

Transferencia de Conocimiento en la Gestión de Calidad de la Ingeniería de Software

Mauricio Rozo^{1 2}, Ines Casanovas^{1 2}, Ma. Florencia Pollo Cattaneo^{1 2},

¹Grupo de Estudio en Metodologías de Ingeniería de Software (GEMIS)
Universidad Tecnológica Nacional. Facultad Regional Buenos Aires.
Medrano 951 (C1179AAQ) Ciudad Autónoma de Argentina. Buenos Aires

²Programa de Maestría en Ingeniería de Sistemas de Información
Universidad Tecnológica Nacional, Facultad Regional Buenos Aires.
Medrano 951 (C1179AAQ) Ciudad Autónoma de Argentina. Buenos Aires

{mauricio.rozo.rodriguez, inescasanovas, flo.pollo}@gmail.com

Resumen. Los procesos y actividades en la gestión de calidad de la ingeniería de software generan un gran volumen de conocimiento, con lo cual es considerado como un factor crítico para la calidad de producto software, por lo tanto, exige una creciente demanda en la mejora de la efectividad y cumplimiento de las tareas que la componen, es ahí donde el uso de métodos y principios de gestión de conocimiento se convierte en la base para gestionarla. Con base en este argumento, se indagan los desafíos de los modelos de gestión de conocimiento existentes en el dominio de la calidad de software para identificar las falencias de los modelos existentes y a partir de estas, proponer una solución para la ingeniería de software en el ámbito de la gestión de conocimiento, basado en la utilización de metodologías ontológicas en el dominio de la fase de pruebas.

Palabras Clave: Gestión de Conocimiento, Gestión de Calidad, Fase de Pruebas, Ingeniería de Software, Ontologías.

1 Introducción

Los procesos y actividades de la gestión de calidad tienen gran importancia en la ingeniería de software debido que permiten evaluar las soluciones informáticas y determinar el nivel de calidad que poseen. Por tanto, esta fase es una de las más importantes en el desarrollo de software, sin ellas, el producto informático contendría multitud de defectos provocando alto impacto y costos en la solución de los defectos, deteriorando la gestión integral en las organizaciones [1-2]. Por otro lado, en [3], entienden que las pruebas de software son un proceso intensivo en conocimiento, y se hace necesario proporcionar soporte computarizado para las tareas de adquisición, procesamiento, análisis y diseminación de conocimiento para su reutilización.

Es indudable que las pruebas de software (PS) es uno de los aspectos que hacen que la ingeniería de software trascienda en los proyectos informáticos con calidad de

producto. Sin embargo, las dificultades que atraviesan para ser efectivas tienen diferentes ámbitos, en ese sentido, el principal problema en las organizaciones de software relacionadas con esta etapa es la baja tasa de reutilización del conocimiento y las barreras en la transferencia de este [4].

A pesar de la importancia de las PS, existen organizaciones que no se les presta la suficiente atención o son, simplemente, omitidas en la planificación del desarrollo de productos software debido a la presión por tener un producto terminado en una fecha establecida, con lo cual provoca que haya actividades, consideradas erróneamente como prescindibles, que son eliminadas o reducidas en la planificación, afectando la calidad de software [3,5].

De acuerdo con lo anteriormente descrito, en [6], se identificaron varios desafíos para la gestión de la calidad de los productos de software, entre los que destacan: baja tasa de reutilización del conocimiento de PS, barreras en la transferencia de conocimientos, baja posibilidad de lograr rápidamente la distribución más eficiente de recursos humanos durante las pruebas. Además, se enuncia que las debilidades en las PS se deben a que existe una pérdida significativa de capital intelectual debido a la rotación del personal y conocimiento limitado. La obtención, difusión, adquisición, evolución y empaquetamiento del conocimiento reciben poca atención debido a que el conocimiento se maneja principalmente durante las pruebas dentro de un proyecto u organización y se dedica menos atención al intercambio de conocimientos [7-8].

Basado en estos desafíos, la gestión de conocimiento (GC) provee, de manera estructurada y sistemática, las herramientas para facilitar que el conocimiento generado permita alcanzar los objetivos estratégicos de las organizaciones y optimizar las decisiones que se tomen para mejorar los procesos, convirtiéndose en un activo [9].

Ahora bien, entendiéndose que, en la ingeniería de software, la GC está desarrollándose en la medida de las necesidades que expresan las organizaciones de este rubro, es necesario expandir el uso de la GC dentro de las diferentes fases de la ingeniería de software, debido a la especificidad que exige la evolución de los procesos de cada una. En consecuencia, surgen como interrogantes cuál es el estado de situación de la GC aplicado a la gestión de calidad de la ingeniería de software y qué técnicas existentes y adaptadas en otros ámbitos informáticos, son aplicables en la GC en la fase de PS y surgen como elementos para potencializar el desarrollo y masificación en el contexto de la ingeniería de software. En vista de lo anterior, es necesario resolver estos cuestionamientos para determinar qué técnica es viable utilizar para potencializar el desarrollo de la GC aplicado a la PS en la ingeniería de software. Es por ello que, aunque reconocida como un instrumento importante por la comunidad de GC, las ontologías no se utilizan ampliamente en estas iniciativas, entendiéndose que una ontología es una especificación explícita de una conceptualización, en ésta se modela el vocabulario del dominio, básicamente usando los conceptos, características y sus relaciones [3,10,11]. En ese sentido en [12,20] se propone estudiar cómo las ontologías se pueden utilizar en la GC en el dominio de la gestión de calidad en el ámbito de la ingeniería de software.

Para cumplir con el objetivo propuesto, el presente trabajo se organiza de la siguiente manera: en la Sección 2 se exponen los materiales y métodos, en la Sección 3 se presenta la propuesta metodológica para la transferencia de conocimiento en la gestión de calidad, en la Sección 4 se presentan los resultados de la evaluación de la

propuesta. Finalmente, en la Sección 5 se exponen las conclusiones y trabajos futuros.

2 Materiales y Métodos

El presente trabajo se basa metodológicamente en la visión del constructivismo, donde el interés se centra en cómo se crean las significaciones y cómo se construye la realidad. De acuerdo con lo anterior, el enfoque metodológico que sustenta el presente trabajo es el cualitativo, donde el investigador hace cuestionamientos basados principalmente en perspectivas constructivistas que demanda estrategias de investigación [13-14].

El diseño utilizado en esta investigación es el descriptivo, y en relación con los métodos, se utiliza la revisión sistemática documental exploratoria [15-16] aplicándolo en el conocimiento existente en el dominio de la GC, en el dominio de las PS y en el dominio de los modelos ontológicos para determinar el estado de la cuestión y fundamentación de los desafíos en la gestión de calidad de software.

A su vez, se realiza un estudio analítico y comparativo de las metodologías existentes para determinar su aplicabilidad a la solución de la problemática, concluyendo en la elaboración de una propuesta metodológica que solucione o minimice la brecha de conocimiento detectada como desafíos en la gestión de calidad dentro de la ingeniería de software. Como solución se desarrolla un modelo de aplicación de GC a la PS de la ingeniería de software.

3 Propuesta Metodológica

El modelo de GC propuesto reúne los descriptores: motor de la GC, uso de las herramientas tecnológicas, contexto, actores, tipo de conocimiento y clima organizacional, a su vez incorpora en su estructura una propuesta ontológica que adopta el método planteado en [17] como pasos para su construcción y define el uso de patrones de diseño, basado en los desafíos de las ontologías y aplicable a la gestión de la calidad de producto de la ingeniería de software [21].

El modelo está basado en los procesos propuestos por los modelos Riesco [9], Paniagua y López [18], Angulo y Negrón [9] y el Modelo ISECO [9], teniendo en cuenta que cumple con lo definido en [18], que plantean como elemento clave en la GC la necesidad de asumir los aspectos relacionados con las personas, los procesos y la tecnología como un todo.

Como lo propone [19], los procesos de GC ocurren cíclicamente y acompañan la evolución de las PS, maximizando la tasa de reutilización, la transferencia de conocimiento, a su vez, minimizan los desafíos que tiene la fase de pruebas en la ingeniería de software. El modelo propuesto incorpora una capa ontológica, que adopta mejoras en la formalización y distribución del conocimiento, a través de la propuesta de una ontología de dominio en el contexto de las PS (ver Fig. 1).



Fig. 1. Propuesta de un modelo de aplicación de Gestión de Conocimiento a la fase de pruebas de la ingeniería de software.

3.1 Estructura

El modelo de GC propuesto está dividido en instancias, entendiendo que una instancia, como lo propone [22], es una entidad que pertenece a un determinado contexto. Las instancias que componen el modelo son: organización, ejecución y conservación o mantenimiento, se describen a continuación. A su vez, cada instancia posee una serie de fases que permiten el desarrollo de las PS en el dominio de la GC.

Instancia de Organización. En esta instancia, lo que se busca es la asignación de roles específicos, para que, en la GC desde la perspectiva de la organización, garantice que las habilidades, experiencias y pericias del equipo del proyecto se utilicen antes, durante y después del proyecto [19,23].

Desde las perspectivas antes descritas, se busca definir en el equipo de pruebas, el rol de gestor de conocimiento y a su vez definir el dominio de conocimiento bajo el enfoque de las PS que va a ser incorporado en la GC.

Como lo proponen en [23], para las entradas de esta parte del modelo se tiene el expertise de los integrantes del equipo de pruebas, que proporciona información sobre competencias y experiencias disponibles del proyecto de software, además ayuda a comprender qué conocimiento está o no disponible; también se requiere la documentación del proyecto como registro de lecciones aprendidas, documentación de requisitos. Las actividades principales consisten en seleccionar el gestor de conocimiento, determinar qué tipos de pruebas se van a utilizar en el proyecto, definir la estrategia para la obtención de elementos de conocimiento básico que exista en el proyecto [23]. Como salida se tiene el gestor y dominio de conocimiento seleccionado que involucra la validación del plan de gestión de calidad y sus métricas.

Instancia de Ejecución. En la instancia de ejecución, se busca cumplir las etapas cíclicas del modelo para que las PS dentro de la gestión de calidad queden inmersas en el contexto de la GC. En [18], se expone que estas acciones disminuyen la redundancia de tareas y el número de errores en su ejecución, al aprovechar la experiencia existente. De acuerdo con lo anterior, esta instancia posee cuatro etapas que involucran: Identificación del conocimiento, Formalización del Conocimiento, Distribución del Conocimiento y Validación e Incorporación del Conocimiento.

Identificación del conocimiento. Se inicia cuando el gestor identifica el conocimiento requerido para que el equipo de pruebas del proyecto realice las PS de acuerdo con el control de gestión de calidad seleccionado. Se establece qué información se tiene disponible y sus fuentes. Se basa en convertir el conocimiento explícito en conocimiento implícito y agregarle experiencia [19,24]. Como entradas, se identifica la necesidad de datos, de información y de conocimiento explícito de las pruebas no formalizado en el dominio del proyecto, se requiere la documentación como el registro de lecciones aprendidas y documentación de requisitos. Se propone como actividades la recopilación de datos, el juicio de expertos y el análisis de datos, para identificar la necesidad de conocimiento en el ámbito de la calidad del proyecto. Como salida se tiene la información y el conocimiento identificado para ser formalizado en la GC [23].

Formalización del conocimiento. Toma la salida de la fase previa y a través de la capa ontológica, identifica las características y/o atributos para categorizar, evaluar, seleccionar y formalizar la información que ingresa en la GC. Como entradas se tiene la información y/o conocimiento no formalizado. Como actividades se propone utilizar la metodología ontológica del modelo Stuart [17], es decir, seguir sus procesos para definir las características y/o atributos para categorizar, evaluar, seleccionar y formalizar la información que ingresa en la GC a través de la creación de la ontología. En [21], indican que la practicidad de este modelo es, ser iterativo y permite la reutilización de las ontologías, lo cual reduce costos y facilita la interoperabilidad de los datos. La ontología de dominio que el modelo de GC aplicado a la fase de PS en la ingeniería de software propone, está inmersa en la capa ontológica. Se busca que esta ontología permita ser más eficiente en resolver la identificación de un concepto o conjuntos de conceptos. Como última actividad se tiene el almacenamiento, dando crecimiento a la memoria corporativa y los sistemas basados en conocimiento [18]. Como salida se tiene el conocimiento implícito formalizado y almacenado, disponible para ser presentado.

Distribución de conocimiento. Comparte el conocimiento entre los miembros del equipo de PS, con lo cual, el conocimiento implícito es convertido a explícito y combina este conocimiento a partir de las experiencias adquiridas en la fase de PS del proyecto, el énfasis es adquirir los mecanismos necesarios para dejar disponible la transmisión de la información adecuada para aplicar y crear conocimiento [9]. Como entradas se tiene el conocimiento implícito formalizado. Dentro de las actividades se propone compartir el conocimiento entre los miembros del equipo, exteriorización del conocimiento a través del intercambio de experiencias, generar conocimiento

explícito a través de las experiencias adquiridas en las PS realizadas en el proyecto, transmitir la información para crear conocimiento, aquí cobra participación de la capa ontológica, para mejorar la eficiencia en la identificación y procesamiento en la GC para las PS que se estén realizando. Como salida se tiene el conocimiento explícito.

Validación e Incorporación del Conocimiento. En esta etapa, el conocimiento explícito desarrollado en la fase de pruebas es validado e incorporado como activo de conocimiento en el dominio de GC para el proyecto de software. Los integrantes del equipo generan conocimientos explícitos al reunir experiencias provenientes de sus aportes y de otras fuentes, es aquí donde se valida y combina el conocimiento, dando como resultado una fácil comprensión del conocimiento para ser utilizado en la generación de nuevas experiencias [19]. Es importante que en la validación participen los miembros del equipo de PS, que junto al gestor de GC, le dan el valor de aceptación, o no, del conocimiento (En el caso de una validación no aprobada, se reinicia el proceso de GC). Tal como se establece en [9], el conocimiento tendrá un valor apreciado cuando su utilización sea direccionada a mejoras en los procesos, toma de decisiones, innovación, resolución de problemas o cualquier otro aspecto que beneficie la fase de PS del proyecto. Por otro lado, [23] propone que se debe realizar el proceso de control de la calidad, en el cual se monitorea y registra los resultados de la ejecución del plan de pruebas, de esta manera se evalúa el desempeño y se asegura que la fase PS del proyecto se encuentre completa, sea correcta y satisfaga las necesidades la calidad de producto del proyecto de software. Como entrada se tiene el conocimiento explícito. Como actividades se define la realización de reuniones de validación. Como salida se tiene el conocimiento explícito incorporado como activo dentro de la PS para el proyecto de ingeniería de software.

Instancia de Conservación o Mantenimiento. Extiende el ciclo de vida del modelo realizando los ajustes en los aspectos que lo requiera, buscando la evolución de este según las necesidades. Es necesario revisar el estado de aplicación del modelo, de esta manera se sigue y controla los resultados, para adelantar en el proyecto la implantación, o no, de mejoras en determinadas etapas [19]. Para [23], este proceso se lleva a cabo a lo largo del proyecto y se realiza a través del control de la calidad, el beneficio es verificar que el trabajo del proyecto cumpla con los requisitos especificados. En esta instancia como entrada se identifican los procesos del modelo de GC propuesto. Como actividades se verifica la aplicabilidad del modelo y se realiza el seguimiento a los procesos, corrección a posibles desvíos en la gestión de los procesos. Como salida se toman las lecciones aprendidas y se reincorporan en la evolución del modelo de GC.

Capa Ontológica del modelo. Para el desarrollo de la ontología de dominio que da sustento a la capa ontológica del modelo de GC, se adopta el método propuesto en [17], constituido por doce pasos que incorpora la identificación del software apropiado y la sustentabilidad de la ontología [21]. Además, se complementa la ontología propuesta, con el fortalecimiento de la reutilización con ontologías en la dimensión de cobertura conceptual, fortaleciendo su evaluación e incorporando la implementación de relaciones no taxonómicas en la dimensión de estructura no

propuesta en la ontología ROoST [8]. Es aquí, donde el presente trabajo avanza en fortalecer e incorporar mejoras en la identificación y procesamiento en la GC para las PS que se estén realizando. Se proponen para cada uno de los pasos, una serie de preguntas de competencia, a su vez equivale a la elicitación de términos, que la ontología debe ser capaz de contestar con todos los conceptos, propiedades entre conceptos y datos, los axiomas y reglas definidas en la ontología [17,24].

4 Resultados obtenidos

La validación de este modelo se divide en dos componentes, el primero evalúa el fortalecimiento de la PS dentro de la gestión del proyecto de software en el cual se definen las siguientes dimensiones: gestión de la información y la documentación, gestión de la comunicación, gestión de la innovación y el cambio, gestión del aprendizaje organizacional [25,26]. El segundo componente define la efectividad de operación al aplicar las fases del modelo de GC en el dominio de PS, lo compone: identificación de conocimiento, formalización de conocimiento, distribución del conocimiento, retención de conocimiento [25-27]. A partir de las dimensiones, se definen los indicadores, lo que permite establecer los ítems que componen el instrumento de investigación. La tabla 1 delimita los componentes, dimensiones e indicadores a fines de determinar la estructura de validación.

Tabla 1. Componentes, Dimensiones e Indicadores de validación del modelo de GC aplicado a la PS de la ingeniería de software.

Componente	Dimensión	Indicadores
Fortalecimiento de la fase de pruebas dentro de la gestión del proyecto de software.	Gestión de la información y la documentación	1.Análisis de requerimientos de información 2.Gestión documental
	Gestión de la comunicación	1.Estructura de entorno adecuado para comunicar la información 2.Facilidad de comunicar a través de materiales digitales documentación de casos de pruebas funcionales, de regresión, de aceptación.
	Gestión de la innovación y el cambio	1.Sistemas de evaluación del desempeño 2.Gestión del cambio de cultura en la mejora de las actividades en las pruebas
	Gestión del aprendizaje organizacional	1.Incorporación de mejores prácticas en el desarrollo de las actividades de pruebas 2.Mejora de gestión de calidad en la fase de pruebas
Efectividad de operación al aplicar las fases del modelo de GC	Identificación de conocimiento	1.Espacio para el fortalecimiento de lecciones aprendidas 2.Incorporación de mapas de conocimiento
	Formalización de conocimiento	1.Sistematizar las tareas de las pruebas 2.Sistema de gestión de la formación
	Distribución del conocimiento	1.Facilidad al acceso a los escenarios de pruebas 2.Efectividad al acceso a los escenarios de pruebas
	Retención de conocimiento	1.Incorpora centros digitales de información 2.Incorpora cuadernos digitales explicativos sobre las actividades de la fase de pruebas

Por otro lado, tabla 2 resume el resultado obtenido para cada uno de los componentes, producto de aplicar el instrumento de investigación a los Expertos.

El cálculo del valor promedio para cada indicador está basado en la escala Likert. El valor referencial, describe la tendencia de valoración de la escala de acuerdo con la puntuación que el Experto aplica sobre las preguntas del instrumento de investigación [15], para valores menores de 2,5 representa estar en desacuerdo, para valores mayores a 3,5 indica estar de acuerdo con lo que propone el modelo de GC para ese indicador. Para los valores que sean mayores de 2,5 y menores de 3,5 no se está ni de acuerdo ni en desacuerdo. Ahora bien, analizando los resultados en un nivel mayor, se tiene que los Expertos coinciden como mayor aporte del modelo de GC a la dimensión “Gestión del aprendizaje organizacional”, el cual obtuvo una valoración de 4,56 puntos. De esta manera, la propuesta del presente trabajo de investigación aporta crecimiento en términos de aprendizaje dentro de las organizaciones. A su vez, los Expertos concuerdan que la dimensión “Formalización de conocimiento” es la segunda en importancia para el modelo propuesto, debido que obtuvo una valoración de 4,39 puntos.

Tabla 2. Resultados para los componentes del modelo de GC producto de la valoración de los Expertos.

Componente	Dimensión	Valor Prom. x Dimen.	Valor Prom. x Componen.	Valor Ref.	En Desacuerdo / De Acuerdo
Fortalecimiento de la fase de pruebas dentro de la gestión del proyecto de software.	Gestión de la información y la documentación	3,94	4,14	2,5 < 3,5	De Acuerdo
	Gestión de la comunicación	3,89			
	Gestión de la innovación y el cambio	4,17			
	Gestión del aprendizaje organizacional	4,56			
Efectividad de operación al aplicar las fases del modelo de GC	Identificación de conocimiento	4,28	4,18	2,5 < 3,5	De Acuerdo
	Formalización de conocimiento	4,39			
	Distribución del conocimiento	4,28			
	Retención de conocimiento	3,78			

5 Conclusiones y Trabajo Futuro

El modelo de aplicación de GC a la fase de pruebas de la ingeniería de software propuesto, se validó y los resultados denotan que, en el primer nivel (que refiere a los indicadores), los Expertos estuvieron de acuerdo con la propuesta, destacando la mejora de gestión de calidad en la fase de pruebas. Para el segundo nivel (que define a las dimensiones), los Expertos destacaron que la gestión del aprendizaje organizacional fue el que más se acentuó en el modelo de GC propuesto. En relación con el tercer nivel (que describe los componentes que hacen a la utilización del

modelo de GC aplicado a las PS), los Expertos indicaron que se expresa efectividad de operación al aplicar las fases del modelo. A su vez están de acuerdo ya que advierten el fortalecimiento de la fase de pruebas dentro de la gestión del proyecto de software.

Como trabajo futuro se continuará en mejorar los aspectos relacionados a la incorporación de cuadernos (“Guías”) digitales explicativas sobre las actividades de la PS, el cual fue uno de los indicadores con menor valoración. A su vez, se fortalecerá la dimensión “Retención de conocimiento”. Por otro lado, es necesario continuar incorporando aspectos en el desempeño del modelo GC desde el enfoque de la gestión del recurso humano.

Referencias

1. Jústiz-Núñez, Dalila, Gómez-Suárez, Darlene, Delgado-Dapena, Marta Dunia: Proceso de pruebas para productos de software en un laboratorio de calidad. *Ingeniería Industrial*, 35(2), 131-145. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59362014000200003&lng=es&tlng=pt (2014).
2. Fernández, F., Ángel, M.: Aplicación de técnicas de pruebas automáticas basadas en propiedades a los diferentes niveles de prueba del software. Recuperado de <http://hdl.handle.net/2183/14814> (2015).
3. Souza, E. F., Falbo, R. A., Vijaykumar, N. L.: Knowledge management applied to software testing: A systematic mapping. In *The 25th International Conference on Software Engineering and Knowledge Engineering (SEKE 2013)*, Boston, USA (pp. 562-567). <https://www.researchgate.net/publication/289638583> (2013).
4. Souza, É. F., Falbo, R. A., Vijaykumar, N. L.: Using the findings of a mapping study to conduct a research project: a case in knowledge management in software testing. In *Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on* (pp. 208-215). IEEE. <http://dx.doi.org/10.1109/SEAA.2015.10> (2015b).
5. Dávila, Abraham, García, Cecilia, Córdor, Sandra: Análisis exploratorio en la adopción de prácticas de pruebas de software de la ISO/IEC 29119-2 en organizaciones de Lima, Perú. *RISTI - Revista Ibérica de Sistemas e Tecnologías de Información*, (21), 1-17. <https://dx.doi.org/10.17013/risti.21.1-17> (2017).
6. Wnuk, K., Garrepalli, T.: Knowledge Management in Software Testing: A Systematic Snowball Literature Review. *e-Informatica Software Engineering Journal*, 12(1), 51-78. <http://io.pwr.edu.pl/eprints/id/eprint/81> (2018).
7. Durán, D. E. S., Gamboa, A. X. R., Builes, J. J.: Aplicación de la Gestión de Conocimiento al proceso de pruebas de software. *Ingenierías Revistas USBMed*, 8(2), 6-13. <http://dx.doi.org/10.21500/20275846.2836> (2017).
8. Souza, É. F. D., Falbo, R. D. A., Vijaykumar, N. L.: ROoST: reference ontology on software testing. *Applied Ontology*, 12(1), 59-90. <https://www.researchgate.net/publication/314268288> (2017).
9. Avendaño Pérez, V., Flores Urbáez, M.: Modelos teóricos de gestión del conocimiento: descriptores, conceptualizaciones y enfoques. *Entreciencias: diálogos en la Sociedad del Conocimiento*, 4(10). <http://www.redalyc.org/articulo.oa?id=457646537004> (2016).
10. Flores, V., Hadfeg, Y.: Un método para generar explicaciones de resultados de un Sistema Experto, usando Patrones de discurso y Ontología. *RISTI-Revista Ibérica de Sistemas e Tecnologías de Información*, (21), 99-114. http://www.scielo.mec.pt/scielo.php?script=sci_arttext&pid=S1646-98952017000100008&lng=pt&nrm=iso (2017).

11. González Gola, F., Sánchez, A., Montejano, G. A.: Asistencia dirigida por ontologías al diseño arquitectónico de videojuegos. In XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires). <http://sedici.unlp.edu.ar/handle/10915/61343> (2017).
12. Rozo Rodríguez M., Casanovas I.: La Gestión de Conocimiento Aplicado a la Fase de Pruebas de la Ingeniería de Software - Revisión Sistemática. 6to Congreso Nacional de Ingeniería Informática / Sistemas de Información. CoNaIISI. Publicación online - ISSN 2347-0372. <https://www.conaiksi2018mdp.org/memorias/memorias.html#> (2018).
13. Creswell, J. W.: Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. University of Nebraska, Lincoln. 4a. ed. Sage Publications, Inc. (2014).
14. Labra, O.: Positivismo y Constructivismo: Un análisis para la investigación social. Rumbos TS. Un espacio crítico para la reflexión en Ciencias Sociales, 0(7), 12-21. <http://revistafaco.ucentral.cl/index.php/rumbos/article/view/55> (2016).
15. Hernández Sampieri, R., Fernández Collado, C., Baptista Lucio, P.: Metodología de la investigación. México: Editorial Mc Graw Hill, 15-40. ISBN 978-92-75-32913-9. (2010).
16. Souza, E. F., de Almeida Falbo, R., Vijaykumar, N. L.: Knowledge management initiatives in software testing: A mapping study. Information and Software Technology, 57, 378-391. <https://doi.org/10.1016/j.infsof.2014.05.016> (2015).
17. Stuart, D.: Practical Ontologies for Information Professionals. Facet. <https://doi.org/10.29085/9781783301522> (2016).
18. Medina Nogueira, Daylin, Medina León, Alberto, Nogueira Rivera, Dianelys.: Procesos y Factores claves de la Gestión del Conocimiento. Revista Universidad y Sociedad, 9(3), 16-23. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202017000300002&lng=es&tlng=es (2017).
19. Linares Pons, N., Piñero Pérez, Y., Rodríguez Stiven, E., Pérez Quintero, L.: Diseño de un modelo de Gestión del Conocimiento para mejorar el desarrollo de equipos de proyectos informáticos. Revista española de Documentación Científica, 37(2), e044. <http://dx.doi.org/10.3989/redc.2014.2.1036> (2014).
20. Rozo Rodríguez M., Casanovas I.: Aplicación de Gestión de Conocimiento a la Fase de Pruebas de la Ingeniería de Software XXII Workshop de Investigadores en Ciencias de la Computación. WICC. <https://drive.google.com/file/d/1vGiNX9ogumSBjnAnH-4d6N5hn5omJPM4/view> (2020).
21. Barber, E., Pisano, S., Romagnoli, S., de Pedro, G., Gregui, C., Blanco, N., Mostaccio, M.: Metodologías para el diseño de ontologías Web. Información, Cultura Y Sociedad, 0(39), 13-36. <http://ppct.caicyt.gov.ar/index.php/ics/article/view/14453/45454575759139> (2018).
22. Fernández Hernández, A.: Modelo ontológico de recuperación de información para la toma de decisiones en gestión de proyectos. <http://hdl.handle.net/10481/43409> (2016).
23. PMI.: Guía de los Fundamentos para la dirección de Proyectos (Guía del PMBOX®). Sexta Edición. Pennsylvania, EEUU.: PEARSON. (2017).
24. Contreras, M., Gutiérrez, F., Ortiz, J., Ramírez, R.: Ingeniería Ontológica Aplicada en el Diseño de un Sistema de Ontologías para la Gestión de Horarios. Pistas Educativas, 39(128). <http://www.itcelaya.edu.mx/ojs/index.php/pistas/article/viewFile/1173/949> (2018).
25. Arias, I. M.: Diseño y validación de un cuestionario de escala formativa para valorar las competencias transversales de los estudiantes universitarios. Una propuesta para dispositivos móviles basada en Android. Universitas Tarraconensis. Revista de Ciències de l'Educació, 1(1), 84-87. <https://revistes.urv.cat/index.php/ute/article/view/661> (2015).
26. Marulanda, Carlos E, Giraldo, Jaime A, López, Marcelo.: Evaluación de la Gestión del Conocimiento en las Organizaciones de la Red de Tecnologías de Información y Comunicaciones del Eje Cafetero en Colombia. Información tecnológica, 24(4), 105-116. <https://dx.doi.org/10.4067/S0718-07642013000400012> (2013).
27. Romero Rodríguez, J. M.: Representación del conocimiento experto de un tutor e-learning a través del mapa conceptual: Un modelo de buenas prácticas docentes. <http://digibug.ugr.es/handle/10481/46413> (2017).