

TESTING ÁGIL DE SOFTWARE CON HERRAMIENTAS LIBRES Y ABIERTAS

FABIO ERNESTO GARCIA RAMÍREZ
Magister en Software Libre
Fundación Universitaria Tecnológico Comfenalco
fgarcia@tecnologicocomfenalco.edu.co

RESUMEN

El presente artículo pretende mostrar una visión de las pruebas de software (Testing) a partir de la experiencia del autor, mediante el uso de metodología Ágil (SCRUM) y herramientas de software libre y abierto aplicadas a un proyecto en particular: SWEST – Software para Empresas de Servicios Temporales. Se describirá como la metodología SCRUM, normalmente aplicada al desarrollo de software, puede ser adaptada al proceso de pruebas en proyectos donde el factor tiempo es una limitante para la obtención de los resultados; igualmente se evidenciará que herramientas de gestión de este tipo de procesos como Testlink, Mantis y PangoScrum, provenientes del Software Libre y Abierto e inclusive de la nube, demuestran ser confiables para la administración del Testing.

Palabras Claves— Pruebas de Software, Testing, Scrum, TestLink, Mantis, PangoScrum

1. INTRODUCCION

El desarrollo de software es un proceso que involucra ambigüedad, suposiciones y fallas en las comunicaciones del equipo. Cada cambio realizado a un componente de software, la adición de una nueva funcionalidad, el intento de reparar un defecto, conlleva a una posibilidad de generar un fallo. Con cada fallo, el riesgo de que el software no cumpla con los requisitos del cliente, se incrementa.

Teniendo en cuenta el anterior panorama, el Testing (prueba) de software tiene como objetivo reducir o minimizar el impacto de estos fallos en la funcionalidad y en los resultados esperados de la aplicación.

Para comprender mejor la importancia del proceso de pruebas en la Ingeniería de Software, se hace necesario precisar algunos conceptos al respecto, para lo cual el desarrollo de este artículo se organiza de la siguiente manera: en la sección II de fundamentos de pruebas de software, se abordarán los conceptos relacionados con: testing,

técnicas, metodología, gestión y herramientas para pruebas de software; en la sección III se abordarán los resultados obtenidos con la aplicación de las herramientas y finalmente en la sección IV las conclusiones y recomendaciones.

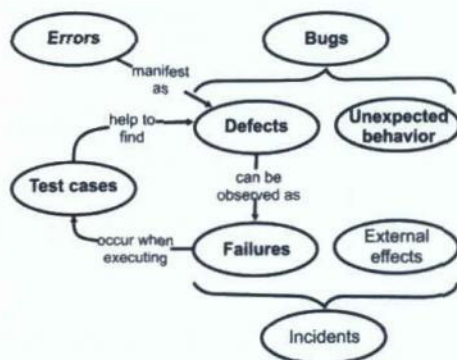
2. FUNDAMENTOS DE PRUEBAS DE SOFTWARE

Las pruebas de software pueden ser consideradas como el Proceso de ejecutar una aplicación en unas condiciones específicas, monitoreando los resultados, evaluando los aspectos del componente, proceso o sistema.(IEE 610.12-1990, citado en Majchrzak, 2012, p.14).

Conceptos de Testing.

La realización de este proceso involucra necesariamente el conocimiento de los siguientes conceptos, mostrados en la Figura 1:

Figura 1. Conceptos relacionados con las pruebas de software



Fuentes: Majchrzak, 2012, p.14

- Casos de Prueba (TestCases): corresponde a un flujo normal y alternativo de pasos a ser probados, de acuerdo a condiciones establecidas y resultados esperados de un determinado requisito o funcionalidad de la aplicación.

- Error: acción humana que produce un resultado incorrecto, como por ejemplo un error en el modelamiento de datos.(IEEE 610.12,1990, p. 31).
- Defectos (Bugs): desperfectos o anomalías que pueden causar que el sistema falle. Normalmente son la manifestación del Error.
- Fallos(Failures): imposibilidad de una aplicación de realizar determinada funcionalidad(es), correspondiente a unos requisitos definidos previamente.(IEEE 610.12, 1990, p.32)
- Incidentes (Incidents): es un evento(s) en la ejecución de un software, que genera fallos que merecen la atención del lider de pruebas.

Técnicas de Pruebas de Software

Existen dos grandes grupos de técnicas de pruebas de software:

- Caja Negra(Black Box): se centra en comparar las entradas con las salidas esperadas, sin detenerse en detalles de su desarrollo(código fuente).(Jenkins, 2008, p. 9)
- Caja Blanca(White Box): se centra en analizar el código fuente y la lógica interna del software. (Jenkins, 2008, p.9)

Los resultados de la experiencia del autor se enfocan en las pruebas de Caja Negra (Black Box), donde se valida requisitos vs funcionalidades esperadas en la aplicación.

Metodología Ágil para Pruebas

Existen varios enfoques sobre la manera se puede abordar la ejecución de las pruebas de software; sin embargo, y para el caso particular del proyecto

SWEST – Software para Empresas de Servicios Temporales, donde el autor está a cargo de este proceso, se tenían las siguientes condiciones:

- Necesidad urgente del Cliente por tener en producción la aplicación, en concreto el módulo de Nómina.
- Antecedentes de fallos e incidencias en la primera versión del módulo de nómina.
- Necesidad de mostrar resultados en el menor tiempo posible.

Las metodologías ágiles en el desarrollo de software, se han caracterizado por ofrecer al Cliente resultados en el menor tiempo posible, de tal manera que facilita la verificación y validación de los requisitos versus las funcionalidades que se van implementando.

Una de las metodologías ágiles más representativas es SCRUM, la cual fue la seleccionada por el autor para la ejecución del proceso de pruebas.

La metodología SCRUM puede resumirse en la siguiente gráfica:

Figura 2. Metodología SCRUM.



Fuentes: <http://www.softwaretestingbees.com/wp-content/uploads/2011/12/Scrum.jpg>

Esta metodología involucra:

- Pruebas iterativas
- Backlog (repositorio) de los procesos o módulos a ser testeados.
- Planificación y desarrollo de Sprints(conjunto de procesos o módulos a ser probados) durante un periodo de 15 o 30 días máximo.
- Seguimiento continuo del progreso de cada uno de los Sprints planificados(Burn Down Chart).
- Reuniones Diarias (15 a 30 minutos) con el equipo de pruebas(Daily Scrum Meeting), con el fin de discutir: Que se ha realizado desde la última reunión, Que se necesita realizar hasta la próxima reunión y que impedimentos existen, si los hay.
- Reunión de Planificación de los Sprints(Sprint Planning Meeting), con el equipo de pruebas, con el fin de distribuir los módulos o procesos a ser probados.
- Reunión de Revisión de los Sprints(Sprint Review Meeting), con el propósito de discutir aspectos que permitan optimizar los Sprints a realizar o pendientes.

Teniendo claridad sobre conceptos, técnicas y metodología de pruebas de software, es necesario tener en cuenta que al igual que en otros procesos de la Ingeniería de Software, se requiere planificar aspectos como actividades, tareas, recursos y el tiempo en el cual se va a desarrollar.

Gestión de las Pruebas de Software

Para el logro de un exitoso proceso de pruebas, se hace necesario planificar aspectos tales como:

- Requisitos de la Aplicación claramente definidos en el documento de especificación de requisitos.
- Procesos de Negocio de la Aplicación a ser sometidos a pruebas
- Conformación del Equipo de Pruebas
- Ambiente(hardware y software) de Pruebas
- Procesos del Plan de Pruebas
- Cronograma de las Pruebas.
- Herramienta para administrar los requisitos, casos de pruebas, ejecución de los casos de pruebas y resultados.
- Herramienta para administrar los defectos/bugs detectados al ejecutar los casos de prueba.
- Herramienta para administrar la metodología ágil de Scrum
- En ambos casos, se definió desde el principio, dar preferencia a herramientas de software libre y abierto, con el fin de no generar rubros adicionales al proyecto.

Todos los anteriores aspectos se consolidan en el denominado Plan Maestro de Pruebas, documento guía del proceso.

Entre todos los aspectos mencionados, se destacan los pertenecientes a los procesos del plan de pruebas, los cuales comprenden (Garcia, 2012, p. 10):

- Planeación de las Pruebas
- Selección y ejecución de los Sprints.
- Aplicación de las Pruebas
- Gestión de los defectos
- Certificación de las pruebas
- Métricas de las pruebas

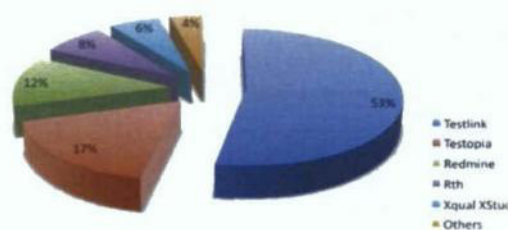
Herramientas de Gestión de las Pruebas

La ejecución de las pruebas de software, requieren la ayuda de herramientas de software que faciliten su gestión y los reportes e informes de resultados.

Concretamente y para el caso en estudio, la necesidad apuntaba a:

Teniendo en cuenta las anteriores necesidades y luego de revisar varias alternativas, se considero idóneo la siguiente estadística de preferencia de uso de estas herramientas, como se muestra en la Figura 3:

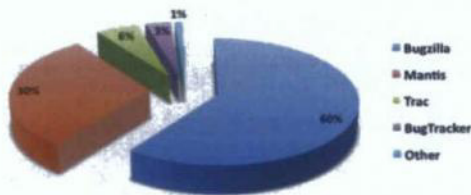
Fig. 3 Popularidad de herramientas de gestión de pruebas de software



Fuentes: *Testing Experience Magazine*, 2010, p. 20

Igualmente lo referente a herramientas de gestión de los defectos, se tiene lo mostrado en la Figura 4:

Fig. 4 Popularidad de herramientas de gestión de defectos de software



Fuentes: *Testing Experience Magazine*, 2010, p. 20

Lo anterior claramente define las herramientas que se encuentran en el “top” de las preferencias de uso:

Fig. 5 Testlink logo.



Fuentes: <http://testlink.sourceforge.net/demo/login.php>

- **TestLink:** Es una herramienta web de administración de las pruebas, que permite la especificación, planeación, ejecución y reporte de los resultados de los casos de prueba, integrándose fácilmente con sistema de gestión de defectos y fallos. (*Testing Experience Magazine*, 2010, p.90).

Fig. 5 Mantis logo



Fuentes: <http://www.mantisbt.org/>

- **Mantis:** Es una herramienta web de código

abierto que permite la gestión de los defectos resultantes de la ejecución de las pruebas de software. Se caracteriza por su facilidad de instalación, uso, multiplataforma, entre otros (*Testing Experience*, 2010, p.19)

Finalmente, quedaba por definir una herramienta que facilitará la administración de la metodología Scrum, la seleccionada fue PangoScrum:

Fig. 6 Pango SCRUM



Fuentes: pangoscrum.com

Pango Scrum: Es una herramienta web, de tipo SaaS (*Software as a Service, Software como Servicio*), que utilizando la nube (Internet), permite la administración de una metodología Ágil como Scrum, permitiendo:

- Gestión del Product Backlog (Repositorio de procesos, requisitos o funcionalidades a ser probados)
- Planeación de los Sprints (Iteraciones con los procesos a ser probados) con unos tiempos especificados.
- Gestión del Calendario de los diferentes Sprints
- Gestión de Usuarios
- Gestión de Proyectos

3. RESULTADOS

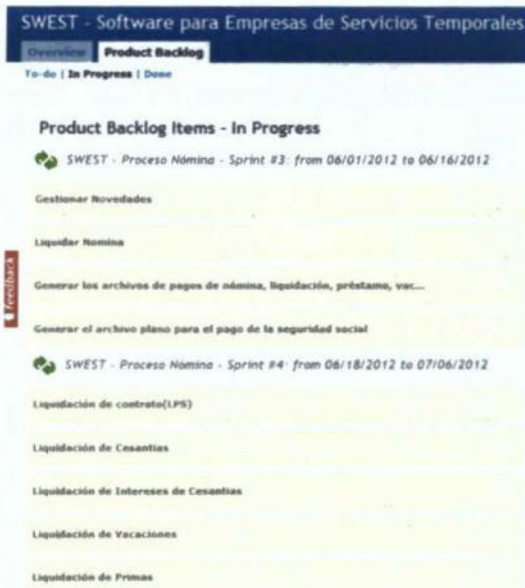
A partir de la elaboración del documento del Plan Maestros de Pruebas, y teniendo las herramientas

instaladas (TestLink y Mantis) se procedió a:

Planeación de las Pruebas

A. Definición del Proyecto (Product) y Product Backlog (Repositorio de los requisitos o funcionalidades) en PangoScrum:

Fig. 7 Product Backlog - Proyecto SWEST.



Fuentes: pangoscrum.com

Definición de los Sprints en PangoScrum:

Fig. 8 Sprints - Proyecto SWEST. Herramienta Pango SCRUM



Fuentes: <http://pangoscrum.com>

Definición del equipo de pruebas en PangoScrum:

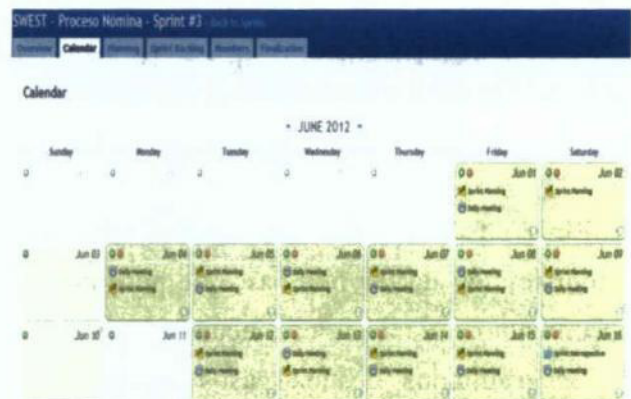
Fig. 8 Equipo de pruebas - Proyecto SWEST. Herramienta Pango SCRUM



Fuentes: <http://pangoscrum.com>

Definición del Cronograma por cada Sprint:

Fig. 9 Cronograma de un Sprint - Proyecto SWEST. Herramienta Pango SCRUM



Fuentes: <http://pangoscrum.com>

Especificación de los Requisitos en TestLink:

Fig. 10 Especificación de Requisitos - proyecto SWEST. Herramienta PangoSCRUM



Fuentes: <http://http://testlink.sourceforge.net/demo/login.php>

Especificación de Casos de Pruebas en TestLink:

Fig. 11 Especificación de Casos de Prueba - Proyecto SWEST, Herramienta PangoSCRUM



Fuentes: <http://testlink.sourceforge.net/demo/login.php>

Selección y Ejecución de los Sprints

Asignación de los Casos de Pruebas al equipo de Testing en TestLink:

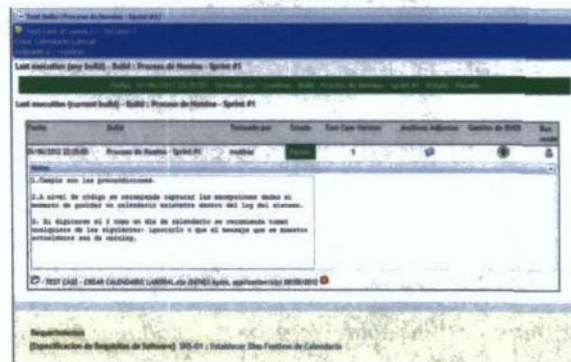
Fig. 12. Asignación de los casos de prueba - Proyecto SWEST. Herramienta testlink



Fuentes: <http://testlink.sourceforge.net/demo/login.php>

Ejecución de cada caso de prueba en TestLink:

Fig. 13. Ejecución de los casos de prueba - Proyecto SWEST. Herramienta testlink



Fuentes: <http://http://testlink.sourceforge.net/demo/login.php>

Gestión de los Resultados

Reporte de métricas de las pruebas de software en TestLink:

Fig. 14. Reporte de Métricas de las Pruebas de Software - Proyecto SWEST. Herramienta testlink

Test Suite	Total	No Ejecutado	[%]	Pasado	[%]	Fallado	[%]	Repetido	[%]	Completado	[%]
Proceso de Nomina - Sprint #1	26	0	0.00	15	57.69	10	38.46	1	3.85	100.00	
Proceso de Nomina - Sprint #2	17	0	0.00	9	52.94	8	47.06	1	5.88	100.00	
Proceso de Nomina - Sprint #3	37	17	45.95	10	27.03	16	42.97	0	0.00	54.05	
Proceso de Nomina - Sprint #4	9	7	77.78	0	0.00	2	22.22	0	0.00	22.22	
Proceso de Contratación - Sprint #1	6	3	50.00	3	50.00	0	0.00	0	0.00	50.00	
Proceso de Nomina - Sprint #5	7	7	100.00	0	0.00	0	0.00	0	0.00	0.00	

Fuentes: <http://http://testlink.sourceforge.net/demo/login.php>

Reporte de Casos de pruebas fallados en TestLink:

Fig. 15. Reporte de Casos de Pruebas Fallados - Proyecto SWEST. Herramienta testlink

Test Case	Estado	Fecha de Ejecución	Descripción del Error
Proceso de Nomina - Sprint #1	Fallado	2013-06-17 22:11:50	Por ser un procedimiento de trabajo automatizado, el error consistió en que al momento de ejecutar el proceso, se generó un error de ejecución. Este error se generó al momento de ejecutar el proceso de trabajo automatizado. El error se generó al momento de ejecutar el proceso de trabajo automatizado. El error se generó al momento de ejecutar el proceso de trabajo automatizado.
Proceso de Nomina - Sprint #1	Fallado	2013-06-17 22:51:51	1. Comprobar con los procedimientos. 2. La subsección del proceso se ejecutó correctamente, pero se generó un error de ejecución. Este error se generó al momento de ejecutar el proceso de trabajo automatizado. El error se generó al momento de ejecutar el proceso de trabajo automatizado. El error se generó al momento de ejecutar el proceso de trabajo automatizado.

Fuentes: <http://testlink.sourceforge.net/demo/login.php>

Reporte de defectos encontrados por caso de prueba en TestLink:

Fig. 16 Reporte de defectos encontrados por caso de prueba - Proyecto SWEST. Herramienta testlink

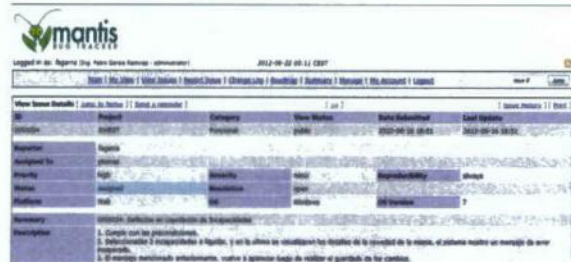
Test Case	Defectos	Estado	Def. Detectado/Ahora
Proceso de Nomina - Sprint #1	26	0	26

Fuentes: <http://testlink.sourceforge.net/demo/login.php>

Gestión de los Defectos

Ingreso de los defectos generados por los casos de prueba ejecutados en TestLink, en la herramienta Mantis:

Fig. 17. Ingreso de Defectos - Proyecto SWEST. Herramienta Mantis



Fuentes: http://www.mantisbt.org/demo/my_view_page.php

Listado de Defectos registrados en la herramienta Mantis:

Fig. 18 Reporte de defectos encontrados por casos de prueba - Proyecto SWEST. Herramienta testlink

ID	Project	Category	Severity	Priority	Assign To	Reported By	Created	Updated	Status
000001	Proyecto SWEST	Procesos	Crítica	Alta	Administrador	Administrador	2013-06-17 22:11:50	2013-06-17 22:11:50	Activo
000002	Proyecto SWEST	Procesos	Crítica	Alta	Administrador	Administrador	2013-06-17 22:51:51	2013-06-17 22:51:51	Activo

Fuentes: http://www.mantisbt.org/demo/my_view_page.php

Los defectos registrados en Mantis se asignan al Líder de Desarrollo del proyecto SWEST, con el fin de que se proceda a realizar las correcciones que se requieran. Durante este tiempo, el defecto/bug permanece en estado activo en la herramienta TestLink

Cuando se ha terminado las correcciones, se debe proceder nuevamente a aplicar los respectivos casos de prueba, con el fin de asegurar que no vuelvan a ocurrir los defectos y fallos.

4. CONCLUSIONES

La experiencia del autor en la realización de las pruebas de software en el proyecto Swest, le permite hacer las siguientes precisiones:

- Las pruebas de software son un proceso tan importante como el desarrollo, de tal manera que tiene alto impacto en la satisfacción final del Cliente/Usuario. Se recomienda su aplicación en lo posible de manera paralela, en la medida que se tengan prototipos funcionales de los requisitos, con el fin de asegurar la calidad en la aplicación que se construye.
- Con los tiempos de entrega que se manejan actualmente en los proyectos, resulta imprescindible el contar con la ayuda de herramientas para la gestión del proceso de pruebas de software, las cuales facilitan considerablemente la generación de informes acerca del avance del proceso.
- TestLink se recomienda como herramienta de software libre para la gestión del proceso de pruebas, puesto que permite llevar la trazabilidad de los requisitos, casos de pruebas, resultados y defectos que se encuentren en la ejecución de los tests.
- Mantis se constituye en una buena alternativa para la gestión de los defectos encontrados en la ejecución de los casos de prueba, permitiendo realizar el seguimiento en su solución y posterior validación.
- PangoScrum, es una herramienta sencilla pero eficaz cuando se trata de aplicar la metodología ágil de Scrum, tanto para el desarrollo como para la planeación y ejecución de pruebas de software.
- Finalmente, el software libre y abierto dispone de herramientas con la confiabilidad, trayectoria e idoneidad reconocida por los usuarios en el terreno de las pruebas de software, que merecen ser tenidas en cuenta al momento de evaluar herramientas de apoyo al proceso de Testing.

5. REFERENCIAS

- Garcia, Fabio (2012). Plan Maestro de Pruebas. Proyecto SWEST. Colombia.
- Herramienta TestLink: <http://xxx.16.240.2xx/testlink/login.php>
- Herramienta Mantis. http://xxx.16.240.2xx/mantisbt/login_page.php?return=%2Fmantisbt%2Fmanage_proj_create_page.php
- Herramienta PangoScrum. <http://fagarra.pangoscrum.com/login>
- IEEE(1990). IEEE Standard Glossary of Software Engineering Terminology, Std 610.12-1990. New York, IEEE.
- Jenkins, Nick(2008). A Software Testing Primer: An Introduction to Software Testing. USA, Creative Commons.
- Majchrzak, Tim A.(2010). Improving Software Testing: Technical and Organizational Developments. Germany, Springer.
- Testing Experience Magazine(December 2010). http://www.testingexperience.com/testingexperience12_12_10.pdf