

# Aplicación de la Gestión de Conocimiento al proceso de pruebas de software

Applying Knowledge Management to the software testing process

Darío Enrique Soto Durán<sup>1\*</sup>, Adriana Xiomara Reyes Gamboa<sup>2</sup>, y Jovani Jiménez Builes<sup>3</sup>

## Resumen

La gestión del conocimiento tiene como propósito aprovechar el conocimiento para generar valor a nivel organizacional, soportado en procesos, herramientas y actividades para lograr el propósito. En este contexto, las pruebas son un proceso relevante y contribuye en gran medida al aseguramiento de la calidad del producto. A nivel académico y empresarial, existen varias iniciativas para articular la gestión del conocimiento a las pruebas de software, teniendo como problemática común, la falta de reutilización del conocimiento que se deriva en múltiples factores, como: las barreras para compartir el conocimiento, la alta rotación de personal, modelos de gestión de conocimiento con enfoque corporativo y compleja implementación entre otros. En consecuencia, se propone un modelo de implementación para gestionar el conocimiento en los proyectos de pruebas de software, desarrollado por seis procesos: definición de objetivos, identificación, incorporación, preservación, distribución y utilización, basado en artefactos con el propósito de mejorar las pruebas de software.

## Abstract

The purpose of knowledge management is harnessing knowledge to generate value at the organizational level, supported in processes, tools, and activities to achieve the purpose. In this context, testing is a relevant process which contributes greatly to the assurance of product quality. To academic and business levels, there are several initiatives to link knowledge management to software testing, having as a common issue the lack of reuse of knowledge that results in multiple factors e.g. knowledge sharing barriers, high rotation of personnel, knowledge management models with corporate focus and complex implementation, among others. Therefore, an implementation model is proposed to manage knowledge in software testing projects developed by six processes: Definition of objectives, Identification, incorporation, preservation, distribution, and use based on artifacts for the purpose of improving the software tests.

## Palabras Clave

Gestión del conocimiento; pruebas de software; estrategia de conocimiento; activos de conocimiento.

## Key words

Knowledge management; software testing; knowledge strategy; knowledge assets.

<sup>1</sup>Facultad de ingeniería, Tecnológico de Antioquia I.U., Medellín, Colombia

<sup>2</sup>Facultad de ingeniería, Politécnico C. Jaime Isaza Cadavid, Medellín, Colombia

<sup>3</sup>Facultad de Minas, Universidad Nacional de Colombia, Medellín, Colombia

\*Corresponding author: jjime@unal.edu.co

Manuscript received 20-11-2016; revised 15-03-2017; accepted 11-05-2017.

## 1. Introduction

Las pruebas de software es un proceso relevante dentro del ciclo de vida de desarrollo de software, debido a que incrementa la calidad de los productos y artefactos producidos. Las pruebas de software son una actividad intensiva en conocimiento, porque dependen de las habilidades, conocimientos, intuición y experiencia de los miembros que integran el equipo de desarrollo de pruebas [1]. Por ello, Xue-Mei y colaboradores identifican algunos problemas en el proceso de pruebas de

software en [2].

1. Baja tasa de reutilización del conocimiento de pruebas de software.
2. Existen barreras en la transferencia de conocimiento de pruebas de software.
3. Falta un entorno adecuado para el intercambio de conocimiento del proceso de pruebas de software.

4. Alto nivel de rotación de personal, generando una pérdida permanente del proceso de pruebas de software.

Esta problemática puede mitigarse, integrando eficazmente la gestión del conocimiento (GC) a actividades de verificación y validación, para que los activos de conocimiento se transmitan y reutilicen, y redunde en un proceso de pruebas de software eficiente [3].

Por lo anterior expuesto, en este artículo se propone un modelo de implementación para la gestión del conocimiento en el proceso de pruebas, teniendo como premisa un dominio unificado derivado del estándar ISO/IEC 29119, basado en tres capas del proceso: estratégico u organizacional, gestión de las pruebas y el proceso fundamental o técnico de la prueba. El modelo apoya la estrategia de prueba basada en riesgos e instancia la tipología de conocimiento relevante para esta disciplina y genera valor al proceso de pruebas.

El artículo se encuentra organizado en tres secciones la primera presenta la contextualización de los temas que se abordan en la investigación, la segunda presenta los resultados y discusión del modelo propuesto, y la tercera presenta las conclusiones.

## 2. Materiales y métodos

### 2.1 Gestión de conocimiento

La Gestión de Conocimiento (GC) se soporta desde la comprensión del concepto de conocimiento. “El conocimiento es una mezcla fluida de experiencia estructurada, valores, información contextualizada y discernimiento experto que provee un marco de referencia para evaluar e incorporar nuevas experiencias e informaciones” [4].

La GC, comprende tres elementos.

1. Conceptualizar las diferencias y la relación existente en el trial de los conceptos: dato, información y conocimiento.
2. La instanciación que realiza un individuo por medio del procesamiento cognitivo de los datos y la información, para ser usados en un contexto específico.
3. La concepción de conocimiento, el cual es propia del individuo, su utilidad está sujeta a los procesos de compartir, interpretar e interiorizar de otras personas.

Los conceptos de dato, información y conocimiento son definidos por Checkland y Holwell en [5].

- Dato: es un valor aislado y representa un hecho observado o creado por las personas.
- Información: es el conjunto de datos que se le atribuye un significado.
- Conocimiento: colección de información que pueden ser apropiadas o interiorizadas por las personas, que puede ser útil para ellas, pero no provee generación de nuevo conocimiento.

En consecuencia, el equipo de pruebas debe estar capacitado para tomar la cadena datos-información-conocimiento y extraer lo útil, para optimizar y mejorar el proceso.

Una definición genérica de la GC, según Gupta y Sharma es el conjunto de procesos que gobiernan la creación, diseminación, y utilización de conocimiento en [6].

Para Moral y Pazos, la GC tiene un espectro amplio y lo define como: el conjunto de principios, métodos, técnicas, herramientas, métricas y tecnologías que permiten obtener los conocimientos precisos, para quienes los necesitan, del modo adecuado, en el tiempo oportuno de forma eficiente y sencilla, con el fin de conseguir una actuación institucional lo más inteligente posible en [7].

### 2.2 Gestión de conocimiento en organizaciones de software

En el ámbito específico de la Ingeniería de Software (IS), Rus y Lindvall proponen que, dependiendo del conjunto de actividades en IS al cual pertenezcan los conocimientos, éstos pueden ser de diferentes clases, tales como: conocimiento organizacional, conocimiento de gestión, conocimiento técnico y conocimiento del dominio en [8]

- Organizacional: Se refiere al conocimiento para gestionar la organización, los objetivos del negocio y la gestión de los recursos humanos.
- De Gestión: Se refiere al conocimiento para planificar, liderar y realizar el seguimiento de un proyecto.
- Técnico: Abarca el conocimiento para realizar análisis de requisitos, diseño, programación y pruebas de software.
- Del Dominio: Hace referencia al conocimiento del dominio de aplicación, tales como: banca, seguros y telecomunicaciones, entre otros.

### 2.3 Pruebas de software

Las pruebas de software, se consideran como: “la actividad en que un sistema o un componente se ejecuta bajo condiciones controladas, los resultados son registrados y la evaluación es realizada sobre algún aspecto del sistema o componente” [9]. Esta es una actividad cognitiva y no mecánica ni repetitiva que involucra varias funciones mentales como el lenguaje, la imaginación y la percepción, entre otros. Un probador desarrolla estas habilidades a medida que obtiene conocimiento y experiencia.

De acuerdo, con Vargas y colaboradores, se puede afirmar que el objetivo de la norma ISO/IEC 29119 de Pruebas de software es proporcionar una norma definitiva para las pruebas de software que define el vocabulario, procesos, documentación, técnicas y un modelo de evaluación del proceso de pruebas de software que se puede utilizar dentro de cualquier ciclo de vida de desarrollo [9].

La norma IEEE, se centra en un modelo de proceso de tres niveles basado en riesgos para las pruebas de software que

orienta el desarrollo de la estrategia de prueba. Estandarizando la forma como se planifica, diseña, ejecuta y mantienen las pruebas, además unificando estándares anteriores y aplicables a diferentes tipos de sistemas de software [10, 11].

### 2.4 Revisión de literatura

De acuerdo con la definición de GC, se analizan los estudios e investigaciones que permiten implementar estrategias relevantes de GC en el contexto de las pruebas de software. Estos trabajos se describen a continuación:

Los autores Kerkhof y colaboradores realizan un análisis de estrategias de gestión de conocimiento en las fábricas de software y considera que la gran mayoría se enfocan en la implementación de los procesos de GC y sin tener en cuenta la utilidad y uso del conocimiento. Plantean un modelo que soporta el aprendizaje organizacional en las organizaciones de pruebas de software [12].

Nogeste y Walker presentan un sistema de gestión de conocimiento centrado en las pruebas de regresión describiendo la forma para obtener el conocimiento tácito y posterior reuso en [13].

Xu-Xiang y Zhang proponen una estrategia que integra la GC y la mejora de procesos desde un enfoque PHVA (plan-hacer-verificar-actuar) focalizándose en el conocimiento explícito en [14].

Abdullah y colaboradores definen un modelo arquitectónico para sistema de GC basada en la estrategia de las comunidades de práctica. La comunidad de practica integra los desarrolladores y los probadores para generar un ambiente colaborativo que resuelve diferencias e inquietudes que surgen durante las pruebas de software [15].

En el trabajo presentado por Desai y Shah exponen las dificultades que se presentan en los procesos de GC aplicados al dominio de pruebas de software; para facilitar los procesos crear, explorar, refinar, almacenar, gestionar y la diseminación del conocimiento proponen el desarrollo de un sistema de GC [16].

Los autores Li y Zhang plantean un modelo para el reuso del conocimiento asociado a los casos de prueba basado en la representación ontológica. El principal propósito es documentar e identificar los elementos relevantes de conocimiento que pueden aportar al reuso. Integra herramientas como mapas de conocimiento, repositorio de datos, ontologías de dominio para la captura del conocimiento tácito y uso del conocimiento explícito [17].

Partiendo de la caracterización del conocimiento para el almacenamiento y recuperación los autores Andrade y colaboradores, implementan un sistema de lecciones aprendidas, con el propósito de crear una memoria organizacional soportada en páginas amarillas en [18].

**Crítica:** las investigaciones presentadas por [13] y [17] se focalizan en las pruebas de regresión y en los casos de prueba, sin integrar las tres capas del proceso de pruebas.

Los trabajos desarrollados por [13, 15, 16, 17, 18], se centran

**Tabla 1**

*Procesos de la Gestión de Conocimiento. Fuente: elaboración propia.*

Procesos	Términos similares utilizados por diferentes autores
Definición de objetivos	Alineación estratégica y lineamientos organizacionales
Identificación	Mapeo y fuentes de información
Incorporación	Adquisición, desarrollo, creación, construcción, externalización, innovación
Preservación	Indexación, formalización, codificación, almacenamiento
Distribución	Transferencia, socialización, integración y compartir
Utilización	Aplicación o uso

en el desarrollo de herramientas informáticas para facilitar el reuso del conocimiento, sin describir un proceso para implementar la estrategia de GC.

De acuerdo con [12] se centra en la conducta y la estrategia corporativa, sin analizar los flujos de trabajo previstos en los equipos de pruebas de software.

Según [14] plantea un modelo que integra la estrategia GC y mejora de procesos de forma paralela al desarrollo de pruebas, sin describir cómo se implementa en los procesos que integran las pruebas de software.

## 3. Resultados y discusión

Históricamente los procesos de GC, reciben diferentes denominaciones en la Tabla 1, se presenta la unificación de estos términos.

El modelo propuesto adopta los procesos de GC, referidos en la Tabla 1. Cada proceso se define a continuación:

- **Definición de Objetivos:** hace referencia a los conocimientos, habilidades y capacidades requeridas por la organización, para alcanzar la estrategia de negocios establecida en los planes y objetivos estratégicos. Partiendo de estos argumentos, se establecen los objetivos de conocimiento estratégico, que soportan la estrategia de la organización.
- **Identificación:** en esta etapa se detectan los activos de conocimiento que tiene la organización (inventario de conocimiento) y el conocimiento requerido para soportar los objetivos de conocimiento, definidos en la fase anterior.
- **Incorporación:** el propósito es adquirir el conocimiento requerido para cumplir con el propósito estratégico y que no está disponible en la organización, según el análisis realizado en el proceso anterior.

- **Preservación:** este proceso está compuesto por las actividades que propende por almacenar y mantener los conocimientos (tácito y explícito) disponibles en la organización.
- **Distribución:** este proceso genera las condiciones para que el conocimiento se pueda compartir en la organización.
- **Utilización:** el objetivo del proceso es garantizar que el conocimiento se aplique en la organización y genere valor. Siendo el fin de cualquier estrategia de GC.

### 3.1 Taxonomía del conocimiento en el contexto de las pruebas de software

El conocimiento es un elemento fundamental, que apoya la identificación de fallos en los productos de software. Específicamente, en el proceso de pruebas es relevante identificar el tipo y clase de conocimiento necesario, para que los probadores tengan mayor probabilidad de éxito en la detección y reconocimiento de defectos y mejoren su desempeño. Según los autores [19], en el estudio empírico realizado a equipos de pruebas en cuatro organizaciones de software se identifican tres categorías de conocimiento:

- **Conocimiento del dominio (CD):** hace referencia a las reglas de negocio que soportan la misión empresarial y los referentes normativos, legales y culturales que influyen en los procesos de negocio.
- **Conocimiento del sistema (CS):** este se refiere al conocimiento embebido en la aplicación y no se puede identificar claramente en la documentación de diseño.
- **Conocimientos generales de la ingeniería de software (CG):** hace referencia al conocimiento y habilidades que deben tener los probadores frente a la cadena de valor que soporta la construcción de software.

Los tipos de conocimiento son utilizados en el modelo para caracterizar los activos de conocimiento disponibles y requeridos (inventarios de conocimiento), asociados al objetivo de conocimiento estratégico de la organización.

### 3.2 Modelo de gestión de conocimiento aplicado a las pruebas de software

Para que las pruebas de software cumplan con los objetivos propuestos se plantean los siguientes componentes fundamentales:

- Cumplimiento con la estructura de procesos definida en el estándar de pruebas de software ISO/IEC 29119.
- Ejecución de una estrategia de pruebas basada en riesgos.
- Integración de la tipología de conocimientos que generan valor en la identificación y reconocimiento de fallos.

El modelo propuesto articula los procesos de GC y pruebas de software, teniendo como insumo los artefactos de las pruebas, para desarrollar las actividades que permiten construir las salidas del modelo.

En la Tabla 2, se describe el modelo tomando un esquema similar al propuesto por [20], donde cada fase del modelo se detalla con los siguientes elementos: descripción general, entradas, actividades y salidas.

**Definición de objetivos:** se debe alinear el propósito de acuerdo al estándar ISO/IEC 29119, establece un nivel de procesos organizacionales, en el cual se desarrollan y gestionan las especificaciones de pruebas para la organización. Estos lineamientos describen los objetivos y alcance global de las pruebas dentro de la organización y establece las prácticas a instanciar dentro de la gestión de proyectos de prueba.

**Identificación:** en esta etapa se analiza el conocimiento estratégico requerido para el proceso y se realiza un diagnóstico con el inventario de activos de conocimiento existentes. Lo que implica construir el inventario de conocimiento, habilidades y experiencias que tiene el equipo de pruebas. En este caso, un elemento esencial es la comprensión del sistema [21], el modelado de negocio y los artefactos de la aplicación. El resultado que se obtiene son los bloques de conocimiento y los gaps de conocimiento del equipo.

**Incorporación:** en esta etapa se agrega al plan de pruebas las actividades requeridas para mitigar los gaps de conocimientos y se cuantifica en tiempo y costo, generando una calendarización del proceso.

**Preservación:** en este punto se incluye en el artefacto denominado registro de incidencias, el evento de conocimiento que permite el registro de los problemas obtenidos en el proceso.

**Utilización:** en esta etapa se debe contar con una cultura y medios tecnológicos que permitan la apropiación de los conocimientos derivados del proces.

Estas etapas se presentan en la Figura 1.

### 3.3 Aplicando el modelo

El modelo se aplica al proceso de pruebas de software que soporta la gestión de prácticas de los programas académicos adscritos a una Institución de Educación Superior.

El producto de software utilizado para el caso de estudio, es una aplicación ubicada en un servidor de producción. La documentación existente suministrada al inicio del proceso es el manual de usuario. Al realizar la revisión del documento se encuentra como hallazgo que esta desactualizado. Es de aclarar que no se cuenta con documentación adicional que soporte el diseño del producto. Otra característica, es que los procesos de las pruebas se realizan en un contexto independiente al proceso de desarrollo.

Tabla 2

Componentes del modelo de conocimiento aplicado a pruebas. Fuente: Elaboración propia

Fases del Modelo	Entradas	Actividades	Salidas
Definición de Objetivos	Políticas y estrategias organizacionales.	Establecer un diagrama de conocimiento según los objetivos estratégicos de las pruebas.	Identificar objetivos conocimiento estratégico.
Identificación	Comprensión del Contexto. Modelado del Negocio. Activos de Proceso. Plan de riesgos.	Realizar inventario de activos de conocimiento y experiencias.	Inventario de conocimiento. Bloques de Conocimiento. Listado de gaps de conocimiento.
Incorporación	Plan de Pruebas.	Definir actividades de conocimiento para mitigar los gaps de conocimiento. Incluirlos requerimientos de conocimiento en el plan.	Plan de adquisición de conocimiento.
Preservación	Plan de pruebas. Gestión de Incidentes.	Registrar los incidentes manteniendo una trazabilidad con los riesgos del proyecto y los gaps de conocimiento. Indexar los incidentes por palabras claves.	Registros de incidencias de conocimiento y lecciones aprendidas postmortem.
Distribución	Registro de incidencias.	Distribuir a los grupos de interés las lecciones aprendidas. Definir socialización de las principales incidencias de conocimiento.	Listas de distribución. Eventos de socialización.
Utilización	Roles de Pruebas. Política de reconocimiento e incentivos.	Identificar casos de éxito. Documentar los casos éxitos. Registrar los activos en el ambiente colaborativo.	Reporte de ingreso a la plataforma virtual.

Figura 1. Modelo de GC aplicado a las pruebas de Software



El equipo de pruebas cuenta con tres integrantes con los siguientes roles: un líder de proyecto, un diseñador de casos de prueba y un probador.

**Definición de objetivos:** para esta fase se realiza la definición del servicio y se lleva a cabo durante tres reuniones con el líder funcional, en las cuales se explica globalmente la aplicación y se describe brevemente las principales funcionalidades.

Con el líder funcional que administra la aplicación se establecen los objetivos de pruebas preliminares:

- Generar información fiable del proceso de prácticas para los diferentes procesos institucionales académicos y administrativos.
- Realizar la prueba del sistema sobre las funcionalidades de mayor criticidad.

El líder del proyecto plantea los objetivos de conocimiento que se derivan del propósito de pruebas establecido:

- Alcanzar un cumplimiento del 100% de los casos de prueba frente a las reglas de negocio planteadas en la normativa institucional.
- Poner a disposición del equipo de pruebas el conocimiento explícito de carácter relevante.
- Propiciar la apropiación y comunicación del conocimiento tácito en los integrantes del equipo de pruebas.

**Identificación:** en esta fase se lleva a cabo la revisión preliminar de los requerimientos, a partir de la documentación existente del producto, se realiza una lista de las principales funcionalidades de la aplicación, basándose en los menús y las funcionalidades de cada menú, que aparecen en el manual de usuario de la aplicación. Se detecta que la documentación existente se encuentra incompleta, por lo que se debe mejorar las especificaciones, esto se tiene en cuenta al realizar el cronograma del proyecto de prueba. Las funcionalidades son listadas en el Inventario de pruebas, se encuentran 42 funcionalidades durante esta actividad, que se presentan en la 3. El inventario de pruebas se valida con el cliente.

También se realiza el análisis preliminar de riesgo del producto, en las reuniones mantenidas con el cliente, se le explica la importancia de realizar el análisis de riesgo del producto y a partir de dicho análisis se define el alcance de las pruebas. Como resultado de estas actividades, se define la periodicidad de las reuniones con el cliente y que se mantendría comunicación fluida por correo electrónico y mensajería electrónica instantánea, semanalmente. Se identifican los responsables por parte del cliente de tomar las decisiones de alto nivel, de contestar dudas respecto al producto, de decidir sobre los cambios del alcance de las pruebas a lo largo del proyecto, de validar los casos de prueba y los resultados de su ejecución. Al finalizar esta etapa se obtienen las salidas de conocimiento: los bloques e inventario de conocimiento y el listado de gaps de conocimiento.

En la Figura 2, se definen los bloques de conocimiento de las fuentes de información requeridas para ejecutar la estrategia de conocimiento implícita en el proceso de pruebas.

En la Tabla 4, se presentan los gaps de conocimiento de manera resumida, donde se realiza un diagnóstico. Identificando el nivel de conocimiento de cada integrante del equipo frente a los bloques de conocimiento establecidos en el proyecto. Para esta caracterización se plantean tres categorías para valorar el nivel de conocimiento del personal frente a cada dimensión de conocimiento requerida para el proyecto.

**Incorporación:** en esta fase se realiza el plan de pruebas con una duración de nueve semanas, como se presenta en la Figura 3.

El plan de pruebas incluye actividades para fortalecer los procesos de apropiación de conocimiento. Estas actividades son:

**Tabla 3**

*Funcionalidades del sistema a probar. Fuente: elaboración propia*

No.	Fun- cionalidad	Descripción de la funcionalidad
1		Generar carta de presentación estudiantes
2		Generar carta de presentación asesores
3		Gestionar practicas x estudiante
4		Gestionar matriculas
5		Consolidar calificaciones
6		Gestionar grupos
7		Gestionar materias
8		Gestionar convenios
9		Gestionar solicitudes de practicantes
10		Gestionar sedes de prácticas
11		Gestionar docentes
12		Gestionar cooperadores
13		Gestionar centros de practicas
14		Gestionar sedes de prácticas
15		Gestionar asesores
16		Gestionar visitas
17		Gestionar administradores
18		Gestionar planes académicos
19		Gestionar coordinadores de prácticas
20		Gestionar invitados de prácticas
21		Gestionar evaluaciones
22		Gestionar novedades
23		Registro de evaluación del seguimiento de estudiantes (masiva)
24		Registro de evaluación del seguimiento x estudiante (individual)
25		Generar informe ARL
26		Generar informe estudiantes por nivel
27		Generar informe de calificaciones
28		Gestionar publicaciones
29		Gestionar configuraciones
30		Postular a solicitudes de practicantes
31		Postular a grupos de prácticas
32		Consultar historial de calificación de estudiante
33		Consultar historial de prácticas estudiante
34		Registrar concepto de evaluación - centro de practica
35		Cierre de periodo
36		Informes de evaluación
37		Informes por grupo
38		Informes por centro de practica
39		Informes por programa
40		Informes de novedades
41		Gestionar usuario y contraseña
42		Gestionar usuarios del sistema

Figura 2. Insertar Caption Figura 2

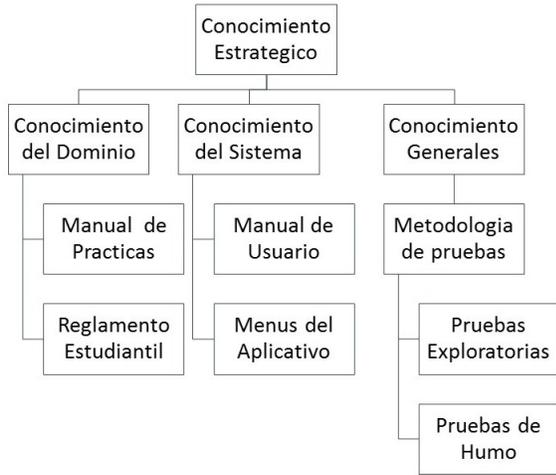
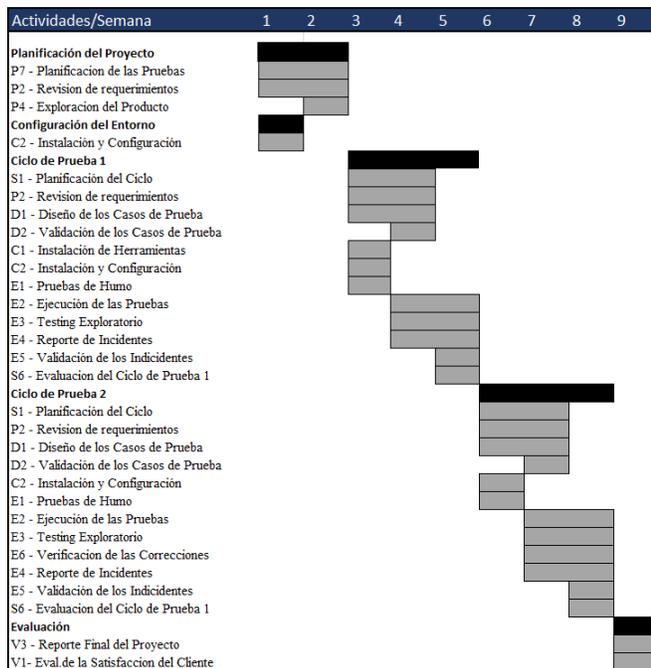


Tabla 4

Gap de Conocimiento. Fuente: elaboración propia

Personal	Conocimiento actual		
	CD	CS	CG
Líder proyecto	Bajo	Medio	Alto
Diseñador casos de prueba	Bajo	Bajo	Medio
Probador	Bajo	Bajo	Medio

Figura 3. Insertar Caption Figura 3



- Revisión de requerimientos: esta actividad se desarrolla con el líder funcional de la aplicación por medio de entrevistas que permiten compilar el conocimiento tácito del cliente.
- Exploración del producto: Se plantea un trabajo cooperativo por el uso de una comunidad de práctica, donde se codifique los elementos claves de la aplicación y la documentación que soporta el dominio.
- Plan de capacitación: Se lleva a cabo un plan de capacitación al equipo del proyecto, con tutoriales.
- Entrevistan al líder funcional: Se desarrollan entrevistas para conocer las reglas del negocio asociadas al proceso de prácticas.

**Preservación:** se realiza la ejecución de las pruebas. Al terminar la ejecución de las pruebas, se realiza la actividad de evaluación de las pruebas, donde se evalúan las actividades de cada ciclo. Se realizan dos ciclos de pruebas. Durante el desarrollo de cada ciclo de pruebas se registran las incidencias, teniendo en cuenta los siguientes elementos: código del incidente, descripción, tipo de incidente, causa del incidente (identifica el bloque de conocimiento asociado), descripción de la solución, palabra clave. Estos campos permiten codificar las experiencias para ser usadas en otros procesos con incidentes similares para reducir tiempos tanto en la ejecución de la prueba como en la solución de los nuevos incidentes.

**Distribución y utilización:** en esta fase resultaron importantes para la distribución y utilización del conocimiento:

- El Plan de pruebas: es de gran utilidad para que el cliente tener claridad respecto al trabajo del proyecto de prueba independiente, para lograr acuerdos respecto a responsabilidades y compartir un vocabulario común.
- Informes de avance: La realización de los mismos al finalizar cada ciclo de prueba mostrando el cubrimiento de las funcionalidades probadas y los incidentes encontrados resulta ser una buena forma de comunicación del trabajo del equipo de pruebas.
- Sistema de seguimiento de incidentes web: En este caso se utiliza la herramienta Bugzilla, que permite que el cliente y los miembros del equipo validen los incidentes a medida que se reportan en la herramienta.
- Lecciones aprendidas durante el proceso de pruebas se registraron en la base de conocimiento.
- La base de conocimiento documental integrada por las actividades prevista en el plan de adquisiciones de conocimiento, el registro de incidentes, las lecciones aprendidas previstas en cada evaluación del ciclo permiten compilar las experiencias y conocimientos asociados al contexto del proyecto para un nuevo ciclo de pruebas o para un nuevo proyecto.

## 4. Conclusiones

El presente artículo describe un proceso que articula la GC y las pruebas fundamentado en el conocimiento explícito representado en los activos del proceso de pruebas, quedan origen a los activos de conocimiento como una estrategia implícita en la gestión del proyecto.

Con el análisis realizado en este artículo al integrar los procesos de la Gestión de Conocimiento a las pruebas de software se genera una mejora sistemática al desempeño del equipo de pruebas, permitiendo la identificación, preservación y uso de las experiencias obtenidas en el proceso.

Al identificar el tipo y clase de conocimiento asociado al proceso de pruebas de software, se permite que los probadores tengan una mayor probabilidad de éxito en la detección y reconocimiento de defectos en el proceso. Teniendo como premisa la definición de los objetivos desde los riesgos asociados al dominio y el objetivo estratégico del equipo de pruebas. Los activos del conocimiento se plantean teniendo como fundamento el contexto estratégico y organizacional de la prueba y los riesgos asociados al dominio del negocio.

A partir del plan de pruebas, se establecen las actividades de conocimiento requeridas para mitigar los riesgos previamente identificados.

El modelo de gestión de conocimiento se enfoca en la mitigación de riesgos y las falencias de conocimiento identificadas en el proyecto de pruebas, con el propósito de mejorar la efectividad del proceso.

## References

- [1] C. Kaner, J. Bach, and B. Pettichord, *Lessons Learned in Software Testing*. New York, NY, USA: John Wiley & Sons, Inc., 2002.
- [2] L. Xue-Mei, G. Guochang, L. Yong-Po, and W. Ji, "Research and implementation of knowledge management methods in software testing process," in *2009 WRI World Congress on Computer Science and Information Engineering*, vol. 7, pp. 739–743, March 2009.
- [3] R. Abdullah, Z. D. Eri, and A. M. Talib, "A model of knowledge management system in managing knowledge of software testing environment," in *2011 Malaysian Conference in Software Engineering*, pp. 229–233, Dec 2011.
- [4] M. Alavi and D. E. Leidner, "Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues," *MIS Quarterly*, vol. 25, no. 1, pp. 107–136, 2001.
- [5] P. Checkland and S. Holwell, *Data, capta, information and knowledge*, pp. 47–55. Elsevier, 2006.
- [6] J. Gupta and S. Sharma, *Creating Knowledge Based Organizations*. Idea Group Publishing, 2004.
- [7] A. D. Moral, J. Pazos, E. Rodríguez, A. Rodríguez-Patón, and S. Suárez, *Gestión del conocimiento*. Paraninfo, 2008.
- [8] I. Rus and M. Lindvall, "Knowledge management in software engineering.," *IEEE Software*, pp. 26–38, 2002.
- [9] F. A. Vargas-Agudelo, D. E. Soto-Duran, J. C. Giraldo-Mejía, and C. E. Durango-Vanegas, "Representación en el núcleo de semat de la norma iso/iec/ieee 29119-2 para identificar patrones en pruebas de software," in *Memooria 4to. Congreso Internacional de Investigación e Innovación en Ingeniería de Software, CONISOFT*, pp. 41–46, 2016.
- [10] "Iso/iec/ieee systems and software engineering – architecture description," *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)*, pp. 1–46, Dec 2011.
- [11] K. Villamizar-Suaza, D. Soto-Durán, J. C. Giraldo-Mejía, and J. Jimenez-Builes, "Modelo de pruebas en proyectos bi," in *LACCEI 2016 Internacional Multi Conference*, 2016.
- [12] C. Kerkhof, J. v. d. Ende, and I. Bogenrieder, "Knowledge management in the professional organization: a model with application to cmg software testing," *Knowledge and Process Management*, vol. 10, no. 2, pp. 77–84, 2003.
- [13] K. Nogeste and D. H. Walker, "Using knowledge management to revise software-testing processes," *Journal of Workplace Learning*, vol. 18, no. 1, pp. 6–27, 2006.
- [14] L. Xu-Xiang and W. N. Zhang, "The pdca-based software testing improvement framework," in *The 2010 International Conference on Apperceiving Computing and Intelligence Analysis Proceeding*, pp. 490–494, Dec 2010.
- [15] R. Abdullah, Z. D. Eri, and A. M. Talib, "A model of knowledge management system in managing knowledge of software testing environment," in *2011 Malaysian Conference in Software Engineering*, pp. 229–233, Dec 2011.
- [16] A. Desai and S. Shah, "Knowledge management and software testing," in *Proceedings of the International Conference &#38; Workshop on Emerging Trends in Technology*, (New York, NY, USA), pp. 767–770, ACM, 2011.
- [17] X. Li and W. Zhang, "Ontology-based testing platform for reusing," in *2012 Sixth International Conference on Internet Computing for Science and Engineering*, pp. 86–89, April 2012.
- [18] J. Andrade, J. Ares, M.-A. Martínez, J. Pazos, S. Rodríguez, J. Romera, and S. Suárez, "An architectural model for software testing lesson learned systems," *Information and Software Technology*, vol. 55, no. 1, pp. 18–34, 2013.
- [19] J. Itkonen, M. V. Mäntylä, and C. Lassenius, "The role of the tester's knowledge in exploratory software testing," *IEEE Transactions on Software Engineering*, vol. 39, pp. 707–724, May 2013.
- [20] J. R. Persse, *Process Improvement Essentials*. O'Reilly Media, Inc, 2006.