

TECNOLOGÍA: RESULTADOS DE INVESTIGACION

Ontología para Transferir Conocimiento en la Etapa de Pruebas de Software

Ontology to Transfer Knowledge in the Software Testing Stage

Edición Nº 39 – Diciembre de 2020

Artículo Recibido: Agosto 17 de 2020

Aprobado: Noviembre 09 de 2020

Autores

Mauricio Rozo Rodríguez

Especialista en Ingeniería de sistemas de Información, Ingeniero de Sistemas. Integrante del Grupo de Estudio en Metodologías de Ingeniería de Software (GEMIS), Universidad Tecnológica Nacional. Facultad Regional Buenos Aires. Ciudad Autónoma de Buenos Aires, Argentina.

<https://orcid.org/0000-0002-7329-9301>

Correo electrónico: mauricio.rozo.rodriquez@gmail.com

Inés Casanovas

Doctora en Informática, Magister en Docencia Universitaria, Magister en informática, Ingeniera en Sistemas de Información. Integrante del Grupo de Estudio en Metodologías de Ingeniería de Software (GEMIS), Universidad Tecnológica Nacional. Facultad Regional Buenos Aires. Ciudad Autónoma de Buenos Aires, Argentina.

<https://orcid.org/0000-0002-3806-2026>

Correo electrónico: inescasanovas@gmail.com

María Florencia Pollo Cattaneo

Doctora en Ciencias Informáticas, Magister en Ingeniería de Software, Especialista en Construcción de Sistemas Expertos, Ingeniera en Sistemas de Información. Directora del Grupo de Estudio en Metodologías de Ingeniería de Software (GEMIS), Universidad Tecnológica Nacional. Facultad Regional Buenos Aires. Ciudad Autónoma de Buenos Aires, Argentina.

<https://orcid.org/0000-0003-4197-3880>

Correo electrónico: 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, considerado como un factor crítico para la calidad de

producto software. En consecuencia, es necesario una mejora significativa de la efectividad y cumplimiento de las tareas que componen este proceso. Es ahí donde el uso de métodos y principios de gestión de conocimiento se convierten en la base para gestionar este proceso. Siguiendo este argumento, se estudió cómo las ontologías pueden utilizarse en la gestión de conocimiento en la fase de pruebas de la ingeniería del software. Se desarrolló un modelo basado en ontologías para ese dominio que abarca la identificación, formalización, distribución, y retención de conocimiento definiendo los indicadores para cada una de estas dimensiones.

Palabras clave: Gestión de Conocimiento; Fase de Pruebas; Ingeniería de Software; Ontologías

Abstract

The processes and activities in quality management of software engineering generate a large volume of knowledge, which is considered a critical factor for the quality of the software product. Therefore, a significant improvement of the effectiveness and fulfillment of the tasks that encompass this process, is required. It is there where the use of knowledge management methods and principles becomes the basis for managing the process. Following this argument, it was studied how ontologies can be used in knowledge management in the testing phase of software engineering. An ontology-based model was developed for this domain that covers the identification, formalization, distribution, and retention of knowledge, defining the indicators for each of these dimensions.

Keywords: Knowledge Management; Testing Phase; Software Engineering; Ontologies

Introducción

El conocimiento en los últimos años ha cobrado importancia dentro de las organizaciones, ya que se lo considera el elemento que genera valor y riqueza, aportando además ventajas competitivas en las economías mundiales. Estas economías

se basan en el conocimiento, incorporándose en los mercados dinámicos, en la competencia global, en la estructura organizativa a través de redes y en el manejo de industrias para una producción flexible. Por lo tanto, el motor del crecimiento es la innovación y el conocimiento (Silva, Cerda y Altamirano, 2017). Ahora bien, el conocimiento y la cultura organizacional de las empresas forman parte fundamental del llamado capital intelectual, el cual es un valor estratégico dentro de los activos intangibles de cualquier organización. Así mismo, existe un amplio acuerdo entre investigadores y empresarios para que el conocimiento constituya uno de los activos intangibles estratégicos para la organización y el factor más importante para el crecimiento económico en la actualidad (Liberona y Ruiz, 2013; Ponjuán, 2015).

De acuerdo con lo anteriormente mencionado, el conocimiento está inmerso dentro del quehacer organizacional y en ese sentido el concepto de gestión de conocimiento (GC de ahora en adelante), toma relevancia y es definido como la disciplina encargada de diseñar e implementar sistemas con el objetivo de identificar, capturar y compartir el conocimiento de una organización, de forma que pueda ser convertido en valor para la misma (Martínez, 2014; Pérez-Montoro, 2016; Durán, Gamboa y Builes, 2017). La GC empieza con la creación, descubrimiento y recolección interna de conocimiento y de las mejores prácticas. Luego se deben compartir y entender las prácticas que la organización puede usar y de esta manera, ajustarlas y aplicarlas a nuevas situaciones en la búsqueda de la mejora en el desempeño organizacional (Avendaño y Flores, 2016; Marulanda, López, y Mejía, 2017). En ese sentido, en la ingeniería de software, la GC está desarrollándose en la medida de las necesidades que expresan estas organizaciones. Por lo tanto, es necesario expandir su uso en las diferentes fases de la ingeniería de software, debido a la especificidad que exige la evolución de sus procesos.

Una de las etapas que genera una gran cantidad de conocimiento y que requiere de una gestión a través de las herramientas informáticas, es la gestión de calidad (Souza, Almeida y Vijaykumar, 2015a).

Es indudable que los procesos y actividades de la gestión de calidad tienen gran importancia en la ingeniería de software debido a que permiten evaluar las soluciones informáticas, determinando el nivel de calidad que poseen. En consecuencia, esta fase de pruebas reviste una especial importancia 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 (Jústiz-Núñez, Gómez-Suárez, y Delgado-Dapena, 2014; Fernández y Angel, 2015). Por otro lado, Souza et al. (2013), entienden que las pruebas de software (PS de ahora en adelante), son un proceso intensivo en conocimiento, y se hace necesario proporcionar soporte computarizado para las tareas de adquisición, procesamiento, análisis y disseminación de conocimiento para su reutilización. Sin embargo, existen organizaciones que no les prestan 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, eliminadas o reducidas en la planificación, afectando la calidad de software (Souza, Falbo, y Vijaykumar, 2013b; Dávila, García, y Córdor, 2017).

Wnuk, y Garrepalli (2018) 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 por fuera de estos ámbitos (Durán, Gamboa, y Builes, 2017; Souza, Falbo, y Vijaykumar, 2017).

La 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 (Avendaño y Flores, 2016).

Entendiendo que, en la ingeniería de software la GC está desarrollándose, surgen como interrogantes cuál es el estado de situación de la GC aplicada 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 a la fase de PS y surgen como elementos para potencializar el desarrollo y masificación en el contexto de la ingeniería de software. Es necesario avanzar en 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.

Por otro lado, aunque son reconocidas como un instrumento importante por la comunidad de GC, las ontologías no se utilizan ampliamente en estas iniciativas. Una ontología es una especificación explícita de una conceptualización, en la que se modela el vocabulario del dominio, básicamente usando los conceptos, características y sus relaciones (Flores, y Hadfeg, 2017; González, Sánchez, y Montejano, 2017; Souza et al., 2013). Vasanthapriyan et al. (2017) definen que una ontología proporciona una vista estructurada del conocimiento del dominio de las PS y actúa como un repositorio de conceptos, por lo que, las ontologías y las tecnologías de la web semántica han recibido más atención y se han utilizado gradualmente en la representación del conocimiento. Zapata et al. (2010) afirman que las ontologías son necesarias para verificar la coherencia de una base de conocimientos, proveer un medio para estructurar las PS y sugerir respuestas apropiadas cuando las PS indican que existen fallas; la literatura reporta pocos casos de aplicación de ontologías en las PS.

Ruy et al. (2016) consideran que la falta de una base de conocimientos adecuada en su propio contexto por parte de los entornos de PS, hace que los evaluadores de software consulten los limitados conocimientos disponibles o consulten a sus pares, lo que tendría

un gran impacto en su proceso de toma de decisiones, es por ello que, dada una gran importancia al conocimiento para las PS y los beneficios potenciales de administrar el conocimiento de PS, utilizando tecnologías web semánticas basadas en ontologías, proponen una solución a estos desafíos.

En este contexto, el objetivo del presente trabajo es proponer una solución basada en la utilización de metodologías ontológicas para la fase de pruebas en el dominio de GC, para ello, se organiza de la siguiente manera: en la Sección 2, se expone el enfoque metodológico elegido, en la Sección 3, se presenta la propuesta ontológica 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.

1. Enfoque Metodológico

El presente trabajo se basa metodológicamente en la visión del constructivismo y, el enfoque metodológico que lo sustenta es el cualitativo, donde el investigador hace cuestionamientos basados principalmente en perspectivas constructivistas que demanda estrategias de investigación (Creswell, 2016; Labra, 2016).

El diseño utilizado en esta investigación es el descriptivo, y en relación con los métodos, se utiliza la revisión sistemática documental exploratoria (Hernández Sampieri, Fernández Collado, Baptista, 2010; Souza et al., 2015a) del conocimiento existente de los modelos ontológicos, para determinar el estado de la cuestión y fundamentación de los desafíos en la gestión de calidad de software en el dominio de la GC.

A su vez, se realiza un estudio analítico y comparativo de las ontologías aplicadas a la gestión de calidad existentes para determinar su aplicabilidad a la solución de la problemática, concluyendo con la elaboración de una propuesta metodológica de orden ontológico que minimice la brecha de conocimiento detectada en la gestión de calidad dentro de la ingeniería de software. Finalmente, se desarrolla una ontología de aplicación para un modelo de GC aplicado a la PS de la ingeniería de software.

Propuesta Ontológica

Las ontologías proporcionan una vista estructurada del conocimiento y actúa como un repositorio de conceptos en el dominio, con lo cual es de gran importancia para las PS. Los beneficios potenciales para administrar este conocimiento, es potencializado utilizando tecnologías de web semántica basado en ontologías (Vasanthapriyanet al., 2017). Ahora bien, las PS son un proceso complejo y de conocimiento intensivo cuya eficiencia se puede mejorar, al reutilizar el conocimiento relacionado con la prueba. En este contexto, Souza et al. (2013c), exponen que una ontología de dominio de referencia en las PS define el vocabulario compartido que puede ser utilizado en un sistema GC, lo cual redundante en facilitar la comunicación, la integración, la búsqueda y representación del conocimiento de pruebas.

Para identificar las ontologías propuestas aplicables a la PS, se realizó una revisión sistemática documental exploratoria (Souza et al., 2015a) del conocimiento existente, buscando trabajos que propongan ontologías en el dominio de las PS. Fueron consultadas las siguientes bases de datos y metabuscadores, seleccionados de acuerdo con la importancia propuesta por Martínez (2006): IEEE Xplore, Science Direct, ACM Digital Library, SAGE Journals, Springerlink, Emeraldinsight, GREDOS, Dialnet y Google Académico. Sobre estas bases de datos, se utilizaron los siguientes términos: "Pruebas de software", "Software testing", "Ontology", " Ontología" (ver la tabla 1) aplicada en tres campos de metadatos (título, resumen y palabras clave).

El proceso de búsqueda consideró los artículos publicados sin definir un período de tiempo y luego de aplicar la cadena referida en la tabla 1, se obtuvieron 47 publicaciones. El proceso de selección para que los artículos fuesen incorporados dentro de la población objeto de estudio, estuvo basado en la aplicación del criterio de inclusión: la publicación científica refiere una Ontología aplicada a la fase de pruebas en la ingeniería de software; y de los criterios de exclusión: en las palabras claves y/o en el título de la publicación no se consideran los términos Ontología y/o pruebas de software. Con las fases de selección aplicadas, se tuvo una reducción de 33 publicaciones, equivalente al 70% del resultado de

la búsqueda, finalmente son 14 los artículos de investigación publicados que fueron analizados en esta investigación (ver tabla 2).

Tabla 1.

Cadena de búsqueda en el estudio ontologías aplicadas a las pruebas de software.

Área	Términos de búsqueda
Ingeniería de Software	"Pruebas de software", "Software testing"
Ingeniería Ontológica	"Ontología", "Ontology"
Cadena de búsqueda: ("Pruebas de software" or "Software testing") and ("Ontology" or " Ontología ")	

Tabla 2.

Investigaciones que proponen Ontologías aplicadas a las Pruebas de Software.

Investigación	Ontología Propuesta	Autor(es)	Año
Developing software testing ontology in UML for a software growth environment of web-based applications	STOWS (Software Testing Ontology for Web Service)	Zhu Hong, Qingning Huo	2005
Towards the establishment of an ontology of software testing	OntoTest	Barbosa Ellen Francine, Elisa Yumi Nakagawa, José Carlos Maldonado	2006
Performing Unit Testing Based on Testing as a Service (TaaS) Approach	TaaS	YuLian, Shuang Su, Jing Zhao	2008
Ontology-Based Test Modeling and Partition Testing of Web Services	modelo de ontología de prueba (TOM)	Xiaoying Bai, Shufang Lee, Wei-Tek Tsai, Yinong Chen	2008
A Strategic Test Process Improvement Approach Using an Ontological Description for MND-TMM	MTO (Ontología MND-TMM)	Hoyeon Ryu, Dong-Kuk Ryu, JongmoonBaik	2008
Knowledge-based software test generation	Arquitectura de ontología basada en Test Generator	Nasser Valeh H., Weichang Du, Dawn Maclsaac	2009
Test Case Reuse Based on Ontology	Testing ontology based SWEBOK and software quality model	Lizhi Cai, Weiqin Tong, Zhenyu Liu, Juan Zhang	2009
An application of ontology to test case reuse	Test case ontology	Shaojie Guo, Juan Zhang, Weiqin Tong, Zongheng Liu	2011
An Ontology Based Approach for Test Scenario Management	ontology for test management.	Sapna P. G., Hrushikesh Mohanty	2011
Semi-automatic Generation of a Software Testing Lightweight Ontology from a	Lightweight ontology of a domain	ArnicansGuntis, Dainis Romans, UldisStraujums	2012

Glossary Based on the ONTO6 Methodology			
Ontology-based Testing Platform for Reusing	knowledge reuse based on the ontology representation	Li Xuexiang, Wenning Zhang.	2012
Development of Ontology-Based Intelligent System For Software Testing	framework of an ontology construction for integrated teaching of programming foundations with testing	Anandaraj A., Kalaivani P., Rameshkumar, V.	2013
ROoST: Ontología de referencia en pruebas de software	ROoST	Souza, É. F. D., Falbo, R. D. A., Vijaykumar, N. L.	2017
Ontology support software testing at the University of Computer Sciences	ontology for Software testing	Aliuska Castañeda Martínez, Carlos Parker Leyva, Yamilis Fernández Pérez, Yoan Antonio López Rodríguez	2018

Souza et al. (2013c), establecen que la calidad de una ontología está basada en las dimensiones de estructura y cobertura conceptual, por ser características medibles y con posibilidades de ser evaluadas objetivamente. En relación con la estructura, en la presente investigación se consideró el cumplimiento de las siguientes características: es formalmente rigurosa, implementa relaciones no taxonómicas, posee un método de evaluación. Con respecto a la cobertura conceptual, se consideró si poseen una cobertura de dominio en las PS, está basado en una norma internacional y tiene en cuenta la reutilización de las ontologías. De acuerdo con lo anteriormente descrito, las características de cada dimensión fueron utilizadas como categorías de agrupación, por lo que, en los 14 artículos analizados, se utilizaron las siguientes categorías de aplicación: es formalmente rigurosa, implementa relaciones no taxonómicas, posee un método de evaluación, posee una cobertura de dominio en las PS, está basado en una norma internacional y si considera la reutilización de las ontologías.

La tabla 3 resume la completitud de cada una de las características para las dos dimensiones analizadas, evaluando cada una de las investigaciones bajo tres criterios: cumple con la característica, parcialmente cumple con la característica y no cumple con la característica, de la cual, surge una matriz de vacancia para las ontologías aplicadas a las PS. De acuerdo con la matriz de vacancia (expuesta en la tabla 3), se puede

observar que ninguna de las ontologías propuestas en las investigaciones analizadas cumple con la totalidad de las características para las dimensiones objeto de estudio.

Ahora bien, analizando cuantitativamente las investigaciones por dimensión, asignando los siguientes valores para cada uno de los criterios: 2 para cuando cumple con las características, 1 para cuando cumple parcialmente y 0 para cuando no cumple, la tabla 4 describe las valoraciones cuantitativas agrupadas por dimensión de los artículos analizados, resultando que la ontología ROoST propuesta en Souza et al. (2017), obtiene mayor puntuación total, aun cuando denota una debilidad en la dimensión de estructura, debido a que no implementa relaciones. Le sigue en la puntuación OntoTest, expresando debilidades en las características de evaluación y reutilización de las ontologías. Cabe destacar que la ontología ROoST, siendo una ontología propuesta recientemente, deja abierto el camino para que se fortalezca la reutilización con ontologías en la dimensión de cobertura conceptual, así como su evaluación, aplicando un método reconocido e incorporando la implementación de relaciones no taxonómicas en la dimensión de estructura.

Tabla 3.

Matriz de Vacancia de las investigaciones analizadas en relación con Ontologías aplicadas a las PS.

	No cumple
	Cumple Parcialmente
	Si Cumple

Investigaciones Analizadas	Estructura			Cobertura Conceptual		
	Formalmente Rigurosa	Implementa Relaciones	Método Evaluación	Cobertura de Dominio	Basado Norma Internacional	Reutiliza Ontologías
Developing software testing ontology in UML for a software growth environment of web-based applications						
Towards the establishment of an ontology of software testing						
Performing Unit Testing Based on Testing as a Service (TaaS) Approach						
Ontology-Based Test Modeling and Partition Testing of Web Services						

A Strategic Test Process Improvement Approach Using an Ontological Description for MND-TMM								
Knowledge-based software test generation								
Test Case Reuse Based on Ontology								
An application of ontology to test case reuse								
An Ontology Based Approach for Test Scenario Management								
Semi-automatic Generation of a Software Testing Lightweight Ontology from a Glossary Based on the ONTO6 Methodology								
Ontology-based Testing Platform for Reusing								
Development of Ontology-Based Intelligent System for Software Testing								
ROoST: Ontología de referencia en pruebas de software								
Ontology support software testing at the University of Computer Sciences								

Tabla 4.

Valoración cuantitativa agrupadas por dimensión de las investigaciones analizadas en relación con las Ontologías aplicadas a las PS.

Investigaciones Analizadas	Estructura				Cobertura Conceptual				TOTAL, PUNTAJACION
	Formalmente Rigurosa	Implementa Relaciones	Método Evaluación	TOTAL	Cobertura de Dominio	Basado Norma Internacional	Reutiliza Ontologías	TOTAL	
ROoST: Ontología de referencia en pruebas de software	2	0	1	3	2	2	1	5	8
OntoTest: Towards the establishment of an ontology of software testing	2	1	0	3	2	2	0	4	7
Ontology-Based Test Modeling and Partition Testing of Web Services	2	2	0	4	1	2	0	3	7
An Ontology Based Approach for Test Scenario Management	1	2	0	3	1	2	0	3	6
Developing software testing ontology in UML for a software growth environment of web-based applications	2	2	0	4	2	0	0	2	6
Knowledge-based software test generation	2	2	0	4	1	0	0	1	5
Semi-automatic Generation of a Software Testing Lightweight Ontology from a Glossary Based on the ONTO6 Methodology	0	0	0	0	2	2	0	4	4
Ontology-based Testing Platform for Reusing	1	2	0	3	1	0	0	1	4
Ontology support software testing at the University of Computer Sciences	0	0	1	1	1	2	0	3	4
A Strategic Test Process Improvement Approach Using an Ontological Description for	0	0	0	0	1	2	0	3	3

MND-TMM									
Test Case Reuse Based on Ontology	0	0	0	0	1	2	0	3	3
An application of ontology to test case reuse	0	2	0	2	1	0	0	1	3

Para el desarrollo de la ontología de dominio que sustente a un modelo de GC, se adopta el método propuesto por Stuart (2016), que incorpora la identificación del software apropiado y la sustentabilidad de la ontología, como lo proponen Barber et al. (2018). Además, se complementa la ontología propuesta, con el fortalecimiento de la reutilización con ontologías en la dimensión de cobertura conceptual, mejorando su evaluación e incorporando la implementación de relaciones no taxonómicas en la dimensión de estructura, no propuesta en la ontología ROoST. Es aquí donde el presente trabajo de investigación avanza en fortalecer e incorporar mejoras en la identificación y procesamiento en la GC para las PS.

A continuación, se listan los objetivos de la ontología de dominio propuesta, que se basa en Souza et al. (2017) e incorpora aspectos establecidos en lo que PMI (2017) recomienda para la gestión de calidad en las PS:

1. Servir de apoyo al aprendizaje humano en el proceso de PS.
2. Servir como base para estructurar y representar el conocimiento relacionado con las PS.
3. Servir como modelo de referencia para integrar herramientas que soportan las PS.
4. Servir de referencia para incorporar recursos de prueba en el enfoque de documentación semántica.
5. Servir de apoyo para la fase de ejecución en la gestión de calidad en relación con el análisis de datos, toma de decisiones y representación de datos.
6. Proveer un entorno adecuado para que el logro de los objetivos de calidad en las PS sea superador.

De acuerdo con la metodología Stuart (2016), se proponen para cada uno de los pasos, una serie de preguntas de competencia (PC) para cumplir con los propósitos de la capa ontológica del modelo GC. A su vez, como lo indica Contreras et al. (2018), estos pasos equivalen a la elicitación de términos para que la ontología sea capaz de contestar con

todos los conceptos, propiedades entre conceptos y datos, los axiomas y reglas definidas. A continuación, se listan las PC consideradas en el presente trabajo:

1. Alcance de la ontología

PC01.1 ¿En qué proyecto se demanda la gestión de calidad?

PC01.2 ¿Cuándo comienza y finaliza la gestión de calidad?

PC01.3 ¿Qué procesos de gestión de calidad del proyecto se va a utilizar?

PC01.4 ¿Cuándo inicia y finaliza cada proceso de gestión de calidad del proyecto se va a utilizar?

PC01.5 ¿Cuándo inicia y finaliza cada actividad de PS?

PC01.6 ¿Qué entornos de pruebas son los utilizados en la PS?

2. Reutilización de la ontología

PC02.1 ¿Qué relación tiene los artefactos entre sí en la PS?

PC02.2 ¿Qué actividades tienen relación con otros procesos de gestión de calidad del proyecto?

3. Identificación del software apropiado

PC03.1 ¿Qué recursos software se utilizan en la PS?

4. Adquisición de conocimiento

PC04.1 ¿Qué tipos de artefactos se producen en la actividad de PS?

5. Identificación de términos importantes

PC05.1 ¿Qué actividades se desarrolla en la ejecución de PS?

PC05.2 ¿Qué técnicas se van a utilizar en la PS?

PC05.3 ¿Qué niveles de prueba son que se utilizan en la PS?

6. Identificación de términos adicionales, atributos y relaciones

PC06.1 ¿Qué actividades son dependientes de otras en la ejecución de PS?

PC06.2 ¿Qué actividades son dependientes de otros procesos de gestión de calidad del proyecto?

7. Especificación de las definiciones

PC07.1 ¿Qué relaciones existen entre las actividades y los artefactos en las PS?

PC07.2 ¿Qué relaciones no taxonómicas se utilizan en la identificación de relaciones entre actividades y artefactos?

8. Integración con ontologías existentes

PC08.1 ¿Qué artefactos se utilizan en la PS?

PC08.2 ¿Qué actividades son reutilizables en la PS?

9. Implementación

PC09.1 ¿Qué etapas se requiere para implantar la ontología en el modelo GC?

PC09.2 ¿Qué técnicas se requieren para implementar la ontología en el modelo GC?

10. Evaluación

PC010.1 ¿Qué actividades de verificación se realizan en la PS?

PC010.2 ¿Qué actividades de evaluación se realizan en la PS?

11. Documentación

PC011.1 ¿Qué artefactos se producen en la de PS?

12. Sustentabilidad

PC012.1 ¿Qué recursos humanos se utilizan en la PS?

PC012.2 ¿Qué recursos hardware se utilizan en la PS?

Tal como lo indica Contreras et al. (2018), estas preguntas abarcan los requisitos funcionales que debe cumplir la ontología propuesta. Ahora bien, para representar estos requisitos, se utiliza el enfoque OPL, como lo fundamenta Quirino et al. (2017). De acuerdo con lo anterior, se procede a especificar cómo está conformada la ontología que hace parte de la GC propuesta.

Las PS constituyen una de las etapas del desarrollo del proyecto. En ese sentido, se considera como un proceso particular dentro del contexto general de gestión del proyecto. El proceso de gestión de calidad como denomina PMI (2017) a las PS, está compuesto por las etapas de planificación, gestión y control. A su vez cada una de estas etapas, la componen actividades para ser ejecutadas dentro de la prueba.

A los efectos del presente trabajo de investigación, la etapa que debe incorporarse en la ontología propuesta es la gestión de calidad, debido a que es la que aborda la calidad, tanto de la gestión del proyecto como la de sus entregables, tal como lo indica PIM (2017).

Las actividades que se consideran realizar en la ontología de GC son: la planificación de pruebas, diseño de pruebas, ejecución de pruebas, finalización de pruebas y análisis de

los casos de pruebas. Es necesario tener presente que las actividades de PS dependen de los niveles de prueba a ejecutar. Para este caso se consideran las pruebas unitarias, pruebas de integración, pruebas de sistema y pruebas de regresión. De acuerdo con los anteriores conceptos, se estarían respondiendo las preguntas relacionadas a PC01, PC05 y PC06.

Como lo afirman Souza et al. (2017), una actividad realizada puede tener como componente una participación de artefactos que pueden ser: de creación (que se generan durante la ocurrencia de la actividad), de uso (lo que significa que el artefacto solo se usa durante la actividad), de cambio (es decir que se va modificando el artefacto durante la actividad).

Por otro lado, se pueden mencionar los documentos que son generados en los diferentes procesos de gestión de la calidad que propone PMI (2017), como el plan de gestión de la calidad, los documentos de prueba y los documentos de evaluación (reportes de casos de pruebas exitosos y fallidos), la información de desempeño del trabajo, las actualizaciones al plan para la dirección del proyecto y las actualizaciones a los documentos del proyecto.

Otro tipo de artefacto es el elemento de software, que se refiere a una pieza de software, producida durante la ejecución de una actividad, en las PS, pero que no se considera un producto de software completo. Es un resultado intermedio.

Por último, otro artefacto a considerar es el elemento de información, que son los datos utilizados o producidos durante el proceso. A su vez se incorpora en la ontología el rol de relaciones no taxonómicas entre las actividades y artefactos, entendiendo que este tipo de relaciones representan cualquier asociación entre conceptos diferentes (De la Villa Moreno, 2016). En ese sentido, el rol a considerar son las reglas de asociación, que vinculan las actividades y artefactos cuando éstas no son explícitas. Basado en lo anterior, se responden las preguntas PC02, PC04, PC07 y PC08.

La secuencialidad de las actividades se inicia con el plan de pruebas, que incorpora el diseño de los casos de prueba, basado en la documentación del proyecto (documentos

funcionales, especificación de requisitos, modelos conceptuales, etc.). En ese sentido, los artefactos requieren estas entradas de información y su producto es el caso de prueba propiamente dicho. A su vez, su ejecución produce un resultado que debiera ser el esperado.

Basado en lo anterior, el diseño de los casos de prueba requiere que se aplique una técnica de prueba, concepto de tipo procedimental incorporado en la ontología propuesta. Para el presente trabajo, se consideran las técnicas de caja negra y caja blanca. Es importante entender que la aplicabilidad de las técnicas de pruebas depende del nivel de prueba. Como ejemplo, se tiene que, para pruebas unitarias, se utiliza la técnica de caja blanca, mientras que la técnica de caja negra aplica a todos los niveles de prueba, según indican Souza et al. (2017).

El caso de prueba tiene en esencia un fragmento de sistema a probar que, dependiendo del nivel, puede ser abarcativo o no, generando posibles propiedades de especialidad. Para la ejecución de los casos de prueba en principio se realiza la secuencia de acciones necesarias para completar el escenario de prueba, por otro lado, también puede utilizarse el llamado de otros módulos objeto de prueba u otros componentes dependientes, denominándose código de prueba.

La ejecución del caso de prueba requiere como entrada el código de prueba y los casos de prueba. En términos de efectividad, si un caso de prueba es ejecutado, requiere el código de prueba y, el resultado es la finalización de la actividad; por consiguiente, se produce el resultado de la prueba.

El resultado de la prueba puede tener dos valoraciones, cuando finaliza correctamente dando el cierre de la prueba, o cuando falla, y se generan los incidentes que pueden ser defectos, fallas o problemas identificados. Hay que tener claro que, en el incidente se documenta la información de la falla o el defecto que lo ocasiona. Este incidente puede estar disponible para futuros análisis, sin embargo, sale del alcance de las PS en el dominio de GC. Al finalizar las pruebas, el siguiente paso es analizar los resultados,

generando un informe del estado de las pruebas, en ese sentido se responden las preguntas PC010, PC011.

Para la ejecución de las actividades de prueba, se requiere de recursos tanto de hardware como de software. Para ello, dentro de la ontología propuesta, se identifica el concepto de entorno de prueba, el cual permite administrar el rol que cumple un recurso tecnológico en una actividad de prueba. Es necesario entender que las PS son realizadas por recursos humanos que pueden tener a su vez un grado de participación diferente, como es el líder del equipo de pruebas, el diseñador de pruebas y el ejecutor o “tester”, los cuales están vinculados a una ejecución de prueba de un proyecto. De esta manera, se responden las preguntas PC03, PC012.

Para contestar las preguntas de competencia PC09, se sugiere incorporar el modelo conceptual propuesto en términos de un lenguaje de ontología operacional (OPL), definido en Quirino et al. (2017).

La figura 1 describe la ontología aplicada a las PS en el dominio de un modelo GC.

3. Resultados Obtenidos

La validación de la ontología embebida en el modelo de GC en el dominio de las PS, 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. El segundo componente define la efectividad de operación al aplicar las fases del modelo de GC en el dominio de PS y abarca la identificación de conocimiento, formalización de conocimiento, distribución del conocimiento, y retención de conocimiento. A partir de las dimensiones, se definen los indicadores, lo que permite establecer los ítems que componen el instrumento de investigación. La tabla 5 delimita los componentes, dimensiones e indicadores a fines de determinar la estructura de validación.

Tabla 5.

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, la tabla 6 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 (Hernández et al., 2010), para valores menores de 2,5 representa estar en desacuerdo, mientras que, 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, lo cual indica que lo que propone la ontología, no aporta, ni resta en el modelo de GC. 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, este trabajo de investigación aporta crecimiento en términos de aprendizaje dentro de las organizaciones. A su vez, los Expertos concuerdan en 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 6.

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

Componente	Dimensión	Valor Prom. x Dimen.	Valor Prom. x Componente	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			

4. Conclusiones y Trabajo Futuro

La ontología propuesta en el dominio de un modelo GC y de las PS, 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, coinciden en el fortalecimiento de la fase de pruebas dentro de la gestión del proyecto de software.

Como trabajo futuro se procederá a mejorar aspectos relacionados con la dimensión “Retención de conocimiento” incorporando el desarrollo de cuadernos (“Guías”) digitales explicativas sobre las actividades de la PS y sobre la ontología, para que reúna este contexto dentro de su funcionalidad, el cual fue uno de los indicadores con menor valoración. Por otro lado, es necesario continuar incorporando aspectos en el desempeño de la ontología en el dominio del modelo GC desde el enfoque de la gestión del recurso humano, para esta misma dimensión.

Referencias Bibliográficas

1. Silva, E. E. J., Cerda, L. A. L., & Altamirano, J. F. L. (2017). *La gestión del conocimiento organizacional basado en las perspectivas del Balanced Scorecard como factor clave para la innovación de las PYMES*. Revista Publicando, 4(12 (2)), 640-657. Recuperado el 15 de marzo de 2020, de bit.ly/31YjhYx
2. Liberona, D., & Ruiz, M. (2013). *Análisis de la implementación de programas de gestión del conocimiento en las empresas chilenas*. Estudios gerenciales, 29(127), 151-160. Recuperado el 6 de enero de 2020, de bit.ly/3q2AHIP
3. Ponjuán Dante, G. (2015). *La gestión del conocimiento desde las ciencias de la información: responsabilidades y oportunidades*. Revista Cubana de Información en Ciencias de la Salud, 26(3), 206-216. Recuperado el 12 de febrero de 2020, de bit.ly/31X0yww

4. Martínez, M. I. R. (2014). *La gestión del conocimiento y la ciencia de la información: relaciones disciplinares y profesionales* (Doctoral dissertation, Universidad Carlos III de Madrid). Recuperado el 15 de marzo de 2020, de bit.ly/30ZbpGY
5. Pérez-Montoro, M. (2016). *Gestión del conocimiento: orígenes y evolución*. *El profesional de la información*, 25(4), 526-534. Recuperado el 8 de enero de 2020, de bit.ly/2Y1eLrh
6. Durán, D. E. S., Gamboa, A. X. R., & Builes, J. J. (2017). *Aplicación de la Gestión de Conocimiento al proceso de pruebas de software*. *Ingenierías Revistas USBMed*, 8(2), 6-13. Recuperado el 12 de febrero de 2020, de bit.ly/30ZyCZB
7. Avendaño Pérez, V., & Flores Urbáez, M. (2016). *Modelos teóricos de gestión del conocimiento: descriptores, conceptualizaciones y enfoques*. *Entre ciencias: diálogos en la Sociedad del Conocimiento*, 4(10). Recuperado el 12 de febrero de 2020, de bit.ly/3iOI2x4
8. Marulanda Echeverry, C. E., López Trujillo, M., & Mejía Salazar, M. H. (2017). *Minería de datos en gestión del conocimiento de pymes de Colombia*. *Revista Virtual Universidad Católica del Norte*, (50). Recuperado el 12 de febrero de 2020, de bit.ly/37irnhn
9. Souza, E. F., de Almeida Falbo, R., & Vijaykumar, N. L. (2015a). *Knowledge management initiatives in software testing: A mapping study*. *Information and Software Technology*, 57, 378-391. Recuperado el 15 de marzo de 2020, de bit.ly/3kNi137
10. Jústiz-Núñez, Dalila, Gómez-Suárez, Darlene, & Delgado-Dapena, Marta Dunia. (2014). *Proceso de pruebas para productos de software en un laboratorio de calidad*. *Ingeniería Industrial*, 35(2), 131-145. Recuperado el 12 de febrero de 2020, de bit.ly/30YxmWP
11. Fernández, F., & Ángel, M. (2015). *Aplicación de técnicas de pruebas automáticas basadas en propiedades a los diferentes niveles de prueba del software*. Recuperado el 15 de marzo de 2020, de bit.ly/343GRWv
12. Souza, E. F., Falbo, R. A., & Vijaykumar, N. L. (2013). *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). Recuperado el 12 de febrero de 2020, de bit.ly/2PVsFIk
13. Souza, E. F., Falbo, R. A., & Vijaykumar, N. L. (2013b). *Ontologies in software testing: a systematic*. In *VI Seminar on Ontology Research in Brazil* (p. 71). Recuperado el 8 de enero de 2020, de bit.ly/2DXRr6T
14. Dávila, Abraham, García, Cecilia, & Córdor, Sandra. (2017). *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. Recuperado el 8 de febrero de 2020, de bit.ly/3asEEW1

15. Wnuk, K., & Garrepalli, T. (2018). *Knowledge Management in Software Testing: A Systematic Snowball Literature Review*. e-Informatica Software Engineering Journal, 12(1), 51-78. Recuperado el 12 de febrero de 2020, de bit.ly/343K08L
16. Souza, É. F. D., Falbo, R. D. A., & Vijaykumar, N. L. (2017). *ROoST: reference ontology on software testing*. Applied Ontology, 12(1), 59-90. Recuperado el 8 de enero de 2020, de bit.ly/3iKB4sZ
17. Flores, V., & Hadfeg, Y. (2017). *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 Informação, (21), 99-114. Recuperado el 8 de enero de 2020, de bit.ly/3aBVX76
18. González Gola, F., Sánchez, A., & Montejano, G. A. (2017). *Asistencia dirigida por ontologías al diseño arquitectónico de videojuegos*. In XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires). Recuperado el 8 de enero de 2020, de bit.ly/3odinRH
19. Vasanthapriyan, S., Tian, J., Zhao, D., Xiong, S., & Xiang, J. (2017). *An Ontology-based Knowledge Management System for Software Testing*. In SEKE (pp. 230-235). Recuperado el 15 de marzo de 2020, de bit.ly/3201eBc
20. Zapata Jaramillo, C. M., Giraldo, G. L., & Urrego Giraldo, G. A. (2010). *Las ontologías en la ingeniería de software: un acercamiento de dos grandes áreas del conocimiento*. Revista Ingenierías Universidad de Medellín, 9(16). Recuperado el 12 de febrero de 2020, de bit.ly/2Y7FuST
21. Ruy, F. B., de Almeida Falbo, R., Barcellos, M. P., Costa, S. D., & Guizzardi, G. (2016). *SEON: A software engineering ontology network*. In European Knowledge Acquisition Workshop (pp. 527-542). Springer, Cham. Recuperado el 6 de enero de 2020, de bit.ly/3kJxDEO
22. Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. University of Nebraska, Lincoln. 4a. ed. Sage Publications, Inc.
23. Labra, O. (2016). *Positivismo y Constructivismo: Un análisis para la investigación social*. Rumbos TS. Un espacio crítico para la reflexión en Ciencias Sociales, 0(7), 12-21. Recuperado el 15 de marzo de 2020, de bit.ly/3arTfB1
24. Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2010). *Metodología de la investigación*. México: Editorial Mc Graw Hill, 15-40. ISBN 978-92-75-32913-9.
25. Souza, E. F., Falbo, R. A., & Vijaykumar, N. L. (2013c). *Using ontology patterns for building a reference software testing ontology*. In 2013 17th IEEE International Enterprise Distributed Object Computing Conference Workshops (pp. 21-30). IEEE. Recuperado el 12 de febrero de 2020, de bit.ly/2PVc9Xh
26. Martínez, Á. M. (Ed.). (2006). *La información especializada en Internet: directorio de recursos de interés académico y profesional*. Editorial CSIC-CSIC Press.

27. Stuart, D. (2016). *Practical Ontologies for Information Professionals*. Facet. Recuperado el 12 de febrero de 2020, de bit.ly/311dkeg
28. Barber, E., Pisano, S., Romagnoli, S., de Pedro, G., Gregui, C., Blanco, N., & Mostaccio, M. (2018). *Metodologías para el diseño de ontologías Web*. Información, Cultura Y Sociedad, 0(39), 13-36. Recuperado el 6 de enero de 2020, de bit.ly/3g5ACUu
29. PMI. (2017). *Guía de los Fundamentos para la dirección de Proyectos* (Guía del PMBOX®). Sexta Edición. Pennsylvania, EEUU.:PEARSON.
30. Contreras, M. C. B., Gutiérrez, F. P., Ortiz, J. A. R., & Ramírez, R. A. A. (2018). *Ingeniería Ontológica Aplicada en el Diseño de un Sistema de Ontologías para la Gestión de Horarios*. Pistas Educativas, 39(128). Recuperado el 12 de febrero de 2020, de bit.ly/3kP3RP5
31. Quirino, G. K., Barcellos, M. P., & Falbo, R. A. (2017). *OPL-ML: A Modeling Language for Representing Ontology Pattern Languages*. In International Conference on Conceptual Modeling (pp. 187-201). Springer, Cham. Recuperado el 8 de enero de 2020, de bit.ly/3kRJ2m0
32. De la Villa Moreno, M. Á. (2016). *Método para la construcción automática de ontologías basado en patrones lingüísticos*. (Doctoral dissertation, Ph. D. thesis, ETSI Informática). Recuperado el 6 de enero de 2020, de bit.ly/3h5oNPq