

Pruebas exploratorias como metodología de enseñanza y complemento a los métodos tradicionales de prueba de software

Gabriela Salazar

Docente
Escuela de Ciencias de la Computación e
Informática
Universidad de Costa Rica
San José, Costa Rica
gabriela.salazar@ecci.ucr.ac.cr
Tel. 89985443
Fax. (506)2511-5527

Oscar Suárez Lechado

Estudiante
Escuela de Ciencias de la Computación e
Informática
Universidad de Costa Rica
San José, Costa Rica
oscar.suarez@ucr.ac.cr
Tel. 8824-6923
Fax. (506) 2297-8013

Resumen

La producción de software es un proceso que evoluciona a grandes velocidades. Asegurar la calidad de este proceso se vuelve vital en un sector productivo que cada vez es más competitivo. Esto redundará finalmente en la importancia de las pruebas como elemento base para garantizar un proceso de producción de calidad. La Ingeniería de Software ha desarrollado diversas técnicas que facilitan el proceso de pruebas de software, entre las cuales están las Pruebas Exploratorias (PE), las cuales le dan un enfoque diferente a la planeación y ejecución de las mismas. El artículo describe las características y metodología de la técnica, y expone dos casos prácticos en donde su aplicación como complemento a los procesos formales de pruebas, aumenta la probabilidad de obtener un producto de software de mayor calidad. Los puntos descritos en este artículo pueden interesar a profesores que desean formar ingenieros de software.

Palabras claves: Calidad de Software, Pruebas Exploratorias, Ingeniería de Software, Desarrollo de Software.

Abstract

Software production is a process that evolves at high speeds. Ensuring the quality of this process becomes vital in a productive sector that is increasingly becoming more competitive. This results in the importance of QA elements to ensure a quality production process. Through the development of software engineering various techniques have evolved to facilitate the systematization of this thread. Such is the case of exploratory tests, which intend to change the way to plan and execute the tests. This paper describes the features and methodology of the technique, and exposes two practical cases where its application as a complement to the formal processes of evidence, increases the likelihood of obtaining a product of higher quality software.). The described points in this paper can interest leaders, professors, and instructors that want to form future software engineers.

Keywords: Software Quality, Exploratory Testing, Software Engineering, Software Development.

1. Introducción

Las pruebas exploratorias son una técnica que pese a ser formalizada a principios de los años 70, dado que no existía fundamentación teórica e investigación, se ha utilizado de manera empírica, como una técnica más dentro del proceso de pruebas. [Ge11] [Itko05]

Las pruebas exploratorias (PE) pueden definirse como el proceso simultáneo de exploración del producto (aprendizaje), diseño y ejecución de pruebas. [Bach02] [Itko05] De acuerdo a [Arza11] “Las PE constituyen una técnica de prueba en la cual el probador controla activamente el diseño mientras son realizadas, y utiliza la información obtenida en la exploración para diseñar nuevas y mejores pruebas. Esta técnica es sumamente útil cuando se tiene un software que nunca se ha probado, es desconocido o es inestable.”

Según [Sant08] algunos aspectos que se deben considerar para poder ejecutar satisfactoriamente este tipo de pruebas son:

- Todo lo que se verifica debe estar orientado a la detección de problemas que impidan satisfacer los requisitos funcionales y los no funcionales. Estas pruebas deben enfocarse en probar en el menor tiempo posible aquellas funcionalidades que son de mayor importancia para el usuario.

- Se debe tener capacidad de análisis elevada para entender los requerimientos con base en la observación y el análisis de la evolución de los mismos. El equipo de pruebas debe tener conocimiento suficiente de la aplicación, principalmente en cuanto a la etapa de modelado y diseño. Esto le permitirá tener claro que probar y con qué prioridad y profundidad hacerlo.
- Poseer capacidad de ampliar caminos de pruebas para mejorar espontáneamente la cobertura, basándose principalmente en el uso intuitivo que tendría un usuario del sistema. Esto significa ponerse en el lugar del usuario.
- Tener amplio conocimiento y dominio del negocio al que pertenece el problema. Muchas veces no basta con tener un conocimiento amplio de los requerimientos y de la aplicación en sí. Para poder proveer una retroalimentación más completa es importante conocer el ámbito del problema para poder innovar.
- Ser proactivo y tener capacidad de comunicación para generar un ambiente colaborativo. Una personalidad activa y extrovertida hace más fácil para el sujeto producir y mejorar pruebas, innovar elementos del sistema, entre otros. Además, la capacidad comunicativa es esencial para el proceso de reporte de fallos, para informar de innovaciones de manera precisa y finalmente, fomentar un ambiente colaborativo y de compañerismo.
- Conocer los procesos del sistema para reconocer las fallas que se manifiestan en los resultados no deseados. El probador debe conocer cuál es la salida y el comportamiento esperado del sistema, así puede discriminar rápidamente si ha encontrado un fallo o no.
- Ser capaz de definir y variar las estrategias, las tácticas y las técnicas de validación y verificación. El probador debe tener un amplio arsenal de estrategias para probar y verificar el sistema.
- Poseer capacidades intuitivas para el desarrollo de las pruebas. Este concepto, pese a ser abstracto y difícil de medir, es fundamental para el probador. Es una capacidad que puede ser desarrollada a través de la experiencia.

En la siguiente sección se explican algunos elementos teóricos fundamentales que facilitan las PE y se detalla el proceso a seguir en la ejecución de las mismas. En la tercera sección se comparan las pruebas tradicionales y las pruebas exploratorias, y se explican los casos en los que es conveniente utilizar esta técnica. En la cuarta se presentan dos casos reales en donde se aplica las PE, el primero es en un laboratorio de calidad de software en una universidad y el segundo describe la experiencia al aplicarla en el curso de Ingeniería de Software del Plan de estudios de Pregrado en la Universidad de Costa Rica. Seguidamente, en la quinta sección se presentan las conclusiones.

2. Conceptos teóricos

2.1 Principales características

A continuación se describen las principales características de las pruebas exploratorias [Arza11]:

- **Velocidad de Respuesta:** La facilidad y simplicidad de las PE le permite a los equipos de calidad del software realizarlas bajo cualquier circunstancia y en cualquier momento. Esto permite que el equipo de pruebas pueda encontrar fallos desde el inicio del proceso, así como detectar tempranamente oportunidades de cambio y la formación de criterios gracias al rápido aprendizaje. De esta manera, se puede cambiar el rumbo del proceso de desarrollo (cuando sea necesario), permitiendo que la retroalimentación de estas pruebas redunde en una mejora del proceso. Además, proporciona una importante reducción en el tiempo que se invierte posteriormente en re-trabajo.
- **Pruebas de valor:** Esta característica es de suma importancia en este tipo de pruebas, pero no es una responsabilidad estrictamente del equipo de pruebas que las realiza. Se busca involucrar a todas las partes del proceso (desarrolladores, probadores y clientes), de forma que las pruebas incluyan la perspectiva y la lógica de los diferentes actores teniendo así un valor agregado importante. Además, el hecho de incrementar el número de involucrados en el proceso de diseño/ejecución de las pruebas, produce mayor calidad en los resultados de éstas, esperando así obtener un software de mayor calidad.
- **Nuevas ideas:** Esta característica es la más importante en el proceso de pruebas porque cuando al probador se le ocurre alguna idea nueva sobre cómo probar un elemento, puede ser incorporada y podría dar origen a un nuevo camino de pruebas dentro del proceso. Aquí radica la flexibilidad de esta modalidad de pruebas. Sin embargo, dado que la cantidad de pruebas que pueden surgir en este proceso puede crecer a tal punto de no ser manejable, es importante que entre en juego un factor externo, el ingenio del probador. Este debe tener la suficiente capacidad para discernir entre las buenas y las malas ideas, pero también debe disponer de la experiencia necesaria para determinar a cuáles ideas debe darle prioridad. Si se carece de esta experiencia existe una regla a seguir para discriminar entre buenas y malas ideas: una buena nueva idea debe ser lo suficientemente distinta del resto de las ideas, de forma que permita comprobar el cumplimiento de los requerimientos del software, sin redundar en lo que los otros caminos de prueba ya comprobaron con anterioridad.
- **Escenarios de Criticidad:** Estos consisten en la base fundamental de las PE. Este enfoque de pruebas suplanta el diseño de casos de prueba de los enfoques tradicionales, por preceptos de simplicidad y practicidad, fundamentales para mantener los beneficios buscados con la utilización de las PE. La base de un escenario de criticidad son los documentos de descripción de requerimientos, los cuales permiten tener una noción de la meta a cumplir del software. Con esta información en mente y el conocimiento del estado actual del software a probar, se plantean los escenarios de criticidad, que son una descripción de los elementos hacia los cuales se dirigen las pruebas. Estas deben centrarse en cumplir a cabalidad y de forma adecuada los requerimientos del software que se determinó probar.
- **Probar y Aprender:** El proceso de PE tiene inmersa en su dinámica la combinación de ambos elementos. Las PE permiten que los encargados de realizarlas se conviertan en expertos en el sistema que se está desarrollando. Esta experticia en la aplicación le permite al grupo de pruebas retroalimentar a los desarrolladores, pero también terminan convirtiéndose en capacitadores y personal de soporte técnico para el usuario.

En un estudio realizado en el Instituto de Negocios e Ingeniería de Software de la Universidad Tecnológica de Helsinki en Finlandia [Itko07], se lograron establecer algunos aspectos fundamentales que diferencian las pruebas tradicionales de las exploratorias:

- Respecto a la eficiencia para detectar errores, ambos métodos tienen un rendimiento similar, es decir, detectan una cantidad de errores similares en tipo y severidad. Sin embargo, las pruebas basadas en casos de uso permiten obtener resultados con un grado mayor de profundidad y completitud.
- En cuanto a eficiencia y calidad de los resultados, ambas metodologías son muy similares, no obstante, aún hay otros aspectos que deben evaluarse como la experiencia del equipo de pruebas. La experiencia del equipo puede afectar la calidad de las pruebas y por ello de la aplicación en general, dado que las PE no tienen una fase formal de diseño de casos de prueba, el tiempo que el equipo le dedicaría a esta tarea se utiliza en la misma ejecución de las pruebas, obteniendo resultados similares a las pruebas tradicionales pero en menos tiempo. Finalmente, según algunas investigaciones, la personalidad de los integrantes del grupo de pruebas incide en los resultados obtenidos. [Akba09]

Existen algunas desventajas documentadas de las PE [Våga02] [Bach00] [Cope04]:

1. Es difícil seguir el progreso individual de cada probador y conocer el esfuerzo global de las pruebas. Esto se debe a que existe una tendencia natural a documentar lo menos posible y dada la libertad que ofrece esta técnica, puede resultar tentador caer en este tipo de malas prácticas.
2. Las PE tienen una menor capacidad para prevenir defectos, a diferencia de las pruebas basadas en casos de prueba que se comienzan a diseñar desde la fase de recopilación de requerimientos y/o diseño, lo que permite encontrar defectos desde fases tempranas.
3. Las pruebas exploratorias deben esperar a que existan módulos implementados para poder probarlos.

2.2 Tipos de PE

Es usual que se crea que las pruebas exploratorias son informales y que no tienen un procedimiento o forma de aplicación. Esto es falso, pese a la libertad que se le concede a quien ejecuta las pruebas, existen ciertos lineamientos que permiten establecer prioridades y “guiones” que mejoran el resultado y mitigan algunos puntos débiles, que se tratarán posteriormente. [Ge11] [Arza11]

De acuerdo a [Pere07] [Arza11] según el tipo de sistema a probar y el estado de madurez del mismo, se pueden distinguir diferentes formas de hacer las pruebas exploratorias. Las más libres son útiles en las primeras fases del desarrollo, cuando aún los requerimientos, las funcionalidades y las especificaciones son inestables. Conforme el desarrollo del sistema avanza, las PE se vuelven más controladas y para ejercer este control existen diferentes aproximaciones, alguna de las cuales son:

- **Basadas en sesiones:** Esta técnica fue propuesta por James Bach. Consiste en organizar las pruebas en sesiones documentadas. Cada sesión posee un itinerario, Este se determina con base en una misión e inclusive algunas heurísticas. Tiene como ventaja que no incurre en un consumo alto de recursos, a pesar de que se elaboran reportes de avance, se registra el itinerario y se gestiona y mide el proceso. Esta forma de PE es flexible y adaptable, lo cual es idóneo para realizar pruebas independientes para un cliente. Como desventaja se puede mencionar la dependencia de las PE de las habilidades y preparación del grupo de probadores.
- **Basadas en funcionalidades parciales:** Se usa para probar funcionalidades individuales después de que han sido implementadas. Se busca probar que el módulo probado cumpla con los requerimientos y las concepciones del diseño. Su principal ventaja radica en proporcionar rápida retroalimentación a los desarrolladores en etapas tempranas del desarrollo.
- **Realizadas por los usuarios:** Los usuarios exploran si las funcionalidades se adecúan a sus necesidades reales. Ellos son quienes poseen conocimiento del negocio y por ello producen una retroalimentación de gran valor.
- **Pruebas de humo exploratorias:** Permiten obtener un vistazo rápido y global del nivel de calidad de una nueva versión (liberada para pruebas) de un sistema. Es especialmente útil cuando se generan actualizaciones frecuentes. Para ejecutarlas se utiliza una lista de funcionalidades prioritarias para detectar defectos en éstas. También se necesita contar con una lista de correcciones realizadas para verificar que se hayan corregido. Finalmente, se consideran las mejoras que se hayan realizado desde el punto de vista del usuario.
- **Pruebas exploratorias de regresión:** A veces resulta imposible realizar pruebas de regresión exhaustivas (por restricciones de tiempo, económicas o de recursos humanos), por ello esta fase puede enfocarse en las correcciones y mejoras desarrolladas. Este tipo de prueba debería ser encargada a personal de pruebas con experiencia, ya que el proceso de exploración debería informar de la forma más completa posible, los nuevos defectos o daños colaterales.

Todas estas generan documentación de utilidad no solo para el proceso de pruebas, si no para la innovación del producto, además de servir para fines de capacitación y soporte técnico.

3. Metodología

3.1 Elementos fundamentales que posibilitan las PE

Es importante conocer algunos elementos fundamentales que las posibilitan las PE [Corr11]:

- **Exploración del producto:** Esta tarea permite descubrir y almacenar los propósitos y las funciones del producto, los tipos de datos procesados y las áreas potenciales de inestabilidad. De esta forma es posible tener una idea de la dirección e intención de las pruebas.
- **Diseño de pruebas:** Es el determinar las estrategias para operar, observar y evaluar el producto. Produce una especie de estándar para trabajar que le permite al equipo de pruebas generar documentos homogéneos para facilitar su unificación y por ello la producción de resultados. También facilita el análisis de posteriores resultados.
- **Ejecución de pruebas:** Son las operaciones sobre el producto, la observación de su comportamiento y la utilización de la información para generar hipótesis sobre cómo trabaja el producto. En especial se le da atención a los elementos definidos como de mayor prioridad.
- **Heurísticas:** Debe contar con guías y reglas de dedo que le ayuden a decidir lo que se debe hacer en casos de ambigüedad o cuando existan diversos caminos para realizar la tarea. Esto consiste en que el probador tenga claro qué regla debe elegir. Por ejemplo: el realizar primero las pruebas que conlleven menor tiempo, o más bien las tareas complejas, y en caso de ser necesario, qué tareas tienen una menor prioridad, o incluso, cuáles de las mismas pueden ser obviadas.
- **Resultados revisables:** Los resultados deben darse en un formato en el cual puedan ser revisados de una forma rápida y eficaz, así como una orientación hacia el cumplimiento de los objetivos o misiones de la aplicación.

3.2 Metodología del proceso de PE

El proceso de pruebas exploratorias está compuesto por las siguientes tareas:

1. **Analizar los requerimientos y el estado del producto:** Debe tenerse en cuenta la etapa de desarrollo en la que se encuentra el producto para determinar cuáles de los requerimientos pueden ser total o parcialmente probados de acuerdo al estado actual del producto.
2. **Analizar los recursos disponibles:** En este paso se definen elementos fundamentales para cualquier proceso de pruebas. Ejemplos de estos recursos son: las aplicaciones de apoyo que podrá utilizar el probador para obtener sus resultados, el tiempo con el que contará el probador para interactuar con el producto a lo largo de cada prueba, la cantidad de miembros que se encargarán de cada misión (tradicionalmente se trabaja de forma individual o en equipos de 2 personas) y finalmente el equipo físico con el que contarán para la realización de las pruebas.
3. **Definir las misiones:** Para hablar sobre las tareas a realizar en este tipo de pruebas se definen los elementos llamados “misiones”, las cuales contienen una descripción de los objetivos y metas que se desean lograr en los procesos de pruebas. Además se define una descripción metodológica que muestre el camino a seguir para alcanzar las metas planteadas, sin que esto se convierta en un elemento altamente restrictivo para el probador.
4. **Realizar las pruebas:** En este paso se procede a dar completa libertad sobre el proceso al probador, de forma que se sienta en libertad de llevar a cabo las pruebas a su propio estilo. Sin embargo, debe mantenerse enfocado en su “misión”, dentro del lapso de tiempo definido, y utilizando los recursos estipulados.
5. **Incorporar los resultados:** En esta última tarea se toman los resultados obtenidos por cada uno de los miembros del equipo de pruebas a través de los reportes, y se analizan con dos objetivos en mente: el primero de ellos es la corrección de errores de la aplicación, objetivo final de cualquier proceso de pruebas, el segundo objetivo es analizar el proceso general con el fin de mejorarlo y tomar decisiones por parte del probador. Esto se da debido a que se incorporan nuevas ideas, obtenidas gracias a la incorporación de las mismas desde las primeras etapas del proceso. El objetivo final de todo esto es que el probador adquiera cada vez más experticia, que pueda tomar mejores decisiones que le permitan lograr una tasa mayor de detección de errores, en un menor tiempo y con una menor cantidad de pruebas, lo que a la vez le facilitará disponer de un mayor espacio para implementar las ideas que el proceso de PE genera.

3.3 ¿Cuándo usar las Pruebas Exploratorias?

Para lograr un buen desempeño de este tipo de pruebas, es necesario conocer bajo qué situaciones es recomendable realizarlas. A continuación se muestran algunas situaciones según [Bach02]:

- Cuando se necesita una rápida retroalimentación de un producto. Por ejemplo en las etapas tempranas de desarrollo es necesario determinar si la orientación del sistema es la adecuada desde el punto de vista del usuario y con esta retroalimentación hacer los ajustes necesarios. Otro caso importante es para decidir si el producto tiene las condiciones necesarias para pasar a la fase de pruebas siguiente o se declaran fallidas las mismas.
- Cuando se requiere decidir si el producto tiene las condiciones necesarias para pasar a la siguiente fase de pruebas. Por ejemplo se pueden aplicar PE y si se considera que está listo se le aplican pruebas funcionales, pero si declaran las pruebas fallidas se suspenden temporalmente las pruebas funcionales.
- Cuando se necesita aprender rápidamente sobre el manejo del producto. Muchas veces el cliente solicita capacitación en cuanto al uso del sistema, una de las maneras más eficientes para realizar esto es tomando al grupo que realizó pruebas exploratorias sobre el sistema y utilizar su experiencia en el uso de este para capacitar al cliente. En síntesis, el proceso de pruebas exploratorias resulta la manera natural de aprender el uso del sistema.
- Cuando se requiere una diversificación de técnicas en el proceso de pruebas.
- Cuando se requiere encontrar un problema en el menor tiempo posible. Usando un enfoque menos controlado (sujeto a un guión) se reduce el tiempo para encontrar fallos.
- Cuando se requiere contrastar el trabajo de un probador con otro. Como se mencionó antes, las pruebas exploratorias tienen un componente subjetivo propio de la libertad que se le otorga a quien realiza las pruebas. Esto permite comparar el trabajo de algunos probadores sobre un mismo módulo. Esta información permite evaluar al personal para determinar las cualidades específicas de cada miembro y poder utilizarlo de acuerdo a éstas.
- Cuando se desea aislar un defecto en particular. Esto se da principalmente en sistemas donde las pruebas basadas en casos de uso no son aplicables. En cualquier caso, resulta más rápido para un experto usar el enfoque exploratorio para aislar un defecto que someterse a una receta que sistematiza este proceso.
- Cuando se requiere evaluar riesgos particulares de un producto. Igual que en el caso anterior, las evaluaciones de riesgo suelen ser llevadas a cabo por personal con experiencia y en ese caso es más natural el enfoque exploratorio.

De estas situaciones podemos concluir que la principal ventaja con la que cuentan las PE, es la velocidad y efectividad de retroalimentación. Esta situación permite un rediseño veloz de las pruebas y del sistema en general, mejorando sustancialmente la capacidad de las mismas para proveer resultados.

4. Casos prácticos

4.1 Aplicación de PE en un Laboratorio de Calidad de Software en una Universidad [Arza11]

La Universidad de Ciencias Informáticas (UCI) de la Habana Cuba, es un centro de estudios vinculado directamente a la producción de soluciones informáticas. Para ello cuenta con el Departamento de Pruebas de Software (DPSW) el cual está constituido por dos grandes grupos: el Grupo de Ingeniería de Pruebas (GIPS) y el Laboratorio Industrial de Pruebas de Software (LIPS). Estos grupos son responsables de verificar que todos los productos elaborados en la UCI y presentados al laboratorio sean comprobados y evaluados según normas y estándares de calidad, antes de ser entregado al cliente, siendo esta evaluación confiable tanto para los equipos de desarrollo como para los clientes de la UCI.

Una vez que entra un producto de software al DPSW para ser evaluado, el mismo se somete a la realización de PE, con el objetivo de decidir si tiene las condiciones para pasar a la fase de prueba siguiente o si se declaran fallidas las mismas. Anteriormente se aplicaban las PE basados en la experiencia y conocimiento del especialista del DPSW encargado de administrarlas, sin un procedimiento que permitiera guiar el trabajo. Esto hacía que en general no se tomara en cuenta la selección de la muestra a probar debido a la magnitud del mismo, provocando desconocimiento a la hora de tomar decisiones durante el proceso de pruebas y un incremento gradual de gastos de recursos humano, de tiempo, de tecnología, de costos y de esfuerzo. Actualmente se rediseñó la ejecución de las PE y se le insertaron nuevos conceptos como las Pruebas Exploratorias Iniciales (PEI) las cuales permiten obtener una retroalimentación rápida

del artefacto a probar, que valida la calidad de éste, permitiendo o no el paso a nuevas etapas. El nuevo procedimiento se describe a continuación:

1. Una vez que se presenta una solicitud de prueba para un artefacto, se confecciona un Pre-Plan de Pruebas que incluye los objetivos y metas.
2. Se convoca a una reunión de inicio donde se define todo el proceso de pruebas.
3. Se ejecuta una revisión y se refina el diseño de las pruebas y el montaje de entorno.
4. Se aplican la PEI.
5. Finalizadas las PEI el entregable puede o no ser abortado o declararse fallida la prueba. Si se considera fallida se convoca a una reunión para determinar las causas. En caso contrario se ejecutan las pruebas funcionales siguiendo cierta cantidad de iteraciones (≤ 3) hasta que el mismo sea liberado o abortado.

La utilización de las PEI no excluye la aplicación de las PE, las cuales se realizan paralelas a las pruebas que se están ejecutando. El proceso de pruebas del DPSW requiere de ambos tipos de pruebas.

De acuerdo a [Arza11] se puede notar como la aplicación de las pruebas exploratorias en LIPS produce los siguientes beneficios:

1. Disminuir gastos de recursos de : tiempo, capital humano, tecnología, costo, esfuerzo, etc..
2. Proporciona rápidamente información sobre el artefacto a probar.
3. Encuentra los principales errores en un plazo breve.
4. Mitiga equivocaciones en el análisis de riesgos del producto.
5. Garantiza un mínimo de calidad para que el artefacto pase a la fase de Pruebas Funcionales.

4.2 Aplicación de PE en un proyecto de un curso universitario

En esta sección se describe la experiencia de enseñar la técnica de PE en el curso de Ingeniería de Software a estudiantes de pregrado en la Escuela de Ciencias de la Computación e Informática (ECCI) en la Universidad de Costa Rica. En la ECCI la ingeniería de software se imparte a través de los cursos Ingeniería de Software I e Ingeniería de Software II con sus respectivos laboratorios. Los cursos son semestrales, uno es requisito del otro y cada uno tiene una duración aproximada de 16 semanas lectivas. Son impartidos durante el sexto y séptimo semestre del plan de estudios. Cuando los estudiantes inician el curso de Ingeniería de Software I no cuentan con experiencia ni conocimiento: en metodologías de desarrollo, en implementación de aplicaciones web en una arquitectura n capas, ni en materia de pruebas.

La aplicación de la técnica de PE en el curso de Ingeniería de Software se realizó durante el año 2011 tomando como base una aplicación web implementada en una arquitectura de 4 capas, que había sido desarrollada y probada formalmente por los estudiantes del año 2010. El equipo de trabajo que la había desarrollado, aseguraba después de un proceso formal de pruebas, que aunque no se satisfacían todos los requerimientos solicitados, estaba libre de fallas.

El objetivo de aplicar las PE era “conocer” la aplicación y determinar su estado en cuanto al cumplimiento de los requerimientos solicitados y a las fallas encontradas.

El propósito general del proyecto consistía en implementar un Sistema de Administración de Requerimientos que permitiera administrar los requerimientos de un proyecto de desarrollo de software durante su ciclo de vida, y de esta manera implementar el área clave de Administración de Requerimientos del Modelo de Capacidad Madurez (CMMI). La aplicación debía cumplir con las siguientes funcionalidades:

- a. Administrar la información básica de los **recursos humanos** que tienen acceso al sistema (administrador, cliente, desarrollador).
- b. Administrar la **seguridad** de la aplicación restringiendo el acceso a la información de acuerdo al rol del usuario.
- c. Administrar la información básica de un **proyecto**, permitiendo además, crear para cada proyecto el equipo de trabajo (desarrolladores y usuario).
- d. Administrar los **requerimientos funcionales** y los **requerimientos no funcionales** para un determinado proyecto.

- e. Para cada requerimiento funcional y no funcional administrar las **dependencias** y los **conflictos** asociados con otros requerimientos del mismo proyecto.
- f. Administrar los **cambios** realizados al requerimiento (quién, cuándo, qué y por qué?) y a partir de una línea base controlar las **versiones** para facilitar el manejo de su historial sin borrar las versiones anteriores.
- g. Algunas **consultas** como por ejemplo: árbol jerárquico que muestre los proyectos organizados por iteraciones y dentro de estas por módulos, con sus correspondientes requerimientos, etc..
- h. Administrar los **cambios realizados** al requerimiento (quién, cuándo, qué y por qué?) y a partir de una línea base controlar las versiones para facilitar el manejo de su historial sin borrar las versiones anteriores.
- i. Administrar el **materia de apoyo** (casos de uso, casos de pruebas, entre otros) asociados a los requerimientos de un determinado proyecto, de manera que permita el acceso a los archivos electrónicos correspondientes.
- j. Administrar el proceso de **verificación y validación de la calidad** de los entregables del proyecto.

El proyecto del curso tenía las siguientes tareas:

1. Conocer la aplicación y determinar su estado en cuanto al cumplimiento de los requerimientos solicitados y a las fallas encontradas.
2. Recomendar mejoras a la aplicación y negociarlas con el profesor (en su rol de usuario).
3. Implementar en la aplicación las mejoras y los requerimientos que quedaron pendientes del año 2010.

Para realizar este proyecto a los estudiantes se les entregó: el código de la aplicación, la especificación de requerimientos y una lista de No conformidades clasificadas que les permitiera determinar el tipo de falla encontrada. La lista de No conformidades incluía los siguientes tipos:

1. **Funcionalidad:** Al realizar una acción determinada el resultado que se muestra no está acorde con el esperado.
2. **Errores de interfaz:** Se encuentran aquellas inconformidades de efecto visual que provoquen las interfaces de las aplicaciones.
3. **Validación:** Existen errores relativos a la falta de validación.
4. **Opciones que no funcionan:** Al realizar una acción determinada no se muestra resultado alguno.
5. **Excepciones:** El sistema muestra un mensaje señalando que ha ocurrido un error inesperado o que no ha sido tratado.

Las pruebas fueron realizadas por 4 equipos de trabajo constituidos por 5 estudiantes cada uno. Cada equipo tenía su propio líder y partiendo de la organización de módulos que componían la aplicación, estos fueron distribuidos entre los miembros del equipo.

Entre los 4 equipos detectaron un total de 56 no conformidades clasificadas de acuerdo al tipo tal como se muestra en la siguiente tabla.

Tabla 1. Cantidad de No conformidades identificadas de acuerdo al tipo.

Funcionales	Interfaz	Validación	Excepciones	Opciones que no Funcionan	Totales
11	26	7	0	12	56

Algunos errores detectados fueron:

1. Problemas en la asignación del líder a un proyecto al permitir que más de una persona fuera líder del mismo proyecto.
2. Admitía que un mismo recurso humano se insertara dos veces.
3. No validaba correctamente la contraseña en el módulo de seguridad.
4. No se controlaba adecuadamente la concurrencia.
5. Falta de validación de datos y de mensajes de aviso que le permitieran al usuario entender y corregir su error
6. Algunas funcionalidades como el manejo de conflictos o la conformación de equipos de trabajo a un proyecto eran poco amigables lo que dificultaba su uso.
7. Interfaces muy recargadas y los campos mal distribuidos.

La distribución de errores se muestra en el siguiente gráfico:

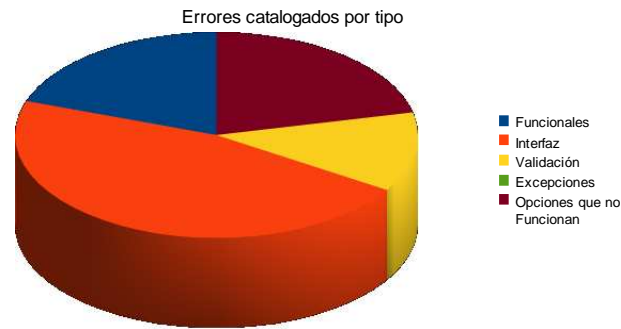


Figura 1. Distribución de no conformidades distribuidas de acuerdo al tipo.

Con la aplicación de las PE se lograron los siguientes beneficios:

1. Conocieron una aplicación desconocida para ellos en poco tiempo.
2. Experimentaron el papel del usuario lo cual les permitió enfrentar problemas de usabilidad por interfaces mal diseñadas.
3. Aprendieron a identificar incumplimiento de funcionalidades ofrecidas al usuario.
4. Aprendieron a detectar y clasificar fallas de una manera práctica y rápida.

Por el desconocimiento que presentaban los estudiantes en materia de pruebas era difícil enseñarles a diseñar y ejecutar pruebas formales, y a automatizar este proceso, por lo que la técnica de PE resultaba de mucha utilidad.

Posterior a la aplicación de PE se especificaron los requerimientos considerando la lista de mejoras elaborada por los estudiantes e iniciaron el proceso de desarrollo. Como metodología de desarrollo utilizaron una combinación de Proceso Unificado y de metodología ágiles. Dentro del proceso de desarrollo realizaron las siguientes tareas: rediseñaron las pantallas, actualizaron los casos de uso y los requerimientos, planificaron el trabajo a realizar, mejoraron el diseño de la arquitectura, rediseñaron los diagramas de secuencia y de clases, mejoraron el código y concluyeron el ciclo con un proceso formal de pruebas utilizando como guía plantillas elaboradas con base en el estándar del IEEE 829.

Dentro del proceso formal de pruebas trabajaron con las siguientes plantillas:

1. Plan de pruebas.
2. Especificación de casos de pruebas.
3. Especificación de pruebas unitarias y de integración.
4. Reporte de No conformidades.
5. Informe de métricas.

Como parte del proceso formal de pruebas ejecutaron pruebas Funcionales y pruebas de Regresión utilizando como herramienta de software Selenium. Los resultados de las pruebas fueron registrados en el Reporte de no conformidades y en el Informe de métricas el cual incluía métricas como: Cobertura de pruebas y Densidad de pruebas.

5. Conclusiones

A continuación se detallan las principales ventajas que como conclusión podemos establecer de las PE:

1. Las PE no son un reemplazo de las pruebas tradicionales basadas en casos de prueba, por el contrario resultan complementarias.
2. Las características que presentan las PE respecto al poder realizarlas en cualquier momento y experimentar nuevas ideas mientras se están realizando, sin seguir un plan riguroso, las hace idóneas cuando se está trabajando en las versiones iniciales de un sistema, ya que esa flexibilidad les permite adaptarse al cambio constante de requerimientos que se da en esta fase.
3. La practicidad que promueve de probar los escenarios de mayor criticidad, con base en el documento de especificación de requerimientos es una característica de mucho provecho aún cuando se realicen pruebas tradicionales.
4. Las PE tienen mucho valor cuando se ejecutan para determinar si una versión está suficientemente madura como para iniciar un proceso formal de pruebas y cuando la aplicación es desconocida.

5. Las PE permiten que los encargados de realizarlas se conviertan en expertos en el sistema que se está desarrollando. Esta experticia le permite al grupo de pruebas retroalimentar a los desarrolladores, y fácilmente pueden convertirse en capacitadores y en personal de soporte técnico para el usuario.
6. Adicionalmente, las PE pueden ser de mucha utilidad en un proceso de desarrollo de software bajo metodologías ágiles, debido a que tienen un enfoque similar pues permite el uso de expertos que pueden reducir la cantidad de tiempo y esfuerzo invertido en planeamiento y diseño.
7. Las PE sirvieron de metodología de enseñanza en el curso de Ingeniería de Software al enfrentar a los estudiantes con una aplicación real, en donde ellos mismos lograron identificarse con el papel del usuario e identificar fallas técnicas.
8. La aplicación de PE como complemento de las pruebas formales permite obtener productos de software de mayor calidad.

6. Referencias bibliográficas

- [Arza11] Arza, Lizandra., Brito, Yanet., León, Yeniset., et al. "Aplicación de la Estrategia de Pruebas Exploratorias en el Departamento de Pruebas de Software", Universidad de las Ciencias Informáticas, Cuba, 2011. Páginas: 4, 5, 8.
- [Akba09] Akbar, Aisha., Nadeem, Aamer., Shoaib, Lozina. "An Empirical Evaluation of the Influence of Human Personality on Exploratory Software Testing", ISBN: 978-1-4244-4873-9 IEEE, Pakistán, 2009. Página 6.
- [Bach00] Bach J., "Session-Based Test Management", STQE, vol. 2, no. 6, 2000, <http://www.satisfice.com/articles/sbtm.pdf> Página 1.
- [Bach02] Bach J., "Exploratory Testing Explained", The Test Practitioner, 2002, <http://www.satisfice.com/articles/et-article.pdf> Página 7.
- [Cope04] Copeland, L. "A Practitioner's Guide to Software Test Design", Boston: Artech House Publishers, 2004. Capítulo 13. 300pp.
- [Corr11] Corrales Vega, Rudy. "Pruebas Exploratorias". Blog Lecciones Aprendidas. URL: <http://rudycorrales.blogspot.com/2011/05/pruebas-exploratorias.html> San José, Costa Rica, Mayo, 2011.
- [Ge11] Gé, Surima., Pérez, Yenly. "Pruebas Exploratorias y Selección de las Muestras", Universidad de las Ciencias Informáticas, Cuba, 2011. Páginas 1, 5.
- [Itko05] Itkonen, Juha., Rautianen, Kristian. "Exploratory Testing: A Multiple Case Study", ISBN: 0-7803-9508-5 IEEE, Finlandia, 2005. Páginas 1, 2-10.
- [Itko07] Itkonen, Juha., Lassenius, Casper., Mäntylä, Mika. "Defect Detection Efficiency: Test Case Based vs. Exploratory Testing", ISBN: 0-7695-2886-4 IEEE, Finlandia, 2007. Páginas 9-10.
- [Pere07] Pérez, B., Pittier, A., Travieso M., Wodzislowski M.. "Testing exploratorio en la práctica", Uruguay, 2007. <http://www.ces.com.uy/documentos/JIISIC-2007.pdf> Páginas 2-3.
- [Pere07] Pérez, B., Pittier, A., Travieso M., Wodzislowski M.. "Testing exploratorio en la práctica", Uruguay, 2007. <http://www.ces.com.uy/documentos/JIISIC-2007.pdf> Páginas 2-3.
- [Sant08] Santillán, Javier. "Olfato e intuición Amplificada, Fortalezas y debilidades de las pruebas exploratorias". Blog Javier Santillán. URL: <http://javiersantillan.wordpress.com/2008/08/05/olfato-e-intuicin-amplificada/> Argentina, Agosto, 2008.