

Tipo de artículo: Artículo original
Temática: Ingeniería y Calidad de Software
Recibido: 25/05/2018 | Aceptado: 10/09/2018

Ontología de apoyo a las pruebas de software en la UCI.

Ontology support software testing at the University of Computer Sciences.

Aliuska Castañeda Martínez ^{[0000-0003-0223-8488]*}, Carlos Parker Leyva ^[0000-0002-3907-8329], Yamilis Fernández Pérez ^[0000-0001-5028-541X], Yoan Antonio López Rodríguez

Universidad de las Ciencias Informáticas. {[acastanedam](mailto:acastanedam@uci.cu), [cparker](mailto:cparker@uci.cu), [yamilisf](mailto:yamilisf@uci.cu), [valopez](mailto:valopez@uci.cu)}@uci.cu

* **Autor para correspondencia:** acastanedam@uci.cu

Resumen

El proceso productivo de la Universidad de las Ciencias Informáticas (UCI) ha sido certificado con CMMi nivel 2. Se tiene definido el área de proceso de Aseguramiento de la Calidad del proceso y del producto y dentro de este los subprocesos de pruebas de software. Estos subprocesos se vuelven complejos debido al gran volumen de documentación generada, la cantidad de personas, recursos, materiales y actividades involucradas. Todo ello, provoca dificultades en el entendimiento y comprensión total del proceso para una adecuada institucionalización, producto la cantidad de términos y conceptos usados, algunos ambiguos y en definición. No existe consenso sobre la terminología utilizada, no se gestiona el conocimiento generado durante la ejecución de estos subprocesos. En búsqueda de paliar estas dificultades surge la presente investigación, que tiene como objetivo desarrollar una ontología que apoye el subproceso de pruebas de software en la UCI, permitiendo la consistencia, organización y comunicación del conocimiento generado. Para la construcción de la ontología se utilizó la metodología Methontology. La ontología incluye el conocimiento tácito del personal involucrado en las pruebas, así como el conocimiento explícito representado en diferentes fuentes como el estándar NC ISO / IEC 25000 y los procedimientos definidos en la entidad. Esta ontología proporciona un vocabulario común y reduce los problemas de inconsistencias e integridad detectados. Se valida la solución ontológica utilizando un esquema para evaluar ontologías únicas para un dominio de conocimiento, evaluando el lenguaje utilizado para la codificación, la exactitud de la estructura taxonómica, el significado de los términos y conceptos representados; y por último la adecuación a los requerimientos especificados al inicio del desarrollo. Además, se utilizó el servicio de validación de W3C para certificar las condiciones lógicas formales de la ontología.

Palabras clave: gestión de conocimiento, metodología, ontologías, pruebas de software

Abstract

The production process of the Information Sciences University has been certified CMMI level 2. The Process and Product Quality Assurance process area has been defined and software testing subprocesses within it. Software testing subprocesses are complex because of the volume of documentation generated and the diverse knowledge levels of professionals involved; causing difficulties in its overall comprehension for an adequate institutionalization. The present paper aims to develop an ontology that supports the software testing subprocesses in the Information Sciences University, allowing the consistency, organization and communication of the generated knowledge. The methodology “Methontology” was applied for the construction of the present ontology. The ontology includes the knowledge of the professionals involved in the software testing, as well as the applicable knowledge represented by various sources, like the NC ISO / IEC 25000 standard, the procedures defined in the entity, among others. This ontology offers a common vocabulary and solves the problems of inconsistencies and integrity detected. To validate the proposal a tool was implemented, allowing to answer the competencies questions previously defined in the investigation, and the validation service of W3C was used to validate the consistency of the ontology. By this mean, the quality of the ontological proposal was guaranteed.

Keywords: Knowledge Management, Ontologies, Software Testing.

Introducción

Para desarrollar un software con la calidad requerida, se tiene en cuenta una serie de etapas o procesos por los que el sistema debe transitar antes de ser entregado al cliente. En estas etapas se comprueba el correcto funcionamiento del software dígase fiabilidad, eficiencia, portabilidad, compatibilidad, usabilidad entre otros aspectos fundamentales. La realización de las pruebas de software es esencial para velar por el correcto desarrollo del sistema, y que este cumpla los requisitos de cada etapa en la que se esté trabajando.

El proceso productivo de la Universidad de las Ciencias Informáticas (UCI) está certificado CMMI nivel 2, por lo que se definen actividades de aseguramiento de la calidad, respondiendo al área de proceso aseguramiento de la calidad de los procesos y productos (PPQA por sus siglas en inglés). Dentro de los subprocessos para la gestión de las actividades de calidad se encuentran la ejecución de las pruebas a nivel de proyectos, de centro y de gerencia. Las pruebas a nivel de gerencia se desarrollan en la Dirección de Calidad de la universidad, al igual que las de liberación de producto.

Estos subprocesos son complejos por la cantidad de personal involucrado de diferentes entidades con diversos niveles de conocimiento (Fernández et al. 2018). Además, influye también el volumen de documentación generada. Todo ello, provoca dificultades en el entendimiento y comprensión total de dichos subprocesos, fundamentalmente de los términos y conceptos utilizados. A través de entrevistas y una encuesta aplicada a administradores y asesores de la calidad de centros de desarrollo y Dirección de Calidad de la UCI, se ha comprobado que frecuentemente se presentan inconsistencias entre diferentes términos en el subproceso de pruebas de software. En particular, no existe consenso sobre los conceptos y terminologías utilizadas. Hay principios y métodos que aún están siendo definidos y consolidados.

Por otra parte, los estándares internacionales desarrollados por importantes organizaciones e instituciones de estandarización, como IEEE e ISO, la problemática se mantiene. Los estándares internacionales ((IEEE Computer Society & Standards Committee 2014), (ISO 2008)) que guían y orientan los aspectos fundamentales de estos subprocesos, existen conceptos un tanto ambiguos y solapados, además de disímiles significados para un mismo término. Pueden encontrarse inconsistencias y conflictos de terminología entre estándares de diferentes organizaciones e incluso en los de una misma organización (ISO/IEC/IEEE. 2017). Lo mismo sucede con los modelos de calidad. La NC ISO/IEC 25000 (NC ISO/IEC 25000 2011), contiene una visión general sobre la calidad de productos de software, pero hay aspectos, como la medición, que se complementan con otros estándares y ofrecen vistas parciales de este dominio. Las pruebas de software, son escasamente tratadas.

Todo lo anterior, conllevó a la búsqueda de técnicas y herramientas que proporcionen un vocabulario común para resolver el problema de integridad e inconsistencia, identificado en los diversos estándares y en el personal involucrado en la Dirección de Calidad de la UCI, con el objetivo de reducir la pérdida o desaprovechamiento del conocimiento.

El análisis realizado a varias investigaciones ((Duarte & Almeida Falbo 2000), (Ferreira et al. 2006), (Echeverría Pérez & Fernández Pérez 2011), (Souza 2013), (Echeverría Pérez et al. 2012)) arrojó un conjunto de ontologías aplicadas al dominio de calidad de software. Las mismas, como solución a la problemática actual, llegan a ser incompletas, generalizadoras de este dominio, no abordan todos los términos o las relaciones surgidas en la Dirección de Calidad, específicamente las que abarcan las pruebas de software. Estas soluciones ontológicas, no incluyen conceptos a nivel de proyecto, lo que limita la toma de decisiones.

Tras un análisis a estas ontologías, no se encontró una que abarcara todo el subproceso de pruebas de software a los diferentes niveles y conceptos que caracterizan el proceso de desarrollo en la UCI. Además, entre las que abordan el

subproceso de pruebas, se presentan términos semejantes, pero existen contradicciones. Ninguna ontología analizada cubre las necesidades de los centros de desarrollo de la UCI y la Dirección de calidad; el conjunto de conceptos cubiertos tampoco es homogéneo.

Estas diferencias detectadas en la bibliografía consultada están dadas porque corresponden a desarrollo e investigaciones efectuadas por separado, en distintos periodos de tiempo. Esta es la razón que guía a la integración para proporcionar una vista común en función de reducir las diferencias y conflictos.

Después de analizar diferentes variantes, se constató las potencialidades para apoyar la toma de decisiones en el subproceso de pruebas de software, teniendo como principal objetivo organizar y comunicar el conocimiento acumulado. Por todo lo anterior, se decide desarrollar una ontología en la UCI, abarcando los conceptos fundamentales y adaptándolas a las nuevas necesidades de los proyectos y la Dirección de Calidad de la UCI, teniendo como principal objetivo organizar y comunicar el conocimiento acumulado en dicho subproceso.

Materiales y métodos.

Para el análisis de las fuentes, ya sean documentales o de expertos, en búsqueda de un vocabulario común, se utilizaron métodos científicos teóricos y empíricos. Entre los primeros, están el histórico lógico, analítico-sintético y modelación. Entre los segundos, la observación, entrevista, observación y experimentación. Para el desarrollo de la ontología se utilizó como metodología METONTHOLGY, la cual guía el desarrollo de la ontología e identifica las tareas que deben realizarse cuando se construye. Como herramienta de modelado para la construcción y creación de las clases, propiedades, instancias, axiomas de la ontología se utilizó Protégé en su versión 5.2. Como lenguaje OWL, se trata de un lenguaje diseñado para usarse cuando la información contenida en los documentos necesita ser procesada por programas o aplicaciones.

Reutilización de Ontologías.

Es de vital importancia para el desarrollo de la ontología analizar la posibilidad de la reutilización o cualquier otra fuente de información existente en el dominio seleccionado. Se han consultado varias fuentes bibliográficas organizadas para su estudio en dos grupos. En el primer grupo, se encuentran los artículos dedicados al uso y desarrollo de las ontologías y el segundo grupo está integrado por aquellos trabajos que muestran ontologías desarrolladas.

En el primer grupo se encuentra el artículo (Lapiente 2013), donde la autora analiza y expone una serie de aspectos relacionados con las ontologías dígame: beneficios del uso de ontologías, definiciones de las mismas, componentes,

características, entre otros aspectos. En (Fernández Hernández 2015), la autora propone el desarrollo de un modelo para el diseño y construcción de un sistema de recuperación de información basado en ontologías.

Para el segundo grupo, el cual constituye el fundamental para el desarrollo de la investigación, fue necesario la búsqueda de ontologías relacionadas con las pruebas de software y la calidad. La revisión documental practicada arrojó un conjunto de ontologías aplicadas al dominio de calidad de software. Las mismas, como solución a la problemática actual, llegan a ser muy generales o incompletas, no abordan todos los términos o las relaciones surgidas en la dirección de calidad.

En (Felipe Alfonso 2011) se hace referencia al uso de una ontología como herramienta para la gestión del conocimiento con el objetivo de apoyar la realización de un Modelo Cubano de Calidad para el Desarrollo de Aplicaciones Informáticas (MCDAI) en la Industria Cubana de Software.

Una ontología que describe la calidad de software la presenta (Duarte & Almeida Falbo 2000). En ella solo aparecen elementos fundamentales de la calidad en general, sin abordar las pruebas de software, ni definir el modelo de calidad. Además, presentan conceptos no explicados correctamente y con ambigüedad.

La ontología desarrollada por (Ferreira et al. 2006) caracteriza el proceso de medición. Es abarcadora y elimina problemas de homonimia, sinonimia y otras dificultades presentes entre estándares. No abarca el subproceso de prueba de software.

Investigaciones como las de (Echeverría Pérez et al. 2012) y (Souza 2013), abarcan las pruebas de software y presentan ontologías generalizadoras de este dominio, pero hay conceptos a nivel de proyecto que no se incluyen, lo que limita la toma de decisiones.

Tras un análisis de dichas ontologías, no se encontró una que abarcara todo el subproceso de pruebas de software a los diferentes niveles y conceptos que caracterizan el proceso de desarrollo en la UCI. Además, entre las que abordan términos semejantes hay contradicciones. A pesar de que existen varias propuestas de investigación relacionadas con el tema no se encontró una ontología para pruebas de software disponible en la red que esté estrechamente vinculada con el dominio del procedimiento de pruebas del Departamento de Pruebas de Software.

Resultados y discusión

Descripción de la propuesta:

En la dirección de calidad de software es donde se llevan a cabo las pruebas de liberación y aceptación a nivel de gerencia en la UCI, estas se realizan de la siguiente forma: Un proyecto tiene especialistas y producto, este producto a su vez tiene módulos, cada módulo responde a determinados requisitos. Al analizarse el cumplimiento de estos requisitos pueden ser detectadas no conformidades, lo cual es realizado por un revisor. De cada no conformidad se analiza el impacto que provoca, el tipo y el estado.

Es esencial controlar el paso de un estado a otro de una no conformidad. Al desarrollar un proyecto informático se obtienen artefactos. Un artefacto se puede dividir en partes. Estos artefactos de proyectos para ser analizados requieren de documentación.

Para detectar las no conformidades se ejecutan diferentes tipos de pruebas, que se realizan a diferentes niveles dígase unidad, sistema, integración, interna, aceptación y liberación. Cada tipo de prueba corresponde a una determinada característica de calidad.

En la figura 1 se muestra el modelo conceptual confeccionado con la herramienta Visual Paradigm UML 8.0 Enterprise Edition, donde se presentan todas las clases y relaciones que se utilizaron para diseñar la ontología.

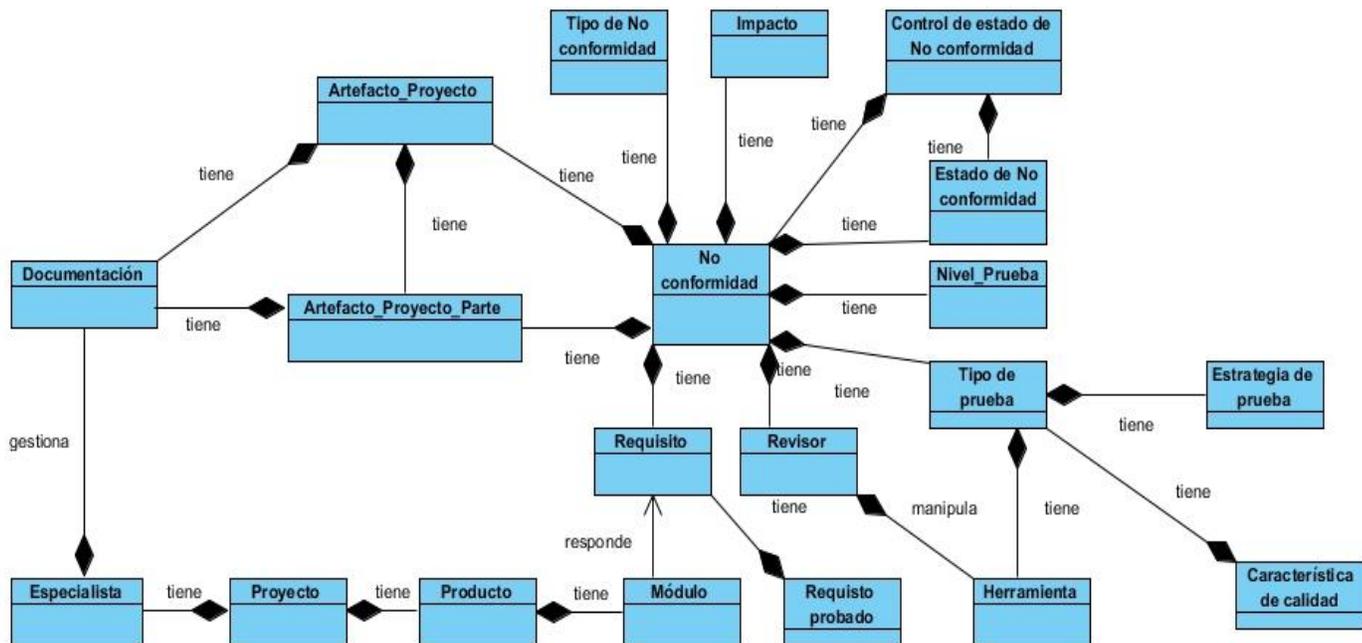


Figura 1. Modelo conceptual del subproceso de prueba.

Desarrollo de la ontología.

La metodología Methontology proporciona guías sobre cómo llevar a cabo el desarrollo de una ontología mediante las actividades de especificación, conceptualización, formalización, implementación y mantenimiento.

Una manera de determinar el alcance de la ontología e identificar sus requisitos funcionales es mediante el uso de preguntas de competencia. Esto consiste en describir una serie de preguntas en lenguaje natural, que el modelo ontológico debe estar capacitado para responder.

La función fundamental de las preguntas de competencia es limitar el alcance del modelo ontológico. A pesar de esto durante el diseño de la ontología las respuestas a estas preguntas pueden sufrir cambios. Este es el resultado principal de la actividad de especificación.

La ontología debe responder las siguientes preguntas:

1. ¿Cuál es la clasificación de una determinada característica de calidad acorde a los tipos de prueba?
2. ¿Qué tipos de prueba corresponden a una determinada característica de calidad?
3. ¿De qué tipo es una determinada no conformidad?
4. ¿Qué no conformidad presentó un determinado artefacto de proyecto y cuál es su clasificación?
5. ¿Qué impacto tiene una determinada no conformidad?
6. ¿Qué módulos tiene un determinado producto?
7. ¿Qué módulos pertenecen a un determinado requisito?
8. ¿Qué no conformidad detectó un determinado revisor?
9. ¿Cuál es el estado de una no conformidad?
10. ¿Qué documentación requiere un determinado artefacto de proyecto para ser analizado?
11. ¿A qué proyecto pertenece un determinado especialista?

Actividad de conceptualización:

Methontology recomienda realizar una serie de tareas de conceptualización en un orden determinado como son:
(Vitelli 2011)

1. Construcción del glosario de términos.

2. Construcción de la taxonomía de conceptos.
3. Construcción del diagrama de relaciones binarias.
4. Construcción del diccionario de conceptos.
5. Describir relaciones binarias.
6. Describir atributos de instancias
7. Describir atributos de clases
8. Describir constantes
9. Describir axiomas formales
10. Describir reglas
11. Describir instancias.

El glosario de términos incluye todos los términos relevantes del dominio dígase (conceptos, instancias y relaciones entre conceptos). En esta solución se propone 55 conceptos, 18 relaciones y 25 instancias.

Las taxonomías de conceptos representan mediante las clases y las instancias de esas clases una realidad contextualizada, que permite la organización y recuperación de los conceptos representados. Para la construcción de la taxonomía de conceptos, se seleccionan del glosario de términos aquellos términos que son conceptos. Dicha tarea se puede llevar a cabo de tres maneras: top-down, bottom-up y mezclando los dos procesos anteriores. En el presente trabajo se utilizó top-down.

Después de haber construido la taxonomía, se construyen los diagramas de relaciones binarias, que tiene como objetivo establecer las relaciones existentes entre clases. Con el modelo conceptual mostrado anteriormente en la figura 1, se puede constatar que todas las relaciones que existen son relaciones binarias.

Después de haber generado las taxonomías de conceptos y haberse construido el diagrama de relaciones binarias, se define cuáles son las propiedades que describen cada concepto de la taxonomía y las instancias de cada uno de los mismos. El diccionario de conceptos contiene todos los conceptos del dominio, sus relaciones e instancias.

En la presente investigación no se identificaron axiomas, solo fueron identificadas reglas. Para cada regla, Methontology se propone incluir la siguiente información: nombre, descripción en lenguaje natural, expresión que describe formalmente la regla, conceptos y relaciones utilizados en la regla. Además, propone especificar las

expresiones de las reglas utilizando el formato si <condiciones> entonces <consecuente>. La parte izquierda de la regla es una conjunción de condiciones simples, mientras que la parte derecha es una simple expresión de un valor de la ontología.

Actividad: Formalización

La actividad de formalización es la encargada de la transformación del modelo conceptual obtenido en la anterior actividad en un modelo formal o semi-computable. Para la transformación se utilizó el editor Protégé en su versión 5.2.

Las Clases son creadas partiendo desde la Clase Padre “Thing”, palabra clave que hace referencia a la W3C. Cada clase presentará: sub-clases, instancias, restricciones, anotaciones, clases equivalentes entre otros elementos necesarios.

Las clases no proveen la información suficiente para dar respuesta a las preguntas de competencia, a su vez las propiedades permiten constituir las relaciones, dado que estas representan las cualidades internas de los conceptos y constituyen las propiedades distintivas de los objetos. Las mismas pueden ser de 2 tipos, “object properties” y “data properties”.

Obtención del grafo de la ontología propuesta.

Para la obtención del grafo de la ontología desarrollada se utiliza OWLViz, el cual es un plugin para Protégé cuya función es permitir la visualización de los conceptos y relaciones creadas mediante grafos. Además de este plugin se debe tener instalado el GraphViz.

Actividad: Implementación

Una vez finalizada la ontología, existen varios lenguajes en los que se puede exportar, en este caso por requerimientos del cliente se exportó en el lenguaje OWL.

OWL facilita un lenguaje para definir ontologías estructuradas basadas en Web con esto procura proporcionar un lenguaje para describir clases, las propiedades de las clases y la relación entre las clases. OWL está construida sobre RDF, por lo que ofrece una base apropiada para desarrollar ontologías.

Actividad: Mantenimiento

Al llevar a cabo el mantenimiento de la ontología se tendrán en cuenta las modificaciones que se puedan haber originado durante el proceso, además de apreciar los cambios que se puedan producir en la estrategia de pruebas de la dirección de calidad. De producirse cambios estos podrán incluirse en la ontología.

En la creación de la ontología se han utilizado las anotaciones donde se describen los componentes y las restricciones. Además, la ontología cuenta con una documentación que permite su comprensión, necesaria para poder realizar cualquier cambio sobre ella.

Validación de la ontología

Ramos (Ramos 2009) expone que los criterios a evaluar en cada fase del ciclo de vida del desarrollo de una ontología son:

- Uso correcto del lenguaje utilizado para la codificación.
- Exactitud de la estructura taxonómica.
- Significado de los términos y conceptos representados.
- Adecuación a los requerimientos especificados al inicio del desarrollo.

Para cumplir con estos criterios es indispensable que se evalúen los resultados parciales de cada una de las fases del ciclo de vida de desarrollo garantizando de esta forma la calidad de la propuesta ontológica.

Por tal motivo propone cuatro fases acordes a los criterios antes descritos con el fin de evaluar ontologías:

Fase 1: Uso correcto del lenguaje

- Validar que el lenguaje cumpla con los estándares de desarrollo ontológicos.
- Evaluar sintácticamente la ontología en cada fase de su desarrollo.

Fase 2: Exactitud de la estructura taxonómica

- Chequeo de inconsistencias, completitud y redundancia de los términos de la taxonomía.

Fase 3: Validez del vocabulario

- Identificar, extraer y organizar los términos significativos del dominio a partir de los documentos.
- Evaluar el vocabulario considerando las medidas de calidad de resultados usando medidas tales como la precisión y el recall (exhaustividad).

Fase 4: Adecuación a los requerimientos

- Verificar que las especificaciones de requerimientos se cumplan.
- Verificar que las respuestas proporcionadas por la ontología a las preguntas de competencia sean correctas y pertinentes.

A continuación, se describe la forma en la que cada una de estas fases fue aplicada a la ontología.

Se utilizó el lenguaje OWL el cual cumple con los estándares para desarrollos ontológicos. Es un lenguaje completo teniendo en cuenta que cualquier expresión que esté lógicamente implícita en la base de conocimiento puede ser derivada, y sólido dado que cualquier expresión puede ser derivada a partir del conocimiento codificado. A raíz de esto se pueden aplicar métodos de razonamiento sobre la ontología de manera satisfactoria. Vale resaltar que en cada fase del ciclo de desarrollo se utilizó el marco de chequeo que provee el editor Protégé-OWL específicamente el razonador pellet, con el objetivo de corregir inconsistencias sintácticas y obtener un código libre de errores. Así se cumplimenta la fase 1.

La evaluación taxonómica considera que las actividades que se llevan a cabo para incrementar la calidad de la ontología del modelo multidimensional son: completitud de los conceptos, identificación de inconsistencias y existencia de redundancia en las clases y relaciones, por lo que se realizó una comprobación manual de cada concepto y de las relaciones entre estos. El resultado de aplicar la fase 2 confirmó que no existen inconsistencias tales como: conceptos que no pertenecen a una clase en particular, ausencia de conceptos relevantes del dominio y clases e instancias con diferentes nombres, pero definiciones similares.

En la fase 3 se chequea que los términos codificados en la ontología existan y sean significativos en otras fuentes de conocimiento independientes. Esto se realiza mediante el cálculo de dos medidas de razón: la precisión y la exhaustividad, medidas numéricas de cubrimiento para determinar la completitud y validez del vocabulario.

El cálculo de la precisión ofrece el porcentaje de los términos que aparecen en el corpus con relación a la cantidad total de términos de la ontología, utilizando la siguiente fórmula:

$$Precision = \frac{CO_C}{COnto} \dots\dots\dots (1)$$

Donde:

CO_C: es la cantidad de términos que se solapan entre la ontología y el corpus,

Conto: es la cantidad total de términos que tiene la ontología.

Al aplicar la ecuación 1 se demostró que el 100% de los términos existentes en la ontología se encuentran en el corpus del dominio.

Por su parte el cálculo de la exhaustividad proporciona el porcentaje de términos del corpus que aparecen en la ontología con relación al total de términos en el corpus, utilizando la siguiente expresión:

$$Exhaustividad = \frac{CO_C}{CCorp} \dots\dots\dots (2)$$

Donde:

CCorp: representa la cantidad total de términos del corpus.

El resultado es 1, semejante a la medida anterior.

La última fase se comprobó, a través del resultado obtenido al revisar el resultado de cada pregunta de competencia.

Además, la propuesta ontológica es validada también mediante el servicio de validación web de la W3C(W3C 2018). Al cargar la ontología y el código generado en RDF, el validador arrojó el resultado que se muestra en la Figura 2, demostrando que ambos funcionan correctamente y sin incongruencias.

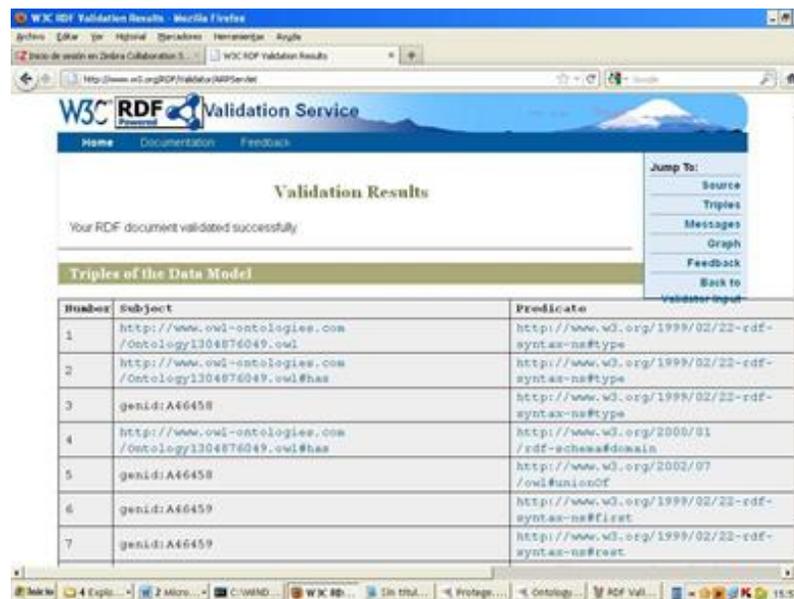


Figura 2. Validación mediante la W3C

Conclusiones

Los resultados obtenidos durante el desarrollo de la investigación permiten llegar a las siguientes conclusiones:

- El subproceso de pruebas de software juega un papel fundamental en las empresas que desarrollan software. Este garantiza que el producto final tenga la calidad requerida.
- Se demostró la potencialidad de las ontologías para apoyar el subproceso de pruebas de software en la Dirección de Calidad de la UCI. El análisis del estado del arte se determinó que las ontologías identificadas son generales e incompletas al no abordar todos los términos y sus relaciones.
- Las herramientas, metodología y tecnologías seleccionadas contribuyeron al desarrollo una propuesta ontológica que abarca el conocimiento que está presente en el dominio de las pruebas de software, permitiendo de esta forma la consistencia, organización y comunicación del conocimiento generado.
- La ontología mejora la consistencia, organización y comunicación del conocimiento generado, apoyando el subproceso de pruebas de software en la Dirección de Calidad de la UCI. Esto se demostró mediante la validación de la propuesta a través del esquema de validación realizado, que evalúa la elaboración correcta de la ontología, el uso correcto del lenguaje utilizado para la codificación, la no existencia de inconsistencias y la capacidad de inferencia. Además, se obtuvo resultados satisfactorios cuando se realizó la validación de la consistencia de la ontología mediante el validador web de la W3C.
- La ontología propuesta abarca todo el subproceso de pruebas de software a los diferentes niveles y conceptos que caracterizan el proceso de desarrollo en la UCI. Además, elimina contradicciones entre conceptos analizados en las ontologías existentes, lo que queda demostrado en la validación realizada.

Agradecimientos

Agradecemos a la Dirección de Calidad de la UCI, al Grupo de Investigación Ingeniería y Calidad de Software, a CALISOFT así como al Programa Especial de Formación Científica en Informática.

Referencias

- Duarte, K.C. & Almeida Falbo, R., 2000. Uma Ontologia de Qualidade de Software. *Repositorio Institucional de Universidade Federal do Espírito Santo. Vitória- Espírito Santo:*
- Echeverría Perez, D. & Fernández Perez, Y., 2011. *Máster en Calidad de Software: Desarrollo de una ontología de apoyo al procedimiento del Departamento de Pruebas de Software.* UCI.

- Echeverría Pérez, D., Fernández Pérez, Y. & Zulueta Pozo, D., 2012. Ontología de apoyo al procedimiento de pruebas del Departamento de Pruebas de Software de Calisoft. In *Tenth LACCEI Latin American and Caribbean Conference (LACCEI'2012)*.
- Felipe Alfonso, R., 2011. Implementación de una ontología para el proceso de ingeniería de requisitos del modelo cubano de calidad para el desarrollo de aplicaciones informáticas en la industria cubana de software. *TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS.*, UCI.
- Fernández Hernández, A., 2015. *TESIS DOCTORAL Modelo Ontológico de recuperación de información para la toma de decisiones en Gestión de Proyectos*, Departamento de Información Y Comunicación, Universidad de Granada.
- Fernández, Y., Cruz, C. & Verdegay, J.L., 2018. A New Model Based on Soft Computing for Evaluation and Selection of Software Products. *IEEE Latin America Transactions*, 16(4).
- Ferreira, M. et al., 2006. Medición del Software Ontología y Metamodelo. *Informe Técnico Universidad de Castilla-La Mancha*.
- IEEE Computer Society & Standards Committee, S.E., 2014. *IEEE Standard for Software Quality Assurance Processes 730-2014*,
- ISO/IEC/IEEE., 2017. ISO/IEC/IEEE 24765 Systems and Software Engineering- Vocabulary. Switzerland: ISO.
- ISO, 2008. *ISO / IEC 15939: 2007 Systems and software engineering — Measurement process*,
- Lapiente, M.J., 2013. Hipertexto, el nuevo concepto de documento en la cultura de la imagen.
- NC ISO/IEC 25000, N., 2011. *NC ISO/IEC 25000:2011 Ingeniería de software — Requisitos de calidad y evaluación de productos software (SQuaRE)*,
- Ramos, E., 2009. Esquema para evaluar ontologías únicas para un dominio de conocimiento. <https://doaj.org/article/2e52e18a85de4c94b7009f89c2d2c697>.
- Souza, E.F., 2013. Using Ontology Patterns for Building a Reference Software Testing Ontology. In *17th IEEE International Enterprise Distributed Object Computing Conference Workshops*. pp. 21–30.
- Vitelli, I.F., 2011. Aplicación de Methontology para la Construcción de una Ontología en el Domino de la Microbiología. *Caracas, Venezuela : s.n.*
- W3C, 2018. W3C RDF Validation Service. <http://www.w3.org/RDF/Validator>.