

VIII Jornadas de Software Libre

**No todo es codificar – Modelos de
Madurez en SL**

Integrantes
A.C. Anahí Rodríguez
A.C. Valeria Soria

Modelos de Madurez

CMMI evalúa el modelo de madurez del proceso de desarrollo del software. El nivel que posee es demasiado complejo y no es aplicable a los procesos de desarrollo de software libre.

En los últimos años se han desarrollado varios métodos para definir un proceso de evaluación de software Free/Open Source.

Selección de un Software

A la hora de seleccionar un software Free/Open Source, es necesario contar con un método de clasificación, examinando las limitaciones y los riesgos que tiene este tipo de software.

Algunas preguntas generales para hacer antes de seleccionar un software son las siguientes:

- ¿Cual es la durabilidad del software?
- ¿Que nivel de estabilidad tiene o se espera?
- ¿Es fácil y factible agregarle funcionalidad al software?
- ¿Tiene una comunidad activa que lo avale?

Estas preguntas llevaron a definir los modelos de madurez de proyectos de software Free/Open Source, los cuales en su mayoría cuentan con:

Una puntuación según las características del software.

Una elección final, según los criterios de puntuación establecidos

OSMM – Capgemini

Primer modelo de madurez para software Open Source.

Posee una licencia privada pero con autorización para su distribución.

Fue desarrollado para comparar y decidir la utilización de un producto.

OSMM – Capgemini (cont.)

Existen 27 parámetros (12 características genéricas del producto y 15 características basadas en las necesidades del usuario.)

La puntuación va de 1 a 5.

VIII Jornadas de Software Libre

Open Source Maturity Model product evaluation form					
***** PRODUCT INDICATORS *****					
					Form Version 1.3
Product Information					Eval. date:
Product name					
Homepage product					
Version				Release date:	
Product group*	Infra	Security	OS	Office	Database
	Development	Portal	Groupware	Web	Maintenance
	Other:				
	Motivation:				
Evaluator Information	Name		Company		
First Product Evaluator (FPE)					
Second Product Evaluator (SPE)					
Product indicators	Score (FPE)	Score (SPE)	Motivation		
Age					
Licensing					
Human hierarchies					
Selling points					
Developer community					
Modularity					
Collaboration with other products					

OSMM – Capgemini (cont.)

Ayuda a determinar la madurez del producto.

Ayuda a determinar si el producto cumple con los requerimientos de la empresa.

Permite comparar los productos con alternativas comerciales.

Muestra la importancia de utilizar productos Open Source.

OSMM – Navica

Su método es más compacto que el propuesto por Capgemini.

Academic Free License

Consta de 4 fases:

Primera Fase: Seleccionar el software que se va a evaluar.

OSMM – Navica (cont.)

Segunda Fase: Ponderación de las siguientes categorías:

Software (4 ptos)

Soporte (2 ptos)

Documentación (1 pto)

Capacitación (1 pto)

Integración (1 pto)

Servicios de profesionales (1 pto)

OSMM – Navica (cont.)

Tercer Fase: aplicación de cada una de las plantillas correspondientes a las categorías nombradas en la fase anterior en las cuales se sugiere qué aspectos dentro de la categoría deben ser evaluados ponderando cada uno de estos aspectos.

Cuarta Fase: consiste en multiplicar la puntuación de cada categoría por su ponderación para producir una puntuación final de 0 a 100

VIII Jornadas de Software Libre

Open Source Maturity Model

Documentation Maturity Assessment Template

OSMM Potential Point Assignments for Documentation

Type	Score
Developer-created	2
Web postings	3
Commercial published documentation	5
Documentation possible maturity score	10

OSMM Documentation Assessment Checklist

Open Source Maturity Model	
Documentation Assessment Checklist	
Product Name:	

Documentation Assessment	
Method	Notes
Reviewed developer-created documentation	
Reviewed web postings	
Reviewed commercially published documentation	

BRR (Business Readiness Rating)

El proyecto fue anunciado en el 2005, este método permite evaluar Software Free/Open Source.

Incluye cuatro pasos:

- Creación una lista con el software a evaluar.
- Clasificación y ponderación de los criterios de selección.
- Recopilación de datos para cada criterio.
- Cálculo y publicación de los resultados.

BRR (cont.)

Tiene 12 criterios que pueden ser utilizados para evaluar el software.

Por lo general sólo se utilizan 6 o 7 para la evaluación.

Funcionalidad: el software satisface las necesidades del usuario?

Usabilidad: el software es intuitivo / fácil de instalar / fácil de configurar / fácil de mantener?

Calidad: el software está bien diseñado, implementado y probado?

BRR (cont.)

Seguridad: qué tan seguro es el software?

Rendimiento: cómo se comporta el software ante ciertas circunstancias?

Escalabilidad: puede hacer frente al grandes volúmenes de datos?

Arquitectura: el software es modular, portátil, flexible, extensible y abierto? Puede ser integrado con otros componentes?

Soporte: la comunidad cuenta con apoyo profesional?

BRR (cont.)

Documentación: existe documentación de buena calidad?

Adopción: el software ha sido adoptado por la comunidad, el mercado, y la industria?

Comunidad: la comunidad del software es activa?

Profesionalismo: que nivel de profesionalismo presentan el proceso de desarrollo y la organización del proyecto?

BRR (cont.)

No ha tenido el nivel de adopción (aceptación) que se esperaba. Desde julio de 2010 que se dio de baja el proyecto.

El proyecto aún no ha creado una comunidad próspera

Qualification and Selection of Open Source Software (QSOS)

Es un método, diseñado para calificar, seleccionar y comparar Software Free/Open Source de una manera objetiva.

GNU Free Documentation License.

QSOS (cont.)

El proceso consiste en 4 pasos. Cada una de estas etapas son independientes entre si.

Paso 1 – Definición: definir elementos tales como:

Clasificación del software y descripción funcional.

Clasificación de Licencias

Clasificación de la comunidad

Paso 2 – Evaluación: consiste en coleccionar información general del software, servicio ofrecidos y aspectos funcionales y técnicos

QSOS (cont.)

Paso 3 – Calificación: Ponderación de los criterios. Esta etapa sirve para eliminar el software que no satisface las necesidades del usuario en un contexto específico.

Paso 4 – Selección: identificar el software que cumple con los requisitos del usuario de dos modos posibles:

Una selección estricta

Una selección más amplia

VIII Jornadas de Software Libre

Intrinsic durability		Score	
		0	1
Maturity	Age	For instance less than 3 months	For instance between 3 months and 3 years
	Stability	Unstable software with numerous releases or patches generating side effects	Stabilized production release existing but old. Difficulties to stabilize development releases
	History, known problems	Software knows several problems which can be prohibitive	No known major problem or crisis
	Fork probability, source of Forking	Software is very likely to be forked in the future	Software comes from a fork but has very few chances of being forked in the future

Intrinsic durability		Score	
		0	1
Adoption	Popularity (related to: general public, niche, ...)	Very few users identified	Detectable use on Internet (sourceforge, freshmeat, google, ...)
	References	None	Few references, non critical usages

NASA's Reuse Readiness Levels (RRL)

La NASA Earth Science Data Systems (ESDS) Software Reuse Working Group (WG) reconoció la necesidad de medir la madurez del software para su reutilización y propone una serie de niveles de preparación para la reutilización (RRLs). La madurez de una tecnología en particular se puede medir de varias maneras, un método conocido es el Technology Readiness Levels (TRLs)

Se identificaron 9 áreas para medir la madurez de reutilización del software:

- Documentación
- Extensibilidad
- Cuestiones de Propiedad Intelectual
- Modularidad
- Empaquetado
- Portabilidad
- Cumplimiento de estándares
- Soporte
- Verificación y Pruebas

RRL (cont.)

Basado en TRL (Technology Readiness Levels), se tienen 9 niveles RRL (1 menos maduro – 9 más maduro):

- RRL 1 – Reutilización limitada.
- RRL 2 – Reutilización inicial.
- RRL 3 – Reutilización básica.
- RRL 4 – La reutilización es posible.
- RRL 5 – La reutilización es práctica.
- RRL 6 – Software es reutilizable.
- RRL 7 – El software es altamente reutilizable.
- RRL 8 – Se puede demostrar una reutilización local.
- RRL 9 – Se puede demostrar una reutilización extensa.

RRL (cont.)

Se creó una tabla (producto cartesiano) donde se tienen las 9 áreas junto a los 9 niveles con una breve descripción (Reuse Readiness Level (RRL) Topic Area Levels Summary).

Table 1 – Reuse Readiness Level (RRL) Topic Area Levels Summary

	Documentation	Extensibility	Intellectual Property Issues	Modularity	Packaging	Portability	Standards Compliance	Support	Verification and Testing
Level 1	Little or no internal or external documentation available	No ability to extend or modify program behavior	Product developers have been identified, but no rights have been determined.	Not designed with modularity	Software or executable available only, no packaging	The software is not portable	No standards compliance	No support available	No testing performed
Level 2	Partially to fully commented source code available	Very difficult to extend the software system, even for application contexts similar to the original application domain	Developers are discussing rights that comply with their organizational policies.			Some parts of the software may be portable	No standards compliance beyond best practices	Minimal support available	Software application formulated and unit testing performed
Level 3	Basic external documentation for sophisticated users available	Extending the software is difficult, even for application contexts similar to the original application domain	Rights agreements have been proposed to developers.	Modularity at major system or subsystem level only	Detailed installation instructions available	The software is only portable with significant costs	Some compliance with local standards and best practices	Some support available	Testing includes testing for error conditions and proof of handling of unknown input
Level 4	Reference manual available	Some extensibility is possible through configuration changes and/or moderate software modification	Developers have negotiated on rights agreements.			The software may be portable at a reasonable cost	Standards compliance, but incomplete and untested	Moderate systematic support is available	Software application demonstrated in a laboratory context
Level 5	User manual available	Consideration for future extensibility designed into the system for a moderate range of application contexts; extensibility approach defined and at least partially documented	Agreement on ownership, limited reuse rights, and recommended citation.	Partial segregation of generic and specific functionality	Software is easily configurable for different contexts	The software is moderately portable	Standards compliance with some testing	Support provided by an informal user community	Software application tested and validated in a laboratory context
Level 6	Tutorials available	Designed to allow extensibility across a moderate to broad range of application contexts, provides many points of extensibility, and a thorough and detailed extensibility plan exists	Developer list, recommended citation, and rights statements have been drafted.			The software is portable	Verified standards compliance with proprietary standards	Formal support available	Software application demonstrated in a relevant context
Level 7	Interface guide available	Demonstrated to be extensible by an external development team in a similar context	Developer list and limited rights statement included in product prototype.	Clear delineations of specific and reusable components	OS detect and auto-build for supported platforms	The software is highly portable	Verified standards compliance with open standards	Organized/defined support by developer available	Software application tested and validated in a relevant context
Level 8	Extension guide and/or design/developers guide available	Demonstrated extensibility on an external program, clear approach for modifying and extending features across a broad range of application domains	Recommended citation and intellectual property rights statement included in product.				Verified standards compliance with recognized standards	Support available by the organization that developed the asset	Software application "qualified" through test and demonstration (meets requirements) and successfully delivered
Level 9	Documentation on design, customization, testing, use, and reuse is available	Demonstrated extensibility in multiple scenarios, provides specific documentation and features to build extensions which are used across a range of domains by multiple user groups	Statements describing unrestricted rights, recommended citation, and developers embedded into product.	All functions and data encapsulated into objects or accessible through web service interfaces	Installation user interface provided	The software is completely portable	Independently verified standards compliance with recognized standards	Large user community with well-defined support available	Actual software application tested and validated through successful use of application output

Nuestro estudio

Se toma como base de conocimiento dichos métodos adoptando algunos criterios de evaluación propuestos por los mismos.

Los puntos más importantes a ser evaluados son:

Por cada herramienta se tiene la siguiente información:

Nombre:

Descripción:

URL:

Luego las siguientes tablas

Datos generales

Herramienta	Versión	Inicio del proyecto	Licencia	Plataforma	Interfaz	Lenguaje

Criterios de documentación

Herramienta	Guía de instalación	Manual de usuario	Preguntas Frecuentes	Soporte Online			Código comentado	Adicional
				Foro	Lista de mail	Blog		

Criterios de madurez (*)

Herramienta	Inicio del proyecto	Grado de actualización	Actividad en lanzamientos	Actividad en el reporte de errores

(*) Se completa con las siguientes siglas:

M = Malo

B = Bueno

R = Regular

MB = Muy bueno

Links

OSMM – Capgemini

<http://www.osspartner.com/portail/sections/accueil-public/evaluation-osmm>

OSMM – Navica

<http://www.oss-watch.ac.uk/resources/osmm.xml>

BRR

<http://www.oss-watch.ac.uk/resources/brr.xml>

QSOS

<http://www.qsos.org/>

RLL

<http://www.esdswg.org/softwarereuse/Resources/rlls/>

VIII Jornadas de **Software Libre**

Muchas Gracias!!!

