

GESTIÓN DE CONOCIMIENTO: LA SOLUCIÓN PARA DISMINUIR EL REPROCESO EN LAS PRUEBAS DE SOFTWARE

Luz A. Perona Ossaba
Universidad de San Buenaventura
luz.perona@gmail.com

Juan E. Velásquez Isaza
Universidad de San Buenaventura
juanes.velasquez@gmail.com

(Tipo de Artículo: **Reflexión**. Recibido el 27/11/2012. Aprobado el 27/12/2012)

RESUMEN

La gestión de conocimiento se ha convertido en una clave fundamental en los procesos de la ingeniería de software, permitiendo mejorar y optimizar cada una de las actividades que se ejecutan durante todo el ciclo de desarrollo de software. Cada una de las etapas de la ingeniería de software ha adoptado de la gestión de conocimiento algunas características, de acuerdo con las necesidades que se presentan, adecuándolos y fortaleciéndolos para reducir los tiempos y disminuir la fuga de información dentro de las organizaciones. Es por esto que las pruebas de software han adoptado la gestión de conocimiento como la alternativa para disminuir el reproceso en las pruebas de software.

Palabras clave

Gestión de Conocimiento (GC), Ingeniería de Software, Desarrollo de Software, Pruebas de Software.

KNOWLEDGE MANAGEMENT: THE SOLUTION TO REDUCE REPROCESSING IN SOFTWARE TESTING.

ABSTRACT

Knowledge Management has become a fundamental key in software engineering processes, allowing the improvement and optimization of all the activities performed during the software development cycle. Each of the stages of software engineering has adopted some features of Knowledge Management, according to the needs, adapting and strengthening them to reduce time and reduce the leakage of information inside companies. For this reason software testing has adopted knowledge management as an alternative to reduce rework.

Keywords

Knowledge Management (KM), Software Engineering, Software Development, Software Testing.

GESTION DE LA CONNAISSANCE: LA SOLUTION POUR DIMINUER LE REPROCESSUS DANS LES TESTS DU LOGICIEL

Résumé

La gestion de la connaissance est devenue comme une clef fondamentale dans les processus du génie logiciel, en permettant d'améliorer et d'optimiser chacun des activités qui s'exécutent pendant tout le cycle du développement logiciel. Chacun des étapes du génie logiciel a adopté quelques caractéristiques, d'après les besoins, en les adaptant et les fortifiantes pour réduire les temps et diminuer la fuite d'information au sein des entreprises. Par conséquent les tests du logiciel on adopté la gestion de la connaissance comme l'alternative pour diminuer le reprocessus dans les tests du logiciel.

Mots-clés

Gestion de la connaissance, génie logiciel, développement logiciel, tests du logiciel.

1. INTRODUCCIÓN

Para el presente artículo nos basaremos en la bibliografía que presentamos durante el desarrollo y en las experiencias que se han obtenido durante la vida laboral, ejerciendo como analistas de pruebas por más de 5 años en varios proyectos de software, desde una perspectiva analítica, interpretativa o crítica. No se hace referencia a las empresas para no comprometer las metodologías y procesos de las mismas, ya que el objetivo es presentar un artículo reflexivo.

Cuando una compañía decide invertir en un proyecto de desarrollo de software, busca entregarle a sus clientes (usuarios internos o externos) un sistema que les ayude a mejorar los procesos o procedimientos que actualmente se están llevando a cabo en la compañía. Para lograr la satisfacción de los clientes, es necesario garantizar la calidad del proyecto de software desarrollado. Por esta razón el proceso de pruebas de software se convierte en clave fundamental para lograr los objetivos planteados al inicio del proyecto. La inyección de defectos a través del ciclo de desarrollo de los proyectos de software es recurrente y no se puede evitar. Durante el proceso de pruebas se identifican los defectos que deben ser corregidos por el equipo de desarrollo, es en este punto donde es importante tener identificados los errores para tomar acciones conocidas dentro o fuera del equipo de desarrollo que permitirán tener una solución rápida y efectiva.

Estas situaciones se presentan en todos los proyectos de desarrollo de software, por lo que los resultados que presentamos de nuestra investigación dentro del artículo, se pueden aplicar tanto en pequeños como en grandes proyectos de software, donde la principal diferencia radica en el número de integrantes del equipo de trabajo y el número de líneas de código generadas al finalizar el proyecto. Dependiendo del tamaño del proyecto, la corrección de defectos puede ser más costosa, por esto es importante que la solución que se plantee pueda ser estándar.

La rotación de personal en los equipos de trabajo y el uso de nuevas tecnologías sin la capacitación adecuada, son algunas de las causas principales para que la presencia de defectos aumente en un equipo de trabajo; en este punto es importante tener una buena gestión de conocimiento dentro del equipo de trabajo o en la compañía, permitiendo el acceso a la información de situaciones reales que se han presentado, las soluciones planteadas y los resultados obtenidos, permitiendo tomar decisiones en cuanto a la mejor solución al momento de abordar la corrección de los defectos identificados [1].

Se han realizado varias aproximaciones en cuanto a sistemas para implementar la gestión de conocimiento en el proceso de pruebas de software, como el QESuite 2.0, plataforma de administración de las pruebas de software desarrollado por Beijing University of Aeronautics and Astronautics (BUAA), por ahora, corresponde a un prototipo que ha sido construido [2].

Dentro de la experiencia, hemos logrado visualizar que para que la gestión de conocimiento sea parte importante del proceso de pruebas de software, no es necesario contar con costosas herramientas que ayuden en el proceso, basta con tener la disposición y una buena metodología para permitir que el conocimiento se transfiera dentro del equipo de trabajo sin limitaciones.

2. EL PROBLEMA DEL REPROCESO EN LAS PRUEBAS DE SOFTWARE

Para las compañías de desarrollo de software, enfrentar nuevos proyectos se convierte en un reto y en cada uno de ellos busca mejorar las prácticas y metodologías que utilizaron en proyectos anteriores.

Dentro del ciclo de vida de los proyectos de software, se desarrollan las pruebas como una actividad que busca garantizar que el producto desarrollado cumpla con las necesidades del cliente, es por esto que se pueden describir como el proceso permite identificar y entregar la calidad de software como un producto basado en la especificación que se ha dado y es requerida por los usuarios. La medición de la calidad del producto basada en las especificaciones de usuario, involucra muchas personas que trabajan juntas para este propósito, como una comunidad de práctica [3].

Una de las preguntas más importantes es cómo integrar eficazmente la gestión del conocimiento en el proceso de pruebas de software para que los activos de conocimiento se puedan transmitir y reutilizar por las organizaciones en el proceso de pruebas [3].

Las pruebas de software son una actividad basada en conocimiento, tanto de negocio como de metodologías orientadas a brindar una solución real a las necesidades del cliente. Sin embargo, el conocimiento acerca de las pruebas de software, habilidades, experiencia e inspiración son muy importantes ya que le permiten a los analistas de calidad alcanzar los objetivos planteados. Si no tiene pensamientos abiertos, experiencias abundantes en pruebas y competencias, la calidad de las pruebas no se puede asegurar [2]. Es importante que los analistas de calidad tengan criterio y argumentos válidos para enfrentar un proceso de pruebas, esto para evitar que por desconocimiento se omitan errores que puedan llegar a ser críticos dentro del sistema implementado.

Las pruebas de software se basan en dos procedimientos, verificación y validación. La verificación consiste en hacer chequeos que certifiquen que el sistema funciona de acuerdo con las especificaciones y la validación consiste en asegurar que el sistema hace lo que se supone deba hacer, es decir que la especificación este correcta [4].

Si miramos el proceso de pruebas de software como el conjunto de actividades relacionadas con 1) verificación del desarrollo del producto, este debe estar basado en las necesidades del cliente, 2) reporte de los defectos

encontrados y 3) solución de los defectos por parte del equipo de desarrollo, nos damos cuenta que este proceso se puede repetir en muchas oportunidades, hasta que se logre encontrar una solución óptima y adecuada para el defecto encontrado. Es por esto que el reproceso es tan evidente dentro de esta etapa del ciclo de vida de desarrollo. La principal razón, que genera tanto reproceso se deriva de que los desarrolladores al momento de enfrentarse con un desarrollo no encuentran a su disposición la información necesaria para hacerlo o porque no se tiene la capacidad de indagar o al entrar a resolver estos incidentes no encuentran información que les pueda minimizar el trabajo debido a que es posible que otros desarrolladores o a ellos mismos les haya pasado algo similar pero ya no se acuerdan cómo lo resolvieron o simplemente la persona o personas que de alguna manera pudieran colaborar ya no se encuentran en la organización.

Esta problemática se presenta en todos los proyectos de software, sin importar el tamaño; pero mientras más grande el proyecto puede llegar a ser más crítico y costoso, ya que cualquier solución implementada para corregir el defecto puede afectar otros artefactos de desarrollo, generando nuevos errores. Es en este punto es importante involucrar la gestión de conocimiento en el proceso de pruebas de software, ya que desde la perspectiva del analista de calidad se puede identificar situaciones previamente encontradas y las soluciones que se han entregado a ellas.

Con la gestión del conocimiento se pueden establecer formas de manejar la información que otras personas conocen y que no se ha dejado almacenada en ningún medio, ayudando a formar grupos de trabajo que se encarguen de entrevistar los diferentes participantes o invitarlos a formar bases de datos de conocimiento acerca de todas las experiencias, de todos los incidentes que se les hayan presentado en algún momento del ciclo de desarrollo. De esta forma se podrían minimizar los costos en la solución de incidentes y permitir que las pruebas de software hagan parte importante de este ciclo de desarrollo. Una vez se identifiquen los incidentes durante el proceso de pruebas, el desarrollador tiene la capacidad y la información a la mano para darle solución de manera más rápida y eficaz a dichos incidentes.

Se debería entonces entender la gestión de conocimiento, como una disciplina que permite consolidar el capital intelectual en las organizaciones. La gestión de conocimiento apoya no sólo los conocimientos técnicos de una empresa, sino también el dónde, quién, qué, cuándo y por qué suceden las cosas [5].

Lo anterior nos lleva a que dentro de las organizaciones, el conocimiento no se encuentre centralizado en un grupo particular de personas, sino que se pueda ir transfiriendo a otras personas que lo requieran. Es aquí, donde la gestión del conocimiento tiene su primer obstáculo, ya que todos sabemos que

dentro de la mentalidad del ser humano está que mientras se sienta dueño del conocimiento, se tendrá asegurado un lugar en la organización. Lograr que el conocimiento se extienda, implica que las personas sean conscientes de la importancia de trabajar en equipo, buscando el logro de objetivos y metas, sin importar quién sea el dueño del conocimiento.

El concepto de gestión de conocimiento, surgió a mediados de 1980 por la necesidad de obtener conocimiento de la "avalancha de información" y fue utilizado principalmente como un término del mundo de los negocios [5]. Más de 10 años de investigación en la gestión de conocimiento promueven y marcan muchas herramientas de tecnologías de información -IT- para la gestión, pero no todas satisfacen los requerimientos particulares de cada empresa, especialmente en campos especializados.

Desde que se comenzó a escuchar el concepto de gestión de conocimiento, se evidenciaba la transformación que estaba ocurriendo en muchas organizaciones [6].

Entonces, ¿por qué es importante que la gestión de conocimiento se adhiera a los procesos de ingeniería de software? Porque los procesos de ingeniería, en especial el desarrollo de software, es muy cambiante, requiere de gran conocimiento del negocio e involucra muchas personas trabajando en diferentes fases y actividades [5]. Es en este punto donde vemos que la gestión de conocimiento puede apoyar el proceso de pruebas de software, porque en proyectos en los que la rotación de personal es muy alta el conocimiento previamente adquirido, se va con las personas y en ocasiones recrear situaciones detectadas con anterioridad puede tener un costo muy alto por el desconocimiento que se tiene tanto del negocio como de las metodologías utilizadas.

Estudios previamente realizados [5], nos llevan a entender que para implementar la gestión de conocimiento en una compañía es necesario enfocarse en la aplicación de siete procesos:

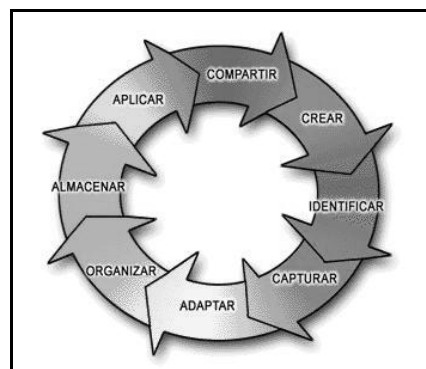


Fig. 1. Integración de los procesos de gestión de conocimiento (Elaboración propia)

Teniendo en cuenta los procesos de la gestión de conocimiento aplicada al proceso de pruebas, en cada uno de ellos obtendríamos resultados que mejorarían el

reproceso que se presenta actualmente en esta etapa del ciclo de desarrollo; la creación, permite crear conocimiento previamente adquirido, sea dentro o fuera del proyecto en curso; la adquisición, permite obtener conocimiento de los demás integrantes de proyecto que ayuden a solucionar incidentes detectados con características similares a otros ya identificados previamente; la identificación puede llegar a ser un poco más compleja, debido a la cantidad de posibles soluciones que surjan dada la similitud en errores previamente detectados; la adaptación, permite mantener actualizadas las bases de conocimiento, para que los miembros de la organización tengan acceso a ella en el momento que los requieran, es importante establecer políticas de actualización de la información, pero sobre todo se debe destinar el tiempo y los recursos para realizar esta actividad que sin ella la gestión de conocimiento no tendría la información disponible; la distribución de conocimiento es el proceso más importante de la gestión de conocimiento, debido a que sin unos canales adecuados de distribución de la información, no sería posible acceder a ella en el momento en que se requiera; la aplicación del conocimiento, permite crear conciencia en los miembros de los equipos de desarrollo de aplicar el conocimiento existente para agilizar los procesos y minimizar el riesgo de reproceso por soluciones no adecuadas.

Luego de implementar gestión de conocimiento dentro de los procesos de ingeniería de software, vemos que es posible mejorar el reproceso en cada una de las etapas, incluyendo el proceso de pruebas de software tema de este artículo.

El análisis realizado del proceso de pruebas de software, en algunas compañías, usando los principios primarios de la gestión de conocimiento, fueron identificados cinco problemas importantes en el proceso [2]:

1. Baja tasa de reuso de conocimiento en las pruebas de software: El conocimiento en las pruebas de software no se ha almacenado concienzudamente. Algunas bases de conocimiento se han implementado pero generan gran discusión en los grupos de trabajo.
2. Barreras en la transferencia de conocimiento en las pruebas de software: La administración del conocimiento de pruebas es muy difícil de transmitir.
3. Ambiente pobre para compartir el conocimiento de las pruebas de software: Los miembros del equipo tienen poco tiempo en común para compartir experiencias.

4. Pérdida significativa en el conocimiento de las pruebas de software: Los conocimientos y experiencias obtenidas por el equipo de trabajo en pocas ocasiones se convierten en conocimiento público. Adicionalmente, la rotación de personal lleva a la pérdida de conocimiento en pruebas.
5. Imposible obtener la más óptima distribución del recurso humano rápidamente: La gestión de conocimiento es una integración entre personas, procesos y tecnología, donde las personas son la parte más importante. Sin tener un conocimiento adecuado de la información de cada persona, los recursos no se asignan adecuadamente dentro de los proyectos de software.

De acuerdo a lo planteado en el artículo Administración de Conocimiento en Desarrollo [1], los autores describen cuál es el conocimiento, cómo lo puede usar un desarrollador y cómo se puede implementar en la práctica una base que lo contenga.

A. Conocimiento sobre la corrección de defectos

Primero se debe definir cuál es el conocimiento asociado a la corrección de defectos. Esto quiere decir que la información que se recopila acerca de un defecto y de su corrección puede ser orientada para facilitar la corrección de defectos similares en el futuro. Por esta razón, se considera apropiado definir un modelo general de defectos, en el cual todos los defectos siguen una caracterización estándar bajo la cual se tipifica su descripción y su solución.

B. Análisis y uso del conocimiento

Una vez se cuenta con una base de casos el siguiente problema a resolver es cómo hacer uso de ella. Para esto se decidió adoptar el modelo de razonamiento basado en casos (Case-based reasoning) o CBR. Este modelo sirve para abordar problemas complejos, cuyo análisis por medio de aproximaciones convencionales no es posible o no da buenos resultados. El supuesto principal del CBR es simple: *problemas similares deben tener soluciones similares*.

En síntesis, este razonamiento consiste en que, dado un problema, obtengamos una solución a éste por medio de la adaptación de una o varias soluciones dadas a problemas parecidos. La figura 2 muestra el esquema general de este razonamiento.

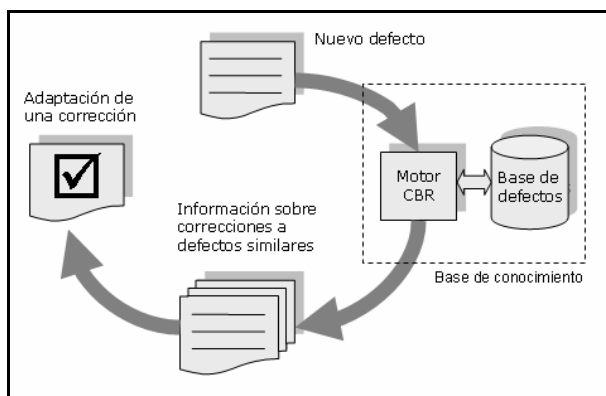


Fig. 2. Esquema general de este razonamiento (Elaboración propia)

La razón principal para su adopción es que un defecto no puede solucionarse siguiendo siempre los mismos métodos, ya que no hay procedimientos sistemáticos que ayuden a localizarlos y a corregirlos. No obstante, la experiencia adquirida cuando se corrigen los defectos sí puede servir para solucionar los defectos similares que aparezcan en el futuro; y es en esa experiencia la que deberían aprovechar los desarrolladores en el momento en que intentan corregir uno en particular.

Si logramos identificar dentro del proceso de pruebas de software, las posibles soluciones que se puedan implementar para los defectos encontrados, el desarrollador tendría una visión más clara de la solución requerida por el analista de calidad, de acuerdo con las necesidades de los clientes.

Como parte de la estrategia para mejorar las prueba de software y parte de ellos está basada en la creación de un sistema de soporte a la toma de decisiones. Sin embargo, mucho del esfuerzo es dedicado al proceso de extracción de conocimiento, para que los errores no se lleven a la base de conocimiento, aunque no es razonable esperar una base de conocimiento libre de errores. Muchas herramientas y procedimientos se han desarrollado para ayudar a detectar errores en los sistemas de apoyo en la toma de decisiones [4]. Pero no sólo se han desarrollado herramientas para detectar errores en las bases de conocimiento, sino que también se han adelantado investigaciones de cómo incluir herramientas de gestión de conocimiento en el proceso de pruebas de software.

Se han invertido grandes esfuerzos tanto en tiempo como en dinero para lograr implementar una herramienta que permita gestionar el conocimiento en las pruebas de software. Grandes empresas como IBM y Microsoft, han invertido dinero y puesto a disposición mano de obra para la investigación en gestión de conocimiento y han presentado una serie de teorías y desarrollado productos de software. Debido a los grandes esfuerzos realizados, sin obtener los resultados esperados se ha instaurado la conferencia, KBSE (Knowledge-Based Engineering Conference Software) que se celebra cada año, en la que se discute el último avance de la gestión del conocimiento en las pruebas de software [7].

En cuanto a software desarrollado para lograr incluir la gestión de conocimiento en las pruebas de software se refieren al prototipo desarrollado por Beijing University of Aeronautics and Astronautics (BUAA), el sistema actual es un subsistema de QESuite2.0. El sistema es una plataforma de software de gestión del conocimiento orientada al proceso de prueba. El sistema utiliza el conocimiento del ciclo de vida como guía. Éste puede ayudar a las empresas a almacenar, gestionar, buscar y compartir todo tipo de conocimientos mediante el uso de los documentos de conocimiento. Se puede evaluar el nivel de conocimientos del personal utilizando mapas de conocimiento.

El sistema confirmará el personal que tenga conocimiento por las estadísticas y que van a mejorar la cultura de intercambio de conocimientos en la empresa. El módulo de mapas de conocimiento es el núcleo del sistema. Éste se divide en dos partes: la red de especialistas y la construcción del equipo de pruebas. Los usuarios comunes pueden editar su experiencia de proyecto. Los analistas de conocimiento tienen derecho a seleccionar a otros usuarios para editar su experiencia de proyecto. Después de editar la experiencia de proyecto, el sistema automáticamente definirá el nivel de conocimiento de los usuarios en función del tiempo que los usuarios utilizan estas tecnologías. Para alcanzar el nivel de especialista, éste debe ser editado por el analista de conocimiento [2].

En el entorno local, se han realizado esfuerzos para involucrar el proceso de pruebas en etapas más tempranas del ciclo de desarrollo, mejorando la calidad de cada uno de los entregables. Como resultado de estas prácticas se han obtenido avances y resultados satisfactorios:

1. Conocimiento del negocio desde el comienzo del ciclo, cuando el analista de calidad se involucra desde la etapa de conceptualización.
2. Especificación de requerimientos apoyada por el analista de calidad orientada a soluciones posibles técnicamente.
3. Desarrollo de software basado en pruebas. Corresponde a una nueva práctica de desarrollo conocida como TDD (Test Driven Development) que consiste en que el desarrollo se realice basado en los casos de prueba diseñados por el analista de calidad, garantizando que el desarrollo realmente está acorde con las necesidades del cliente.

Si miramos estos puntos como orientados a la gestión de conocimiento aplicada al proceso de pruebas de software, vemos que:

1. Se minimiza el reproceso por las diferencias en la interpretación de los requerimientos, entre el analista de calidad y el desarrollador.
2. Se permite la integración de cada uno de los actores del ciclo de desarrollo, especialmente entre los roles analista de calidad y desarrollador.

3. La utilización de los TDD, permiten que el desarrollador retroalimente los casos de prueba diseñados por el analista de calidad o diseñar nuevos casos que permitan garantizar las necesidades del cliente en el proyecto de software.
4. El número de errores encontrados en la etapa de pruebas se disminuye, debido a que el desarrollador ejecuta los casos de prueba al finalizar la etapa de desarrollo.

3. CONCLUSIONES

La gestión de conocimiento viene a apoyar a las organizaciones, para que el proceso de administración del conocimiento sea más efectivo y eficaz, permitiendo que éste pueda estar disponible en todo momento para los miembros de la organización.

Las investigaciones desarrolladas para integrar la gestión de conocimiento con las pruebas de software, han arrojado resultados tanto teóricos como prácticos, pero aún hace falta realizar más investigaciones que permitan obtener mejores resultados en el proceso, que realmente garanticen que el reproceso que se genera en las pruebas de software se disminuye en todos los proyectos, sin importar el tamaño o características particulares.

Cuando una compañía logre implementar de manera exitosa un buen proceso de gestión de conocimiento aplicado al proceso de pruebas de software, se podría garantizar que los tiempos de ejecución de la etapa de pruebas se verían reducidos significativamente y la capacidad de identificación y solución de los mismos será más efectiva.

REFERENCIAS

- [1] F. Romero, R. Rueda, N. López & R. Casallas. [“Knowledge Management in Software Development to Reduce Defect Correction Effort”](#), presentado a III Congreso Colombiano de Computación, Medellín, 2008.
- [2] L. Xue-Mei, G. Guochan, L. Yong-Po & W. Ji. [Research and Implementation of Knowledge Management Methods in Software Testing Process](#). IEEE 978-0-7695-3688-0/09. International Conference on Information Technology and Computer Science, 2009.
- [3] R. Abdullah, Z. Darleena Eri & A. Mohamed Talib. [A model of knowledge management system in managing knowledge of software testing environment](#). IEEE 978-1-4577-1531-0/11. 5th Malaysian Conference in Software Engineering (MySEC), 2011.
- [4] A. Terry Bahill, Fellow, IEEE, K. Bharathan & Richard F. Curlee. [How the testing techniques for a decision support system changed over nine years](#). IEEE Transactions on Systems, Man, and Cybernetics, Vol. 25, No. 12, December 1995.
- [5] I. Rus & M. Lindvall. [Knowledge Management in Software Engineering](#). IEEE Software, pp. 26-38, 2002.
- [6] J. Ward & A. Aurum. [Knowledge Management in Software Engineering – Describing the Process](#). Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04), 2004.
- [7] Y. L. Ji Wu, & X. L. Guochang Gu, [Investigation of Knowledge Management Methods in Software Testing Process](#), International Conference on Information Technology and Computer Science, IEEE 978-0-7695-3688-0/09, 2009.