



ARTÍCULO ORIGINAL
CALIDAD

Proceso de pruebas para productos de software en un laboratorio de calidad

Testing process for software products at a quality laboratory

Dalila Jústiz-Núñez^I, Darlene Gómez-Suárez^{II}, Marta Dunia Delgado-Dapena^{II}

^I Complejo de Investigaciones Tecnológicas Integradas. La Habana, Cuba.

E-mail: djustiz@udio.cujae.edu.cu

^{II} Facultad de Ingeniería Informática. Instituto Superior Politécnico José Antonio Echeverría, Cujae. La Habana, Cuba.

E-mail: dgomez@ceis.cujae.edu.cu, marta@ceis.cujae.edu.cu

Recibido: 28/12/2011

Aprobado: 11/04/2013

RESUMEN

La calidad en sentido general, tanto de *software* como de otros tipos de productos, es un elemento que cada día se tiene más en cuenta a nivel mundial y su logro se relaciona directamente con el proceso que se emplee para obtenerla. Este trabajo presenta una propuesta de proceso de pruebas de *software*, para un Laboratorio de Calidad, inmerso en un ambiente universitario. Se detallan las actividades de los procesos fundamentales y los artefactos de salida, los niveles de prueba que se aplican y otros elementos de interés. Además se muestra una experiencia práctica de aplicación del proceso y los resultados de varios casos de estudio. Esta propuesta incluye la definición de los aspectos metodológicos y la selección de herramientas que automaticen el proceso.

Palabras clave: calidad de *software*, pruebas de *software*, proceso de pruebas, niveles de prueba.

ABSTRACT

In general terms, the quality of the software as of other products, is an element of increasing importance worldwide and it is strongly linked to its obtaining process. This work presents a proposal of a software testing process for a Quality Laboratory, integrated into an academic environment. The activities of the main processes and the output artifacts were detailed, as well as the testing levels applied, among other elements of interest. It was also showed a practical experience related to the process implementation and the results of several study cases. This proposal includes the definition of the methodological issues and the selection of the tools for the process automation.

Key words: *software quality, software tests, testing process, testing level.*

I. INTRODUCCIÓN

Los temas de calidad de *software* y en particular lo referente al proceso de pruebas son abordados en las investigaciones y los trabajos de múltiples autores en la actualidad [1; 2; 3; 4]. La producción de *software* cada vez más complejo y con integración de tecnologías varias exige la definición de procesos que garanticen la calidad del producto final [5; 6]. El escenario de la producción de *software* en las universidades se torna aún más complejo, pues los estudiantes asumen un papel protagónico en la actividad de desarrollo. Esta fuerza de trabajo es muy inestable porque está inmersa en el proceso de formación, por tanto se hace necesario concebir procesos para garantizar la calidad del producto final.

El presente trabajo se realiza en uno de los centros de investigación y desarrollo del Instituto Superior Politécnico "José Antonio Echeverría", en el cual se ha definido un proceso para desarrollar *software* y un conjunto de herramientas que le dan soporte. A pesar de los esfuerzos realizados en la institución, los productos no tienen la calidad requerida y en ocasiones no son certificados como listos para su explotación por las entidades certificadoras externas. La inexistencia de un proceso específicamente definido para realizar pruebas de software trae como consecuencia que esta actividad no se realice o se lleve a cabo inadecuadamente. Además, impide la uniformidad en cuanto a los aspectos de la calidad medidos en soluciones semejantes de proyectos diferentes y dificulta el procesamiento de la información que se obtiene.

La calidad del *software* debe tenerse en cuenta durante todo el proceso de producción del mismo, por tanto deben existir actividades que velen por su cumplimiento durante todo el ciclo de vida. Es una mala práctica esperar a que el software esté terminado para entonces aspirar a obtener la calidad en el mismo. Uno de los principios de Humphrey (2005 y 2008), dicta que "la calidad de un producto está determinada por la calidad del proceso que se emplea para producirlo" [7; 8; 9].

Algunos de los autores más reconocidos en el área de la ingeniería de software como: Beth (2009), Piattini (2009) y Pressman (2005), y especialmente, en la calidad de software, han establecido lo que sería el término propiamente dicho. Según Pressman, calidad no es más que el cumplimiento de los requisitos de funcionalidad y de desempeño establecidos explícitamente; normas de desarrollo documentadas explícitamente y características implícitas que son esperadas en todos los programas desarrollados profesionalmente [5; 8; 9]. Por su parte Piattini, establece que es el conjunto de propiedades o características de un producto o servicio que le confieren aptitud para satisfacer unas necesidades expresadas o implícitas [8].

La gestión de dicha calidad, según apunta, debe incluir varios aspectos fundamentales como la planificación, el aseguramiento y el control de la calidad [10]. En sentido general estos aspectos plantean la identificación de las normativas y estándares relacionados con el proyecto específico al que se le esté gestionando la calidad y a partir de ellas establecer las actividades que será necesario realizar, además de ejecutar las tareas de control, garantizando entregas de productos con cierto nivel de calidad [10].

Según IEEE (2004), las pruebas de software "*consisten en verificar el comportamiento de un programa dinámicamente a través de un grupo finito de casos de prueba, debidamente seleccionados del, típicamente, ámbito de ejecuciones infinito, en relación al comportamiento esperado*" [11]. Contrario a lo que se pueda pensar una prueba exitosa es aquella en la que se consigue que el sistema falle y por tanto, se encuentran errores, aunque estos no sean todos los

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

que posee el producto. De igual forma se puede considerar una prueba exitosa cuando es posible asegurar que no existen más errores en el sistema, cuestión que es mucho más difícil de garantizar [12]. Este es un aspecto del que advierte muy a menudo la teoría de pruebas, ya que no es confiable asumir que se han superado todas las pruebas posibles que pueden efectuarse a un software.

Específicamente las pruebas de *software* permiten evaluar las soluciones y determinar el nivel de calidad que poseen, por lo que se debe definir un proceso que se pueda emplear en el entorno de desarrollo de aplicaciones informáticas en la universidad.

En la organización que se analiza se lleva a cabo en estos momentos la implantación del modelo CMMI (*Capability Maturity Model Integration*), por sus siglas en inglés, es un modelo de madurez de mejora de los procesos para el desarrollo de productos y de servicios. Se enfoca tanto en procesos de Administración como de Ingeniería de Sistemas y *Software*. La aplicación del modelo permite evaluar la madurez de los procesos de desarrollo de *software* y proponer un plan de mejoras para dichos procesos. Todo con la finalidad de ir escalando a través de cinco niveles de capacidad: Nivel 1: inicial; Nivel 2: administrado; Nivel 3: definido; Nivel 4: cuantitativamente administrado y Nivel 5: optimizado. Se comienza por un nivel inicial o caótico, en el que la productividad y la calidad son escasas y los riesgos son máximos, los presupuestos se disparan, no es posible entregar los proyectos en fechas y los empleados trabajan más de lo planificado para terminar un proyecto. Se culmina en un nivel de mejora continua o de optimización en el que los procesos de los proyectos y de la organización están orientados a la mejora de las actividades y, en el que no existen riesgos y la calidad es total [13].

Evidentemente la definición del proceso de pruebas en el marco del laboratorio de calidad de la organización debe estar en total concordancia con lo que establece este modelo de madurez. Con motivo de la implantación en la organización del modelo CMMI es necesario establecer un conjunto de elementos que conforman la infraestructura y que hay que tener en cuenta a lo largo de toda la definición posterior [14]. Tal es el caso de los roles, con las responsabilidades, conocimientos y habilidades de cada uno, y las áreas de procesos en que se ve involucrado cada rol; el plan de mitigación y contingencia de los riesgos de la organización; el cronograma de trabajo; el ciclo de vida para los proyectos; entre otros elementos que se definen. Es fundamental conocer y comprender el ciclo de vida del *software* en la organización de manera que se pueda establecer un proceso de pruebas y calidad de sistemas informáticos acorde con las actividades que en la misma tienen lugar.

El ciclo de vida definido en la organización agrupa los procesos de acuerdo a lo que establecen los estándares internacionales, basado fundamentalmente en la ISO 12207 y la ISO/IEC 15288 [15; 16]. Los grupos son Procesos de Ingeniería, Procesos de Gestión de Proyectos, Procesos Empresariales, y Procesos de Soporte. Las disciplinas que se siguen en el grupo de Procesos de Ingeniería fueron definidas tomando como base la propuesta de metodología de desarrollo para la organización, ellas son: Modelado (que incluye Modelado del Negocio, Requisitos y Análisis y Diseño), Implementación, Prueba y Despliegue. El ciclo de vida está compuesto por cuatro fases, según se definió en el marco de la implantación del modelo CMMI y en la propuesta del proceso de desarrollo: identificación, elaboración, ejecución y transferencia [17; 18]. Los proyectos en la organización se desarrollan de forma iterativa e incremental, en cada fase se pueden definir varias iteraciones y se trabaja incrementalmente en la solución pasando por los procesos definidos, en mayor o menor medida en dependencia de la fase. Todos los proyectos, con independencia de su tipo, deben pasar por las cuatro fases. En función de los objetivos y los resultados que se vayan alcanzando en el proyecto se condiciona el pase a la fase siguiente, a la anterior o incluso, la decisión de terminar el proyecto.

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

Esta investigación tiene como objetivo principal redefinir la disciplina de pruebas que se ha propuesto como parte del proceso de desarrollo de software en un entorno universitario con el fin de generalizar la definición propuesta, de manera que pueda ser empleada tanto en un laboratorio de pruebas de *software* como internamente en el equipo de proyecto [18]. Además se busca poner en uso en la organización un proceso que permita controlar las actividades de prueba y así gestionar la calidad de los productos. Para esto es necesario definir las actividades, los roles y los artefactos que componen el proceso de pruebas. Los objetivos antes planteados pretenden evitar y solucionar los problemas detectados al inicio de la investigación con respecto a la calidad de las soluciones y la gestión de las pruebas que se realicen en la organización en sentido general.

II. MÉTODOS

Proceso de Pruebas para el laboratorio en el entorno universitario

Los niveles de prueba que conforman la propuesta se han determinado tomando como base los propuestos en la bibliografía consultada [17; 19]. Además se han tenido en cuenta las características propias del entorno en que se enmarca la solución como por ejemplo el estado en que se encuentran los proyectos de investigación y desarrollo, los que en su mayoría ya han superado las etapas iniciales, algunos incluso están obteniendo sus soluciones finales por lo cual la realización de pruebas desde fases tempranas no es posible. Es necesario entonces definir un proceso flexible que permite adaptarse a esta situación y por supuesto que contemple los aspectos de las fases iniciales. Teniendo en cuenta lo anterior los niveles de prueba para el proceso que se propone quedarían como se muestra en la figura 1.

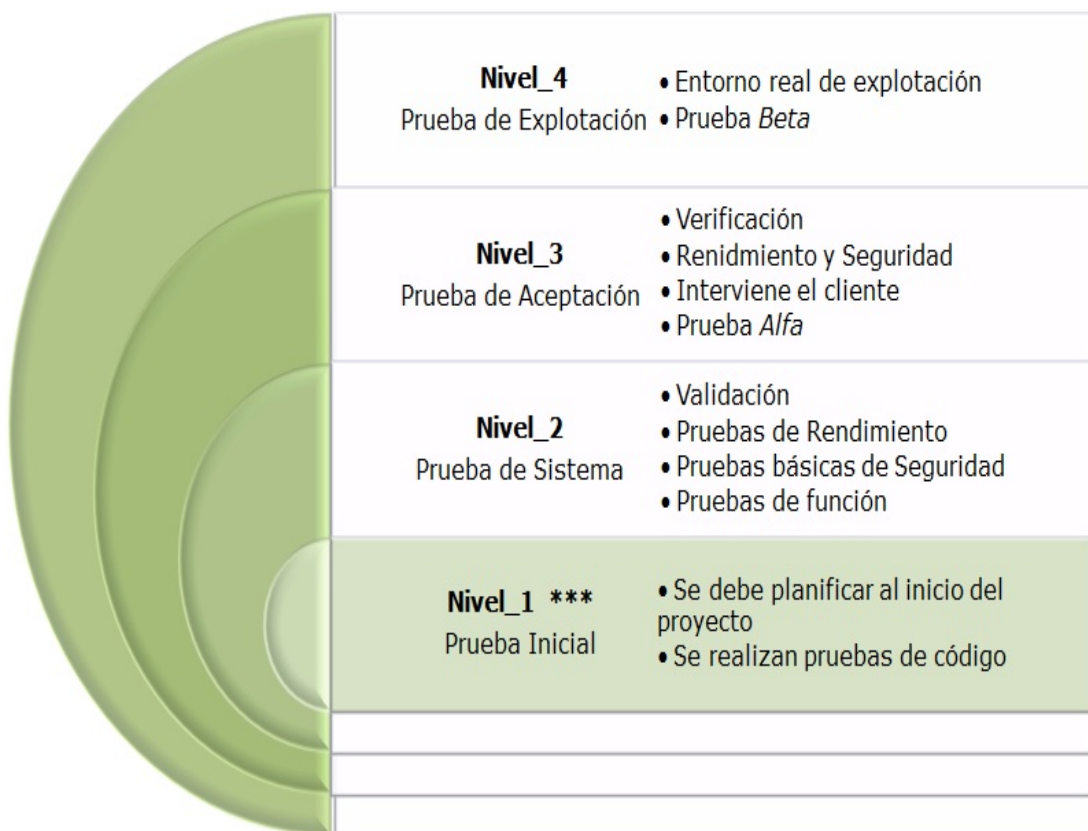


Figura 1. Niveles de pruebas del proceso definido.

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

Nivel 1: Prueba inicial. Este nivel se ejecuta en el marco del proyecto por parte del equipo de desarrollo y debe estar enfocado a los subsistemas independientes de la solución. En el mismo se deben realizar pruebas encaminadas a comprobar cada uno de los caminos posibles buscando errores en la definición de límites para los diferentes ciclos, en la ejecución correcta de decisiones lógicas, etc. La técnica indicada para el diseño de los Casos de Prueba (CP) que se emplearán en la ejecución de este nivel será la de Caja Blanca. El objetivo fundamental de pruebas iniciales es la depuración de las soluciones a nivel de subsistemas o módulos.

Nivel 2: Prueba de Sistema. Esta es una fusión del segundo y tercer nivel que propone la bibliografía consultada, y que por las características del entorno se ha decidido unir en esta propuesta. En este nivel se realizan las pruebas funcionales de la solución y se comprueba la correcta integración de los diferentes subsistemas. La técnica a emplear en este caso para el diseño de los CP será la de Caja Negra. Una vez definidos los mismos se ejecutan, por parte de los integrantes del equipo de pruebas, asumiendo los diferentes roles definidos. El objetivo fundamental de este nivel es validar el producto, o sea, comprobar si cumple con las necesidades del cliente que están especificadas en los requerimientos.

La automatización de las pruebas en este nivel depende de numerosos factores. Hay que tener en cuenta la tecnología empleada para la construcción de las diferentes soluciones, ya que no se automatizan de la misma forma las pruebas de función a sistemas web y de escritorio. En este momento están en estudio un conjunto de herramientas que permitan este fin.

Nota: El procedimiento para la realización del Nivel 1 de pruebas empleando la técnica de Caja Blanca no forma parte de la propuesta que se realiza en este trabajo. Es una definición con características particulares que aún se encuentra en desarrollo.

Además, en este nivel se realizan pruebas básicas de seguridad que incluyen la correcta autenticación en el sistema, el control de acceso o autorización de los distintos tipos de usuarios, el manejo adecuado de sesiones y la gestión de usuarios y permisos de manera general. Se llevan a cabo pruebas de rendimiento a sistemas *web*, diferentes sistemas de bases de datos y servidores FTP (*File Transfer Protocol*), aplicando procedimientos de carga y *stress*. Además, se ejecutan comprobaciones de configuración de la solución. En este último caso la tarea sí está soportada en una herramienta informática que se encuentra en asimilación en estos momentos.

Las pruebas de rendimiento y configuración se incluyen dentro de la comprobación del conjunto de requisitos no funcionales o de calidad [13]. El entorno indicado para realizar este nivel de pruebas será el especificado por el equipo de desarrollo como ideal para su solución. En función de los recursos físicos disponibles en el laboratorio de pruebas y de las características generales del equipamiento con se cuenta, el entorno real en que se realicen las pruebas será lo más cercano posible al ideal propuesto.

Nivel 3: Prueba de Aceptación. En este nivel las pruebas se llevan a cabo conjuntamente con los funcionales o clientes de la solución que se construye. Debido a la diversidad de temáticas en los diferentes desarrollos de la organización la especialización en alguna de ellas, de los integrantes del equipo de pruebas resulta muy complicada. Es por eso que en este nivel se propone la inclusión de los clientes o funcionales. El entorno en que se desarrollan las pruebas sigue estando enmarcado en el laboratorio propio de la organización por lo cual estas comprobaciones constituyen lo que se denomina en la bibliografía como prueba Alfa. En este momento comienzan las pruebas de aceptación de la solución por el cliente que no concluyen sino hasta el próximo nivel.

La técnica a emplear en este caso para el diseño de los CP será igualmente la de Caja Negra. Una vez definidos los mismos se ejecutan por parte de los integrantes del equipo de pruebas, asumiendo los diferentes roles definidos. El objetivo fundamental de este nivel es verificar la solución, o sea, comprobar que el producto

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

que se ha construido funciona correctamente, que las funcionalidades realicen de la manera adecuada lo que el cliente especificó en los requerimientos. En este nivel también se ejecutan pruebas de rendimiento y seguridad básica de la misma forma que se hace en el nivel anterior, con el objetivo de detectar cualquier defecto que pueda haberse introducido a la solución en el pase de un nivel a otro.

Nivel 4: Prueba de Explotación. Este nivel propone la realización de pruebas de explotación de la solución, en su entorno real de despliegue. El objetivo fundamental que se persigue es la comprobación de que el comportamiento de la solución es el indicado en las condiciones reales en que va a ser utilizada. Una vez más, en este nivel la técnica a emplear para el diseño de los CP es la de Caja Negra y la ejecución de los mismos corre a cargo de los integrantes del equipo de pruebas, asumiendo los diferentes roles definidos. Como el entorno empleado en este caso es el correspondiente a los clientes, estas constituyen las Pruebas Beta de la solución.

Como puede observarse en la descripción de cada uno de los niveles propuestos, la aplicación del Nivel 2 y Nivel 3 conjuntamente es posible gracias a la flexibilidad de la propuesta, fundamentalmente porque comparten el entorno y porque las diferentes comprobaciones pueden subsistir de conjunto. A pesar de lo anterior es imprescindible, mantener las perspectivas separadas garantizando la correcta validación y verificación del sistema que son conceptos independientes. Esta es una práctica útil para la organización en los casos en que se necesite acortar el tiempo de entrega de la solución. En la tabla 1 se muestran los roles que intervienen en el proceso.

Tabla 1. Roles y responsabilidades definidos.

Líder del Laboratorio de Pruebas:
Es el encargado de velar porque todas las tareas que se ejecuten en el Laboratorio de pruebas, se lleven a cabo de la manera más eficiente posible, aprovechando el tiempo y los recursos disponibles. Todo el tiempo debe velar porque el proceso de pruebas en el laboratorio se realice basado en las políticas de calidad, definidas en la organización. Responsabilidades: Revisar las solicitudes de prueba, Revisar el Plan de Pruebas, Asignar el especialista encargado de la prueba, Supervisar el montaje del entorno para la ejecución de las pruebas, Supervisar la realización de las pruebas exploratorias, Comunicar los resultados de todas las pruebas a la Alta Gerencia.
Especialista de Pruebas:
Es el máximo responsable de la prueba que le ha sido asignada. Debe garantizar la gestión de los recursos necesarios para llevar a cabo la misma y resolver los problemas que puedan presentarse durante su ejecución. Responsabilidades: Elaborar el Plan de Pruebas, Revisar los CP definidos en el proyecto, Revisar y unificar las NC detectadas durante la realización de la prueba (si existe más de un probador), Montar el entorno para la ejecución de las pruebas, Realizar las pruebas exploratorias, Supervisar el registro y gestión de las NC detectadas, Comunicar los resultados de las pruebas al equipo de proyecto.
Representante del Proyecto:
Es el encargado de mediar entre el equipo de proyecto y el equipo de pruebas. Responsabilidades: Realizar la solicitud de prueba, Proveer toda la documentación del proyecto que sea necesaria para la realización de la prueba, Garantizar que se dé respuesta a las NC detectadas, Montar el entorno para la ejecución de las pruebas.
Probador:
Es el encargado de probar la aplicación y de realizar un levantamiento de NC de la manera más clara, concisa e íntegra posible. Responsabilidades: Ejecutar los CP definidos en el equipo de proyecto, Registrar las NC detectadas durante la prueba en el formato definido para ello, Montar el entorno para la ejecución de las pruebas.

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

Flujo de Trabajo

La definición de esta propuesta está orientada a procesos ya que la obtención de resultados en cada uno de los proyectos, constituye uno de los objetivos fundamentales y prioritarios dentro de la organización. La base del proceso definido ha sido la disciplina de pruebas que propone RUP (Rational Unified Process), eliminando algunas actividades y artefactos que no eran de interés para la organización ni el equipo de calidad. Además se tomaron en cuenta aspectos de la propuesta de proceso de desarrollo para la organización, y añadiendo por supuesto los elementos particulares que no estaban contemplados en los procesos y metodologías estudiadas 16. El flujo de trabajo para el proceso está compuesto por diferentes subprocesos y actividades que garantizan la correcta ejecución del mismo, como se observa en la figura 2.

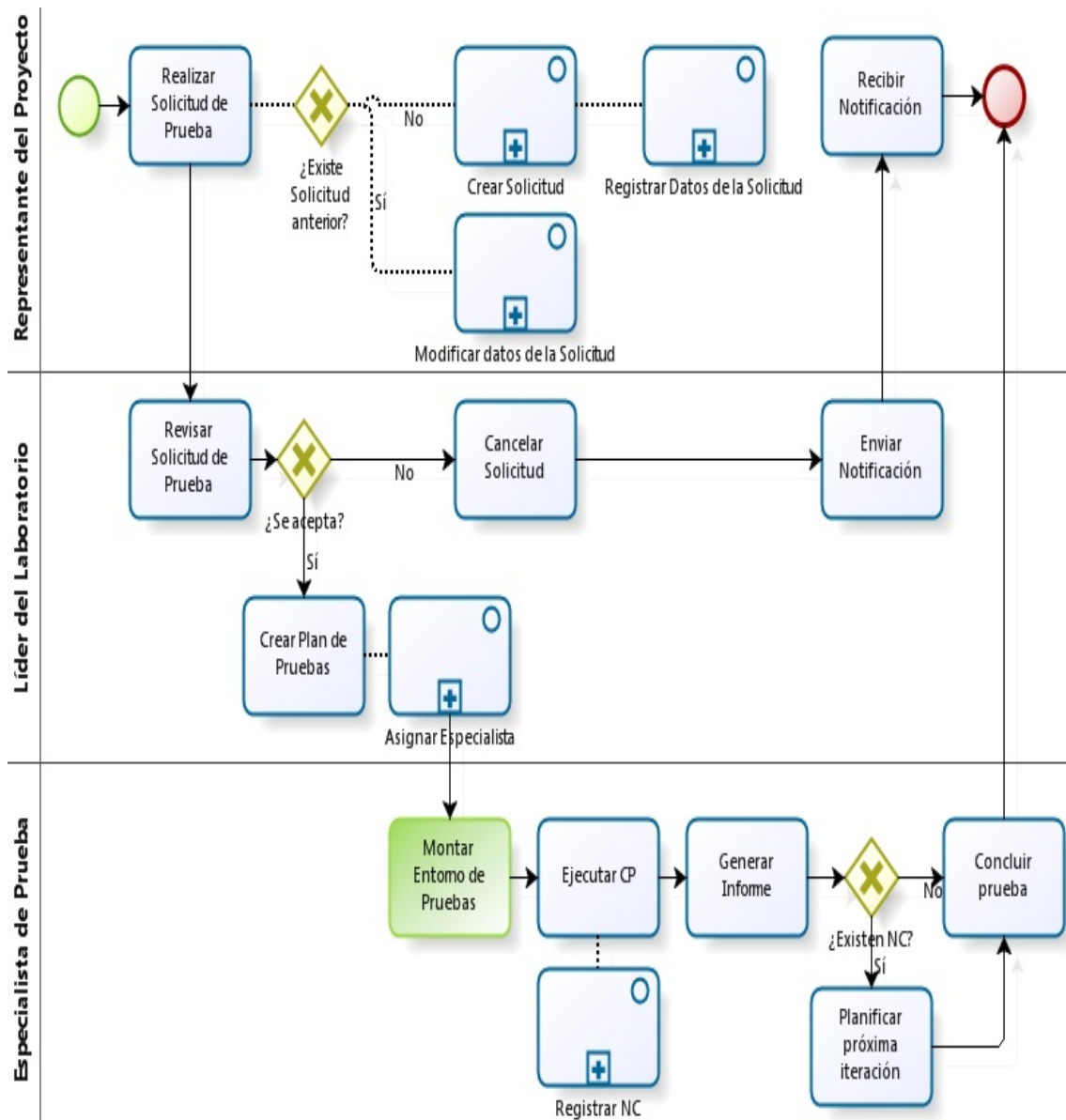


Figura 2. Modelación general del flujo de trabajo del proceso de pruebas.

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

El modelo BPMN del proceso de negocio "Solicitud de Prueba", que se observa en la Figura 3, es el que da inicio al proceso de pruebas definido. Es responsabilidad del líder de proyecto elaborar esta solicitud facilitando toda la documentación que se requiere, como los CP que se quieren probar y sus requisitos asociados, los ejecutables de la aplicación o módulo, los manuales de instalación, configuración y usuario. El artefacto que se genera como resultado de este subproceso es la Solicitud de Prueba, que se crea con el estado Propuesto hasta tanto el Líder del Laboratorio de Pruebas la revise y decida si se acepta o se rechaza. La siguiente figura ilustra la Modelación del flujo de trabajo del subproceso Realizar Solicitud de Pruebas.

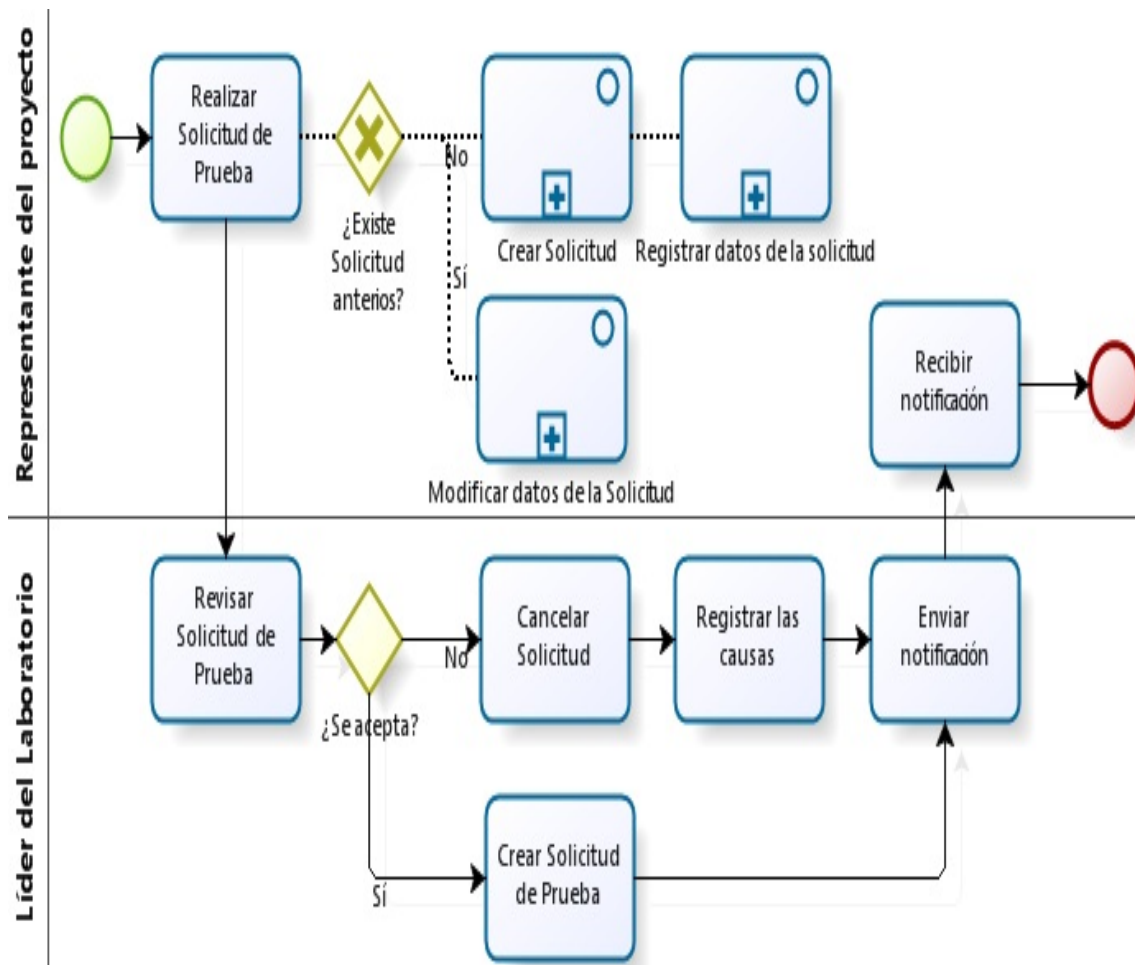


Figura 3. Modelo BPMN del proceso de negocio "Solicitud de Prueba".

La actividad Revisar Solicitud de prueba comienza cuando el Líder del Laboratorio recibe la notificación de la existencia de una nueva Solicitud de Prueba. En este momento se deben revisar todos los elementos que componen la solicitud garantizando que se han completado los datos y que son correctos, que se han adjuntado todos los documentos de apoyo a la prueba y que son válidos los CP diseñados. La definición de los CP es una tarea que se lleva a cabo por parte del equipo de desarrollo. Los elementos fundamentales que lo componen son los pasos a seguir y los requisitos que evalúa.

En la Figura 4 se observa el modelo BPMN del proceso de negocio "Crear Plan de Prueba"- El cual tiene el objetivo fundamental de establecer la planificación en general del proceso de pruebas en el laboratorio de calidad. En este momento se

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

registran los datos iniciales del plan, se asigna el miembro del equipo que asumirá el rol de Especialista de Pruebas, se especifican las propiedades del plan y se definen los conjuntos de prueba y los CP que se incluirán en cada uno de ellos. El artefacto que se obtiene con la ejecución de este subproceso es el Plan de Prueba que contiene diferentes aspectos organizativos importantes concernientes a la prueba que se está iniciando.

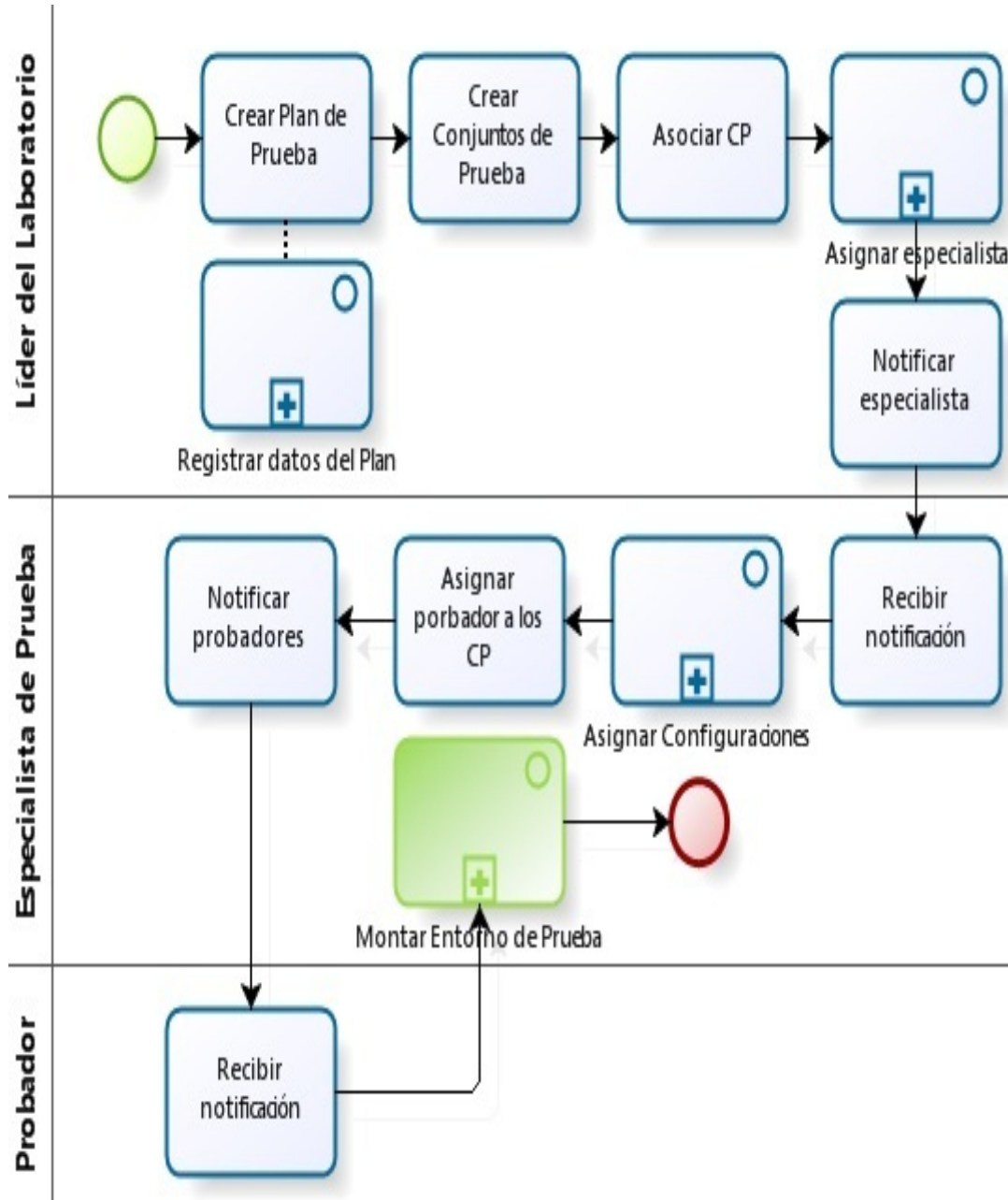


Figura 4. Modelo BPMN del proceso de negocio "Creación de Plan de Prueba".

El modelo BPMN del proceso de negocio "Ejecución de CP" que se observa en la Figura 5, es uno de los más importantes dentro de todo el proceso ya que es el encargado de realizar la comprobación de la solución en sí. El responsable de su ejecución en este caso es el probador, o los probadores que integren el equipo de pruebas. La primera actividad que se debe realizar es la definición de las opciones de ejecución del CP, que establecen cuál será el entorno en el que se ejecuten y las configuraciones, definidas anteriormente. Luego se ejecuta el CP, de manera

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

manual; pero grabando los pasos que se realizan, utilizando el primer juego o combinación de datos que se indicó en los pasos definidos en el CP. Del segundo juego de datos en adelante se reproduce la grabación y se evalúan todas las combinaciones restantes. Si se detecta algún error durante la realización de las comprobaciones se debe entonces generar una No Conformidad (NC) y son estas precisamente el principal artefacto que se obtiene de este subproceso.

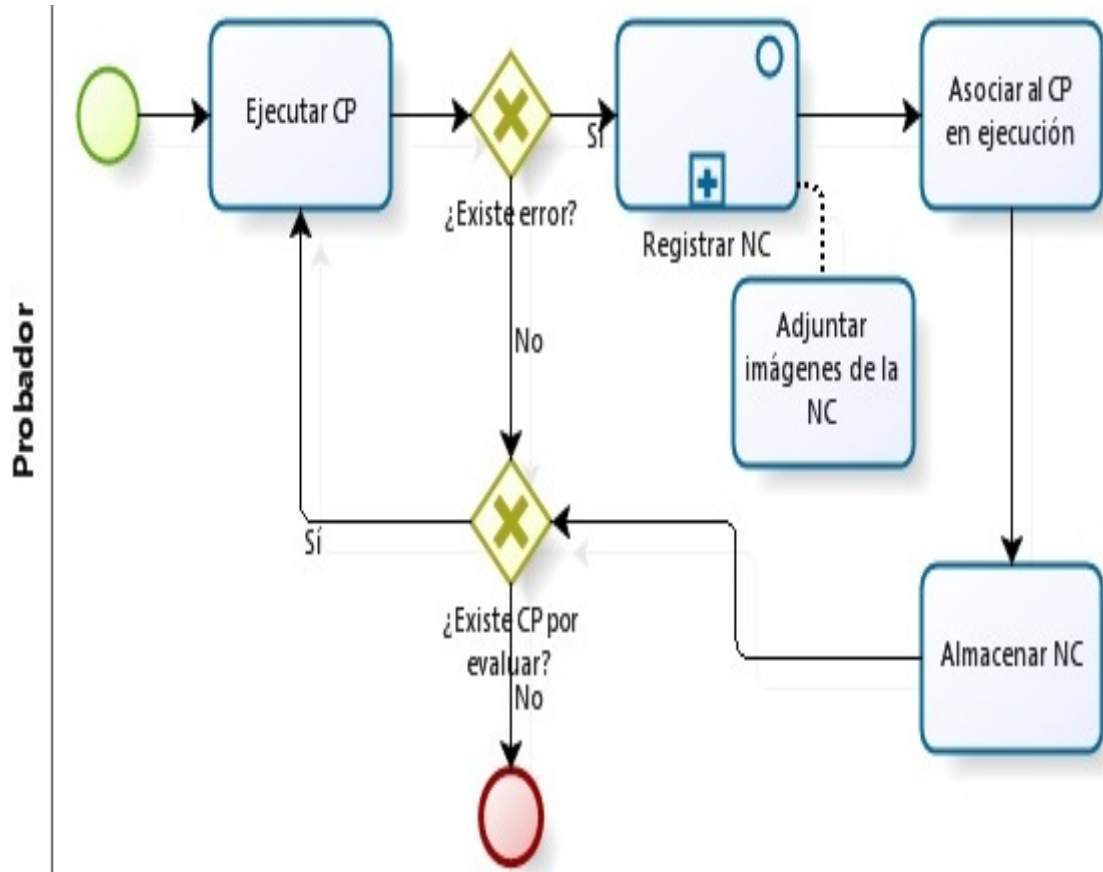


Figura 5. Modelo BPMN del proceso de negocio "Ejecución de CP".

El modelo BPMN del proceso de negocio "Generar Informe" que se observa en la Figura 6, tiene como objetivo fundamental Agrupar todos los resultados relevantes obtenidos durante la evaluación y presentar las conclusiones al respecto a los directivos y demás interesados.

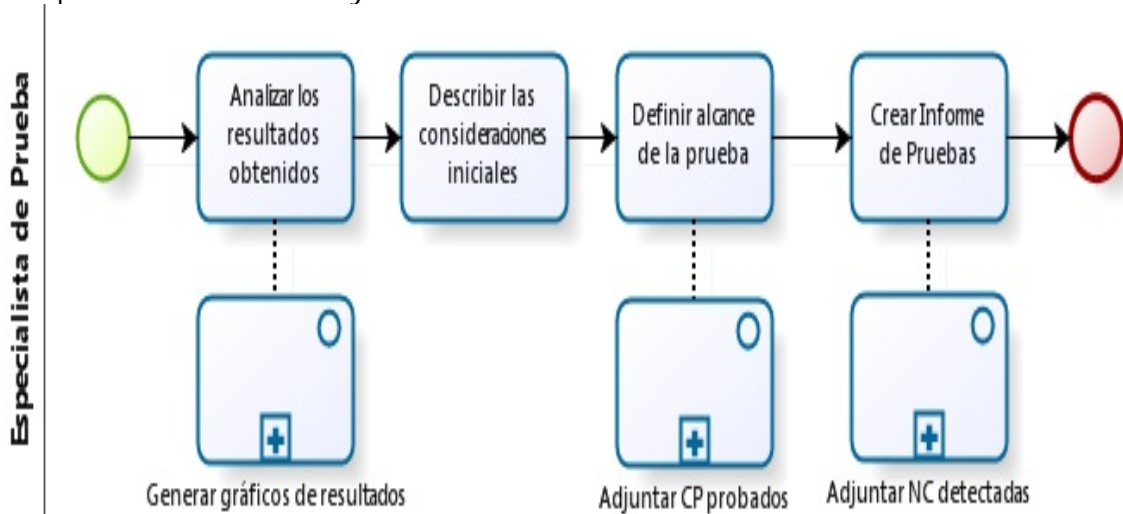


Figura 6. Modelo BPMN del proceso de negocio "Generación de Informe".

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

El proceso se inicia cuando el Especialista de prueba analiza los resultados obtenidos con la ejecución de los CP. Con estos datos el especialista debe generar las gráficas de resultados de la solución y emitir las conclusiones de la prueba. Estas conclusiones deben incluir las valoraciones de calidad que considere oportuno señalar a la solución que se evaluó. Este informe que se genera una vez que se concluye una iteración o una prueba completa de alguna solución incluye aspectos como las consideraciones iniciales que se tuvieron en cuenta en la evaluación. Además, debe especificar el alcance que tuvo la prueba realizada de manera que queden claras cuáles fueron las funcionalidades que se evaluaron por el equipo de calidad. También, se adjuntan por lo general las NC detectadas para que el equipo de desarrollo y los directivos tengan una percepción más clara del volumen de defectos que se detectaron. El artefacto que se genera de este proceso es el Informe Parcial/Final de Pruebas.

III. RESULTADOS

Antes de entrar en el análisis de la aplicación del proceso definido es necesario observar algunos aspectos previos que resultan de interés. Se aplicó una pequeña encuesta a un conjunto de diez proyectos de la organización con el fin de determinar las condiciones en que se realizaban las actividades de prueba en el marco del equipo de proyecto, ante la carencia de un laboratorio especializado. Las preguntas formuladas se listan a continuación y los resultados se observan en el Figura 7.

Listado de preguntas formuladas.

a. ¿Se emplea alguna metodología o proceso para el desarrollo de <i>software</i> ?
b. ¿Se ejecutan al menos algunas actividades de las que propone una de las metodologías o procesos de desarrollo existentes?
c. ¿Se emplea algún procedimiento definido para la etapa de pruebas?
d. ¿Se documentan las pruebas?
e. ¿Se realiza algún seguimiento a las NC detectadas?
f. ¿El equipo de pruebas está compuesto por los propios desarrolladores?
g. ¿Se presentan los resultados de las pruebas al líder del programa de investigación?
h. ¿Se presentan los resultados de las pruebas a los directivos de la organización?
i. ¿Cuál es el tiempo promedio que se invierte en las pruebas de una solución?

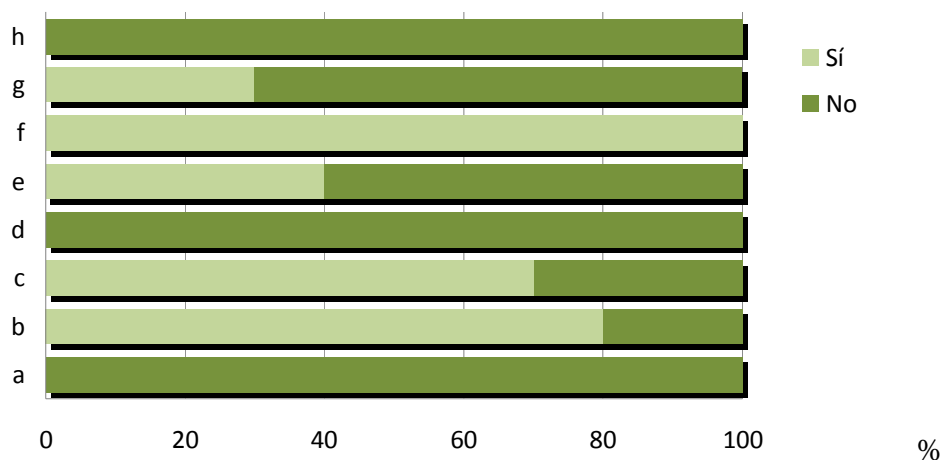


Figura 7. Resultados de la encuesta a un conjunto de proyectos de la organización.

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

Como se puede observar en la figura 7, la totalidad de los proyectos analizados no empleaba una metodología para el desarrollo de sus soluciones ni escalaba los resultados obtenidos en las pruebas que se realizaban en el marco del proyecto. Entre el 70 y 80% de los casos sí se realizan algunas actividades propuestas en diferentes procesos de desarrollo y documentan las pruebas de acuerdo a lo que en ellas se plantea, sin embargo en ningún caso el equipo que realiza las pruebas está compuesto por personal ajeno al proyecto; lo cual es lo recomendado, como se ha visto anteriormente. Además existe un 40% que realiza de alguna forma determinado seguimiento a las NC que se detectan durante las pruebas.

Por otra parte, en la Tabla 2 se muestran los resultados de la aplicación del proceso en un conjunto proyectos seleccionados en la organización en cuanto a: Cantidad de CP diseñados y Cantidad de No Conformidades detectadas (NCD). Los CP diseñados en todos los proyectos abarcan la totalidad de las funcionalidades de la solución en prueba; permite observar la relación de NC solucionadas con respecto al total que se detectó.

Tabla 2. Resultados de la aplicación del proceso en el diseño de CP y detección de NC.

	Iteraciones	CP	NCD
Proyecto A	2	45	39
Proyecto B	2	31 + 7 (rendimiento)	23
Proyecto C	2	17	20

En la figura 8 se muestra el estado de las NC detectadas en los proyectos con la aplicación del proceso.

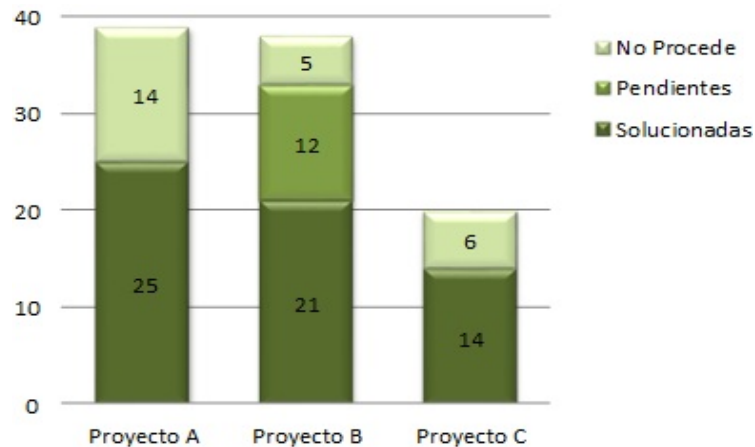


Figura 8. Estado de las NC detectadas en los proyectos con la aplicación del proceso.

Nota: Existen un conjunto de NC, de la totalidad que se detecta durante la realización de las pruebas que son clasificadas como No Procede de acuerdo a los criterios descritos anteriormente y como es lógico estas no forman parte de las NC pendientes de solución.

IV. DISCUSIÓN

Todos los aspectos analizados en la encuesta aplicada, que se describió en el apartado anterior, realizada al conjunto de proyectos mencionados, son importantes para la realización de las pruebas a las diferentes soluciones que se obtienen en la organización.

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

El empleo de esta propuesta permitirá elevar al 100% todos estos indicadores en la totalidad de los proyectos en ejecución. Antes de la existencia de esta propuesta se realizaban las pruebas sin un procedimiento formal por lo que la gestión de las mismas se hacía un tanto complicada y diferente en cada caso particular. Esto trajo como consecuencia dificultades a la hora de documentar los resultados y de comunicarlos a la gerencia de la organización. Además, se ejecutaban los CP sin un orden definido lo cual no es recomendado ya que puede impedir la correcta comprobación de las funcionalidades del producto. Adicionales a las pruebas de funcionalidades del sistema o solución no se ejecutaban otro tipo de comprobaciones como: rendimiento, seguridad, carga, instalación y configuración entre otras. Otro aspecto importante que constituye una dificultad a tener en cuenta cuando se realizan las pruebas sin emplear un procedimiento formal es la resolución de las NC que se detectan, un ejemplo de esto se observa en la Figura 14. Si no se tiene el control adecuado sobre las NC detectadas y las solucionadas se puede trabajar mucho más de lo necesario ya que, por una parte, los desarrolladores invierten tiempo de más en buscar errores que ya se solucionaron y por otra parte, el equipo de pruebas pierde el dominio de lo que se ha probado, y está en realidad listo para su despliegue. La documentación de estas pruebas no se posee; por lo cual no es posible realizar una comparación gráfica con respecto a los resultados obtenidos luego, con la implementación de la propuesta actual.

Por su parte, la aplicación del proceso definido ha permitido la realización de pruebas más objetivas a las soluciones obtenidas en los proyectos. La documentación del proceso se hace de esta forma más efectiva, se le da seguimiento a los defectos que se detectan y solo se cierran cuando se ha comprobado realmente que se solucionaron. Se presentan los resultados a los directivos de la organización, lo cual permite obtener una percepción clara del estado real de los productos de *software* y su nivel de calidad. Con la guía establecida y el empleo de las técnicas definidas con el proceso propuesto se diseñan un mayor número de CP que permiten abarcar mejor las funcionalidades de la solución que se prueba.

Todos los resultados recopilados de la realización del proceso de pruebas en los diferentes proyectos analizados para el presente trabajo permiten elaborar el informe de pruebas del laboratorio de calidad. El informe es la vía para comunicar a los directivos de la organización cuál es el estado de la solución que se ha probado y esto se hace uniformemente para todos los productos de *software* de la entidad. Esto constituye sin dudas una ventaja con respecto a la situación anterior, cuando la información referente a la evaluación de cualquier solución llegaba de las más diversas formas a las manos de la gerencia y en algunos casos ni siquiera se le presentaba.

Actualmente se trabaja en la automatización del proceso con la finalidad de agilizar las tareas que se ejecutan en el mismo. Todos los formatos definidos para el registro de la información asociada a la realización del proceso deberán soportarse en una herramienta que sea capaz de gestionarla y darle seguimiento. Es importante que esta herramienta se encuentre en alineación con la utilizada para la gestión del ciclo de vida ya que esto permitiría mantener las relaciones necesarias entre los diferentes artefactos que se generan durante el proceso de producción. Un primer estudio ha arrojado como resultado más indicado la herramienta *Visual Studio Team Foundation Server 2010*, que provee una solución adicional para la gestión de las pruebas.

V. CONCLUSIONES

Hasta el momento, en la organización la realización de las pruebas de *software* no se lleva a cabo sobre la base de ningún proceso definido. Es tarea era responsabilidad de cada líder de proyecto diseñar y ejecutar las pruebas y luego enviar los resultados a la gerencia para mantenerlos al tanto del estado de la calidad de sus soluciones. Este proceder evidentemente no reportaba los beneficios

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

necesarios ya que en algunos casos las pruebas eran insuficientes e incluso nulas, la información que se reportaba de las mismas era muy diversa y por lo tanto, imposible de procesar para los directivos de la organización. Luego de la investigación llevada a cabo en el marco de este trabajo y de los experimentos realizados se puede arribar a las siguientes conclusiones:

1. La implementación de un proceso de pruebas y su puesta en marcha en un Laboratorio de Calidad, permite elevar la calidad, valga la redundancia, de los productos obtenidos durante el proceso de producción así como organizar y gestionar la actividad de las pruebas funcionales en el ciclo de vida.
2. La información que se obtiene de este proceso ahora se encuentra controlada y es uniforme en soluciones similares lo que permite un control de la calidad más efectivo.
3. Con este procedimiento establecido es posible detectar y dar seguimiento a los defectos de manera que las soluciones que se entregan a los clientes de la organización poseen un mayor nivel de calidad.
4. Los roles involucrados en el proceso tienen claro cuáles son sus responsabilidades y el equipo de pruebas está integrado en su mayoría por especialistas externos al equipo de desarrollo. Con esto se logra una mayor objetividad en la comprobación de los productos ya que se disminuye la posibilidad de omitir la prueba de alguna funcionalidad, como suele suceder cuando es el propio desarrollador quien prueba lo que ha concebido.

En sentido general, la definición del proceso de pruebas constituye un paso de avance en la tarea de garantizar, al menos, cierto nivel de calidad para los productos que se construyen en la organización.

El alcance de esta propuesta no se limita a la organización en cuestión si no que puede aplicarse en cualquier laboratorio de pruebas, ya sea interno, en un proyecto de desarrollo, o externo, a nivel de organización o empresa. En principio está pensado para empresas de desarrollo de *software* ya que sus actividades comprenden la evaluación de soluciones de *software*, sin embargo pudiera aplicarse a otros procesos de producción si se realizan las modificaciones pertinentes en la modelación de las actividades y los artefactos. 📄

VI. REFERENCIAS

1. DERAMAN, A. and YAHAYA, J. "Measuring Unmeasurable Attributes of Software Quality Using Pragmatic Quality Factor". En: *3rd IEEE International Conference (Chengdu Computer Science and Information Technology (ICCSIT) 2010)* 197 - 202 [fecha de consulta: Disponible en: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5564077&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F5550976%2F5563682%2F0564077.pdf%3Farnumber%3D5564077>]
2. YI, Q.; Zhou Bo and XIAOCHUN, Z.,. "Early Estimate the Size of Test Suites from Use Cases ". (Beijing Software Engineering Conference, 2008. APSEC '08. 15th Asia-Pacific 2008) 487 - 492 [fecha de consulta: Disponible en: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4037101&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4037101]
3. KAMDE, P. "Value of Test Cases in Software Testing". En: *IEEE International Conference Management of Innovation and Technology, 2006)* 668-672. [fecha de consulta: Disponible en: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1331307&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F52%2F29396%2F01331307.pdf%3Farnumber%3D1331307>]
4. ZHANG, Y, «Test-driven modeling for model-driven development », [en línea], 2004, vol. 21, no. 5, pp. 80-86 [consulta: 15/10/2012], ISSN 0740-7459 Disponible en: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1331307&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F52%2F29396%2F01331307.pdf%3Farnumber%3D1331307>

PROCESO DE PRUEBAS PARA PRODUCTOS DE SOFTWARE EN UN LABORATORIO DE CALIDAD

5. BETH, M. ; KONRAD, M. ; SHRUM, S., *CMMI. Guía para la integración de procesos y la mejora de productos*, 2da. ed., Madrid, Editorial Pearson Educación, 2009, ISBN 9788478290963, pp. 149-438.
6. HUMPHREY, W., «Best Practices:Acquiring Quality Software», [en línea], 2005, 18, 12, 19-23 [consulta: 2160-1593. Disponible en: <<http://www.worldbooklibrary.org/eBooks/WPLBN0002112533-Crosstalk--The-Journal-of-Defense-Software-Engineering--December-2005--Volume-18--Issue-12---T-by-Johnstun--Kase.aspx>>
7. HUMPHREY, W., «Software Quality: The Software Quality Challenge», [en línea], 2008, 21, 6, 4-9 [consulta: 2160-157. Disponible en: <<http://www.worldbooklibrary.org/eBooks/WPLBN0002112533-Crosstalk--The-Journal-of-Defense-Software-Engineering--June-2008--Volume-21--Issue-8---T-by-Johnstun--Kase.aspx>>
8. PIATTINI, M.G., *Calidad de Sistemas de Información*, 2da ed., Madrid, RA-MA, 2011, 978-84-9964-070-9.
9. PRESSMAN, R.S., *Ingeniería del Software. Un enfoque Práctico*, 5ta ed., Madrid, McGraw - Hill, 2005, 84-481-3214-9.
10. PMBOK, *Guía de los fundamentos para la dirección de proyectos*, Cuarta, Project Management Institute, 2008, 978-1-933890-72-2.
11. IEEE, «Guide to the Software Engineering Body of Knowledge», [en línea], 2004, [consulta: 02-09-2013], Disponible en: <ieeexplore.ieee.org/ielE/4425811/4425812/04425813.pdf>
12. MYERS, G., *The Art of the Software Testing*, Second, Hoboken, NJ, John Wiley & Sons, 2011, 978-1-118-03196-4.
13. ISO, «Systems and software engineering-System life cycle processes», [en línea], 2008, 15288, [consulta: 12-06-2012], Disponible en: <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?cnumber=43564>
14. ISO, «System and software engineering-Software life cycle processes», [en línea], 2008, 12207, [consulta: 14-06-2012], Disponible en: <<http://www.standardsbis.in/Gemini/scoperef/SR16124.pdf>>
15. CITI, «PM-INF-07. Ciclo de Vida», [en línea], 2011, [consulta: Disponible en: <<http://intranet/Nuestro%20Centro/Manual%20de%20Procesos/Forms/AllItems.aspx>>
16. DÍAZ, D., «Definición de un proceso de desarrollo de software en un entorno universitario», Instituto Superior Politécnico "José Antonio Echaverría", Ingeniería Informática, 2011.
17. BLACK, R., *Pragmatic Software Testing-Becoming an Effective and Efficient Test Professional*, Wiley Publishing, Inc., 2007, 978-0-470-12790-2
18. CITI. *Propuesta de Metodología de Desarrollo para el CITI» Manual para el desarrollo de software.*, Complejo de Investigaciones y Tecnologías Integradas, 2010. <http://intranet/Nuestro%20Centro/Manual%20de%20Procesos/Forms/AllItems.aspx>
19. HASS, A., *Guide to Advanced Software Testing*, Artech House, 2008, 978-1-59693-285-2.