	Sí
Selenium IDE	Sí	Sí	Sí	Sí	No	Sí	Sí
JMeter	Sí	Sí	Sí	Sí	Sí	No	No
WebInject	No	Sí	Sí	No	No	No	No
SoapUI	Sí	Sí	Sí	Sí	Sí	No	Sí

Tabla 6: soporte y documentación de las herramientas

La Tabla 7 representa el grado de satisfacción.

Herramienta	Número de descargas	Numero de bugs	Nivel de participación	Respuesta a los usuarios	Ultimo año de publicación
XML Test Suite	7593	10	11	1 semana	2004
XENU	56980	101	1628	1 a 2 días	2013
Link Evaluator	N/A	2852	N/A	N/A	N/A
LinkChecker	142,797	1866	392	1 mes	2013
TestPlan	1611	64	9	Más de 6 meses	2012
Selenium IDE	891	750	31445	15 días	2013
JMeter	11439	1824	16222	1 a 2 días	2013
WebInject	123915	11	N/A	N/A	N/A
SoapUI	4.623.981	699	1861	1 a 2 días	2013

Tabla 7: datos para la estabilidad y satisfacción de las herramientas

Las Tablas 8 y 9 describen el nivel de madurez y el grado de satisfacción.

Herramienta	Antigüedad	Grado de actualización	Número de descargas	Nivel de documentación	Número de Bugs
XML Test Suite	B	R	B	R	MB
XENU	MB	B	B	MB	MB
Link Evaluator	B	B	NA	R	M
LinkChecker	B	MB	MB	B	M
TestPlan	R	B	R	B	MB
Selenium IDE	R	MB	R	MB	B
JMeter	MB	MB	B	B	M
WebInject	B	B	MB	R	MB
soapUI	R	MB	MB	MB	B

Tabla 8: nivel de madurez de las herramientas

Herramienta	Nivel de participación	Nivel de respuesta a los usuarios	Año de publicación
XML Test Suite	M	MB	R
XENU	B	MB	MB
Link Evaluator	N/A	N/A	N/A
LinkChecker	R	MB	MB
TestPlan	M	R	MB
Selenium IDE	MB	MB	MB
JMeter	MB	MB	MB
WebInject	N/A	N/A	N/A
soapUI	B	MB	MB

Tabla 9: grado de satisfacción de las herramientas

IV. DISCUSIÓN

Con base en estos resultados se observa varias características similares entre las herramientas. Las que logran mayor nivel de madurez son las que tienen mayor antigüedad y a lo largo del tiempo han sido constantemente actualizadas. Esto les ha permitido lograr una amplia documentación y una alta participación de los usuarios, lo que ha hecho que los autores corrijan las debilidades y refuercen la prevención de los defectos que presentan. De esta manera se mejoran sus funcionalidades y la eficacia de los procesos del desarrollo. Las herramientas con un nivel de madurez bajo son relativamente recientes y no han tenido un recorrido amplio, o que por el contrario fueron lanzadas hace mucho tiempo pero no tuvieron mejoras en el transcurso del tiempo; además, generalmente presentan documentación y soporte valorados como deficientes.

V. CONCLUSIONES

El mundo del software libre es sorprendentemente dinámico y la velocidad con la que se anuncia nuevos desarrollos es

realmente alta. En este trabajo se estudiaron características importantes de algunas herramientas de este mundo para la realización de pruebas funcionales en aplicaciones web, y se analizaron aplicando ciertos criterios con el objetivo de determinar el nivel de madurez y el grado de satisfacción de cada una de ellas. El objetivo no es recomendar una u otra, porque elegir la más adecuada no es una tarea sencilla. Además, para poder determinar si una aplicación es mejor que otra se debe tener en cuenta el contexto en el cual se desarrolla y la diversidad de usuarios que la utilizan, para de esta manera poder recopilar la información necesaria desde los distintos puntos de vista. Una de las características del software libre es la satisfacción que infunde en los usuarios, por lo que el grado de satisfacción es un factor importante para que la herramienta alcance el éxito.

La evaluación a través de este modelo le brinda a las organizaciones elementos de juicio para decidir entre una u otra herramienta, porque permite establecer una medida del progreso y de avance a través del panorama general que presenta. Este estudio hace un aporte a las intenciones de ofrecer las alternativas libres a las empresas, para que decidan implementar o utilizar las herramientas que se desarrollan bajo esta filosofía.

REFERENCIAS

- [1] Serna, M.E., & Arango, I.F. (2011). "Software Testing: More than a Stage in the Life Cycle." *Revista de Ingeniería. U. de los Andes*, No. 35, pp. 34-40.
- [2] Serna, M.E., & Serna, A.A. (2012). "An Evaluation of Free Tools for Software Testing." *Revista Virtual Universidad Católica del Norte*, No. 37, pp. 44-61.
- [3] Kitchenham, B. et al. (2009). "Systematic literature reviews in software engineering – A systematic literature review." *Journal Information and Software Technology*, Vol. 51 No 1, pp. 7-15.
- [4] Kitchenham, B. (2004). "Procedures for Undertaking Systematic Literature Reviews." *Joint Technical Report Computer. Computer Science Department, Keele University*.
- [5] Díaz, F.J. et al. (2010). "Evaluación de herramientas Free/Open Source para pruebas de Software." *Laboratorio de Investigación de Nuevas Tecnologías Informáticas, Facultad de Informática, Universidad de La Plata. Argentina*.
- [6] Hawking, D., & Craswell, N. (2001). "Which Search Engine is best at finding Online Services?" Online: <http://www10.org/cdrom/posters/1089.pdf> [Nov 2012].
- [7] Glosario Eduteka. Online: <http://www.eduteka.org/glosario/tiki-index.php?page=Bit%C3%A1cora+de+b%C3%BAscueda> [Nov 2012].
- [8] Maglione, C., & Varlotta, N. (2012). "Investigación gestión y búsqueda de información en Internet." Online: <http://bibliotecadigital.educ.ar/uploads/contents/investigacion0.pdf> [Dic 2012].

ANEXO 1

INDICADORES DE INCLUSIÓN/EXCLUSIÓN

Característica	Descripción	Objetivo	Indicadores
Formalidad	Responsable del sitio, puede ser una persona, un grupo de personas o una entidad, además de su prestigio y las fuentes utilizadas	Seleccionar recursos y sitios provenientes de fuentes confiables, específicas, actualizadas y reconocidas	<ul style="list-style-type: none"> ▪ Es claro y explícito quién es el responsable del sitio ▪ Describe el propósito de quien promueve el sitio ▪ La información del sitio es verificable ▪ Exhibe el número de visitas
Actualización	Incorporación periódica de nuevos recursos, o a la modificación de los existentes como respuesta a la aparición de nuevos aportes al tema. La última actualización del sitio debe tener dos meses o menos	Identificar sitios y recursos que contengan información actualizada	<ul style="list-style-type: none"> ▪ Indica claramente la fecha de actualización ▪ Es posible encontrar otros indicios de que la información está actualizada
Navegabilidad	Facilidad con la que el usuario se ubica y navega en el sitio	Seleccionar sitios con una estructura clara y ordenada y un diseño que facilite la orientación del usuario durante la navegación	<ul style="list-style-type: none"> ▪ Incluye un mapa del sitio ▪ Se puede identificar y ubicar fácilmente la página a visitar ▪ Hay un enlace a la página inicial desde cualquier otra página ▪ Se puede volver fácilmente a una página ya visitada ▪ Los enlaces se agrupan con algún criterio reconocible ▪ Los íconos representan claramente su propósito
Organización	El sitio se encuentra ordenado lógicamente y los segmentos de información tienen relaciones bien definidas.	Seleccionar sitios que presenten una adecuada organización de contenidos, que cumplen una función didáctica para la consulta y la exploración	<ul style="list-style-type: none"> ▪ Los encabezamientos, títulos, o capítulos son pertinentes ▪ La información sobre cada tema es completa y lógica ▪ Presenta herramientas de ordenamiento (índices, esquemas)
Selección de contenidos	Los contenidos del sitio tratan adecuadamente el enfoque dado al desarrollo de un tema, tópico o teoría de un campo disciplinar o área del conocimiento	Identificar sitios que incluyan contenidos consistentes, relevantes y significativos	<ul style="list-style-type: none"> ▪ La información se presenta con claridad ▪ La información es relevante ▪ Existe coherencia entre los objetivos, los contenidos y los procedimientos ▪ Existe coherencia entre el texto, las

			<p>imágenes y el sonido</p> <ul style="list-style-type: none"> ▪ La información está libre de errores gramaticales y ortográficos
Legibilidad	Tiene una adecuada combinación de colores, tamaños y tipos de letras, fondos e ilustraciones	Identificar sitios de fácil lectura y navegabilidad	<ul style="list-style-type: none"> ▪ El tamaño de letra utilizada es adecuado ▪ La imagen de fondo ofrece un buen contraste con el tipo de letra ▪ El texto y los gráficos están bien distribuidos ▪ Las páginas no están sobrecargadas
Adecuación al destinatario	Se adecua para cada tipo de usuario	Verificar que el contenido sea adecuado para el destinatario	<ul style="list-style-type: none"> ▪ Ofrece claramente la especificación del tipo de usuario al que se dirige El contenido no ofende ▪ El contenido es adecuado para cada usuario