

Práctica Empresarial en Pruebas de Software

Trabajo de grado para optar por el título de Ingeniero en Informática

Juan Esteban Herrera Morales

Asesor

Jesús Andrés Hincapié

Ingeniero en Sistemas

Corporación Universitaria Lasallista

Facultad de Ingenierías

Ingeniería Informática

Caldas – Antioquia

2013

Resumen

Un proceso de pruebas requiere de varias etapas que reúnen diferentes aspectos del sistema que se busca evaluar. Identificaremos aquellos aspectos de mayor relevancia para cada etapa explicando aquellas tareas que se llevaron a cabo en cada una. Además, se busca identificar cuáles son aquellos factores de riesgo que pueden influir en la calidad de un proceso de pruebas. Se mostrara como se compone un plan de prueba manejando una determinada metodología y se describirán los procesos e implicaciones a tomar en cuenta en una etapa de corrección de defectos.

Palabras Clave: Casos de Prueba, Defectos, Planes de Pruebas, Metodología, Riesgo

Abstract

A Testing process is composed by several phases that include various aspects about what is going to be evaluated. We are going to identify the aspects with the higher relevance for each phase, while explaining which tasks were executed. Additionally, we want to identify some risk factors that can affect the quality of the testing process.

We are showing how a Testing Plan is composed according to the methodology that is being implemented. Also we are going to describe the processes and implications that have to be taking into account in the Defect Resolution phases.

Key Words: Test Cases, Defects, Testing Plan, Framework, Risk

Contenido

	Pág.
Introducción	5
Impacto tecnológico	6
Objetivo General	7
Objetivos Específicos	7
Marco Teórico	8
Metodología y Resultados	10
Conclusiones	19
Recomendaciones	20
Bibliografía (Referencias)	21

Lista de Tablas

	Pág.
Tabla de Fechas de Creación y Diseño de Casos de Prueba	11
Tabla de Cantidad de Casos de Prueba Ejecutados	14
Tabla de Revisión de Defectos	17

Culminar un proceso tan solo indica el comienzo de uno nuevo. La vida está en constante progreso y cada vez se busca más el generar productos que faciliten y mejoren la calidad de vida, el rendimiento de un negocio o los mismos procesos internos de una organización. Este trabajo es la última fase de mi proceso de formación profesional y, como tal, pone a prueba mi capacidad como ingeniero para enfrentarme en un entorno laboral donde se aplican los conocimientos adquiridos durante mi experiencia académica. Sin embargo, también nos dice que es tan solo el primer paso de un proceso de aprendizaje nuevo, guiado por la experiencia, pero teniendo como base los conocimientos adquiridos durante mí proceso de pregrado.

Busco entonces con este trabajo, cerrar el capítulo más enriquecedor de mi carrera académica, compartiendo mis experiencias y aprendizajes en el entorno laboral, haciendo énfasis en los procesos de pruebas de software en los cuales he participado los últimos meses.

Impacto Tecnológico:

Como ingeniero es imprescindible encontrar maneras de mejorar aquello que ya existe, pues no sólo el enfocarse en crear o generar ideas nuevas puede abrir paso al progreso. Como ingeniero informático, más concretamente, es necesario encontrar formas de mejorar aquellos procesos o productos tecnológicos que contribuyen a mejorar la calidad de vida, la satisfacción de los usuarios y a prevenir que problemas de diversas naturalezas puedan interferir en el normal desarrollo de la tecnología y la innovación.

Dentro de la empresa se buscaba una persona que cumpliera con el perfil necesario para llevar a cabo la creación, diseño y ejecución de planes de pruebas dentro del proyecto al cual fui vinculado. Trabajando con todo el equipo de desarrollo y pruebas, se logró cumplir con las actividades propuestas en los tiempos estipulados lo cual conllevó a que la empresa Productora de Software, PSL, fuera tomada en cuenta para futuros proyectos por parte de la empresa estadounidense BMC Software (Antes Numara Software). Se logró además llevar a cabo con éxito la adaptación hacia una metodología ágil de desarrollo, en contraste a la que se había usado hasta ahora en el proyecto.

Objetivo General

Desarrollar planes de pruebas que contribuyan a mejorar la calidad del software, la mitigación de riesgos y el cumplimiento de las expectativas de los usuarios y clientes del producto.

Objetivos Específicos

Diseñar planes y casos de prueba que permitan asegurar el correcto funcionamiento de las características de un software

Ejecutar planes de prueba que posibiliten la identificación de defectos y errores que puedan desembocar en fallas o daños de gran importancia para los usuarios o para el sistema en general

Realizar un seguimiento constante a los defectos encontrados, así como a las soluciones propuestas para los mismos con el fin de crear planes de prevención y contención en caso de que situaciones similares puedan presentarse en el futuro.

Marco Teórico

RUP:

El Proceso Unificado Racional, es un ejemplo de un modelo de proceso moderno que se derivó del trabajo sobre el UML y el proceso asociado de desarrollo de software unificado. El RUP reconoce que los modelos del proceso convencionales presentan una sola visión del proceso. (Sommerville, 2011)

El RUP consta de cuatro fases principales:

- **Concepción:** Donde se establece el caso empresarial para el sistema. Se identifican las entidades externas que interactuarán con el sistema y definirán dichas interacciones.
- **Elaboración:** Desarrollar la comprensión del problema de dominio, establecer un marco conceptual arquitectónico para el sistema, diseñar el plan del proyecto e identificar los riesgos clave. Al finalizar esta fase, se obtiene un modelo de requerimientos para el sistema.
- **Construcción:** Incluye diseño, programación y pruebas del sistema. Partes del sistema se desarrollan en paralelo y se integran durante esta fase.
- **Transición:** Cuando el sistema de software, listo y documentado, se pone a funcionar en un ambiente real.

SCRUM:

El método Scrum es un método ágil que ofrece un marco de referencia para la administración del proyecto. Se centra alrededor de un conjunto de sprints, que son periodos fijos cuando se desarrolla un incremento de sistema. La planeación se basa en priorizar un atraso de trabajo y seleccionar las tareas de importancia más alta para un sprint. (Sommerville 2011)

Esta metodología ágil ofrece un ritmo de trabajo mucho más acelerado y los incrementos y cambios del sistema se notan con mayor facilidad, sin embargo, durante el desarrollo de cada sprint, puede darse el caso que se reduzca el alcance acordado para los diversos incrementos con el fin de cumplir con los tiempos.

Pruebas de Sistema:

Los componentes del sistema se integran para crear un sistema completo. Este proceso tiene la finalidad de descubrir errores que resulten de interacciones no anticipadas entre componentes y problemas de interfaz de componentes, así como de mostrar que el sistema cubre sus requerimientos funcionales y no funcionales y poner a prueba las propiedades emergentes del sistema. Para sistemas grandes, esto puede ser un proceso de múltiples etapas, donde los componentes se conjuntan para formar subsistemas que se ponen a prueba de manera individual, antes de que dichos subsistemas se integren para establecer el sistema final. (Sommerville, 2011)

Pruebas de Regresión:

Un conjunto de pruebas se desarrolla incrementalmente conforme se desarrolla un programa. Siempre es posible correr pruebas de regresión para demostrar que los cambios al programa no introdujeron nuevos bugs. (Sommerville, 2011)

Son un conjunto de pruebas previamente ejecutadas que se vuelen a correr con el fin de identificar si los cambios producidos en el programa después de un incremento no generan nuevos defectos.

Pruebas de Integración:

La prueba de integración es una técnica sistemática para construir la arquitectura de software mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar componentes a los que se aplicó una prueba de unidad y construir una estructura de programa que determine el diseño. (Pressman, 2007)

Lo que se busca con este tipo de pruebas es comprobar que si todo funciona, de manera individual, como se espera que funcione, también lo haga cuando se junta con los demás componentes del sistema. Razón por la cual, estas pruebas se llevan a cabo antes de realizar las pruebas de sistema.

Metodología y Resultados

A continuación se muestra el progreso que se ha llevado a cabo en el cumplimiento de los objetivos del proceso de práctica. Adicionalmente, se habla de los procesos y conceptos claves a tener en cuenta, mencionados en el Marco Teórico, para el cumplimiento de los objetivos propuestos.

Diseñar planes y casos de prueba que permitan asegurar el correcto funcionamiento de las características de un software

En la siguiente tabla se pueden apreciar las fechas en las cuales se llevó a cabo una tarea de creación y diseño de casos de prueba, pertenecientes a los diferentes planes de prueba que fueron creados. Se muestra además el promedio de casos de prueba creados por día.

Tabla 1: Tabla de Fechas de Creación y Diseño de Casos de Prueba

Fecha	Cantidad de Casos de Prueba creados diariamente
7/10/2012	4
7/11/2012	1
7/17/2012	4
7/23/2012	5
7/24/2012	1
8/3/2012	2
8/10/2012	1
8/13/2012	2
8/14/2012	3
8/15/2012	2
8/16/2012	1
8/17/2012	2

8/30/2012	3
8/31/2012	1
9/3/2012	4
9/4/2012	4
9/5/2012	75
9/6/2012	7
9/7/2012	8
9/10/2012	51
9/11/2012	36
9/12/2012	17
9/13/2012	15
9/14/2012	1
9/17/2012	10
9/18/2012	5
9/19/2012	20
9/20/2012	3
9/21/2012	4
9/25/2012	7
9/26/2012	9
9/28/2012	12
10/1/2012	3
Promedio	10

Adicionalmente, la siguiente gráfica manifiesta de forma acumulativa, el progreso de las tareas de diseño, es decir, la cantidad de casos de prueba totalmente diseñados. Se representa entonces la cantidad acumulada de Casos de Prueba que fueron elaborados para cada fecha en que se ejecuto una tarea de Creación y Diseño de Casos de Prueba:

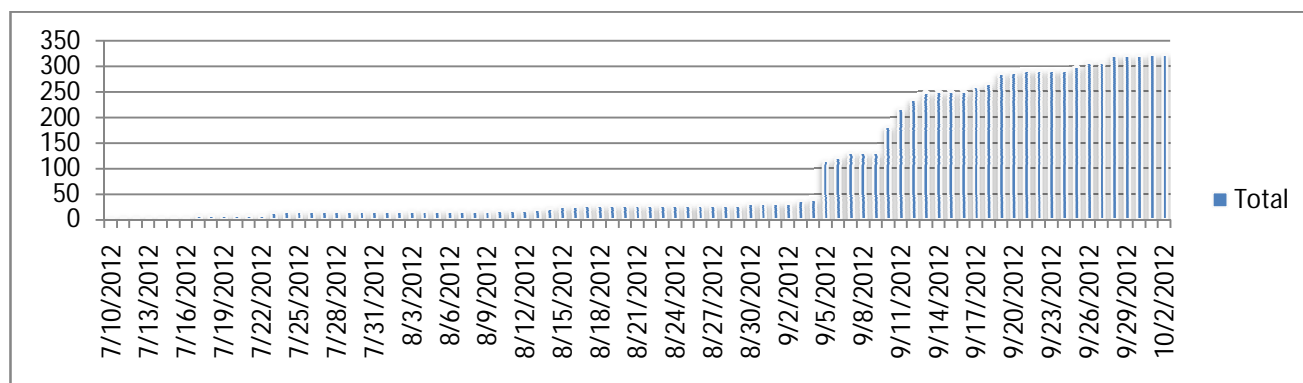


Ilustración 1: Fechas de Creación y Diseño de Casos de Prueba

A partir de la información anterior, concluyo que el progreso sobre este objetivo se muestra alto debido a que se han aprendido los aspectos básicos que deben tenerse en cuenta para crear un plan de pruebas y, así mismo, casos de prueba que permitan cubrir los aspectos esenciales del funcionamiento y comportamiento de una determinada parte del sistema o bien, del sistema completo y su relación con los demás.

Durante el desarrollo del proyecto, se pasó por un proceso de cambio entre dos metodologías diferentes, con el fin de cumplir con las expectativas del usuario en las fechas acordadas. Razón por lo cual, se ha tomado en cuenta los principios esenciales de *RUP* y *SCRUM* para la creación de dichos casos de prueba.

Sin importar cual fuese la metodología empleada en su momento, siempre se ha perseguido el ideal de mitigar riesgos, problemas o defectos dentro del producto.

- **Aspectos a Mejorar:**

Algunos casos de prueba, tanto en la creación como en el diseño, pueden mostrar cierta ambigüedad. Esto no necesariamente se debe a que el requerimiento sea ambiguo, pues en algunos casos, los problemas en la creación y diseño pueden darse por la capacidad del Tester para no extraer de manera explícita información, de lo que es más necesario o riesgoso probar, desde los requerimientos.

Ejecutar planes de prueba que posibiliten la identificación de defectos y errores que puedan desembocar en fallas o daños de gran importancia para los usuarios o para el sistema en general

La siguiente tabla presenta la cantidad de casos de prueba, en total y en promedio, que fueron ejecutados, así como las fechas en las cuales, las tareas de ejecución correspondientes, fueron llevadas a cabo:

Tabla 2: Tabla de Cantidad de Casos de Prueba Ejecutados

Fecha	Casos de Prueba Ejecutados
7/11/2012	1
7/12/2012	16
7/13/2012	8
7/16/2012	3
7/17/2012	12
7/18/2012	6
7/23/2012	37
7/24/2012	27
7/25/2012	2
7/26/2012	3
7/27/2012	3
8/1/2012	1
8/10/2012	13
9/5/2012	13
9/14/2012	24
9/17/2012	9
9/18/2012	8
9/19/2012	14
9/20/2012	27
9/21/2012	34

9/24/2012	32
9/25/2012	12
9/27/2012	16
10/1/2012	24
10/2/2012	1
Total	346
Promedio	13

La siguiente gráfica, muestra el comportamiento en cada una de las fases de ejecución. Se pueden apreciar aumentos y descensos notables en la cantidad de casos de prueba que fueron ejecutados, en las diferentes fechas, producidos por cambios en las estrategias de pruebas y que se llevaron a cabo por los deseos o las necesidades inmediatas del cliente:

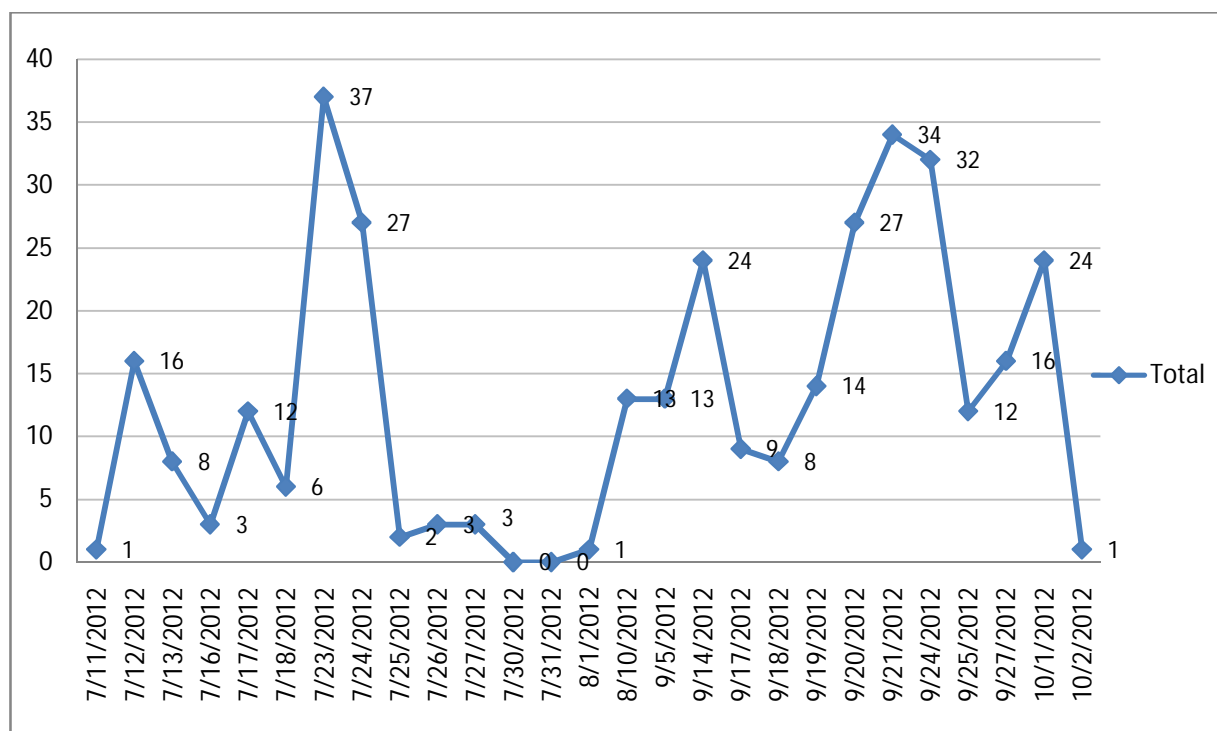


Ilustración 2: Cantidad de Casos de Prueba Ejecutados

Las ejecuciones mostradas dentro de la gráfica incluyen la ejecución de *Pruebas de Sistema*, *Pruebas de Regresión* y *Pruebas de Integración*. Como en cada una de esas fases el proceso y el enfoque varía, la cantidad de casos de prueba ejecutados en cada fase también tiende

a variar. Para cada fase se deben tener consideraciones especiales con respecto a lo que el usuario necesita o desea en un determinado momento, por lo cual, este también es un factor que tiende a influir en el proceso de pruebas y, por consiguiente, tiene una directa relación con la ejecución de ciertos casos de prueba. De este proceso, se aprende y se concluye que probar primero lo más riesgoso, permite dar una visión más amplia del contexto en el que se prueba, por lo cual puede ser un buen punto para retroalimentación sobre el proyecto o el proceso, que desemboca en la posible creación de nuevos casos de prueba o bien, la corrección de algunos casos de prueba ya creados.

- **Aspectos a Mejorar:**

Durante la planeación y estimación de tiempos para las tareas de ejecución, en varias ocasiones, no se tuvieron en cuenta muchos factores que podían influir en el desarrollo de las tareas. Por ejemplo, si se necesitaba probar una funcionalidad que estaba sujeta a un grupo de aspectos preliminares, o precondiciones, y alguna de ellas no se tomaba en cuenta en el momento de la estimación, se invertía parte del tiempo de la tarea de ejecución en hacer cumplir dichas precondiciones. Razón por la cual, en muchas ocasiones, los tiempos resultaban insuficientes para dichas tareas.

Por otro lado, habían otras tareas donde se estimaba una cantidad de tiempo considerablemente mayor a la cual se esperaba invertir, debido a que en el momento de hacer la estimación se presentaban situaciones diversas, como cambios en los requerimientos o falta de total comprensión de los mismos, que llevaban al Tester a programar una cantidad de tiempo poco precisa.

Realizar un seguimiento constante a los defectos encontrados, así como a las soluciones propuestas para los mismos con el fin de crear planes de prevención y contención en caso de que situaciones similares puedan presentarse en el futuro

A continuación, se muestra la cantidad de defectos que fueron creados y cerrados. Se dice que un defecto fue creado cuando se reporta al equipo de desarrollo. Un defecto se cierra cuando el Tester confirma que se le ha dado una solución apropiada, o bien, que el equipo de desarrollo advierte que el defecto no puede reproducirse nuevamente y dicha situación es confirmada por el Tester. En cualquiera de los casos, aquellos defectos que fueron cerrados por el Tester, por cualquier razón, son presentados, dado que en el proceso de seguimiento se deben volver a ejecutar los casos de prueba donde el defecto fue encontrado para comprobar si la solución propuesta es aplicable:

Tabla 3: Tabla de Revisión de Defectos

Fecha	Defectos Creados	Defectos Cerrados
7/11/2012	1	
7/12/2012	1	
7/13/2012	1	
7/17/2012	2	
7/18/2012	2	
7/23/2012	2	
7/24/2012	1	
7/25/2012	2	
7/26/2012	3	
7/27/2012	1	
8/1/2012	1	
8/10/2012	2	
8/13/2012		1
8/14/2012		6
8/15/2012	2	1

9/5/2012	2	
9/14/2012	3	
9/18/2012	1	
9/20/2012	4	
9/21/2012		1
9/25/2012	3	
9/27/2012		2
9/28/2012	3	
Total	37	11

El siguiente gráfico nos presenta la proporción y los totales de los defectos que fueron modificados para cada situación, es decir, la cantidad de defectos que fueron creados y que fueron cerrados con su respectiva fecha. Las barras azules representan la cantidad de defectos que fueron creados y las barras rojas nos indican la cantidad de defectos que fueron cerrados, independientemente de la razón. Estas cantidades se muestran solo para aquellas fechas donde se llevo a cabo una tarea de verificación de defectos:

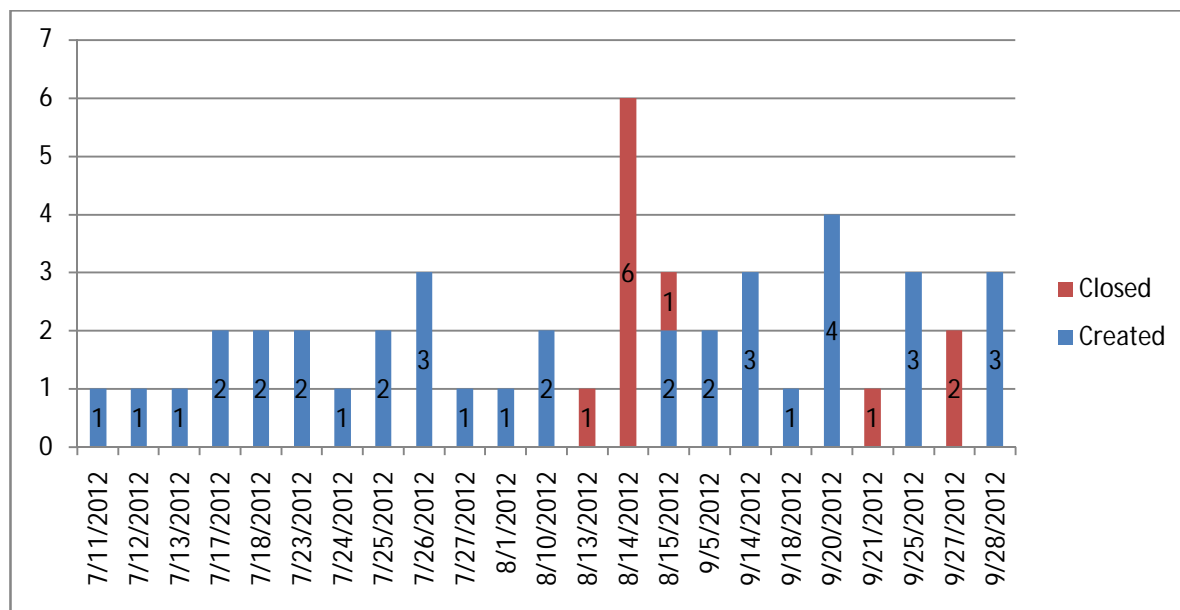


Ilustración 3: Revisión de Defectos

En el proceso de seguimiento que se le da a los defectos, se tiene en cuenta que la comprobación de las soluciones propuestas puede estar sujeta a condiciones muy variadas. Algunos defectos pueden ser particulares dada las condiciones en las cuales se efectúan las

pruebas. Otros pueden ser más importantes que los demás si, el defecto en cuestión, impide que otros casos de prueba pueden llevarse a cabo. Algunos más pueden ser defectos relacionados con la interfaz de usuario. Lo importante de este proceso es llevar a cabo un seguimiento que permita identificar a tiempo amenazas, vulnerabilidades u otros defectos que puedan estar presentes en el producto y que a futuro puedan afectar negativamente en el desarrollo de la aplicación o bien, su funcionamiento una vez sea entregada al usuario.

- **Aspectos a Mejorar:**

Debido a las limitaciones de tiempo existentes, no se ha participado de un espacio o proceso totalmente enfocado a la corrección de defectos. Actualmente, ejecutar pruebas es el único momento donde los defectos toman parte del protagonismo, puesto que es durante esta fase donde se detectan la mayoría de los defectos existentes. Cuando un defecto es solucionado, se crea el espacio para comprobar dicha solución, pero, los defectos se dividen por prioridades y, muchos de los defectos que se reportan, pueden tener una solución un poco más tardía. De cualquier forma, se busca siempre generar el espacio para comprobar la solución de aquellos problemas actuales o potenciales que impliquen un mayor riesgo para el cliente.

Conclusiones

El proceso de pruebas de software no solo sirve para contribuir a la calidad del producto final en cuanto a su rendimiento y desempeño, sino también a la mitigación de riesgos.

El proceso de pruebas hace parte de un proceso de integración más amplio y, por lo tanto, un analista de pruebas debe conocer el funcionamiento completo del sistema y no solo de los componentes que se prueban.

Los procesos de pruebas varían según el proyecto y la metodología general usada en cada proyecto particular, pero eso no quiere decir que un proyecto deba limitarse a una sola metodología.

Recomendaciones

Idealmente deberían poder abarcarse los conceptos más generales de todos los temas relacionados con la ingeniería de software, pero por la forma como se ha estructurado el programa actualmente, no hay tiempo suficiente para esto. Se recomienda abarcar entonces un poco más en el tema de pruebas de software con el fin de darles una perspectiva global a los estudiantes acerca de las diferentes áreas por las que pueden encaminarse en el mundo laboral.

Se recomienda, en la medida de lo posible, impartir los contenidos relacionados con la ingeniería de software desde los primeros semestres y no solo como parte de los énfasis del programa con el fin de abarcar desde temprano una de las áreas más completas y extensas de la carrera.

Finalmente, se sugiere tener en cuenta los diferentes campos de acción que posee el programa en el mundo laboral con el fin de que los estudiantes fomenten el aprecio por el programa y les permita explotar al máximo la calidad del mismo. Lo que se busca es que no se limiten siempre a los mismos roles dentro de una empresa, sino que busquen la manera de tener diferentes opciones a las acostumbradas para que, una vez en el mundo laboral, sean profesionales más competentes y con mayores posibilidades y oportunidades de trabajo.

Bibliografía

SOMMERVILLE, Ian. “Ingeniería de Software” – Novena Edición. México: Editorial Pearson, 2011. P. 42, 50, 51, 78,223

PRESSMAN, Roger S. “Ingeniería del Software” – Un Enfoque Práctico. Sexta Edición. México – Editorial McGraw-Hill Interamericana, 2007. P. 395