

Proposed Application Of Testing Based On BS 7925-2 Standard For Degree Projects Focused On Development Software

Jhonatan Álvarez Caicedo, Jhonny Garcés Tabares, Marta Cecilia Camacho

Institución Universitaria Colegio Mayor del Cauca

Popayán, Cauca

Jalvarez@unimayor.edu.co

jgarces@unimayor.edu.co

cecamacho@unimayor.edu.co

Abstract— Despite the importance of the tests are in the process of development and therefore in the software industry, there is a conceptual gap on testing by developers and engineers who must implement them. The expert staff in this area is scarce, software testing becomes an optional exercise, casual or even not documented in many companies, this situation arises from the universities where the scope of projects includes the development of no evidence so timely and formal. This paper presents an easy and practical proposal to apply the tests in development projects and projects Grade option in IUnimayor, the workflow focuses on unit tests and acceptance tests selected according to a previous diagnostic tests. The thread specified in SPEM process modeling language, enabling a specification deliver HTML readily available and applicable development projects.

Keywords-component; software engineering, process engineering, software testing, process model, testing process

I. INTRODUCCIÓN

Las pruebas de software es un proceso de verificación del software mediante el uso de un conjunto de herramientas, técnicas y métodos que verifican el correcto desempeño de un programa, herramienta o aplicación software [1]. Las técnicas para encontrar errores en un programa son extensamente variadas y van desde el uso del ingenio por parte del personal de prueba hasta herramientas automatizadas, el auge de diversos métodos, técnicas y herramientas para la aplicación de pruebas, hizo evidente la necesidad de unificar y regular la aplicación de pruebas, mediante estándares como son la norma británica BS 7925 [2] y la IEEE 1008, IEEE 829 [3], estas normas permiten dar guía al equipo de pruebas en un correcto proceso que asegure mediante buenas prácticas la calidad de la aplicación desarrollada.

En un estudio sobre las tendencias competitivas del software realizado por Castrillon y Giraldo [4], se plantea que las pruebas de software son una de las tendencias ocupacionales en la industria del software, sin embargo Parra [5] plantea una discusión sobre la incorporación adecuada de los conceptos y competencias requeridas para la aplicación de pruebas (testing) en los planes de estudio de la ingeniería de software.

Los estándares de calidad buscan por medio de la aplicación de buenas prácticas en el proceso de pruebas de software lograr un aseguramiento de la calidad en el producto entregado. A pesar de la importancia de las pruebas en la industria del software, existe un vacío conceptual de ellas por parte de los desarrolladores, e ingenieros que deben aplicarlas, el personal experto en dicha área es escaso, las pruebas de software se convierten en un ejercicio opcional, informal o no documentado, esta situación se da incluso desde las universidades donde el alcance de los proyectos no abarca el desarrollo de pruebas de manera puntual y formal [6]. Este proyecto presenta una propuesta fácil y práctica de aplicar las pruebas en los proyectos de desarrollo de software a desarrollar como opción de proyectos de grado para optar al título de desarrollador de software en la Institución Universitaria Colegio Mayor del Cauca, el flujo de trabajo de pruebas se enfoca a las pruebas unitarias y pruebas de aceptación, necesidad que fue detectada del análisis realizado a los proyectos de desarrollo realizados por estudiantes de último semestre del programa. El subproceso se especifica en SPEM lenguaje de modelado de procesos, lo que permite entregar una especificación en HTML fácilmente disponible y aplicable por los proyectos de desarrollo.

Una prueba de software involucra la ejecución de determinadas operaciones del sistema, bajo condiciones controladas y evaluando los resultados. Las condiciones controladas puede ser usuales o irregulares, además la prueba puede intencionalmente forzar al programa y verificar qué errores se producen en las respuestas bajo condiciones críticas, se busca determinar si los sucesos ocurren cuando tendrían que ocurrir, si el tiempo de respuesta es incrementado considerablemente, o qué módulos no cumplen con los requisitos previamente establecidos, en general busca verificar todas las especificaciones y requerimientos del software, a partir de estos resultados el equipo de desarrollo toma las decisiones necesarias para determinar si el software ha sido desarrollado de forma correcta mediante la evaluación de criterios previamente determinados, en caso de obtener un resultado no satisfactorio es necesario evaluar la forma en que se desarrolló el software en búsqueda de la solución al problema, este proceso se realiza de forma iterativa comunicando y evaluando los posibles cambios entre el equipo de desarrollo. La mayoría de las grandes empresas productoras de software asumen la responsabilidad del control de calidad y prueba de sus productos, incluyendo en su equipo de trabajo un grupo dedicado exclusivamente a las pruebas de software, los cuales mediante un trabajo en conjunto determinan las condiciones necesarias para el

debido cumplimiento del estándar de calidad de sus productos SQA (Software Quality Assurance) [1].

El proyecto fue abordado en cinco fases, en primer lugar se realizó una búsqueda bibliográfica para elaborar el marco teórico soporte del proyecto, incluyendo una revisión de la normatividad general sobre las pruebas de software especialmente de la norma británica BS 7925-2 con el fin de establecer elementos para especificar el proceso de pruebas; Se realizó un estudio referente a la aplicación de pruebas en los proyectos de grado presentados en los últimos tres años agregando un cuarto año para la evaluación de proyectos desarrollados con la metodología UP, para poder evaluar el impacto se realizó un estudio evaluando el desarrollo de pruebas desde cuatro indicadores tipos de pruebas, documentación, hallazgos, y momento de aplicación de las pruebas. Basado en el diagnóstico, el proceso de desarrollo más seguido, la base teórica y la norma británica el grupo selecciona realizar el proceso enfocado en las pruebas unitarias y pruebas de aceptación, en la cuarta fase se especifica y modela un proceso de pruebas en SPEM en EPF Composer, la correctitud del modelo de proceso fue evaluado con la herramienta AVISPA.

II. ESTUDIO SOBRE LA APLICACIÓN DE PRUEBAS EN LOS PROYECTOS DE GRADO

Para poder identificar los procesos y elementos prioritarios dentro del proceso de pruebas, se hizo necesario la realización del estudio sobre la aplicación de pruebas en los proyectos de grado. El propósito de este estudio fue caracterizar la aplicación de pruebas dentro de los proyectos desarrollados en la Institución Universitaria Colegio Mayor del Cauca en la facultad de ingeniería, para lograr responder a las siguientes preguntas ¿Qué tipo pruebas se realizan dentro de los proyectos de grado enfocados al desarrollo de software en la facultad de ingeniería de la Institución Universitaria Colegio Mayor del Cauca?, ¿Qué documentación existe del proceso de pruebas realizado dentro de cada una de las monografías?, ¿La documentación permite evidenciar los hallazgos negativos?. ¿Existe evidencias de retroalimentación al realizar el proceso de pruebas de software en el proyecto?

Para dar respuesta a la pregunta de investigación, ¿los proyectos de grado aplican pruebas dentro de sus procesos de desarrollo de software? se identificaron las siguientes variables:

Documentación de pruebas: Las monografías permiten evidenciar la aplicación de pruebas dentro de los proyectos de grado.

Plan de pruebas: Determinar si los proyectos de grado inician su proceso de pruebas a partir de un plan de pruebas debidamente definido.

Resultados: Determinar si los proyectos documentan resultados posteriores a la ejecución de las pruebas de software utilizadas.

Número de hallazgos negativos documentados: Identificar la relación entre el número de hallazgos negativos encontrados y la ejecución de pruebas en cada proyecto de grado.

Número de soluciones documentadas encontradas: Determinar mediante el número de soluciones establecidas por proyecto la correcta realización de pruebas de software.

La población de este estudio, son las monografías de los proyectos enfocados al desarrollo de software presentados entre los años 2009 a 2012. Esta información se obtuvo dentro de la biblioteca Jaime Macías de la Institución Universitaria Colegio Mayor del Cauca en total se encontraron 27 proyectos de los cuales se tomaron 21 proyectos que cumplían el enfoque de desarrollo de software.

Los resultados obtenidos permitieron establecer que los procesos más seguidos son XP y UP, el estudiante que realiza su proyecto de grado en desarrollo de software encuentra dificultades para documentar fácilmente los resultados de las pruebas realizadas, se identificó un desconocimiento de los estándares relacionados con

pruebas de software. Además se estableció que las pruebas más necesarias son las pruebas de aceptación dado el número de proyectos que se realizan para empresas e instituciones, también las pruebas de unidad y de integración. Por medio de este estudio se decidió enfocar un proceso de pruebas empleable sobre la metodología XP, y enfocado a pruebas de unidad y de aceptación.

III. NORMATIVA PARA EL DISEÑO DE CASOS DE PRUEBA

Para cimentar el proceso de pruebas fue necesario escoger una base que nos permitiera formalizar el proceso de pruebas de software para ello se decidió utilizar la norma de la sociedad británica de computación BS 7925-2 Standard for Software Component Testing, el objetivo de esta norma es permitir la medición y comparación de las pruebas realizadas sobre componentes de software, permitiendo a los usuarios de esta norma mejorar directamente la calidad en la ejecución de sus pruebas de software, el estándar describe un número de técnicas para el diseño de casos de prueba y su medición soportando el proceso de pruebas de software, tomando como base el proceso definido dentro del estándar que establece las tareas de planeación, especificación, ejecución, análisis y validación de los componentes de prueba. La decisión de tomar como base esta normativa en pruebas aparece como una respuesta a futuro donde el proceso mantenga coherencia con la futura norma de pruebas general ISO-IEC 29119 que se encuentra en evaluación, tiene como objetivo cubrir todo el ciclo de vida de las pruebas de sistemas software incluyendo los aspectos relativos a la organización, gestión, diseño y ejecución de las pruebas, teniendo como base estándares IEEE y BSI sobre pruebas de software [16], que se espera en un futuro sea la norma general para formalización y estandarización de procesos de prueba, esta norma usa como base para el diseño de casos de prueba la norma británica BS 7925-2 por lo cual la migración del proceso sería mucho más natural. El enfoque del estándar de diseño de casos de prueba permite establecer las características del proceso de pruebas de software mediante la descripción de técnicas para el diseño y la medición de casos de prueba que permitan apoyar el proceso de desarrollo de pruebas de software [2] [10]. La norma además define ciertas técnicas de diseño de casos de prueba que facilitan la elaboración en la práctica del desarrollo de software.

I. TRABAJOS RELACIONADOS

A continuación se relacionan trabajos que han enfocado a la especificación de procesos de pruebas; también fue necesario consultar otras referencias buscando conocer las principales actividades, las estrategias y enfoques del proceso de pruebas en el ciclo de vida del desarrollo del software [7].

A. Proceso de pruebas para pequeñas organizaciones desarrolladoras de software

Este documento presenta un proceso liviano definido para soportar y guiar las pruebas en pequeñas organizaciones desarrolladoras de software. Con la incorporación de técnicas de pruebas funcionales, se hacen más llevaderas las actividades requeridas para evaluar un software de manera organizada y sistemática, especialmente aquellas que tienen que ver con el diseño y ejecución de las pruebas. Este trabajo ha sido desarrollado a partir del análisis de los procesos de pruebas existentes en los modelos de referencia y en la literatura, y ha sido exitoso tras su ejecución en una pequeña empresa de desarrollo software [8].

B. Proceso de testing funcional independiente.

Define un proceso de prueba funcional de un producto de software, teniendo como objeto validar el comportamiento del producto y verificar si cumple o no con sus especificaciones. El proceso se denomina Protest a ser empleado en el laboratorio de pruebas del

Centro de Ensayos de Software (CES). Protest es validado en un caso de estudio en una aplicación de gestión [9].

C. Propuesta metodológica para la realización de pruebas de software en ambientes productivos.

Plantea un método para la integración de las pruebas en el ciclo de desarrollo de software enfocado a integrar el proceso de pruebas en todo el ciclo productivo de software [18].

D. Procedimiento para la aplicación de pruebas de unidad

Describe un proceso de pruebas de unidad a nivel de clases, de software orientado por objetos, busca plantear una estrategia que cubra un numero representativo de casos de prueba y que sin embargo sea el más pequeño posible, y se plantea que se emplearía en proyectos de grado sin embargo no logro encontrarse resultados reportados de esas experiencias [19].

II. PROCESO DE PRUEBAS IUNIMAYOR

La especificación del proceso de pruebas IUnimayor busca identificar los elementos del proceso, tareas a desarrollar, roles y productos de trabajo, definiendo un flujo de trabajo (workflow), de tal forma que se brinde una guía a los estudiantes de proyectos de grado enfocados al desarrollo de software, en la aplicación de pruebas.

El proceso de pruebas IUnimayor tiene como objetivo facilitar la correcta ejecución de pruebas y validación de los proyectos de desarrollo de software realizados por los estudiantes en su trabajo de proyecto de grado, ofreciéndoles herramientas fáciles de utilizar sustentadas en un estándar de procesos en pruebas de software.

El Proceso Pruebas IUnimayor es un modelo de proceso que busca plantear lineamientos para la aplicación de pruebas en proyectos de desarrollo de productos software como opción de proyecto de grado, este proceso busca ser simple y liviano, combinando las exigencias de la aplicación de pruebas con las características de los grupos de desarrollo de proyectos de grado (recursos limitados, poca experiencia, tiempos limitados) en modalidad desarrollo de software, identificando prácticas y artefactos mínimos necesarios para la aplicación de pruebas unitarias [11] y pruebas de aceptación [11]. Se determinó que la mayoría de los proyectos siguen lineamientos del proceso programación extrema, para lo cual se utilizan las fases del desarrollo del proceso de programación extrema para acoplarlas junto con las tareas y los artefactos de entrada definidos para las pruebas de unidad y pruebas de aceptación [17].

Se identificó un flujo de pruebas general, y un flujo independiente para cada uno de los dos tipos de pruebas, el objetivo de la división en flujos independientes es permitirle al estudiante la elección de las pruebas a realizar según sus necesidades, además de esto la autonomía de los subprocesos de pruebas disminuye la complejidad y curva de aprendizaje del proceso facilitando su aplicación.

El proceso de pruebas Iunimayor, se presenta en tres fases planificación, iteraciones y entrega, para las cuales presenta tareas, roles y artefactos que permiten integrar y realizar las pruebas unitarias y de aceptación en el proceso de desarrollo, también se realizó la especificación de manera separada, permitiendo seleccionar si usar el proceso completo o uno de los subprocesos como se explicara en las siguientes secciones. La figura 2, corresponde a las tareas propuestas para la fase de planeación del proceso de pruebas.

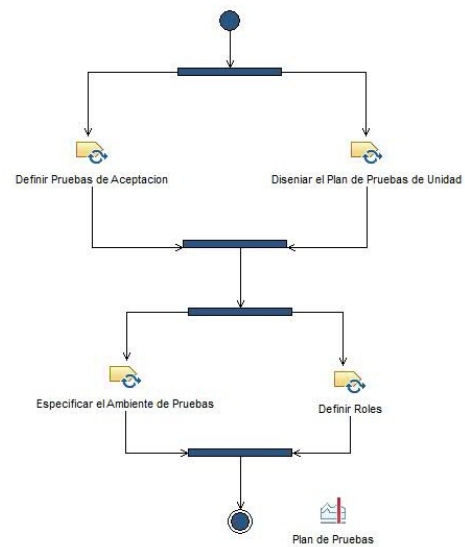


Figura 1. Fase de Planeación del proceso de pruebas Iunimayor.

III. SUBPROCESO PRUEBAS UNITARIAS

Dentro de los hallazgos encontrados a partir del estudio realizado se determinó que las pruebas unitarias son esenciales dentro de los productos desarrollados, tomando como evidencia la tendencia vista al realizar el análisis de las pruebas ejecutadas dentro de los proyectos evaluados. La especificación de un subproceso de pruebas unitarias busca dar las herramientas suficientes para establecer el correcto funcionamiento de los módulos evaluados.

El objetivo del modelado del subproceso de pruebas unitarias es identificar las principales tareas de pruebas unitarias dentro del flujo del proceso de la programación extrema, necesarias para planificar, ejecutar y documentar las pruebas unitarias dentro del proceso de desarrollo de software realizable por los estudiantes de proyecto de grado. El proceso de pruebas unitarias esta descrito en tres flujos de trabajo, a continuación en la figura 1 y en la figura 2 se presenta los dos primeros flujos de trabajo propuestos para las pruebas unitarias, donde la figura 1 nos presenta las tareas a realizar dentro de la fase de planificación y a continuación el flujo de trabajo para el diseño de casos de pruebas unitarias dentro de la fase plantear pruebas unitarias. Dentro de cada tarea se especifican los pasos a seguir además de los artefactos de entrada con sus respectivos ejemplos basado en plantillas las cuales se definieron a partir de los diferentes métodos de diseño planteados dentro del estándar BS7925-2 y diferentes datos para lograr mantener un correcto control de versiones de cada documento, generando las diferentes salidas que se requieran en cada flujo de trabajo, la figura 2 resume los elementos de proceso propuestos para el subproceso de pruebas unitarias, las figuras 3 y 4 presentan los flujos de trabajo.

Roles	Jefe de Proyecto Tester Analista	Diseñador Tester	Programador Tester Jefe de Proyecto
Entradas	Anteproyecto Historias de usuario	Diagramas de diseño Especificación de Requisitos Plan de pruebas de unidad Historias de usuario	Formato de Casos de Prueba Plan de pruebas de unidad Historias de usuario Ambiente de Pruebas
Fases	Planificación		Iteraciones
Tareas	Diseñar el Plan de Pruebas de Unidad Especificar el Ambiente de Pruebas Definir Roles	Diseñar Caso de Prueba	Escribir Caso de Prueba Ejecutar los Resultados de Pruebas Reporte de errores de Pruebas
Salidas	Plan de Pruebas de unidad	Plan de Pruebas de unidad Formato de casos de prueba	Formato de Casos de prueba Formato de Resultados de Pruebas

Figura 2. Elementos del subproceso de pruebas unitarias.

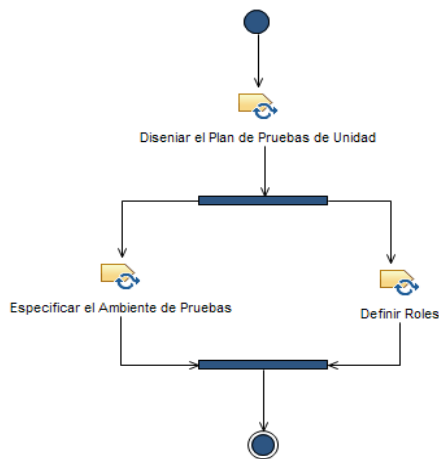


Figura3 Flujo de trabajo fase de planificación en pruebas unitarias

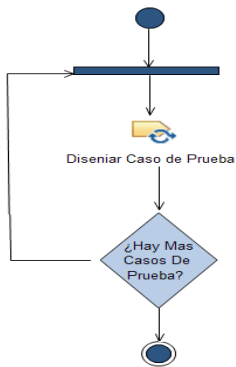


Figura 4 Flujo de trabajo fase de planificar pruebas unitarias

IV. SUBPROCESO PRUEBAS DE ACEPTACIÓN

De igual modo se encontró una necesidad particular en la ejecución de pruebas de aceptación dentro del desarrollo de productos software evidenciado mediante el estudio realizado, este hecho se presenta en gran medida debido a la necesidad de los estudiantes de validar el producto desarrollado por un cliente o en el caso particular del proceso de pruebas IUNIMAYOR, un jurado evaluador. Así el producto desarrollado logra satisfacer el comportamiento desde la perspectiva del cliente bajo un escenario de utilización del sistema esperado.

Las pruebas de aceptación buscan determinar el grado de aprobación de un software desarrollado teniendo como base una especificación y requerimientos, el proceso de pruebas de aceptación como parte del ciclo de vida de XP busca dar al estudiante las tareas y artefactos mínimos necesarios para validar el proceso de desarrollo de software mediante las pruebas de aceptación.

El objetivo del modelado del sub proceso de pruebas de aceptación es detallar las tareas requeridas para planificar, ejecutar y documentar el proceso de pruebas de aceptación, esto facilita la tarea al estudiante de proyecto de grado enfocado al desarrollo de software de presentar y lograr la aprobación en casos donde el software requiere ser evaluado y validado por un cliente o por un jurado evaluador.

El objetivo de las pruebas de aceptación es demostrar al cliente que un producto software cumple con los requisitos estipulados, consiste en formalizar la aceptación de los productos entregables de un proyecto que ha finalizado, para asegurarse que se ha completado satisfactoriamente y obtener la aceptación formal del producto. Las pruebas de aceptación representan los intereses del cliente,

proporcionan confianza de que la aplicación software cuenta con todas las características necesarias y solicitadas y que se comporta de correctamente. El subproceso de pruebas de aceptación está caracterizado por dos flujos de trabajo como se muestran en la figura 5 y en la figura 7, donde cada flujo muestra el orden de ejecución de tareas en las fases del proceso de desarrollo del software para poder obtener los artefactos de salida.

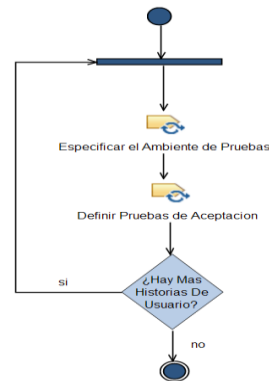


Figura 5 Flujo de trabajo fase de planificación – pruebas de aceptación

La figura 6 presenta un resumen de los elementos de proceso en la fase de pruebas para el subproceso pruebas de aceptación, donde puede apreciarse los productos de entrada y salida, y los roles participantes en cada una de las tareas planteadas.

Roles				
Entradas	Convenio de desarrollo Formato de Resultados de pruebas Historias de usuario Plan de pruebas aceptación	Lista de chequeo del producto Plan de pruebas de aceptación	Historias de usuario Plan de pruebas aceptación Lista de chequeo del producto	Lista de chequeo del producto
Tareas				
Salidas	Lista de Chequeo del producto 	Lista de chequeo del producto 	Lista de chequeo del producto 	Reporte de cambios

Figura 6 Elementos en la fase de pruebas de aceptación

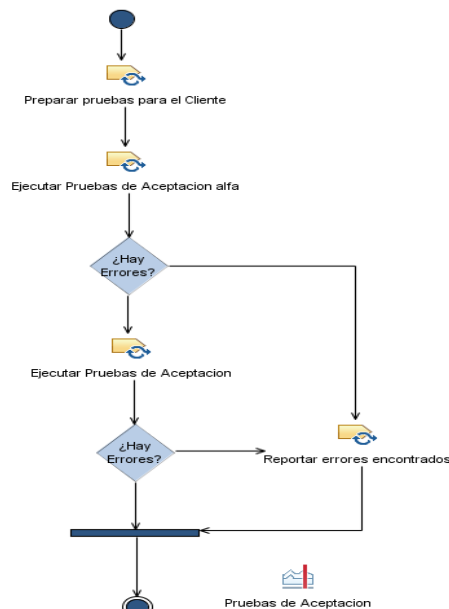


Figura 4 Flujo de trabajo fase de pruebas – pruebas de aceptación

Además de la descripción de cada una de las tareas el proceso de pruebas IUnimayor, se especifica los roles, entradas y salidas de cada una de las tareas que intervienen dentro de las fases de planeación y ejecución de pruebas de software.

V. EVALUACIÓN DE LA CORRECTITUD DEL MODELO DE PROCESO.

La especificación de un modelo de proceso es una tarea en que pueden cometerse errores, por lo cual es necesario evaluar la correctitud del modelo de proceso propuesto, para hacerlo se recurrió a AVISPA, es una herramienta que analiza gráficamente aspectos de un modelo de proceso y que expone posibles errores cometidos en la especificación del proceso con indicadores intuitivos y comprensibles [12]. Existen un conjunto de errores recurrentes en las especificaciones de un proceso de software; los desarrolladores de AVISPA los han clasificado como patrones de error. Los patrones de error establecidos por avispa presentan oportunidades de mejora en la especificación de un proceso de desarrollo de software [12] [13].

La evaluación de la correctitud del modelo de proceso con AVISPA se realizó sobre diferentes versiones del modelo, permitiendo mejorar la especificación. La figura 5 presenta los resultados obtenidos al evaluar dos versiones diferentes, la vista corresponde productos de trabajo, puede observarse que en la imagen de la izquierda aparecen resaltados con color azul 5 productos basura, mientras en la imagen de la derecha ya se ha logrado mejorar la especificación por lo cual no aparecen productos resaltados.

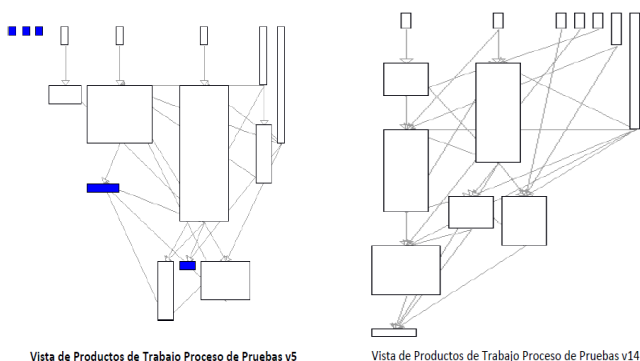


Figura 5 Resultados obtenidos con AVISPA

La tabla 1 presenta los hallazgos obtenidos para cada patrón de error en las diferentes versiones del proceso, donde puede observarse las correcciones realizadas.

	Patrón de error	Proceso de Pruebas v5	Proceso de Pruebas v13	Proceso de Pruebas v15
P1	Rol Aislado	2	0	0
P2	Rol sobrecargado	1	1	1
P3	Sub proyectos independientes (roles)	2	0	0
P4	Tarea multipropósito	3	2	2
P5	Sub proyectos independientes (tareas)	0	0	0
P6	Producto de trabajo basura	5	2	0
P7	Producto de trabajo sobre demandado	1	1	0
P8	Sub proyectos independientes (productos de trabajo)	3	1	0

Tabla 1 Patrones de error vs hallazgos versiones proceso

El análisis de la correctitud del modelo de proceso permitió mejorar la especificación del proceso. La última versión presenta aún un rol sobrecargado que se concluyó que no era un error de modelamiento sino que respondía al hecho de ser una parte de la totalidad del proceso de desarrollo.

VI. EVALUACIÓN DE LA USABILIDAD DEL MODELO DE PROCESO

La correctitud de un modelo de proceso no es la única variable de evaluación, también debe evaluarse que tan fácil es seguir el modelo y que tan útil resulta [15] [16]. Para evaluar la usabilidad del modelo de proceso se realizaron dos casos de estudio, siguiendo el método de estudios de caso propuesto por Runeson y Höst [14].

El primer caso de estudio se realizó en una charla de pruebas con la participación voluntaria de 20 estudiantes de diferentes semestres de la tecnología en Desarrollo de Software y de Ingeniería Informática, la duración del ejercicio fue de tres horas, durante la primera hora se realizó una capacitación general sobre pruebas, en la segunda hora se presentó el modelo de proceso de ejecución de pruebas y los formatos incluidos como plantillas para la ejecución de las tareas de cada flujograma presentado, además se incluyó un ejemplo para cada tarea del subproceso de unidad o aceptación, para finalizar se planteó un taller donde se entregaba el código, objetivo de las pruebas y las plantillas. El objetivo del ejercicio era verificar que tan comprensibles eran los formatos propuestos (plantillas). Gracias al ejercicio realizado, las observaciones realizadas por los participantes como el análisis realizado de las plantillas del proceso de pruebas utilizadas en el ejercicio, los formatos fueron mejorados, además se decidió incluir ejemplos con las plantillas diligenciadas, e indicaciones para su empleo.

El segundo caso de estudio se realizó dentro del proyecto final de una materia de Ingeniería de software en el programa Tecnología en Desarrollo de Software, con la participación de 16 estudiantes de tercer semestre, los cuales debían realizar el proceso completo de desarrollo de software de un producto software de mediana complejidad como proyecto final. Los estudiantes recibieron una capacitación en pruebas de 3 horas, dividida entre la parte teórica y la presentación del modelo de proceso, teniendo un tiempo correspondiente a dos semanas para su aplicación y con la opción de asesorías por parte del docente y los autores del modelo de proceso.

Inicialmente se encontró un rechazo a la aplicación de pruebas y especialmente a la documentación, dado que según los estudiantes se aplicaban “fácilmente”, considerando que el aplicativo era demasiado “sencillo” para aplicar pruebas, o que las pruebas pueden realizarse

de manera intuitiva; Se realizó un seguimiento sobre la aplicación del modelo de proceso por medio de trabajo en clase y puntos de control en el ejercicio. Al finalizar el proyecto se revisaron los formatos, se indagó sobre ellos. El objetivo de este segundo caso era evaluar que tan fácil de usar es el modelo de proceso de pruebas, como indicadores se planteó el nivel de entendimiento, nivel de adherencia y el nivel de utilidad de las prácticas propuestas, de los 16 estudiantes 12 entregaron el proyecto, y de estos 10 entregaron los artefactos relacionados con pruebas, los indicadores se midieron revisando los artefactos entregados, se encontró que el nivel de entendimiento fue del 80%, el nivel de adherencia fue de un 70, 5%, el nivel de entendimiento se apreció a través de la realización de los artefactos y plantillas propuestas, diligenciadas y la manera como fueron empleadas y resueltas, otorgando una medida, en el nivel de adherencia se verifico la realización de las tareas propuestas, plantillas y uso de los artefactos de entrada.

Para el nivel de utilidad se realizó una pequeña encuesta a los estudiantes, los estudiantes consideraron que la aplicación del modelo de proceso les permitió esbozar un plan de pruebas antes de codificar, lo que hizo que identificaran posibles errores que podían presentarse y realizar ajustes al diseño y al código para evitarlos, segundo que los formatos facilitaban el diseño de los casos de prueba, como también el registro de errores facilitaba buscar el fallo y luego el ajuste del código. Finalmente la mayoría de los estudiantes reconocieron la importancia de las pruebas, y consideraron que el modelo de proceso facilitaba la realización de las pruebas.

VII. CONCLUSIONES Y TRABAJO FUTURO

El modelo de proceso de pruebas propuesto contribuye a la ejecución de pruebas de software sin excederse en esfuerzos y tiempo requerido y ajustándose a recursos humanos y técnicos de los proyectos de grado de desarrollo de software, el detalle de cada tarea y función a realizar permite al desarrollador que aplica las pruebas de software reconocer el estado del producto o componente que se está probando y evaluar sus necesidades frente a los resultados obtenidos.

El modelo de proceso de pruebas especificado y presentado en una página web permite comunicar las actividades de pruebas adecuadas a los recursos en un ámbito formativo, permitiendo de una forma clara y concisa visualizar el flujo de trabajo, detallando los pasos necesarios en las fases importantes del proceso y utilizando herramientas sencillas y detalladas de trabajo para la documentación y retroalimentación de los resultados al producto desarrollado.

La evaluación del modelo de proceso propuesto con una herramienta como es AVISPA permite mejorar y validar la correctitud del proceso propuesto, potencializando el proceso y los resultados obtenidos al aplicarlo, gracias el modelado puede verificarse antes de ser empleado.

Este proceso de pruebas de software pretende dar facilidades al desarrollador de software que lo ejecute, por medio de un proceso liviano y ajustado a las necesidades de una metodología ágil, pretendiendo ser una herramienta para el desarrollo de software mejor sustentado y no una imposición para el desarrollo de software, al tiempo que las pruebas unitarias presentadas son soportadas en el estándar de norma británica BS 7925-2, esto permite mejorar directamente la validación y verificación del software desarrollado por medio de herramientas de fácil aplicación, Las pruebas unitarias permiten verificar la funcionalidad de un componente, lo cual las convierte en la base de mayor importancia para determinar el grado de completitud de un componente, a pesar de ello no son aplicadas correctamente, el flujo de proceso propuesto pretende dar una orientación al estudiante donde los elementos utilizados como tareas, roles y productos de trabajo faciliten entender y aplicar este tipo de pruebas, los elementos propuestos buscan ser simples y no recargar de trabajo, logrando facilitar el hallazgo de errores aplicando casos de pruebas propuestos por la norma.

Es importante continuar con los otros tipos de pruebas, de acuerdo a las necesidades detectadas en los trabajos de grado y especialmente a las debilidades detectadas, lo que permitiría mejorar la calidad de los productos obtenidos, fortaleciendo la preparación de los desarrolladores de software y sus competencias respecto a las pruebas de software y el rol de Tester en la industria productora de software.

VIII. REFERENCIAS

- [1] R. Pressman, *Ingeniería del software, Un enfoque practico*. s.l. : McGraw-Hill, 2010.
- [2] British Computer Society. *Standard for Software Component Testing*. 2001.
- [3] S. Committee, *Draft IEEE Standard for software and system test documentation*. Institute of Electrical and Electronics Engineers. New York : s.n., 2008.
- [4] E. Giraldo, E y E. Casstrillon, *Resultados de investigación Impacto del egresado del programa de ingeniería informática y su desempeño con respecto a la industria del software*. 2007, Revista Virtual Universidad Católica del Norte, págs. 1-16.
- [5] J. F. Parra Castrillon, *Factores Críticos de Éxito e Hipótesis sobre la Industria del Software en Colombia. Consideraciones Contextuales y Académicas*. 2008, Revista Avances en Sistemas e Informática, págs. 185-193.
- [6] J. Rojas Rojas y E. J. Barrios, *Investigación sobre estado del arte en diseño y aplicacion de pruebas de software*. Bogota : s.n., 2007.
- [7] M. L. Rojas Montes, *Proceso de pruebas para pequeñas organizaciones desarrolladoras de software*. Popayán : s.n., 2013.
- [8] B. Perez Lamancha, *Proceso de Testing Funcional Independiente*. Instituto de Computación, Uruguay. 2006. Tesis de Maestría.
- [9] British Computer Society. *Glossary of terms used in software testing*. 1998.
- [10] J. Myers, *The Art of Software Testing*. [ed.] Hoboken. 2004. Vol. Segunda edición.
- [11] J. A. Hurtado, M. C. Bastarrica, A. Bergel, *A Tool for Analyzing Software Process Models*, 2013. Journal of software maintenance and evolution: Research and practice.
- [12] J. A. Hurtado, M. C. Bastarrica, A. Bergel, *Analyzing software process models with AVISPA*, 2011. ICSSP '11 Proceedings of the 2011 International Conference on Software and Systems Process.
- [13] I. Sommerville, *Ingeniería de Software*. s.l. : Pearson Educacio, 2005.
- [14] P. Runeson & M. Höst, *Guidelines for conducting and reporting case study research in software engineering*. [D. Sjoberg, Ed.) Empirical Software Engineering , 131-164 [2009].
- [15] G. Canfora, F. Garcia, M. Piattini, F. Ruiz, and C. A. Visaggio, "A family of experiments to validate metrics for software process models", Journal of Systems and Software, vol. 77, no. 2, pp. 113–129, 2005.
- [16] Murnane Taflin, ISO/IEC/IEEE 29119 Software Testing - The international standard for software testing. [Citado el: 22 de agosto de 2014.] <http://www.softwaretestingstandard.org/>.
- [17] James Newkirk, R. C. (2002). *La Programación Extrema en la práctica*
- [18] C. Cardona, *Propuesta metodológica para la realización de pruebas de software en ambientes productivo*, Universidad Nacional de Colombia, 2009.
- M. Castro, Procedimiento para la realización de pruebas de unidad de software orientados por objetos a nivel de clase, Repositorio Institucional de la Universidad Nacional de Colombia, 2011