

Tipo de artículo: Artículo original
Temática: Software libre
Recibido: 01/08/2016 | Aceptado: 06/11/2016

Atributos de calidad para la evaluación y selección de aplicaciones de código abierto en CUBA

Quality attributes for open source applications selection and assessment in Cuba

María Leisy González Carrera¹, Yoandy Pérez Villazón¹, Cesar González Hernández², Marielis González Muñoz¹

¹Centro de Software Libre, Universidad de las Ciencias Informáticas, Cuba, {mlcarrera, yvillazon, mmuno}@uci.cu

²Dirección de Informatización, Universidad de las Ciencias Informáticas, Cuba, cgonzalezh@uci.cu

*Autor para correspondencia: mlcarrera@uci.cu

Resumen

Desde el año 2004 Cuba anunció su intención de migrar paulatina y ordenadamente hacia tecnologías de código abierto, para ello se creó como soporte tecnológico el sistema operativo Nova y la Guía Cubana para la migración a código abierto. Este último documento posee las aplicaciones informáticas recomendadas para ser usadas en el proceso de migración. La selección de las mismas está basada en las experiencias de los usuarios, sin embargo, no siempre son las más adecuadas para las instituciones. El presente trabajo propone un conjunto de atributos de calidad que deben tenerse en cuenta para la evaluación y selección de aplicaciones de código abierto en Cuba. Como parte del estudio del estado del arte se analizan tres modelos empleados sobre el tema y que poseen una amplia documentación. Los atributos propuestos son documentados. Finalmente se propone un caso de estudio para la selección de la alternativa libre a la herramienta Microsoft Office Word, teniendo como punto de partida un proyecto de migración real que tuvo lugar en el año 2014.

Palabras clave: atributos de calidad, código abierto, evaluación, selección

Abstract

Since 2004 Cuba stand its interest in an ordered and incremental migration to free and open source technologies, therefore was created as a technological basement the GNU/Linux distribution Nova and the "Cuban Guide for migration to open source". This document contains a list of the recommend applications to be used in the migration process. The selection of such applications is based on the user experience, but this isn't always the best choice for

institutions. In this paper is presented a set of quality attributes that must be considered in the assessment and selection of open source applications in Cuba. As part of the State of the Art study are analyzed three well documented models. The proposed attributes are documented. In conclusion is proposed a study case for the selection of a free alternative to Microsoft Office Word (R), which is based on the experiences acquired in a real migration project held in 2014.

Keywords: *quality attributes, open source, assessment, selection*

Introducción

Cuba, a partir del año 2004 comenzó de manera intencionada gubernamentalmente la migración ordenada y paulatina hacia aplicaciones de código abierto. Para guiar el proceso se creó una estructura nacional conformada por cuatro grupos de trabajo (ver Figura 1).

El núcleo principal del Grupo Técnico Nacional está encabezado por la Universidad de las Ciencias Informáticas (UCI), la cual tiene la misión de desarrollar el sistema operativo cubano (Nova GNU/Linux) y definir los lineamientos nacionales que guiarán el proceso de cambio tecnológico.



Figura 1: Grupos de trabajo para la migración.

Como parte de esta última tarea se desarrolló la Guía Cubana para la migración a código abierto en el año 2009, y actualmente se trabaja en la edición del libro “Buenas prácticas para la migración a código abierto”. En este último documento se ilustran los procedimientos a seguir para migrar una institución a software de fuentes abiertas. Una etapa importante del proceso es proponer las alternativas libres al software privativo, para ello debe elegirse adecuadamente la solución que mejor se ajuste al entorno. Existen en la actualidad varios modelos descritos en la bibliografía para la evaluación y selección del software de código abierto, sin embargo, nuestro país tiene condiciones tecnológicas propias, por lo que los atributos de calidad definidos en estos modelos deben ser revisados y ajustados/ cambiados para una mejor aplicación en la selección del software a usar en las nuestras instituciones. La presente investigación tiene como objetivo principal: definir atributos de calidad para la evaluación y selección de aplicaciones de código abierto en Cuba.

Materiales y métodos

Instituciones internacionales han formulado modelos para la evaluación y selección de software de fuentes abiertas, la mayoría de ellas se centran en la madurez, durabilidad y la estrategia de organización del desarrollo, otras están focalizadas en los aspectos funcionales.

Los modelos para evaluar aplicaciones de código abierto se pueden dividir en dos generaciones, la diferencia de los descritos como de “segunda generación” es que incluyen herramientas de soporte para automatizar al menos parcialmente el proceso de evaluación (R. Glott, 2010) (M. Ciolkowski, 2009). En el ámbito de este artículo serán analizados los pertenecientes a la segunda generación.

A. QSOS

El modelo de calificación y selección de software (QSOS, por sus siglas en inglés) es una propuesta desarrollada de forma colaborativa por varios autores que se compone: del modelo en sí, herramientas de apoyo y un conjunto de plantillas para el trabajo de evaluación (QSOS, 2013). Este propone cuatro etapas:

1) Definición y organización de los criterios que serán evaluados:

1.1) Definición de los criterios de acuerdo a las necesidades objetivas.

1.2) Organización los criterios.

1.3) Clasificación de los criterios en funcionales y no funcionales.

2) Evaluación, aplicando los criterios definidos previamente y asignando valores a cada indicador establecido:

2.1) Definición de las aplicaciones a evaluar y sus datos respectivos.

2.2) Otorgamiento de puntuación por criterios en cada aplicación en particular.

2.3) Evaluación de los riesgos propios asociados al software (teniendo en cuenta la naturaleza del código abierto).

2.4) Determinación de la puntuación general obtenida por cada software teniendo en cuenta los elementos funcionales.

3) Calificación de la evaluación mediante la organización de los criterios, de acuerdo al grado de importancia de cada uno, teniendo en cuenta las necesidades particulares:

3.1) Definición del valor total de la puntuación por aplicaciones.

3.2) Definición de algún nuevo criterio si fuera necesario.

3.3) Determinación de los criterios de obligatoriedad (más importantes en dependencia de los objetivos).

4) Selección del software más adecuado de acuerdo a la puntuación obtenida:

4.1) Organización de los resultados.

4.2) Comparación de varias soluciones usando los criterios establecidos usando para ello una matriz.

4.3) Selección de la solución de mayor puntuación.

QSOS es un modelo flexible que posibilita la incorporación de criterios, permite además iterar múltiples veces para refinar el resultado de la selección. Este modelo propone un conjunto de plantillas de apoyo a la evaluación, las cuales se componen de matrices que permiten establecer comparaciones entre las soluciones evaluadas. Se brindan de apoyo además tres herramientas informáticas de escritorio que automatizan parcialmente el proceso de pruebas (A. Rodríguez, 2013).

- QSOS XUL Editor.
- QSOS Qt Editor
- QSOS Java Editor

Desarrolladas para ser ejecutadas como una extensión del navegador Firefox, usando las librerías Qt y en el lenguaje Java respectivamente. La primera y última solución son multiplataformas (QSOS, 2013).

B. OMM QualiPSo

OMM contiene los elementos que han de ser evaluados, pero también tiene un conjunto de normas y directrices que describen cómo llevar a cabo los procesos de evaluación, por lo que se le denomina indistintamente como el modelo y la metodología (GROUP, 2015). Actualmente su nivel de aplicación sigue siendo bajo. Para el diseño del modelo se tuvieron en cuenta:

- Los criterios de desarrolladores y usuarios.
- El modelo CMMI (M. Wittmann, 2010).

Está organizado en niveles y cada nivel se basa en el nivel inferior (ver Figura 2).

1) *Nivel básico*, se evalúan los criterios:

- STD: Utilización de normas establecidas y generalizadas.
- QTP: Calidad del plan de pruebas.
- PDOC: Documentación del producto.
- ENV: Medio ambiente técnico.
- LCS: Licencias.
- MST: Mantenibilidad y estabilidad.

- DFCT: Número de confirmaciones e informes de errores.
- RDMP1: Hoja de ruta del producto (parte 1).
- CM: Gestión de la configuración.
- REQM: Gestión de requisitos.
- PP1: Proyecto de planificación (parte 1).

2) *Nivel medio*, se evalúan los criterios:

- STK: Relación entre las partes interesadas.
- RDMP2: Hoja de ruta del producto (parte 2).
- PP2: Planificación de proyecto (parte 2).
- PMC: Monitoreo y control.
- TST1: Pruebas (parte 1).
- DSN1: Diseño (parte 1).
- PPQA: Aseguramiento de la calidad.

3) *Nivel avanzado*, se evalúan los criterios:

- PI: Integración del producto.
- TST2: Pruebas (parte 2).
- DSN2: Diseño (parte 2)
- RSKM: Gestión de riesgos.
- RASM: Resultados del análisis de un tercero
- REP: Reputación
- CONT: Contribución

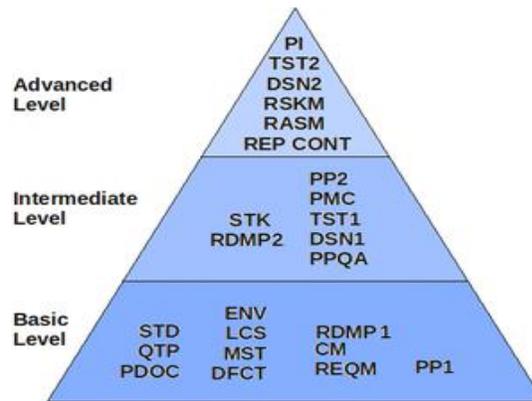


Figura 2: Niveles propuestos por OMM QualiPSo

C. *QUALLOSS*

QUALLOSS es un proyecto coordinado por Jean Christophe del CETIC (Bélgica). El proyecto evalúa las soluciones informáticas desde un punto de vista enfocado a la aplicabilidad para los negocios (Deprez, 2009). Propone 2 y 3 niveles para la evaluación del software.

1) *Productos de trabajo:*

- 1.1) Producto: Mantenibilidad, Confiabilidad, Seguridad.
- 1.2) Documentación: Disponibilidad.
- 1.3) Pruebas: Disponibilidad y alcance, Repetitividad.

2) *Comunidad:*

- 2.1) Tamaño.
- 2.2) Grado de interactividad.
- 2.3) Composición.

3) *Proceso de desarrollo:*

- 3.1) Capacidad de los requisitos.
- 3.2) Capacidad del soporte.
- 3.3) Capacidad de la gestión de cambios.

D. *Comparación entre los modelos*

Independientemente a las cuestiones técnicas y atributos definidos en los modelos y documentados con anterioridad es importante determinar cómo se puede apreciar en la Tabla 1, la flexibilidad (R. Glott, 2010), licencia de uso (M. Ciolkowski, 2009), grado de actividad/actualización (QSOS, 2013), tipo de evaluación que realiza sobre el software

(teórica/práctica) (A. Rodríguez, 2013), si emplea esquema de puntuación o no (GROUP, 2015), si emplea un modelo iterativo o de una sola vuelta (M. Wittmann, 2010) y el responsable del modelo (persona o institución) (Deprez, 2009).

Luego de revisar la bibliografía y de realizar un análisis de los modelos anteriores se considera:

- QSOS ofrece un modelo sencillo y flexible, pero la amplia flexibilidad que propone es a la vez una desventaja pues carece de atributos directos que orienten a los especialistas con poca experiencia en el cómo aplicarlo.
- El modelo OMM QualiPSo propone un excelente marco de trabajo para la evaluación de FOSS, pero es complejo de aplicar debido a que evalúa muchos elementos asociados al proceso de desarrollo de la herramienta. Además, está más centrado en el cómo se desarrolló el producto que en la aplicabilidad y funcionalidad final de la solución.
- QUALOSS es un modelo robusto, con un amplio fondo de financiamiento (799.465,00 €) para su desarrollo y está soportado por instituciones como Zea Partners, AdaCore, MERIT, la Universidad del Rey Juan Carlos, entre otras empresas e instituciones (CETIC, 2013). La principal dificultad para aplicarlo correctamente radica en que se necesita mucha información que en ocasiones no está disponible en las aplicaciones que se someten a la evaluación. A consideración de los autores de la presente investigación este modelo se basa más en la práctica que en elementos teóricos. Para evaluar cada uno de los elementos anteriores se definen métricas a tener en cuenta.

Tabla 1: C(2013)omparativa entre los modelos

criterio	QSOS	OMM QualiPSo	QUALOSS
1	Sí	Parcial	Sí
2	FDL	CC	GPL
3	2013	2013	2014
4	Práctica	Teórica	Práctica
5	Sí	No	No
6	Iterativo	Iterativo	Iterativo
7	Comunitario	Institucional	Institucional

Resultados y discusión

Atributos de calidad para la evaluación y selección de aplicaciones de código abierto en Cuba

La evaluación y selección de las alternativas de código abierto a usar en Cuba tiene necesariamente que estar en concordancia con las condiciones tecnológicas y sociales que caracterizan a las instituciones donde serán desplegadas.

Estudios realizados por el Centro de Software Libre de la Universidad de las Ciencias Informáticas permiten caracterizar tecnológica y socialmente a las instituciones. El estudio no es una conclusión generalizada por el tamaño de la muestra estudiada, pero permite obtener una medida que posibilita caracterizar el sector público.

Condiciones sociales:

- El promedio de edad de los usuarios es de aproximadamente 42 años.
- El 87% de los usuarios llevan 5 años o más usando Microsoft Windows.
- Los usuarios poseen bajo conocimiento sobre el software de fuentes abiertas y sus ventajas potenciales.

Condiciones tecnológicas:

- El 47% de las computadoras poseen un perfil bajo¹ de configuración en el hardware.
- La mayoría de las instituciones poseen sistema operativo Microsoft Windows en diferentes versiones, existen computadoras con la versión de Windows 98 y otras con el reciente Windows 10.
- La mayoría de los usuarios no conocen las aplicaciones con las que trabajan, solo las funciones básicas que necesitan conocer para el desarrollo de su trabajo.

A partir de lo anterior y por lo general se cumple la regla del 80-20, donde solo se emplean el 20% de las funcionalidades de las aplicaciones actualmente instaladas.

Teniendo en cuenta las condiciones anteriores la “Guía Cubana para la migración a código abierto” define un conjunto de interrogantes que ayudan a los especialistas de migración a elegir la herramienta alternativa a una solución privativa. Para la selección de la aplicación equivalente deben formularse, según el documento anteriormente mencionado las siguientes preguntas:

- ¿Cuál es el proceso que automatizará la aplicación?
- ¿Cuál es el nivel de preparación y conocimientos que tienen los usuarios sobre cómo aplicar ese proceso?
- ¿Cuál es el nivel de dominio que posee el usuario de la herramienta privativa que automatiza ese proceso?
- ¿Está ese proceso automatizado o no?
- ¿Qué funcionalidades debe tener la futura aplicación a usar?
- ¿Cuáles son las características técnicas que dispone el hardware en el que se va a ejecutar la nueva aplicación?

Se recomienda además en la Guía:

¹ Perfil bajo: En el proceso de migración a código abierto se define de perfil bajo a una computadora cuya memoria RAM no supera los 512MB.

- No usar varias herramientas que correspondan a una misma categoría (por ejemplo, LibreOffice Writer y Abiword), pues esto dificulta las actividades de soporte, entrenamiento y probablemente la interoperabilidad.
- Seleccionar de las herramientas evaluadas, la más simple de comprender para el usuario y que satisfaga las respuestas a las preguntas anteriores.
- Siempre pensar antes de proponer la solución, en el usuario y el desarrollo del proceso, que es pensar en la futura productividad.

Los elementos anteriores son claves para garantizar una adopción más fácil y en un menor tiempo de las tecnologías por parte de los usuarios. Estos tienen como inconveniente que formulan elementos generales a tener en cuenta para la evaluación y selección de las aplicaciones, pero no proponen los atributos con precisión que deben ser evaluados, ni cómo deben ser valorados los mismos.

A. *Atributos*

Para la selección de las aplicaciones de código abierto se propone tener en cuenta los siguientes atributos.

- Licencia de uso/distribución: permite conocer si el producto es realmente de software libre o no, para verificar se recomienda emplear el sitio de la Iniciativa para el Código Abierto², donde se enuncian aquellas licencias que son compatibles con el software libre.
- Propietario: permite conocer si el producto es desarrollado por una persona o una institución. Los productos desarrollados por instituciones son más recomendados que los desarrollados por individuos independientes.
- Procedencia: no se recomienda productos provenientes de empresas/individuos norteamericanos. En caso de ser seleccionados deben ser sometidos a un riguroso proceso de revisión del código fuente.
- Antigüedad: los productos más antiguos poseen mayor madurez, son menos propensos a vulnerabilidades.
- Cantidad de errores: el estudio de los errores (y su tendencia) permite conocer aquellos softwares que han incrementado su confiabilidad en el tiempo.
- Soporte al idioma español: al ser el español la lengua materna en Cuba, es prioritario que la solución a seleccionar esté disponible en este idioma.
- Impacto, grado de uso: las aplicaciones más usadas, posen por lo general una mayor aceptación de los usuarios (búsqueda en Google Trends para ayudar a determinar el valor de esta variable).
- Comunidad / Rapidez en el desarrollo: la solución a seleccionar debe poseer un desarrollo activo, si alrededor de una aplicación existe una comunidad debe ser un elemento de peso a tener en cuenta. Las comunidades

² <https://opensource.org/licenses>

activas garantizarán la mantenibilidad en el tiempo del producto y mejores tiempos de respuesta ante posibles problemas.

- Rendimiento: por las condiciones tecnológicas de Cuba se recomiendan aquellas aplicaciones que realicen un mejor aprovechamiento del hardware. Esta condición será prevaleciente mientras no afecte la productividad final.
- Documentación: el grado de disponibilidad de la documentación debe ser tenido en cuenta, aquellas aplicaciones poco documentadas por lo general son complejas y poco usadas. La existencia de foros de discusión, listas de correo, manuales de uso y de desarrollo son elementos a tener en cuenta.
- Disponibilidad de las pruebas: las aplicaciones libres cuyas pruebas aplicadas estén disponibles y puedan ser ejecutadas.
- Soporte: existen varios niveles de soporte, los más populares para productos de fuentes abiertas son el soporte comunitario y el profesional. Se debe tener en cuenta que exista un adecuado soporte comunitario. Este atributo está vinculado a la existencia de una comunidad.

Los atributos o cualidades anteriormente mencionados son generales y deben ser aplicados independientemente de la aplicación a la que se le esté buscando una alternativa.

Se propone además de las características genéricas especificadas anteriormente tener en cuenta atributos que son particulares de la solución en cuestión, y que están vinculados directamente a las funcionalidades de la aplicación privativa.

Este último grupo de características serán evaluadas a través de una escala numérica con valores que oscilan de 1 a 5 puntos, siendo las funcionalidades de 5 puntos las de mejor resultado. Se proponen como elementos a tener en cuenta para evaluar cada funcionalidad algunos de los elementos descritos en la norma ISO/IEC 25000-2014 (ISO/IEC, 2014).

- Adecuación: definido como la existencia de la funcionalidad de la aplicación privativa en la solución libre.
- Exactitud: mide que el resultado brindado por la funcionalidad de la aplicación libre sea el mismo que el ofrecido por la privativa.
- Productividad: la funcionalidad en cuestión es fácil de ejecutar y contribuye a la productividad en el trabajo.
- Satisfacción: el usuario final está satisfecho con el resultado.

Las aplicaciones seleccionadas como alternativas libres más adecuadas serán aquellas que tengan un mayor grado de recubrimiento funcional (*Rf*), como se puede apreciar en la ecuación (R. Glott, 2010), entendiéndose el mismo como el resultado de la suma de las puntuaciones asignadas a cada funcionalidad evaluada.

$$Rf = \sum_{i=0}^n Fi \quad Fi \text{ representa a cada una de las funcionalidades evaluadas en la aplicación.}$$

CASO DE ESTUDIO: evaluación y selección de la alternativa de código a abierto a la herramienta Microsoft Office Word

Para ilustrar la pertinencia de los atributos definidos se propone en esta investigación realizar el proceso de selección de la alternativa libre más adecuada para la herramienta Microsoft Office Word. En este caso se debe buscar el equivalente libre para la Empresa Constructoras de Obras de Arquitectura e Industriales No. 3 (ECOAIND3). En esta institución los usuarios emplean Microsoft Office Word para realizar las actividades siguientes:

1) Realizar informes sobre el cumplimiento de los planes constructivos:

- Trabajo con estilos de páginas y carácter.
- Manejo de tablas.
- Trabajo con herramientas de formato.
- Generación de gráficos de barras y pastel.
- Copiar desde el Microsoft Excel hacia el Word tablas de forma directa.

2) Realizar cartas en formato electrónico para un grupo determinado de usuarios:

- Trabajo con estilos de páginas y carácter.
- Trabajo con las herramientas de formato.
- Empleo de la utilidad de combinación de correspondencia.

3) Imprimir documentos:

- Empleo de la utilidad de vista previa para ajustar las características de impresión.
- Imprimir directamente.

Además de las actividades anteriores, se pudo verificar que los documentos generados por algunos trabajadores eran enviados a sus jefes superiores para revisión, y se constató además la existencia de macros empleadas en algunos documentos de uso interno del ECOAIND3.

Lo anterior conlleva a que la solución a seleccionar debe tener capacidades para:

- Gestión de cambios al contenido.
- Gestión de notas en el documento.
- Soporte para algún lenguaje de macros (Preferiblemente Visual Basic³).

³ Visual Basic es el lenguaje de macros soportado por la suite Microsoft Office.

A partir de las funcionalidades empleadas en ECOAIND3 en el manejo de Microsoft Office Word se procede a seleccionar (ver Tabla 2) la alternativa de código abierto más adecuada a esta solución teniendo en cuenta los atributos definidos en la sección anterior.

Tabla 2: Selección de la alternativa a Microsoft Office Word aplicando los atributos definidos.

Atributo	LO Writer	Abiword	WPSOffice
Genéricos			
Licencia	GPL	GPL	Gratis ⁴
Propietario	The document foundation	Gnome community	Kingsoft Corp.
Procedencia	Alemania	ND	R.P. China
Antigüedad	30 años ⁵	17 años	27 años
Errores	224	48	ND
En español	Sí	Sí	Sí ⁶
Impacto	Alto	Medio	Alto
Comunidad	Amplia y activa	No muy activa	Ninguna
Rendimiento	256Mb	64Mb	256Mb
Documentación	Sí	Sí	Sí
Pruebas	Sí	Sí	No
Soporte	Ambos	Comunity	Profesional
Funcionales			
Estilos de página y de carácter	5	3	3
Tablas	4	4	4
Formato	5	5	5
Generación de gráficas	4	3	0
Interoperabilidad de los componentes	5	0	5
Combinación de correspondencia	5	0	0

⁴ La licencia de WPS Office no está clara, en su web oficial se ofrece gratuitamente, pero sin código fuente disponible.

⁵ La antigüedad es teniendo en cuenta su predecesor StarOffice.

⁶ Requiere configuraciones adicionales.

Configuración de la impresión	5	5	5
Impresión directa	5	5	5
Gestión de cambios	5	0	0
Gestión de notas	5	5	5
Soporte VBA	3	0	0
Recubrimiento funcional (Rf)	51	30	32

Desde el punto de vista funcional, la herramienta LibreOffice Writer es superior a las soluciones disponibles para GNU/Linux evaluando las funcionalidades que se necesitan en ECOAIND3. En cuanto a las características genéricas es el producto más antiguo de los existentes, completamente libre, desarrollado por una comunidad y posee una amplia documentación en idioma español, al igual que su interfaz de trabajo. El atributo cuestionable en LibreOffice Writer que pudiera resultar de peso en su selección es que requiere de al menos 256Mb de RAM para su correcta ejecución, pero para el caso de estudio en cuestión resulta aceptable por las características tecnológicas que posee.

Conclusiones

Al terminar la presente investigación se arriba a las siguientes conclusiones:

- Los modelos para la evaluación y selección de software de código abierto estudiados no son estrictamente aplicables para la selección de software de código abierto alternativo en Cuba.
- Se propusieron 12 atributos genéricos a tener en cuenta para la evaluación y selección del software.
- Para la evaluación y selección del software existen atributos particulares en dependencia de la solución privativa de que se está evaluando y de los procesos que la misma automatiza en la institución objeto del proceso de migración.
- Las condiciones tecnológicas y de los recursos humanos en las instituciones inciden directamente en la propuesta de software alternativo.
- No siempre la aplicación con mayor recubrimiento funcional es la elegible, los atributos genéricos tienen una importante influencia en la selección final de la alternativa.

Agradecimientos

Agradecimiento especial a los especialistas del Centro de Software Libre de la Universidad de las Ciencias Informáticas por los años de trabajo en equipo y la experiencia acumulada en la ejecución de procesos de migración a código abierto.

Referencias

- A. Rodríguez, Valeria Soria. 2013. “No todo es codificar – Modelos de Madures en SL”. s.l. : VIII Jornadas de Software Libre, 2013.
- CETIC. 2013. [En línea] 2013. <https://www.cetic.be/QualOSS,500>.
- Deprez, E-C. 2009. “QUALity of Open Source Software”. s.l. : QUALOSS, 2009.
- GROUP, QUALIPSO. 2015. OMM. [En línea] 2015. <http://qualipso.icmc.usp.br/OMM/>.
- ISO/IEC. 2014. ISO. [En línea] 2014. http://www.iso.org/iso/catalogue_detail?csnumber=35683.
- M. Ciolkowski, M. Soto. 2009. “The QualOSS Process Evaluation: Initial Experiences with Assessing Open Source Processes”. 2009. págs. 105–116. Vol. 42.
- M. Wittmann, Nambakam, Ranganatahm. 2010. [En línea] 2010. <http://www.qualipso.org/sites/default/files/A6.D1.6.3CMM-LIKEMODELFOROSS.pdf>.
- QSOS, TEAM. 2013. “Collaborative technological watch”. [En línea] 2013. http://backend.qsos.org/download/qsos-2.0_en.pdf.
- R. Glott, A.-K. Groven, K. Haaland, A. Tannenber. 2010. Quality Models for Free/Libre Open Source Software Towards the “Silver Bull”. s.l. : IEEE Xplore Digital Library , 2010. págs. 439–446.