

Prácticas de Pruebas desde la Industria de Software. La Plataforma ASISTO como Caso de Estudio

Luis A. Blanquicett, María C. Bonfante* y Jairo Acosta-Solano

Facultad de Ingeniería, Programa de Ingeniería de Sistemas, Corporación Universitaria Rafael Núñez, Cartagena-Colombia (e-mail: luis.blanquicett@curnvirtual.edu.co; mariaclaudia.bonfante@curn.edu.co; jairo.acosta@curnvirtual.edu.co)

* Autor a quien debe ser dirigida la correspondencia.

Recibido Jun. 28, 2017; Aceptado Ago. 31, 2017; Versión final Nov. 1, 2017, Publicado Feb. 2018

Resumen

Este artículo revisa conceptos, metodologías y estándares de calidad en procesos de pruebas de software. Se presenta el proceso utilizado para validar y verificar el grado de cumplimiento de requerimientos levantados inicialmente para el desarrollo de la Plataforma Integradora para la Prestación de Servicios y Planes de Seguimiento para la Salud en el Trabajo, ASISTO. Esta fue desarrollada por la firma SYSNET, en el marco de la convocatoria de Colciencias 709 de 2015 para un grupo de cuatro instituciones prestadoras de salud - IPS. La metodología de los procesos de pruebas del software puede considerarse como una buena práctica de la Ingeniería de Software para la verificación y validación de requerimientos a través de las pruebas del producto. Esto facilita el desarrollo de las aplicaciones y el aseguramiento de la calidad del producto.

Palabras clave: pruebas, ingeniería de software, verificación, validación, requisitos

Testing Practices from the Software Industry. The ASISTO Platform as a Case Study

Abstract

This article reviews concepts, methodologies and quality standards in software testing processes. The process used to validate and verify the degree of fulfillment of requirements initially raised for the development of the Integrated Platform for the Provision of Services and Health Monitoring Plans in the Work - ASSIST, is presented. The platform was developed by the company SYSNET, within the framework of the competition 709 done by Colciencias in 2015 for a group of four Health Provider Institutions - IPS. This methodology for software testing processes can be considered as a good practice in Software Engineering for the verification and validation of requirements through product testing. This facilitates the development of applications and the quality assurance of the software product.

Keywords: testing, software engineering, verification, validation, requirements

INTRODUCCIÓN

A través, de la historia de la ingeniería del software y la industria, las pruebas del producto se han convertido en una importante herramienta para el aseguramiento de la calidad del producto final, lo cual permite identificar si satisface con los requisitos iniciales del cliente. Según los resultados de Standish Group, solo el 29% de los proyectos analizados en grandes y medianas empresas desarrolladoras de software resultaron exitosos. Es decir, aquellos proyectos que son entregados en tiempo, no exceden el presupuesto y que cumplen con los requisitos establecidos por el cliente (Capote-García et al., 2015). Existen estudios en España sobre madurez y buenas prácticas del proceso de pruebas, se proponen cinco niveles de clasificación: arte, habilidad individual, proceso definido, organización avanzada de pruebas y calidad. Los ítems que revelan una mayor implantación (mayor o cercana al 50% de las respuestas) en la práctica de las organizaciones de desarrollo son: validación del cumplimiento de las expectativas del cliente (65,69%), especificaciones bien documentadas (49,02%), informe de los resultados de las pruebas al equipo desarrollador (49,02%) (Sanz, 2005).

En un estudio previo se identifica que el 64% de las empresas encuestados llevan procedimientos documentados para las pruebas, el 25% dispone de personal encargado para pruebas, por lo que se concluye, que los que participan en los procesos de desarrollo son los mismos encargados de las pruebas y un 67% afirman certificar el nivel de calidad del producto, aunque se señala que la externalización de las pruebas de software constituye una alternativa fiable y proporciona un catálogo de servicios con los niveles de calidad predefinidos; además de contar con un personal con nivel óptimo de experiencia y la optimización en los costos vinculados al proceso de pruebas (Castilla et al., 2015). En cuanto a las metodologías de pruebas de software, existen diferentes modelos de referencia que definen el conjunto de buenas prácticas al momento de realizar pruebas de un producto dentro de una organización, llevando a cabo una mejora de procesos. CMMI es el modelo de referencia más utilizado en la industria del software, este proporciona la cobertura necesaria para el desarrollo de actividades aplicadas, tanto a productos, como a servicios (Schots et al., 2015). Además, existen modelos de referencia dirigidos especialmente al proceso de pruebas de software, que ofrecen herramientas para establecer niveles de madurez de desarrollo y mantenimiento del software; entre ellos se encuentran los modelos: TMap (*Test Management Approach*), TPI (*Test Process Improvement*), TMM (*Test Maturity Model*) (Castilla et al., 2015) y TMMI (*Testing Maturity Model Integrated*) (Escobar-Sánchez y Fuertes-Díaz, 2015), que permiten la evaluación y mejora de procesos de pruebas.

El proceso de prueba software para el caso de estudio, se enfoca en las pruebas funcionales, estas permiten validar cuando el comportamiento observado del software cumple o no con las operaciones y/o acciones descritas en los requerimientos. En esta prueba las funciones son probadas ingresando las entradas y examinando las salidas (Myers et al., 2011). También se realiza una verificación dinámica del comportamiento de un sistema, basada en la observación de un conjunto seleccionado de ejecuciones controladas o casos de prueba. Estas pruebas se aplican al producto final, y permiten detectar en qué puntos el producto no cumple sus especificaciones, es decir, comprobar su funcionalidad. La planificación, que consiste en definir los aspectos a examinar y la forma de verificar su correcto funcionamiento, debe ser implementada a partir de un ensamble entre las metodologías adoptadas y la experiencia de la empresa con la cual se desarrolla el proyecto (González, 2009).

Otra prueba frecuentemente aplicada en el proceso, es la prueba de Interfaz gráfica, la cual verifica los componentes físicos de la interfaz y los mensajes de información del cliente, cumpliendo con los diseños y los requisitos definidos por las partes interesadas (Aristegui, 2010). La prueba del sistema se hace para probar los requerimientos no funcionales del sistema, tales como desempeño, exactitud, confiabilidad. Las interfaces externas, los dispositivos de hardware o el ambiente de funcionamiento, también se evalúan en este nivel. Para las pruebas de rendimiento, se registra el tiempo que demoran las transacciones de las operaciones en realizarse (González, 2009). Durante las pruebas de regresión, se hace énfasis en detectar varios comportamientos importantes del software con el objetivo de verificar el comportamiento de los requisitos (Marculescu et al., 2016), esto implica la reejecución de algunas o todas las pruebas realizadas anteriormente.

Nuestro caso de estudio de pruebas de software es la Plataforma Integradora para la Prestación de Servicios y Planes de Seguimiento para la Salud en el Trabajo ASISTO, sistema que pretende solucionar la falta de información adecuada y completa que impide que autoridades de salud y los empresarios tomen decisiones para disminuir los riesgos, prevenir accidentes y enfermedades, y fomentar una cultura del autocuidado, además el cumplimiento de la legislación vigente colombiana. Este trabajo describe el proceso de pruebas de software aplicado por la firma desarrolladora SYSNET, la cual está estructurado en fases con roles definidos e instrumentos de pruebas de interfaz y de funcionalidad para validar la calidad del producto desarrollado.

PROCESO DE PRUEBAS DE SOFTWARE DE SYSNET

Las organizaciones de desarrollo de software se han interesado en alcanzar niveles de capacidad en sus procesos para obtener la madurez organizacional (Galvis-Lista y Sánchez-Torres, 2013) y se presenta una relación positiva entre los modelos de procesos de certificación como CMMI y ISO 9000, y características de software de calidad tales como fiabilidad, prueba, usabilidad, eficiencia e integridad (García-Mireles et al., 2013). Por esta razón, el diagrama de procesos de prueba de software que aplica SYSNET para sus productos, donde se identifican los roles de desarrollador, verificador (*tester*) y cliente, el cual se fundamenta en una integración de los estándares CMMI-DEV3, e ISO 9001 enfocada a la calidad del producto, y para la fase de pruebas del producto se considera un nivel de madurez grado tres según el estándar CMMI, que describe requisitos necesarios para definir, evaluar y mejorar los procesos implementado buenas prácticas de ingeniería de software.

En la figura 1, se muestra el proceso de desarrollo de software de la empresa SYSNET, integrado por los procesos de Modelación del negocio y del producto, Codificación y Pruebas.

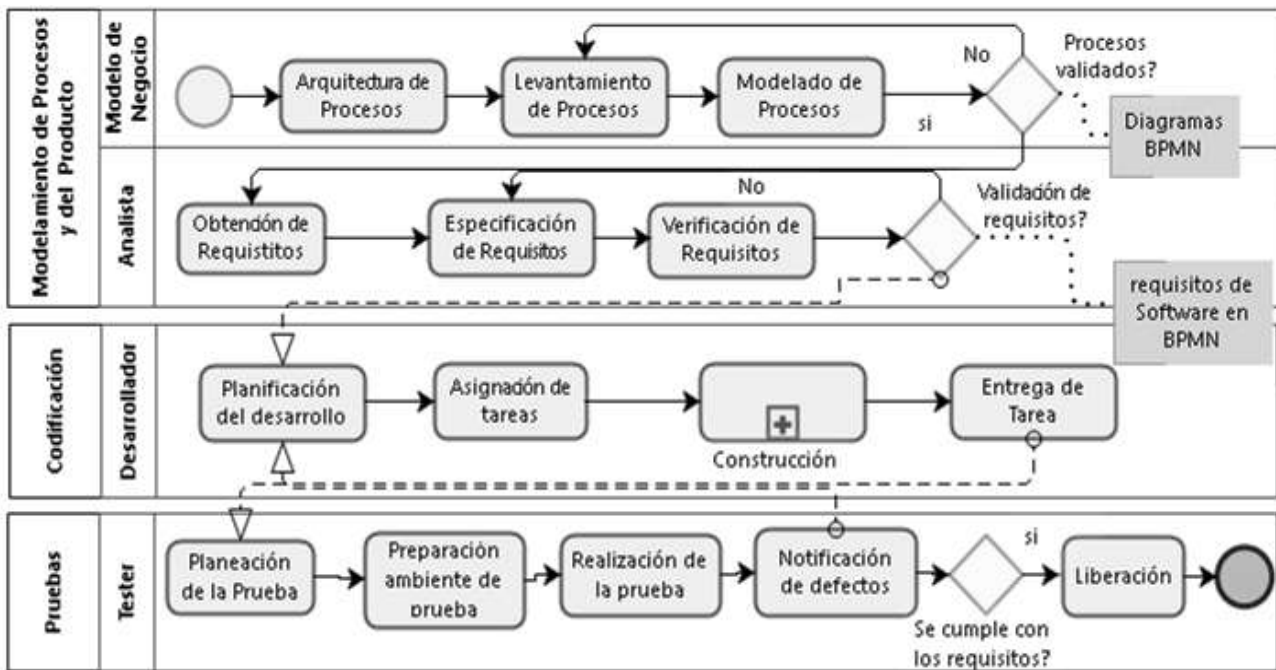


Fig. 1: Diagrama de Proceso de Desarrollo de software de *Software*.

El proceso de modelamiento de procesos y producto, permite descubrir más conocimiento iniciando con la especificación de la arquitectura de procesos, luego el modelamiento de cada proceso y su modelado a partir de diagramas BPMN, lo que permite especificación de requerimientos más acertada, es decir, las captaciones de los requisitos están alineadas con las características y propiedades de los procesos de negocios de la organización cliente. La actividad de Obtención de Requerimientos por parte del analista, se hace de manera sistemática soportándose en fases, plantillas y criterios de verificación del modelo propuesto, con el fin de minimizar el riesgo de fracaso de un proyecto de software. Una vez validados los requisitos se pasa al proceso de Codificación. De esta manera se reduce el tiempo en el análisis y desarrollo de software. En la actividad de Entrega de Tarea del Procesos de Codificación consiste en que el desarrollador notifica la entrega a pruebas por email, en su cuerpo debe contener el nombre y número del requerimiento y tarea de construcción.

La tarea de construcción es la herramienta para gestión de tareas, contiene la ruta de los archivos correspondientes a la entrega; incluyendo el registro Entrega de Codificación con la descripción de los archivos que fueron agregados o modificados. La ruta que se especifica en la entrega debe contener carpetas con (él) o los entregables junto con los SCRIPT de bases de datos. En las entregas de versiones y/o *reléase*, el coordinador de Construcción copia el paquete de despliegue (*Script*, Sitios) en el repositorio indicado para tal fin. La actividad Planeación de la Prueba, consiste en que, una vez recibida la entrega, el verificador actualiza la columna "Pruebas" del formato y procede a diligenciar el registro Programación de Pruebas, consignando la siguiente información: Módulo, Nombre del requerimiento, Tipo de prueba, Estado, Fecha Inicio, Fecha Fin Estimada, Fecha Fin Real como se muestra en la Tabla 1; luego de realizadas las pruebas se actualiza el registro con los resultados obtenidos y la duración real.

Tabla 1: Plan de Pruebas Plataforma Asisto.

Módulo	Nombre del Requerimiento	Tipo de Prueba Funcional	Estado	Fecha Inicio	Fecha Fin Estimada	Fecha Fin Real	Tiempo
Cliente/IPS	Gestión de Usuario y roles del sistema	Interfaz Grafica	Realizada	25/10/16	25/10/16	25/10/16	40 min
Cliente	Portal Cliente-administración	Funcionalidad	Realizada	25/10/16	25/10/16	25/10/16	2 horas
Cliente	Portal cliente-orden de servicio	Funcionalidad	Realizada	25/10/16	25/10/16	25/10/16	3 horas
IPS	Portal IPS-Administración	Interfaz Grafica	Realizada	26/10/16	26/10/16	26/10/16	4 horas
IPS	Portal IPS-Orden de Servicio	Interfaz Grafica	Realizada	26/10/16	26/10/16	27/10/16	4 horas
IPS	Agendamiento	Funcionalidad	Realizada	27/10/16	27/10/16	27/10/16	4 horas

Tabla 2: Requerimientos identificados antes del desarrollo de la plataforma ASISTO.

Requerimientos Funcionales	Descripción
R1. Gestión de usuarios y roles del sistema	Debe permitir la creación de usuarios y configurar los roles del sistemas.
R2. Portal Cliente – Administración	Las empresas clientes de las IPS de salud ocupacional deben configurar y visualizar la información de sus empleados, exámenes que se le practican en cuanto a salud ocupacional, tipos de exámenes, cargos, factores de riesgo de los cargos, departamentos que maneja la empresa, deben definir su profesiograma y configurar los roles y usuarios que tendrán acceso al software.
R3. Portal Cliente – Orden de servicio	Los clientes deben crear órdenes de servicio para la atención de sus empleados y/o candidatos, las cuales serán enviadas a las IPS, para que estas tengan en cuenta los servicios pendientes a realizar. En base a las órdenes de servicios se realizará la asignación de citas, la atención y la respectiva facturación.
R4. Portal IPS-Administración	Las IPS deben configurar y visualizar la información de sus empresas clientes, pacientes, prestadores, proveedores, servicios, cargos, factores de riesgo de esos cargos, exámenes, tipos de exámenes, podrán definir los profesiogramas de sus empresas clientes y configurar los roles y permisos los usuarios que tendrán acceso al software.
R5. Portal IPS – Orden de servicio	Las IPS deben visualizar y gestionar las órdenes de servicios que le emiten sus empresas clientes, ya que en base a estas se realizará la asignación de citas, la atención y la respectiva facturación.
R6. Agendamiento	En el portal IPS se debe estipular su capacidad instalada, asignación de citas a empresas clientes y de esta forma optimizar tiempos y recursos. Las empresas clientes, tendrán en su portal, a través de un menú podrán agendar cita para sus trabajadores y/o candidatos. Para generar el agendamiento se debe cumplir el requisito R5 Orden de Servicio.
R7. Creación de Formularios Dinámicos	Las IPS deben configurar de forma dinámica los registros con información médica de los pacientes durante la atención. Se consignará la información y los resultados, producto de la evaluación médica realizada por el especialista.
R8. Gestión de historias clínicas ocupacionales	En esta especificación las IPS deben tener acceso a los registros médicos de los diferentes exámenes realizados de cada uno de sus pacientes.
R9. Registro de Historia Clínica Ocupacional	En esta especificación los médicos de las IPS deben registrar los resultados de acuerdo al formato de historial clínico de la IPS. Para este requisito existe un prerrequisito R7.
R10. Registro de Visiometría y Optometría	En esta especificación el médico de la IPS debe registrar los resultados de los exámenes de visiometría y optometría aplicados a los pacientes.
R11. Registro De Espirometría	Las IPS en deben registrar los resultados de la evaluación del examen de Espirometría que se realiza los pacientes.
R12. Registro de Audiometría	En esta especificación las IPS deben registrar los resultados de la evaluación del examen de audiometría que se realiza a sus pacientes.
R13. Registros Clínicos Dinámicos	Las IPS deben poder configurar la información clínica obtenida del paciente durante su atención y exámenes de forma dinámica.
R14. Cambios al Certificado de Aptitud	Las IPS deben expedir en el Certificado de Aptitud el cual aparezcan las recomendaciones clasificadas por tipos para que este se interprete mejor.
R15. Crear Remisión a la EPS	Se debe generar una remisión a la EPS por cualquier patología o hallazgo encontrado en la evaluación médica realizada.

La actividad Preparación Ambiente de Pruebas, el verificador organiza los entregables a probar y actualiza los sitios. Los Script se deben correr en la base de datos según su consecutivo y el siguiente orden: Estructura – Procedimiento – Datos. Para la realización de la prueba el verificador debe buscar el registro ERS para leer y entender lo solicitado por el Cliente y el flujo que sigue el requerimiento. Luego se procede a realizar las siguientes pruebas: Funcionalidad, operaciones y/o acciones descritas en el registro Especificación de Requerimientos de Software; Interfaz Gráfica, ortografía, alineación de campos, tamaños en ventanas y alertas utilizadas; Sistemas, funcionalidad en varias Plataformas; Rendimiento, que tanto tardan las operaciones en realizarse; y Regresión, asegura que los cambios realizados no causen un impacto negativo en el resto de la aplicación y que el producto se integra de manera correcta. Para cada una de estas pruebas se aplica una lista de chequeo revisando cada uno de sus ítems y los resultados obtenidos. La notificación de los defectos y/o no conformidades se realiza al desarrollador que hizo la entrega, haciendo uso de un registro donde se relacionan los resultados de las pruebas. Si el requerimiento cumple con cada uno de los requisitos de calidad se procede con su liberación. Finalmente, una vez liberado el producto el funcionario de soporte designado realiza la validación del producto con el Cliente y se asegura que cumpla con todos los requisitos estipulados.

RESULTADOS DE LAS PRUEBAS DEL SOFTWARE ASISTO

En la Tabla 2, se muestran los requerimientos identificados por el grupo de analistas de SYSNET antes del desarrollo de la plataforma ASISTO. En la primera columna se describen los requerimientos funcionales definidos entre el analista y las partes interesadas. Luego en la segunda columna se detalla la función que debe cumplir dentro del software ese requerimiento.

Tabla 3: Lista de chequeo Prueba de Interfaz Gráfica. Fuente: SYSNET Ltda.

<i>Crterios de verificación</i>
¿Se visualiza los controles como activos cuando esto es necesario?
¿En la barra de título aparece el nombre representativo de la interfaz?
¿Primera letra en mayúsculas y las demás en minúscula?
¿Los cuadros de texto están alineados correctamente?
¿La cantidad de caracteres que permite ingresar, es la cantidad estipulada en el requerimiento o en la tabla de base de datos?
¿El ancho es correspondiente con la información que digitará el usuario?
¿Las cajas de texto ubican al lado derecho de las etiquetas?
¿Entre las etiquetas y los cuadros de texto se deja un espacio prudente?
¿El ancho del campo deber ser correspondiente con la información que se va a ingresar?
¿Se ubica al lado derecho de las etiquetas?
¿El formato manejado es de dd/mm/yyyy?
¿Al final de estas ahí dos puntos?
¿Las etiquetas están alineadas correctamente?
¿Las etiquetas están escritas con buena ortografía?
¿El color de las etiquetas es legible?
¿La longitud del combo corresponde a la información a mostrar?
¿Se ubican al lado derecho de las etiquetas?
¿Los combos están alineados correctamente?
¿El nombre del botón primera letra en mayúscula y las demás en minúsculas?
¿Si tiene imagen este debe ir al lado izquierdo del texto?
¿Si el botón tiene que abrir una nueva ventana lo hace correctamente?
¿El tamaño de las grillas es el adecuado para la información que se desea mostrar?
¿La repartición de las celdas que están dentro de la grilla está bien distribuida?
¿Los botones que se encuentran dentro de las grillas cumplen con la función necesaria?
¿Los mensajes brindan información conforme a lo requerido?
¿Se utiliza el icono correspondiente al momento de desplegar los mensajes de información?
¿Se utilizan para alertar al usuario?
¿Se utilizan para brindar información cuando haya ocurrido un error?
¿Al momento de brindar información cuando haya ocurrido un error, el mensaje mostrado es claro, explica dónde y porque se está generando el error o como arreglarlo?
¿Los mensajes desplegados están escritos con buena ortografía?

La tabla 3, muestra la lista de chequeo para verificar y validar la Interfaz Gráfica, la cual incluye criterios para verificar los títulos principales, los cuadros de texto, los campos fecha, las etiquetas, los combos, los botones, las grillas y los mensajes de información.

En la tabla 4, muestra la lista de chequeo de pruebas de Funcionalidad aplicada a cada uno de los requerimientos indicados. La respuesta dada por el proceso de prueba debe ser conforme o no conforme. Para cada una de estas pruebas (de la Tabla 3) se aplica una lista de chequeo revisando cada uno de sus ítems y los resultados obtenidos en la figura 2.

Tabla 4: Prueba de Funcionalidad. Fuente: SYSNET Ltda.

Criterios de verificación
¿Los botones funcionan adecuadamente?
¿Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento?
¿Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente?
¿Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento?
¿Los combos cargan la información adecuada según el requerimiento?
¿Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento?
¿Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado?
¿Cuándo se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error?
¿El requerimiento permite visualizar la ventana o el formulario con los datos precargados?
¿Se muestra un mensaje cuando se ha cumplido una operación?



Fig. 2: Cantidad de errores por prueba de funcionalidad y de interfaz gráfica.

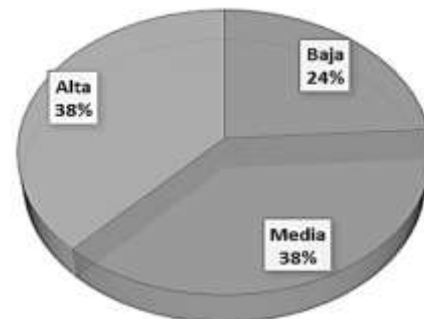


Fig. 3: Resumen de fallo por severidad.

De quince (15) requerimientos, se hallaron siete (7) requerimientos no conformes en las pruebas aplicadas a la Interfaz Gráfica, y un total de veintidós requerimientos no conformes para las pruebas de funcionalidad. En figura 3, se muestran las no conformidades encontradas por severidad, las cuales están clasificados de la siguiente forma: ALTA, falla del sistema que no completa su proceso funcional o el sistema no entrega la salida esperada por el usuario; MEDIA, falla del sistema que muestra su parte funcional, pero con datos incorrectos (datos de salida inconclusos); BAJA, falla del sistema superficial como los de interfaz que son notorios por el usuario pero que no incurren en el funcionamiento del sistema.

En total se encontraron once (11) fallos de severidad Alta, once (11) de severidad media y siete (7) con severidad baja. Se puede decir que de acuerdo a los resultados obtenidos no se puede liberar el producto al cliente hasta que no se corrija los fallos funcionales y de interfaz de acuerdo al nivel de severidad. Después que se evidencia los resultados de las pruebas, se envían a los desarrolladores las no conformidades encontradas en el software, con los requisitos que no se cumplieron con el objetivo de corregir el error en el código. Durante la liberación, se verifica por medio de pruebas, si se corrigieron los errores y el producto no presenta fallas en cada uno de los requisitos, se procede a liberar el producto al cliente. En la validación del producto, se procede a la verificación con el cliente y se asegura que cumpla con los requisitos estipulados del software.

Existe documentación relacionada con las pruebas del software y calidad del software como TMM, TMMI, TPI, TMAP y CMMI; pero en estos no se detallan los procesos, las actividades y la manera correcta de implementar las pruebas del software a un producto para múltiples clientes (4 IPS). A continuación, se relacionan algunos artículos donde se describen aspectos relacionados al proceso de prueba de software:

León Perdomo, Góngora Rodríguez y Febles Estrada (2013), el departamento de pruebas del software (DPSW), describe un enfoque a las pruebas exploratorias (PE) el cual explica el conceptos de las pruebas exploratorias y las métricas de calidad de estas, muestra los resultados de dos proyectos A y B en el cual se le hacen pruebas exploratorias concluyendo que el proyecto A puede pasar a la primera iteración de las pruebas de liberación; el procedimiento de pruebas de SYSNET se enfoca directamente a las pruebas de liberación del producto; Rojas-Montes, Pino-Correa y Martínez (2015), se especifica la adaptación de las técnicas de pruebas funcionales en la fase de diseño y descripción de las actividades, ajustadas a pequeñas empresas desarrolladoras de software que no están certificadas bajo un estándar de calidad del software o en proceso de certificación, en el cual describe los resultados obtenidos como la consolidación de los procesos de pruebas de software en la organización, los procesos de pruebas de SYSNET son similares, con la diferencia que la empresa SYSNET es categorizada como mediana empresa y sus procesos tiene un nivel de madurez tres definidos según CMMI; Callejas-Cuervo y otros (2017), presenta el estado de la cuestión, de los modelos de calidad del software, realiza una comparación de diferentes modelos, la estructura de cada modelo de calidad y sus enfoques, pero no describe un estándar de calidad de pruebas del software; Monroy, Arciniegas y Rodríguez (2017) hacen énfasis en la ingeniería inversa como una excelente herramienta de control de calidad al momento de verificar la trazabilidad del código fuente y los modelos de diseño, los resultados obtenidos en las pruebas de SYSNET, figuras 2 y 3, muestran la trazabilidad del código fuente con los modelos de diseño y los requerimientos del producto, por medio de las pruebas de funcionalidad y de interfaz gráfica. Estos trabajos servirán de insumo para futuros proyectos de mejora de los procesos de prueba en la empresa, una vez se evalué la posibilidad de aplicar algunas de las propuestas metodológicas plasmadas en ellos.

CONCLUSIONES

Hay muchos factores que determinan el éxito y fracaso de un proyecto de software; conocerlos es de interés de la industria de software para el desarrollo exitoso del producto; por tal motivo las pruebas del software son un proceso muy importante dentro del ciclo de desarrollo del software.

Las pruebas deben ser exactas, tienen que demostrar que su descripción se puede probar. Las Pruebas deben ser confiables y repetibles, se debe obtener el mismo resultado cada vez que se ejecute, sin importar que se pruebe. Las pruebas deben ser rastreables, hay que conocer que requisitos se prueban y revisar que el producto no haga lo que no se espera. Dentro del proceso de prueba aplicado se identificaron fallas del sistema catalogadas con severidad alta, media y baja; estas fallas son identificadas tanto en la entrega y presentación de la información.

Actualmente en los proyectos se manejan unos tiempos de entregas el cual hay que cumplir la entrega del producto, y en muchos casos omiten el proceso de pruebas del producto con el fin de cumplir con la fecha de entrega, por esta razón resulta imprescindible contar con la ayuda de herramientas para la gestión de procesos de pruebas del software, las cuales ayudan a verificar el avance del proceso por medio de generación de informes.

El proceso de pruebas del software de la empresa SYSNET se basa en la integración de la norma ISO 9001 y el modelo CMMI el cual, valida las especificaciones funcionales evaluando los riesgos y realizando ciclos de pruebas. La plataforma ASISTO como caso de estudio permitió establecer el alcance de las pruebas, la calidad del producto y reducir tiempo y costos en mejorar los requerimientos. En la actividad de Recepción el desarrollador notifica cuando cada módulo del producto se encuentra conforme para la fase de pruebas. En la fase de plan de pruebas, preparación del ambiente y resultados de las pruebas, se establece la magnitud de las pruebas y su tiempo de duración por cada requerimientos de las cuatro IPS, los resultados de las pruebas permiten a la empresa conocer el porcentaje de cumplimiento de la entrega del producto; de acuerdo a los resultados de severidad, la notificación de los defectos y/o no conformidades, permite crear un repositorio de no conformidades y la solución de la fallas de acuerdo a los resultados de las pruebas por proyecto. Para las pruebas se diseñó una lista de chequeo que permite medir la trazabilidad entre los requisitos y los casos de pruebas generados. Se recomienda a la empresa SYSNET aplicar en su proceso de Pruebas del Software el estándar de TMMI como una extensión a su modelo de desarrollo del software ya sus procesos están enfocados al estándar CMMI, la articulación se logrará sin ningún tipo de inconveniente.

Este caso debe considerarse como un referente de buenas prácticas de Ingeniería de Software para la verificación y validación de requerimientos a través de las pruebas del producto y facilitar el desarrollo de las aplicaciones y el aseguramiento de la calidad del producto.

REFERENCIAS

- Aristegui, J.L., Los casos de prueba en la prueba del software, ISSN: 2145-4086 (en línea), Lámpsakos, (3), 27–34, 2010. <https://goo.gl/3zRi6X>. Acceso: 19 de mayo (2017)
- Callejas-Cuervo, M.; A.C. Alarcón-Aldana; A.M. Álvarez-Carreño; M. Callejas-Cuervo; A. C. Alarcón-Aldana y A.M. Álvarez-Carreño, Software quality models, a state of the art, doi: 10.18041/entramado.2017v13n1.25125, Entramado (en línea), 13(1), 236–250 (2017)
- Capote-García, T.; Y. Pérez-Moreno; R. Yzquierdo-Herrera; A. Febles-Estrada y V. Estrada-Sentí, Perspectivas del Cuadro de Mando Integral personalizadas para laboratorios de pruebas de software, Ingeniería Industrial, ISSN: 1815-5936 36 (en línea), (2), 138-150, 2015. <https://goo.gl/u3D8B5>. Acceso: 6 de abril (2017)
- Castilla, R.C.; O. Barrera; L.G. Fernández; M. Cabrera y L. González, Proceso de pruebas y suite de herramientas de soluciones informáticas para la salud, ISSN: 1684-1859 (en línea), Revista Cubana de Informática Médica, 7(1), 56–72, (2015). <https://goo.gl/q7dtN1>. Acceso: 12 de mayo (2017)
- Escobar-Sánchez, M.E. y W.M. Fuertes-Díaz, Modelo formal de pruebas funcionales de software para alcanzar el Nivel de Madurez Integrado 2, ISSN: 0121-1129 (en línea), Revista Facultad de Ingeniería, 24(39), 31-42, 2015. <https://goo.gl/Cbkeuj>. Acceso: 8 de junio (2017)
- Galvis-Lista, E. y J.M. Sánchez-Torres, A critical review of knowledge management in software process reference models, ISSN: 1807-1775 (en línea), JISTEM - Journal of Information Systems and Technology Management, 10(2), 323-338, 2013. <https://goo.gl/hf7sTQ>. Acceso: 24 de mayo (2017)
- García-Mireles, G.A.; M.A. Moraga; F. García y M. Piattini, The Influence of Process Quality on Product Usability: A Systematic Review, ISSN: 0717-5000 (en línea), CLEI Electronic Journal, 16(2), 6-6, 2013. <https://goo.gl/cKE6kv>. Acceso: 14 de abril (2017)
- González, L., Método para generar casos de prueba funcional en el desarrollo de software, ISSN: 1692-3324 (en línea), Revista Ingenierías Universidad de Medellín, 8(15), 29–36, 2009. <https://goo.gl/ixdsZg>. Acceso: 25 de abril (2017)
- León Perdomo, Y.; E. A. Góngora Rodríguez y A. Febles Estrada, Aplicando métricas de calidad a proyectos y procesos durante las pruebas exploratorias, ISSN: 2227-1899 (en línea), Revista Cubana de Ciencias Informáticas, 7, 193-205, 2013. <https://goo.gl/SP12Nb>. Acceso: 17 de mayo (2017)
- Marculescu, B.; S. Poulding; R. Feldt; K. Petersen y R. Torkar, Tester interactivity makes a difference in search-based software testing: A controlled experiment, doi: 10.1016/j.infsof.2016.05.00, Information and Software Technology (en línea), 78, 66-82, (2016)
- Monroy, M. E.; J. L. Arciniegas y J. C. Rodríguez, Caracterización de los Contextos de Uso de la Ingeniería Inversa, doi: 10.4067/S0718-07642017000400010, Información Tecnológica, 28(4), 75–84 (2017)
- Myers, G. J.; C. Sandler y T. Badgett, The art of software testing, 3ª Ed., John Wiley & Sons, Inc., Ontario, Canada, 32-33 (2011)
- Rojas-Montes, M.L.; F.J. Pino-Correa y J.M. Martínez, J.M., Proceso de pruebas para pequeñas organizaciones desarrolladoras de software, ISSN: 0121-1129 (en línea), Facultad de Ingeniería, 24(39), 55-70, 2015. <https://goo.gl/Jsj7mh>. Acceso: 24 de junio (2017)
- Sanz, L. F., Un sondeo sobre la práctica actual de pruebas de software en España, ISSN: 1885-4486, Revista Española de Innovación, Calidad e Ingeniería del Software, 1(2), 50, 2005. <https://goo.gl/BtfRbj>. Acceso: 21 de mayo (2017)
- Schots, N.C.; R. Figueiredo; T. Guidini; R. de Holanda y A. Regina, A Knowledge-based Environment for Software Process Performance Analysis, ISSN: 0717-5000, CLEI Electronic Journal, 18(2), paper 4, agosto de 2015. <https://goo.gl/s4UtgB>. Acceso: 17 de abril (2017)