

December 2006

Modelo para estimar la calidad en los Sistemas de Información Geográfica

Mirla Tahhán

CONATEL. Caracas - Venezuela

Raymond Sánchez

CONATEL. Caracas - Venezuela

Maryoly Ortega

Universidad Simón Bolívar. Caracas - Venezuela

Maeía Pérez

Universidad Simón Bolívar. Caracas - Venezuela

Luis Mendoza

Universidad Simón Bolívar. Caracas - Venezuela

Follow this and additional works at: <http://aisel.aisnet.org/amcis2006>

Recommended Citation

Tahhán, Mirla; Sánchez, Raymond; Ortega, Maryoly; Pérez, Maeía; and Mendoza, Luis, "Modelo para estimar la calidad en los Sistemas de Información Geográfica" (2006). *AMCIS 2006 Proceedings*. 512.

<http://aisel.aisnet.org/amcis2006/512>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2006 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Modelo para estimar la calidad en los Sistemas de Información Geográfica

Mirla Tahhánm

CONATEL. Caracas – Venezuela
mirlat9@gmail.com

Raymond Sánchez

CONATEL. Caracas – Venezuela
ray.harraka@gmail.com

Maryoly Ortega

Departamento de Procesos y Sistemas – LISI
Universidad Simón Bolívar. Caracas –
Venezuela
marortega@usb.ve

María Pérez

Departamento de Procesos y Sistemas – LISI
Universidad Simón Bolívar. Caracas –
Venezuela
movalles@usb.ve

Luis Mendoza

Departamento de Procesos y Sistemas – LISI
Universidad Simón Bolívar. Caracas – Venezuela
lmendoza@usb.ve

RESUMEN

La complejidad de las tecnologías de los Sistemas de Información Geográfica (SIG) basados en Software Libre (SL), la amplia variedad y el nivel de funcionalidad de este tipo de sistemas en el mercado, hace compleja su selección. Por ello, este artículo tiene por objetivo presentar un conjunto de características de calidad para evaluar los SIG-SL. El conjunto de características está presentado sobre la base de la perspectiva producto del Modelo Sistémico de Calidad de Software (MOSCA), inspirado en el estándar ISO/IEC 9126 y el modelo de calidad de Dromey. Este modelo fue aplicado a un estudio de caso en una organización del estado venezolano, la cual estableció como requerimientos de calidad para un SIG-SL las características de funcionalidad, portabilidad y mantenibilidad. Finalmente, se presenta la validación de la propuesta a través de su aplicación a tres SIG-SL.

Palabras Claves

Modelo de calidad de software, ISO 9126, Modelo de Dromey, MOSCA, Sistemas de información geográfica, Software libre.

INTRODUCCIÓN

El argumento fundamental en la construcción de las premisas del SL está referido a las cuatro libertades de los usuarios del software (GNU, 2005): (1) Libertad para ejecutar el programa en cualquier sitio, con cualquier propósito; (2) Libertad para estudiarlo y adaptarlo a necesidades, esto exige el acceso al código fuente; (3) Libertad de redistribución, de modo que se permita la colaboración y (4) Libertad para mejorar el programa y publicar las mejoras del código fuente, de manera que toda la comunidad se beneficie.

El Ministerio Ciencia y Tecnología de Venezuela (MCT, 2005) publicó el decreto presidencial n° 3.390 el 28 de diciembre del 2004, en el cual insta a la administración pública a emplear prioritariamente software libre desarrollado con estándares abiertos, en sus sistemas, proyectos y servicios informáticos. Para tales fines, todos los órganos y entes de la administración pública iniciarán los procesos de migración gradual y progresiva de éstos hacia el software libre desarrollado con Estándares abiertos.

Esta ley ha comprometido a las organizaciones del estado a estudiar y analizar las tecnologías libres existentes para llevar a cabo su plan de migración a SL en los sistemas que utilizan para su operación. Entre estos sistemas se encuentran los Sistemas de Información Geográfica (SIG), cuya tecnología basada en SL juega un importante papel en diferentes

instituciones, existiendo en la actualidad una gran variedad de desarrollos considerados como herramientas complejas y difíciles de seleccionar, debido al nivel de funcionalidad que disponen. Al respecto Cowen (1990) señala que los SIG son herramientas que se basan en hardware, software y procedimientos que están diseñados para soportar la captura, gestión, manipulación, análisis, modelado y visualización de datos espacialmente referenciados para resolver problemas complejos de planeamiento y gestión; otros autores que soportan esta definición son (Chorley, 1987; Clarke, 1990; Demers, 2000). Por otra parte, los SIG están diseñados para el manejo de información de dos tipos de datos, ellos son los vectoriales y los ráster. Esta investigación está enfocada en los vectoriales, debido a que en el contexto donde se realizó la evaluación no es prioritario el manejo de datos tipo ráster.

Se tiene como objetivo proponer un modelo de especificación de la calidad de SIG basado en SL, que sirva de instrumento a las organizaciones venezolanas para soportar la selección del SIG-SL más adecuado.

Este artículo consta de siete secciones. Además de la introducción, conclusiones y recomendaciones, en la segunda sección se describen los antecedentes; en la tercera se documenta la metodología que se siguió; en la cuarta se hace la propuesta del modelo de calidad, en la quinta se describe el estudio de caso y finalmente en la sexta se analizan los resultados.

ANTECEDENTES

Entre los beneficios que el SL provee se tiene que no existen costos por concepto del software, el código fuente está disponible y puede ser modificado, la comunidad de desarrolladores crece y el ciclo de desarrollo es rápido. Sin embargo, no escapan a los problemas, por ejemplo, no siempre el costo del software es nulo, muchas de las aplicaciones no están tan acabadas en comparación con el software propietario, la compatibilidad con el software propietario puede ser un problema.

Cada día más y más países se agregan a la lista de usuarios de SL, estudios que proporcionan herramientas intelectuales para auxiliar las tareas de adquisición, implementación y monitoreo de sistemas de información basados en SL, son necesarios para impulsar y fortalecer la cultura de uso de código libre. Sánchez (1999) propone una metodología para la implantación de SIG donde incluye diferentes elementos a ser considerados, en los que resalta la importancia de definir: contexto de la organización, información de la adopción tecnológica, requerimientos, educación, alcance del SIG, conversión de datos, contratación de servicios, carga de datos, evaluación y selección tecnológica, adquisición e instalación de la plataforma, diseño de bases de datos, programación, proyecto piloto, pruebas, entrenamiento, implantación, operación y mantenimiento. Por otro lado, Jiménez (2005) presenta un análisis de diferentes sistemas, actualmente disponibles en el ámbito de los SIG-SL, estableciendo para cada uno de ellos una breve descripción de debilidades y puntos fuertes en cuanto a las funcionalidades; sin embargo no presentan criterios definidos de comparación, simplemente muestran una breve descripción de algunos aspectos que poseen los sistemas considerados.

En tal sentido, el proceso de análisis para seleccionar un SIG apropiado en SL y la elaboración de un plan de migración para este tipo de sistema de información, requerirá de la aplicación de un modelo de calidad de software que soporte la selección de SIG en SL de acuerdo a los requerimientos de calidad partiendo de un conjunto de características del estándar ISO 9126: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. Dado esto surge la siguiente interrogante: ¿Cuál(es) será(n) el(los) SIG más adecuado(s) en SL que satisface(n) las necesidades de una organización estatal venezolana? Para dar respuesta a esta interrogante en este artículo se propone un modelo de calidad para soportar la selección de SIG basado en SL.

Modelo sistémico de calidad MOSCA

El modelo sistémico de calidad propuesto por Mendoza, Pérez y Grimán (2005) tiene como finalidad especificar la calidad de los sistemas de software; MOSCA integra dos modelos de calidad: producto (Ortega, Pérez y Rojas, 2002; Ortega, Pérez y Rojas, 2003) y proceso de desarrollo (Pérez, Rojas, Mendoza y Grimán, 2001). Recientemente ha sido considerada la perspectiva humana en MOSCA (Pérez, Domínguez, Mendoza y Grimán, 2006). Además MOSCA está soportado por los conceptos de la calidad total sistémica (Callaos y Callaos, 1996).

MOSCA consta de cuatro niveles a saber, ver figura 1:

Nivel 0. Dimensiones. Las dimensiones son: aspectos internos, contextuales del proceso, aspectos internos y contextuales del producto y aspectos internos y contextuales de la perspectiva humana.

Nivel 1. Categorías. Contempla 14 categorías: 6 pertenecientes al producto, 5 al proceso de desarrollo y 3 a la perspectiva humana.

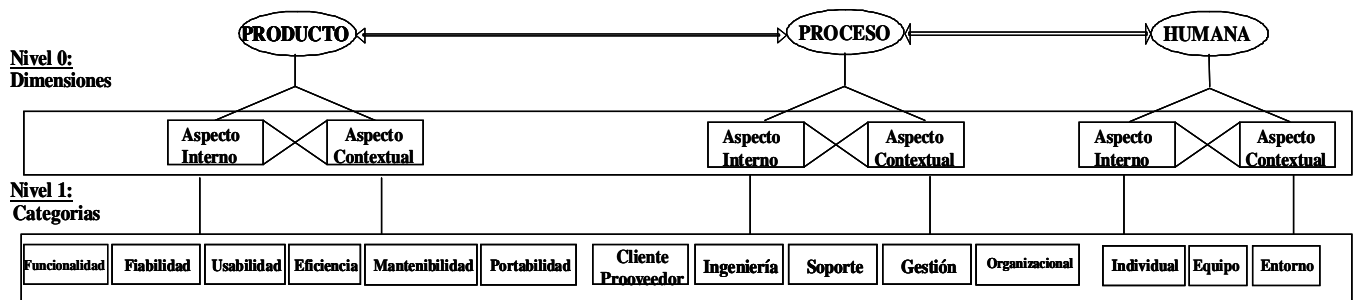


Figura 1. Modelo sistémico de calidad MOSCA

Categorías de la dimensión producto

Funcionalidad (FUN): es la capacidad del producto del software para proveer funciones que cumplan con necesidades específicas o implícitas, cuando el software es utilizado bajo ciertas condiciones.

Confiabilidad (FIA): es la capacidad del producto de software para mantener un nivel especificado de rendimiento cuando es utilizado bajo condiciones especificadas.

Usabilidad (USA): es la capacidad del producto de software para ser atractivo, entendido, aprendido y utilizado por el usuario bajo condiciones específicas.

Eficiencia (EFI): es la capacidad del producto de software para proveer un rendimiento apropiado, relativo a la cantidad de recursos utilizados, bajo condiciones específicas.

Mantenibilidad (MAB): es la capacidad del producto para ser modificado.

Portabilidad (POR): es la capacidad del producto de software para ser transferido de un ambiente a otro.

Categorías de la dimensión proceso

Cliente - Proveedor (CUS): son procesos que impactan directamente al cliente, apoya el desarrollo y la transición del software hasta el cliente, y provee la correcta operación y uso del producto o servicio de software.

Ingeniería (ENG): son procesos que especifican, implementan o mantienen el producto de software, su relación con el sistema y su documentación.

Soporte (SUP): son procesos que son empleados por cualquiera de los otros procesos (incluyendo a los de soporte) en varios niveles del ciclo de vida de adquisición.

Gestión (MAN): son procesos que contienen prácticas de naturaleza genérica, que pueden ser utilizadas por cualquier personaje que dirija algún tipo de proyecto o proceso.

Organizacional (ORG): son procesos que establecen las metas comerciales de la organización y desarrollan bienes (valores) de proceso, producto y recurso, que ayudarán a la organización a alcanzar sus metas en los proyectos.

Categorías de la dimensión perspectiva humana

Aspectos Individuales (IND): considera la calidad humana del individuo, nivel de educación, autogestión, disposición al trabajo en equipo, comunicación y liderazgo.

Equipo (EQU): se refiere a la calidad de los procesos de comunicación y gestión dentro del equipo de trabajo, además considera las actitudes de los miembros del equipo y el nivel de las relaciones entre los mismos.

Entorno Empresarial (ENT): Considera desde un nivel macro las tendencias de la empresa en lo que se refiere a la difusión de su misión y visión, el entrenamiento, la comunicación y el proceso de gestión.

Nivel 2. Características. El siguiente nivel de profundidad corresponde a un conjunto de características que definen las áreas claves a satisfacer para lograr, asegurar y controlar la calidad, tanto en el producto como en el proceso.

Nivel 3. Métricas. Para cada característica se propone una serie de métricas utilizadas para medir la calidad sistémica.

Mendoza y sus colegas ((Mendoza, Pérez, Grimán y Rojas, 2002)) proponen un algoritmo para evaluar la calidad del software usando MOSCA, que se resume en:

1) Estimar la calidad de la funcionalidad del producto. Siempre se debe medir primero la categoría Funcionalidad del producto. Si cumple con el 75% de las características necesarias que se proponen para esta categoría, entonces se procede a la segunda actividad.

2) Instanciación del sub-modelo del producto. En esta actividad el cliente debe seleccionar dos categorías de las cinco restantes del sub-modelo del producto, aquellas que consideran que su software debe cumplir y que desea que sean evaluadas. Luego, se deberán evaluar cada una de las categorías seleccionadas por el cliente. Cabe destacar en este momento que el algoritmo recomienda trabajar con un máximo de tres características del producto (incluyendo la funcionalidad), ya que, si se seleccionan más de tres características del producto, algunas de ellas podrían entrar en conflicto.

3) Estimación de calidad para cada categoría. Para las dos categorías seleccionadas previamente, se debe:

- Aplicar las métricas propuestas en el sub-modelo del producto para las categorías seleccionadas.
- Verificar que el 75% de las métricas se encuentran dentro de los valores óptimos (mayor o igual a 3) para cada una de sus características.
- Evaluar la categoría. Para que una categoría sea satisfecha, al menos el 75% de sus características son altamente satisfechas; de esta manera se garantiza coherencia y consistencia en relación a los niveles de aceptabilidad establecidos por el modelo.

4) Estimar la calidad del producto partiendo de las categorías evaluadas. En este punto es preciso recordar que si no se satisface la categoría funcionalidad el algoritmo finaliza y la calidad del producto de software será nula. Si un producto de software cumple con el objetivo para el cual fue creado (funcionalidad), éste tendrá una calidad Básica. Si satisface sólo una de las categorías seleccionadas, además de la funcionalidad, tendrá un nivel de calidad Intermedio, mientras que si satisface todas las categorías seleccionadas, tendrá un nivel de calidad Avanzado.

METODOLOGÍA

La metodología utilizada se denomina Framework metodológico sistémico para investigar sistemas de información (Pérez, Grimán, Mendoza y Rojas, 2004) y está basada en el método de Investigación-Acción (Checkland, 1993, Baskerville 1999), en la metodología DESMET (Kitchenham, 1996) y en estudios de caso; este framework es el resultado de un proceso evolutivo de aprendizaje con diferentes evaluaciones de productos y procesos de software y consta de 11 actividades (Pérez et al., 2004). La adecuación del marco metodológico sistémico para este trabajo, constó de once actividades que se detallan en la tabla 1.

PROPUESTA DE UN MODELO DE CALIDAD PARA SIG

Para la formulación del modelo fue necesario definir las características de funcionalidad de los SIG según lo establece la metodología en la actividad 5 (Pérez et al., 2004), contando con el apoyo de referencias bibliográficas y del personal experto en el área. Para lograr esto se hizo una consulta con expertos del sector gobierno y con los investigadores del Laboratorio de Investigación en Sistemas de Información (LISI) de la Universidad Simón Bolívar; se convocó a una reunión a gerentes, jefes y otras personas relacionadas con el área, para que eligieran las tres categorías de calidad de acuerdo a los fines del decreto 3.390; se acordó seleccionar funcionalidad, mantenibilidad y portabilidad. La categoría funcionalidad fue seleccionada porque el software a evaluar debe proveer funciones que cumplan con necesidades específicas o implícitas correspondientes a los SIG, este aspecto es indicado en el modelo MOSCA como esencial obligatorio; mientras que en el caso de la mantenibilidad corresponde a la capacidad que debe tener el software para ser modificado ante una necesidad, recordando que el decreto 3.390 busca liberar al estado de su dependencia con el proveedor, esta categoría garantizará mejoras en el SIG a un menor costo. Por último, se encuentra la característica portabilidad, la cual ha sido seleccionada para que el SIG tenga la capacidad de ser transferido de un ambiente a otro, dada la gran variedad de ambientes operativos que tiene la plataforma estatal.

Es de hacer notar que el modelo MOSCA puede ser adaptado para especificar la calidad de los SIG en otro contexto, diferente a SL, seleccionando diferentes categorías y características acordes a los requerimientos de calidad del caso. Esto obedece al enfoque sistémico del modelo, permitiendo adaptar la evaluación al ambiente donde es aplicado, tomando en cuenta la calidad requerida por el usuario.

Esta investigación se enfocó en la estimación de la calidad del producto y no del proceso, ya que no se tiene la documentación requerida para la evaluación del proceso en la empresa donde fueron desarrollados los productos de software; por ello se evaluaron los diferentes SIG basados en SL ya desarrollados y liberados (Producto).

El modelo propuesto consta entonces de las categorías funcionalidad, mantenibilidad y portabilidad, en las tablas 2,3 y 4 se describen estas características.

ACTIVIDAD	DESCRIPCIÓN
1. Investigación documental.	Etapa de la fase diagnosticar, consiste de la revisión del material bibliográfico relacionado con los SIG y calidad.
2. Análisis de antecedentes de MOSCA	Etapa de la fase diagnosticar, consiste en la descomposición de MOSCA y sus relaciones a fin de evaluar la cobertura que posee actualmente respecto a los SIG.
3. Formulación de los objetivos y el alcance de la investigación	Etapa de la fase planificar la acción, consiste en la definición clara de los objetivos de la investigación. El objetivo es delimitar el área de investigación.
4. Adecuación del marco metodológico	En esta actividad el Framework metodológico es especificado, considerando los aspectos relacionados con el propósito de esa investigación.
5. Propuesta de MOSCA para los SIG	Etapa de la fase tomar la acción, en esta etapa se realizan las modificaciones y adaptaciones a MOSCA. El objetivo es obtener una nueva versión de MOSCA que incluya a los SIG, elaborando las métricas requeridas según Basili, Caldiera y Rombach (1994).
6. Análisis de contexto	Etapa de la fase tomar la acción, donde se determinan las especificaciones y acuerdos necesarios para evaluar la nueva versión de MOSCA. El objetivo es preparar las herramientas y el contexto donde será evaluada la propuesta de diseño.
7. Aplicación del método DESMET	Etapa de la fase tomar la acción, donde se ejecuta la metodología DESMET. El objetivo es obtener un método fiable e imparcial para la evaluación.
8. Evaluación a través del método arrojado por DESMET.	Esta es la primera actividad de la fase de evaluación, donde el producto propuesto es evaluado usando el método de evaluación seleccionado según DESMET, en la actividad previa; la cual siguió el análisis de características por estudio de caso.
9. Análisis de los resultados.	Etapa de la fase evaluar, consiste en estudiar los resultados a partir de los objetivos planteados en el trabajo de investigación, en términos de: el modelo propuesto y los productos tangibles alcanzados. Al salir de esta actividad se evalúa si el resultado es satisfactorio, en caso afirmativo se pasa a la actividad 11, en caso negativo se pasa a la actividad 10.
10. Definición del alcance de la siguiente iteración.	Etapa de la fase especificar el aprendizaje, consiste en delimitar el alcance de las modificaciones que se deben realizar al modelo propuesto para aumentar la confiabilidad del mismo y sus posibilidades de éxito. Para efectos de esta investigación, no se realizó esta actividad porque se llevaron a cabo el cumplimiento de los objetivos establecidos.
11. Conclusiones y recomendaciones.	Etapa de la fase especificar el aprendizaje, donde se establecen algunas conclusiones relativas al modelo propuesto aplicado y sus resultados. Finalmente, se sugieren algunas recomendaciones para futuros refinamientos del modelo propuesto y para investigaciones relacionadas.

Tabla 1. Descripción de la metodología. Adaptado de Pérez et al. (2004)

CARACTERÍSTICA	SUB-CARACTERÍSTICA
FUN1. Ajuste a los propósitos	Herramientas, Edición, Elaboración de cartografía temática, Elaboración de cartografía básica, Composición de mapas, Integración de varios mapas, Almacenar, Sistemas de proyección, Topología, Sistemas de coordenadas, Modelos en 3D, Análisis, Curvas de nivel, Despliegue.
FUN2. Precisión	Consultar, Conversiones, Modelo digital del terreno.
FUN3. Interoperabilidad	Importar, Tipos de archivos de datos digitales ráster, Tipos de archivos de datos digitales vector, Multiplataforma, GNSS, Salida de Información.
FUN4. Seguridad	Control de acceso
FUN5. Correctitud	Geocodificación, Manejo de grandes volúmenes de información. Capacidad de cómputo, completitud y consistencia (Dromey, 1995, 1996)
FUN6. Encapsulado	Propias de MOSCA. Las variables, las constantes y los tipos deben ser utilizados sólo en el contexto en el que son definidos (Dromey, 1995, 1996)
FUN7. Especificado	Propias de MOSCA. La funcionalidad es descrita con pre-condiciones y post-condiciones (Dromey, 1995, 1996).

Tabla 2. Características de la categoría funcionalidad. Adaptado de (Mendoza et al., 2005; ISO 9126, 2001)

CARÁCTERÍSTICA	DEFINICIÓN
MAB1. Capacidad de análisis	Capacidad del software para ser diagnosticado por deficiencias o fallas
MAB2. Capacidad de cambio	Capacidad del software para facilitar una modificación específica a ser implementada
MAB3. Estabilidad	Capacidad del software para evitar efectos inesperados después de modificaciones
MAB4. Capacidad de prueba	Capacidad del software para permitir que las modificaciones sean validadas.
MAB5. Acoplamiento	Mide la interconexión entre los módulos de una estructura de programa.
MAB6. Cohesivo	Una forma estructural es cohesiva si todos sus elementos están enlazados uno a los otros y contribuyen a llevar a cabo un simple objetivo o función.
MAB7. Encapsulado	Se refiere a que las variables las constantes y los tipos deben ser utilizados sólo en el contexto en el que son definidos
MAB8. Atributos de madurez del Software	Son las características físicas y las medidas asociadas a la edad y uso del sistema de software objetivo
MAB9. Sistemas de estructuras de control	Son las propiedades relacionadas a la modularidad los atributos de control inter-modular la selección y uso de construcciones de flujo de control la manera en la cual el programa o sistema es descompuesto en algoritmos y el método con el cual los algoritmos son implementados.
MAB10. Sistemas de estructura de información	Son aquellas propiedades asociadas al almacenamiento y flujo de información en un programa o sistema incluyendo definiciones de datos y las características de las entradas y salidas del sistema.
MAB11. Descriptivo	El código del software debe ser fácil de entender.
MAB12. Correctitud	Está dividida en tres propiedades relacionadas con la capacidad de cómputo completitud y consistencia.
MAB13 Efectivo	Una forma estructural es efectiva cuando presenta sólo los elementos necesarios para definir e implementar la forma estructural
MAB14. Directo	Una expresión es directa cuando la abstracción, representación y la estructura del cálculo es congruente con el problema original.

Tabla 3. Características de la categoría mantenibilidad (Mendoza et al., 2005; ISO 9126, 2001)

CARACTERÍSTICA	DEFINICIÓN
POR1. Adaptabilidad	Capacidad del producto de software para ser adaptado a diferentes ambientes especificados sin aplicar acciones u otros medios que no sean los provistos para este propósito en el software considerado
POR2. Capacidad de Instalación	Capacidad del producto de software para ser instalado en un ambiente especificado.
POR3. Co-existencia	Capacidad del producto de software para co-existir con otro software independiente en un ambiente común compartiendo recursos comunes.
POR4. Capacidad de reemplazo	Capacidad del producto de software para ser usado en lugar de otro producto de software especificado para el mismo propósito en un mismo ambiente
POR5. Consistente	Una forma estructural es consistente si su uso mantiene sus propiedades o funcionalidad y si todos sus elementos contribuyen a reforzar su intención o efecto
POR6. Encapsulado	Las variables, las constantes y los tipos deben ser utilizados sólo en el contexto en el que son definidos
POR7. Cohesivo	Una forma estructural es cohesiva si todos sus elementos están enlazados uno a los otros y contribuyen a llevar a cabo un simple objetivo o función
POR8. Especificado	Un módulo, programa u otra forma estructural están especificados si su funcionalidad es descrita con pre-condiciones y post-condiciones
POR9. Documentado	Una forma estructural está documentada cuando su propósito y propiedades están explícitamente definidas en el contexto de la forma estructural. Aplicable a módulos, ciclos, estructuras de datos, variables, constantes, tipos, etc.
POR10. Auto-descriptivo	Una forma estructural es auto-descriptiva si su propósito es evidente en los nombres de los módulos y los identificadores tienen significado en el contexto de la aplicación

Tabla 4. Características de la categoría portabilidad (Mendoza et al., 2005; ISO 9126, 2001)

El modelo propuesto para evaluar SIG-SL consta de trescientas setenta y cinco métricas (375), de las cuales doscientas quince (215) fueron elaboradas para esta investigación correspondientes a la categoría funcionalidad, necesarias para la estimación de la calidad de los SIG-SL, se complementan con cuarenta y seis (46) de funcionalidad, setenta y nueve (79) son de mantenibilidad y treinta y cinco (35) de portabilidad originales del modelo MOSCA

ESTUDIO DE CASO: CONATEL

Este trabajo de investigación contempla la evaluación de los SIG, en una organización del estado venezolano, conocida como Comisión Nacional de Telecomunicaciones (CONATEL). CONATEL tiene aproximadamente 600 empleados y cuenta con una Gerencia de Análisis e Información Geográfica (GAIG), encargada de planificar y controlar el desarrollo y actualización del sistema de información geográfica, a fin de evaluar a través del análisis espacial la infraestructura de los servicios de telecomunicaciones, servicios públicos y variables socioeconómicas para determinar el mercado de servicios de telecomunicaciones en Venezuela.

Se definieron los siguientes roles para la evaluación:

- **Patrocinantes:** la evaluación fue patrocinada por CONATEL y por el laboratorio LISI que es el laboratorio de investigación financiado por el gobierno para el desarrollo y mejoras de MOSCA.
- **Evaluador:** es el responsable de llevar a cabo la evaluación, en este caso se trata de los autores del presente trabajo, junto a dos Licenciados en Geografía y un Ingeniero de Sistemas. Sus responsabilidades fueron: preparación del plan de evaluación, identificación del objeto de la evaluación, identificación y definición de las características a evaluar en la adecuación de MOSCA en los SIG, organización del objeto de evaluación, recopilación y análisis de los resultados obtenidos, preparación del informe final y recomendaciones.
- **Usuarios del modelo:** organizaciones que desean identificar las fortalezas y debilidades de los SIG-SL con el objetivo de seleccionar uno y efectuar planes de migración.

El proyecto contempló la evaluación de tres SIG-SL (ver tabla 5).

SIG	DESCRIPCIÓN
GRASS	Desarrollado en sus inicios por el cuerpo de ingenieros del ejército de los Estados Unidos y posteriormente liberado. Basado en UNIX y soporta prácticamente todas las plataformas y sistemas operativos. Versión 6.0, 2005; Leguaje C++, http://grass.itc.it/
Qgis	Apoya la visión con archivos shape y el almacenaje de datos espacial con PostgreSQL/PostGIS. Posee un gran número de características vía un interfaz con plugin. Versión 0.7, 2005, C++, http://qgis.org/
Thuban	Es un visor de datos geográficos interactivo. Es extensible y multi-plataforma (GNU/Linux, Windows). Versión 1.0, 2005, C++, http://thuban.intevation.org/

Tabla 5. Descripción de los SIG evaluados

Es importante resaltar que solamente se evaluaron tres (3) SIG en SL porque de los encontrados, son los que cumplen exactamente con las cuatros libertades asociadas a SL.

ANÁLISIS DE RESULTADOS

Una vez realizada las mediciones basadas en las características de funcionalidad, mantenibilidad y portabilidad; en los tres (3) SIG en SL, se obtuvieron los siguientes resultados.

Funcionalidad

La figura 2, muestra que el software Grass cumple con 7 de las 8 características de la categoría de funcionalidad; es decir, este software posee al menos un 75% de las métricas asociadas; ya que en las características FUN3 (Interoperabilidad) y FUN4 (Seguridad) no fue satisfecha. Se debe recordar que si 6 de las características del producto de software son altamente satisfechas, entonces la categoría funcionalidad está presente en el producto de software evaluado. No obstante, los otros dos software no cumplen con el 75% de las métricas asociadas, Qgis cumple sólo 2 de las 8 características de la categoría de funcionalidad, las cuales son: FUN3 (Interoperabilidad) y FUN8 (Especificado); a pesar de que el software Grass es el que satisface las características de funcionalidad ante los otros dos, muestra una debilidad en comparación con el Qgis en FUN3 (Interoperabilidad). Por su parte el software Thuban cumple con 2 de las características de funcionalidad, las cuales son: FUN5 (Correctitud) y FUN 8 (especificado).

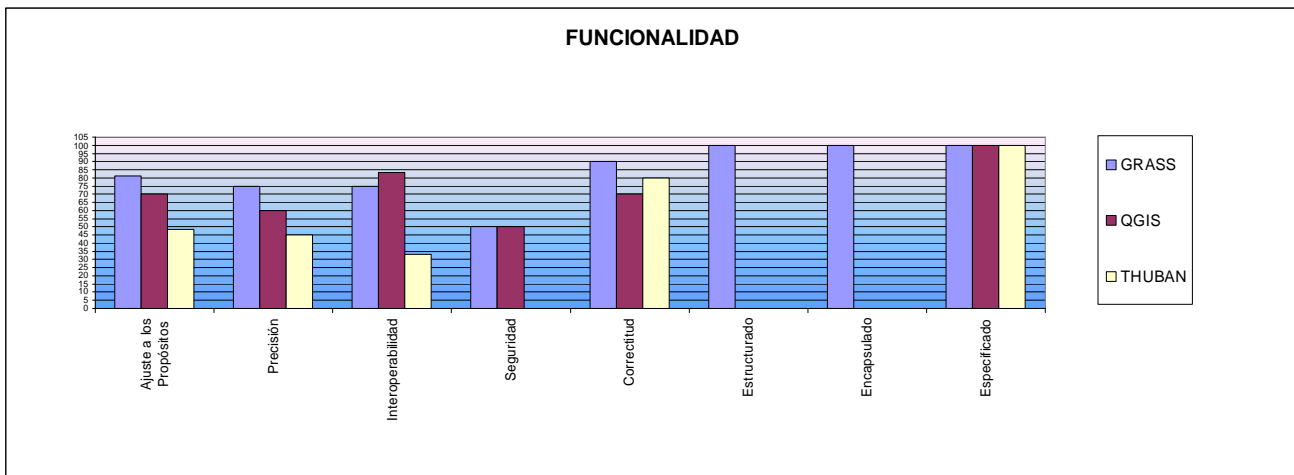


Figura 2. Comparación de los SIG-SL en la categoría de Funcionalidad

Mantenibilidad

Se puede observar en la figura 3, que el software Grass, Qgis y Thuban cumple con 12 de las 14 características de la categoría de mantenibilidad. Dos características que muestran debilidad en estos tres software son en MAB4 (Capacidad de prueba) y MAB7 (Encapsulamiento), siendo similares los resultados obtenidos.

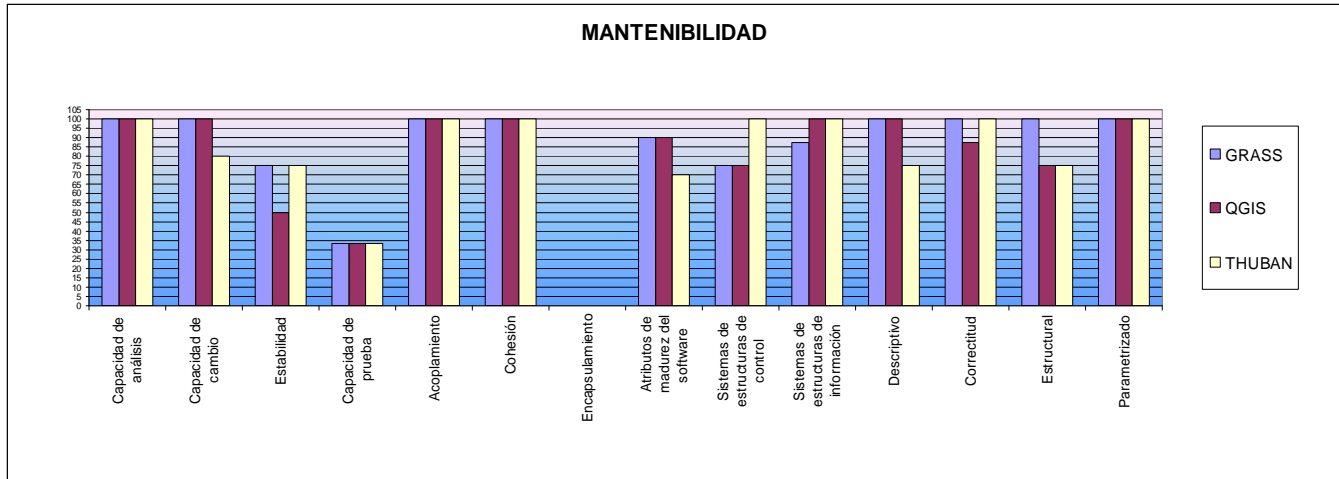


Figura 3. Comparación de los diferentes SIG en SL evaluados en la categoría de Mantenibilidad

Portabilidad

La figura 4, el software Qgis y Thuban cumplen con 11 de las 12 características de la categoría portabilidad; mientras que Grass cumple con 9 de las 12 características para esta categoría. Es importante destacar que Grass es débil en POR 2 (Capacidad de instalación), POR 4(Capacidad de reemplazo) y no tiene POR 9 (Especificado), lo que lo convierte menos competitivo con los software Thuban y Qgis que si presentan la característica POR 9 correspondiente a esta categoría.

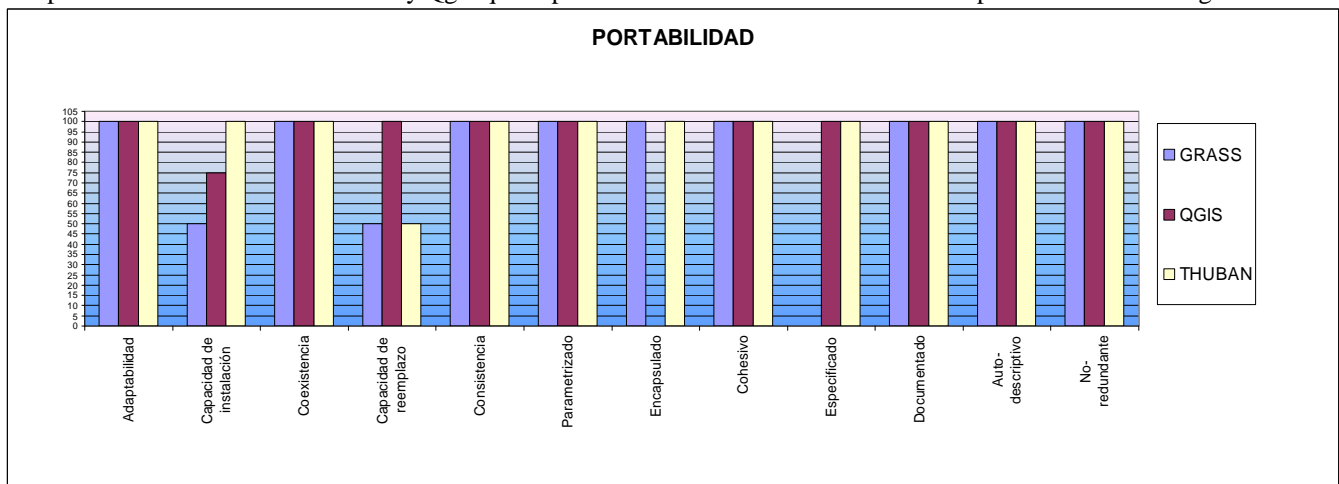


Figura 4. Comparación de los diferentes SIG en SL evaluados en la categoría de Portabilidad

Los licenciados en geografía consideraron *pertinentes* las métricas propuestas para la categoría funcionalidad, resaltando la completitud de la clasificación en las características de este tipo de software; a su vez señalaron lo relevante de haber desarrollado métricas que evalúen la amigabilidad de las interfaces de los diversos SIG, ya que para los usuarios de estos sistemas, la amigabilidad es un factor clave, porque de ella depende la facilidad para operar el SIG a seleccionar. Estos señalaron lo apropiado del modelo para la organización. Por su parte, el Ingeniero de Sistemas que también formó parte del proceso de evaluación presentó una opinión similar, resaltando que las métricas establecidas para la categoría de mantenibilidad son pertinentes, ya que se evalúa el código fuente de los software desde un nivel bajo de líneas de instrucción hasta un nivel más alto de módulos y dependencias entre ellos, siendo esto un factor a tomar en cuenta al momento de realizar desarrollos sobre el código fuente del SIG a seleccionar, indicando el nivel de mantenibilidad que muestran los SIG; en la categoría portabilidad también fueron consideradas pertinentes las sub-características que consideran la capacidad para ser instalado entre otras. Estos argumentos permitieron deducir al evaluador que MOSCA es conveniente para la organización.

CONCLUSIONES Y RECOMENDACIONES

Este trabajo ofrece un aporte esencial a las organizaciones que hacen uso de los SIG y buscan migrar a SL, pues no se encontraron trabajos similares como éste en la bibliografía consultada. Además, este trabajo proporciona una base sólida de la evaluación permitiendo ser adaptado a la realidad de Venezuela. Con este trabajo se logró especificar un modelo de calidad para los SIG, necesario para las organizaciones venezolanas que requieren de este tipo de modelo para sustentar/justificar la selección del SIG basado en SL mas adecuado. De igual manera, el desarrollo de esta investigación permitió realizar mejoras en el Modelo Sistemico de Calidad (MOSCA). Los SIG-SL evaluados fueron: Grass, QGSI y THUBAN. De los tres GRASS fue el que alcanzó los mejores niveles de funcionalidad y mantenibilidad. En portabilidad los tres SIG demostraron aproximadamente los mismos niveles de calidad. Lo que lleva a recomendar Grass para el estudio de caso presentado en el artículo. Dado el ambiente en el que este tipo de software es desarrollo se recomienda la evaluación de las dimensiones proceso (que incluye actividades de soporte) y perspectiva humana.

REFERENCIAS

1. Basili, V. R., Caldiera, G., Rombach, H. D. (1994) *Goal Question Metric Paradigm*. In J. J. Marciniak (ed.), *Encyclopedia of Software Engineering*, John Wiley & Sons
2. Baskerville, R. (1999) Investigating Information Systems with Action Research. *Communications of the Association for Information Systems*, Octubre, 2, 19, 1-32.
3. Callaos, N. y Callaos, B. (1996) Designing with Systemic Total Quality, *International Conference on Information Systems*, Orlando, Florida, July, 548-560.
4. Checkland, P. (1993). *Pensamiento de Sistemas, Práctica de Sistemas*. Grupo Noriega Editores.
5. Chorley, R. (1987) *Handling Geographic Information*. HMSO
6. Clarke, M. (1990) *Geographical Information Systems and Model Based Analysis* in Scholten, H.K. and Stillwell, C.H. (eds) *Geographical Information Systems for Urban and Regional Planning*. Kluwer Academic Publishers
7. Cowen, D.J. (1990) *GIS versus CAD versus DBMS: what are the differences?* in Perquet, D. & Marble, D. (eds)
8. Demers, M (2000). *Fundamentals of Geographic Information Systems*. John Wiley & Sons.
9. Dromey, G. (1995) A Model for Software Product Quality, *IEEE Transactions on Software Engineering*, February, Vol 21, Nº 2, 146-162.
10. Dromey, G. (1996) Cornering the Chimera, *IEEE Software*, January, 1996, 33-43.
11. GNU (2005) Sistema operativo GNU - Fundación para el software libre, <http://www.gnu.org/philosophy/free-sw.es.html>
12. Grass (2005) Geographic Resources Analysis Support System, <http://grass.itc.it/>
13. ISO 9126 (2001) ISO/IEC 9126-1:2001(E) Software engineering — Product quality —Part 1: Quality model. First edition,
14. Jiménez, J. (2005) Alternativas de Software Libre a los Sistemas Información Geográfica Comerciales. [Homepage]. Consultado el día 4 de enero del 2006 de la World Wide Web: <http://adm.ing.unibo.it/ADM%20Ingegraf%202005/pdf/COMUNICACIONES%20ACEPTADAS/GIS10.pdf>
15. Kitchenham, B. (1996). Evaluating Software Engineering Methods and Tools. Part 1: The Evaluation Context and Evaluation Methods. *ACM SIGSOFT - Software Engineering Notes*, 21, 1, 11-14.
16. Mendoza, L., Pérez, M., Grimán, A. and Rojas, T. (2002) Algoritmo para la Evaluación de la Calidad Sistemica del Software *Memorias de las 2das. Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC 2002)* Salvador, Brasil.
17. Mendoza, L; Pérez, M. y Grimán, A. (2005). Prototipo de Modelo Sistemico de Calidad (MOSCA) del Software: *Computación y Sistemas*, 8, 3, 196-217.
18. MCT, Ministerio de Ciencia y Tecnología (2005) *Plan nacional de ciencia y tecnología*. <http://www.mct.gov.ve> Caracas Venezuela
19. Ortega, M., Pérez, M. y Rojas, T. (2000) A Model for Software Product Quality with a Systemic Focus, *4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2000 and The 6th International Conference on*

- Information Systems, Analysis and Synthesis ISAS 2000*, Orlando, Florida, July, 395-401, <http://www.lisi.usb.ve/publicaciones>.
20. Ortega, M.; Pérez, M. y Rojas, T. (2002) A Systemic Quality Model for Evaluating Software Products, *6th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, Florida, Vol. I, 371-376.
 21. Ortega, M., Pérez, M. y Rojas, T. (2003) Construction of a Systemic Quality Model for evaluating a Software Product, *Software Quality Journal*, Kluwer Academic Publishers, Julio, 11:3, 219-242
 22. Pérez, M. Grimán, A., Mendoza, L. y Rojas, T. (2004) A Systemic Methodological Framework for IS Research. *Proceedings of the Tenth Americas Conference on Information Systems(AMCIS 2004)* , August, USA, 4374-4383.
 23. Pérez, M., Rojas T., Mendoza, L. y Grimán, A. (2001) Systemic Quality Model for System Development Process: Case Study, *Seventh Americas Conference on Information Systems – AMCIS*, Boston, Massachusetts, August, 1297-1304, <http://www.lisi.usb.ve/publicaciones>.
 24. Pérez, M., Domínguez, K., Mendoza, L. y Grimán, A. (2006) Human Perspective in System Development Quality. *12th Americas Conference on Information Systems (AMCIS)*. Acapulco, México. Agosto.
 25. Quantum (2005) Quantum GIS (QGIS), <http://qgis.org/>
 26. Sánchez, E. (1999) Una metodología sistémica para la implantación de SIG. *ESRI Latin America User Conference Proceedings*. April 12–16, Venezuela, <http://gis.esri.com/library/userconf/latinproc99/ponencias/ponencia13.html>.
 27. Thuban (2005) Interactive Geographic Data Viewer, <http://thuban.intevation.org/>

Quality Estimation Model for Geographic Information Systems

Mirla Tahhán

CONATEL. Caracas – Venezuela

mirlat9@gmail.com

Raymond Sánchez

CONATEL. Caracas – Venezuela

ray.harraka@gmail.com

Maryoly Ortega

Departamento de Procesos y Sistemas – LISI
Universidad Simón Bolívar. Caracas –
Venezuela

marortega@usb.ve

María Pérez

Departamento de Procesos y Sistemas – LISI
Universidad Simón Bolívar. Caracas –
Venezuela

movalles@usb.ve

Luis Mendoza

Departamento de Procesos y Sistemas – LISI
Universidad Simón Bolívar. Caracas – Venezuela
lmendoza@usb.ve

ABSTRACT

Selecting a Geographic Information System (GIS) based on Free Software (FS) is a difficult task vis-à-vis its technological complexity, the wide range of this type of systems and their functionalities. Therefore, provides a set of quality characteristics used to evaluate FS-GIS. This set of characteristics is presented on the product perspective of the Systemic Software Quality Model (MOSCA), which, in turn, is inspired in the ISO/IEC 9126 standard and the Dromey's quality model. This model was applied to a case study in a State entity, which set functionality, portability, and maintainability as the quality requirements for FS-GIS. Finally, the validation of the proposal through its application to three FS-GIS is presented.

Keywords

Software quality model, ISO 9126, Dromey's model, MOSCA, Geographic Information Systems, Free Software.

INTRODUCTION

Free software (FS) is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom for the users of the software (GNU, 2005): (1) The freedom to run the program, for any purpose; (2) The freedom to study how the program works, and adapt it to your needs. Access to the source code is a precondition for this; (3) The freedom to redistribute copies so you can help your neighbor and (4) The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. Access to the source code is a precondition for this.

The Venezuelan Ministry of Science and Technology (MCT, 2005) issued the presidential decree No. 3.390 on December 28th, 2004, in which the public administration is urged to use mainly FS developed with open standards for all governmental systems, projects and information technology services.

This decree has committed State institutions to study and analyze the existing free technologies in order to implement the migration plan to FS in their systems. These systems include Geographic Information Systems (GIS), whose FS-based technology plays a major role in different institutions. Today there is a wide range of developments considered complex tools which are difficult to be selected due to their functionality level. In this regard Cowen (1989) points out that GIS are tools based on hardware, software and procedures designed to support the collection, management, handling, analysis, modeling and visualization of spatially referenced data to solve complex planning and management issues; other authors who endorse this definition are (Chorley, 1987; Clarke, 1990; Demers, 2000). Furthermore, GIS are designed for handling information in

two kinds of data: vector and raster data. This research is focused on vector data, because raster data handling is not a priority within the evaluation context.

The purpose of this work is to propose a quality specification model for FS-based GIS. This model will assist Venezuelan organizations in selecting the most suitable FS-GIS.

This paper consists of seven sections: introduction; conclusions and recommendations; the background of the research is described in section 2; section 3 presents the methodology followed in the research; section 4 presents the quality model; section 5 includes the case study; and section 6 is devoted to the results analysis.

BACKGROUND

Some of the advantages of FS are that no costs are derived from the software, the source code is available, and can be modified; developers' community grows; and the development cycle is fast. However, there are also some disadvantages: software is not always free of cost, a number of applications are not as finished as the proprietary software, and compatibility with proprietary software can be a problem.

An increasing number of countries are added every day to the list of FS users; therefore studies that provide the intellectual tools for acquiring, implementing, and monitoring FS-based information systems are necessary to boost and strengthen the culture of free software use.

Sánchez (1999) proposes a methodology for GIS implementation, considering: organization context, adoption information, requirements, education, SIG scope, data conversion, services contract, data entry, technological evaluation and selection, platform acquisition and installation, data base design, programming, initial project, test, training, implementation, operation and maintenance. Jiménez (2005) presents an analysis of the different systems which are currently available within the context of FS-GIS and offers a brief description of their weaknesses and strengths as to functionalities. However, Jiménez work (2005) does not provide defined comparison criteria, presenting only a brief description of some aspects of the software studied.

In this sense, the process to select a suitable FS-GIS shall imply the application of a software quality model. The model will support this selection according to the quality requirements based on a set of characteristics such as: functionality, reliability, usability, efficiency, maintainability, and portability (ISO 9126). The following question arises: Which would be the most suitable FS-GIS that would meet the needs of a Venezuelan State organization?

A quality model which can be used to select the FS-GIS is proposed in this paper as an answer to the previous question.

MOSCA Systemic Quality Model

The systemic quality model proposed by Mendoza, Pérez and Grimán (2005) is used to specify software system quality; it combines the Product Quality Model (Ortega, Pérez and Rojas, 2002; Ortega, Pérez and Rojas, 2003) and the Development Process Quality model (Pérez, Rojas, Mendoza and Grimán, 2001) and is based on the concepts of Total Systemic Quality (Callaos and Callaos, 1996). Recently has been considered the human perspective in MOSCA (Pérez, Domínguez, Mendoza and Grimán, 2006).

MOSCA consists of four levels (Figure 1):

Level 0. Dimensions. Dimensions are internal and contextual aspects of the process, internal and contextual aspects of the product, and internal and contextual aspects of human perspective.

Level 1. Categories: 14 categories are considered: 6 are referred to the product, 5 to the development process and 3 to human perspective.

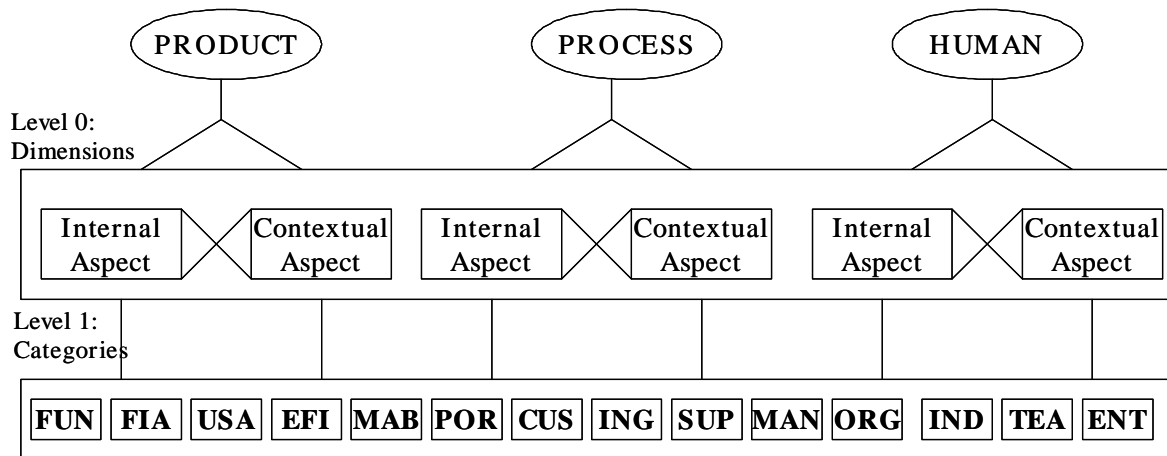


Figure 1. Systemic Quality Model MOSCA

Product dimension categories

Functionality (FUN): The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions.

Reliability (FIA): The capability of the software product to maintain a specified level of performance when used under specified conditions.

Usability (USA): The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions.

Efficiency (EFI): The capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions.

Maintainability (MAB): The capability of the software product to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.

Portability (POR): The capability of the software product to be transferred from one environment to another. The environment may include organizational, hardware or software environment.

Process dimension categories

Client-Provider (CUS): Processes that have a direct impact on the client; support software development and its transition to the client; and provide the proper operation and use of the software product or service.

Engineering (ENG): Processes that specify, implement, or maintain the software product, its relationship with the system and its documentation.

Support (SUP): Processes that are used by any of the other processes (including those in support) at different levels of the acquisition life cycle.

Management (MAN): Processes that contain generic practices which can be used by any person that is conducting any kind of project or process.

Organizational (ORG): Processes that establish the commercial goals of an organization and develop process, product, and resource goods (values), which will allow the organization to attain the goals, set in each project.

Human Perspective dimension categories

Individual Aspects (IND): The individual human quality, educational level, and self-management, as well as the attitude to team work, communication and leadership.

Team (EQU): Quality of the communication and management processes within the working team. This category considers also the attitudes of the team members and the level of the relationships among them.

Corporate Environment (ENT): This category considers, at a macro level, the company trends regarding the dissemination of its mission, its vision, training, communication, and management process.

Level 2. Characteristics. The next depth level corresponds to a number of characteristics which define the key areas to be addressed to achieve, assure, and control quality both in the product as well as in the process.

Level 3. Metrics. A series of metrics used to measure systemic quality is proposed for each characteristic.

Mendoza and his colleagues (Mendoza, Pérez, Grimán and Rojas, 2002) propose an algorithm to evaluate software quality using MOSCA; the algorithm can be summarized as follows:

1) Estimation of the product functionality quality. The Functionality product category should always be the first to be measured. If it meets 75% of the required characteristics proposed for this category, the second activity can be started.

2) Instantiation of the product sub-model. In this activity the client has to select two categories out of the five remaining for the product sub-model, those that in the opinion of the client have to be met by the software and have to be evaluated. Then, each one of the categories selected by the client has to undergo evaluation. It is worth mentioning that the algorithm recommends working with a maximum of three product characteristics (including functionality), because if more than three characteristics are selected, any could come into conflict with another.

3) Quality estimation for each category. For the two categories selected it is necessary to:

- Apply the metrics proposed in the product sub-model for the selected categories.
- Verify that 75% of the metrics are within the optimal values (≥ 3) for each one of their characteristics.
- Evaluate this category. For a category to be met, at least 75% of its characteristics should be satisfied, so that coherence and consistency in relation to the acceptability levels established by the model can be guaranteed.

4) Estimation of product quality based on the evaluated categories. It is necessary to remind at this point that if the Functionality category is not satisfied, the algorithm comes to an end and the software product quality will be null. If a software product meets the purpose for which it was created (Functionality), its quality will be Basic. If the product meets only one of the selected categories, besides Functionality, it will have an Intermediate quality level; whereas if it satisfies all of the selected categories, its quality level will be Advanced.

METHODOLOGY

The methodology applied is known as systemic methodological framework for information systems research (Pérez, Grimán, Mendoza and Rojas, 2004) and is based on the Research-Action method (Checkland, 1993; Baskerville, 1999), on the DESMET methodology (Kirchenham, 1996), and on case studies. This framework is the result of an evolutive learning process with evaluation of different software product and process, and consists of 11 activities (Pérez et al., 2004). The adaptation of the systemic methodological framework for this work consisted of eleven activities which are detailed in Table 1.

PROPOSAL OF A QUALITY MODEL FOR GIS

With the aim of formulating the model, GIS functionality characteristics were defined according to activity 5 of the methodological framework (Pérez et al., 2004), based on bibliographic references and with the support of experts in the area. To this end, consultations were held with experts of the governmental sector and researchers of the Information Systems Research Laboratory (LISI for its acronym in Spanish) of the Universidad Simón Bolívar. In addition, a meeting was held with managers, directors and other persons related to the area. At this meeting they selected the three quality categories in accordance with the purposes of Decree 3.390; the categories chosen were functionality, maintainability, and portability. Functionality was selected because the software to be evaluated should provide functions that meet the specific or implicit needs of the GIS; this aspect is described in the MOSCA model as essential-compulsory. Maintainability is referred to the ability of the software to undergo modifications to respond to a specific need; considering that Decree 3.390 this category will guarantee that the GIS is improved at a lower cost. Portability was selected so that the GIS would have the capability to be transferred from one environment to another, given the wide range of operational environments of the State platform.

It is worth mentioning that the MOSCA model can be adapted to specify the GIS quality in another context, different from

that of the FS, by selecting different categories and characteristics in accordance with the quality requirements of the case. This process follows the model systemic approach and makes it possible to adapt the evaluation to the environment where the model is applied, taking into account the quality requirements stated by the users.

This research was focused on the quality estimation of the product rather than the process, due to the lack of documentation required to evaluate the process in the company where the software products were developed. Thus, the different FS-GIS already developed and liberated were evaluated.

The model proposed consists of the functionality, maintainability and portability categories. The characteristics proposed for each category are described in tables 2,3 and 4.

ACTIVITY	DESCRIPTION
1. Documental research	Stage of the diagnosis phase; it consists of reviewing bibliographical material related to GIS and quality.
2. Analysis of MOSCA	Stage of the diagnosis phase; it consists of decomposing MOSCA and its relationships in order to evaluate its present coverage regarding GIS.
3. Objectives and scope Setting	Stage of the action planning phase; it consists of clearly defining the research objectives. The purpose is to specifically identify the research area.
4. Methodological framework adaptation	The methodological framework is specified in this activity, considering those aspects related to the purpose of this research.
5. Proposal of MOSCA for GIS	Stage of the action-taking phase; MOSCA modifications and adaptations are performed in this stage. The aim is to obtain a new MOSCA version that includes GIS, by developing the metrics required according to Basili, Caldiera and Rombach (1994).
6. Contextual analysis	Stage of the action-taking phase, where specifications and agreements required to evaluate the new MOSCA version are determined. The aim is to prepare the tools and context where the model proposal will be evaluated.
7. Application of DESMET	Stage of the action-taking phase, where the DESMET methodology is applied. The aim is to obtain a reliable and impartial evaluation method.
8. Evaluation through the method selected by DESMET	This is the first activity of the evaluation phase, where the proposed product is evaluated by means of the evaluation method selected according to DESMET in the previous activity which followed the analysis of characteristics based on case study.
9. Results analysis	Stage of the evaluation phase; it consists of studying the results based on the objectives set in the research work, in terms of the proposed model and the tangible products obtained. When this activity is completed, the result is evaluated for its compliance; if the result is affirmative, activity No. 11 is started; if it is negative, activity No. 10 should be started.
10. Definition of the scope of the next iteration	Stage of the learning specification phase; it consists of defining the scope of the modifications to be incorporated into the proposed model to increase its reliability and success probabilities. For the purposes of this research, this activity was not performed because the objectives established were attained.
11. Conclusions and recommendations	Stage of the learning specification phase, where a series of conclusions are drawn regarding the proposed model and its results. Finally, a number of recommendations are made for further improvement of the proposed model and for related research works.

Table 1. Description of the Methodological Framework. Adapted from Pérez et al. (2004)

CHARACTERISTIC	SUB-CHARACTERISTIC
FUN1. Suitability	Tools, Editing, Development of the theme cartography, Development of the basic cartography, Composition of maps, Integration of different maps, Storing, Projection systems, Topology, System of coordinates, 3D models, Analysis, Level curves, Display
FUN2. Accuracy	Consultation, Conversions, Terrain digital model
FUN3. Interoperability	Import, Types o digital raster data files, Types o digital vector data files, Multi-platform, GNSS, Information output
FUN4. Safety	Access control
FUN5. Correct	Geo-coding, Handling of large information volumes, Calculation ability, completeness, and consistency, Dromey).
FUN6. Encapsulated	Inherent to MOSCA. Variables, constant and types should be used only in the context within which they are defined (Dromey).
FUN7. Specified	Inherent to MOSCA. Functionality is described with preconditions and postconditions (Dromey).

Table 2. Functionality category characteristics. Adapted from ((Mendoza et al., 2005; ISO 9126, 2001)

CHARACTERISTIC	DEFINITION
MAB1. Analyzability	The capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for the parts to be modified to be identified.
MAB2. Changeability	The capability of the software product to enable a specified modification to be implemented.
MAB3. Stability	The capability of the software product to avoid unexpected effects from modifications of the software.
MAB4. Testability	The capability of the software product to enable modified software to be validated.
MAB5. Coupling	This characteristic measures interconnection between the modules of a program structure.
MAB6. Cohesion	A structural form is cohesive when all its elements are linked to each other and contribute to perform a simple objective or function.
MAB7. Encapsulation	Variables, constants and types should be used only in the context within which they are defined.
MAB8. Software maturity attributes:	Physical characteristics and measures associated to age and use of the target software system.
MAB9. Control structure systems	Properties related to modularity; attributes of inter-modular control; selection and use of control flow constructions; the way how the program or system is decomposed into algorithms; and the method through which algorithms are implemented.
MAB10. Information structure systems	Properties related to information flow and storage in a program or system, including data definitions and the characteristics of the system inputs and outputs.
MAB11. Descriptive	The software code should be easily understood.
MAB12. Correct	This characteristic is divided into three properties related to calculability, completeness, and consistency.
MAB13. Effective	A structural form is effective when it presents only those elements necessary to define and implement the structural form.
MAB14. Direct	An expression is direct when the abstraction, representation, and structure of the calculation are coherent with the original problem.

Table 3. Maintainability category characteristics (Mendoza et al., 2005; ISO 9126, 2001)

CHARACTERISTIC	DEFINITION
POR1. Adaptability	The capability of the software product to be adapted for different specified environments without applying actions or means other than those provided for this purpose for the software considered.
POR2. Installability	The capability of the software product to be installed in a specified environment.
POR3. Coexistence	The capability of the software product to co-exist with other independent software in a common environment sharing common resources.
POR4. Replaceability	The capability of the software product to be used in place of another specified software product for the same purpose in the same environment.
POR5. Consistent	A structural form is consistent if its use maintains its properties or functionality and if all its elements contribute to the strengthening of its intention or effect.
POR6. Encapsulated	Variables, constants and types should be used only in the context within which they are defined.
POR7. Cohesion	A structural form is cohesive when all its elements are linked to each other and contribute to perform a simple objective or function.
POR8. Specified	A module, program or another structural form are specified if their functionality is described with preconditions and post-conditions.
POR9. Documented	A structural form is documented when its purpose and properties are explicitly defined in the context of the structural form. This is applicable to modules, cycles, data structures, variables, constants, types, etc.
POR10. Self-descriptive	A structural form is self-descriptive if its purpose is evident from the names of the modules and if the identifiers have a meaning within the context of the application.

Table 4. Portability category characteristics (Mendoza et al., 2005; ISO 9126, 2001)

The model proposed for the evaluation of the FS-GIS consists of 375 metric, from which 215 were prepared for this research and correspond to the functionality category, necessary for the quality estimation of FS-GIS; these are complemented with 46 for functionality, 79 for maintainability and 35 for portability, which are originally taken from the MOSCA model.

CASE STUDY: CONATEL

This research work studies the evaluation of GIS in a Venezuelan State entity, known as Comisión Nacional de Telecomunicaciones CONATEL (National Telecommunications Commission) that has approximately 600 workers. One of its divisions is the Direction of Analysis and Geographic Information (GAIG, for its acronym in Spanish), responsible for planning and controlling the development and updating of the geographic information system with the purpose of evaluating through the spatial analysis, the infrastructure of telecommunications services, utilities, and socioeconomic variables in order to define the telecommunication service market in Venezuela.

The following roles were defined for the evaluation:

- Sponsors: The evaluation was sponsored by CONATEL and the LISI, which is the research laboratory financed by the government for the development and improvement of MOSCA.
- Evaluator: Persons responsible for carrying out the evaluation; in this case these persons are the authors of the present work, plus two BAs in Geography and a system engineer. Their responsibilities were: design of the evaluation plan; identification of the evaluation objective; identification and definition of the characteristics to be evaluated in the MOSCA adaptation in the GIS; organization of the evaluation purpose; collection and analysis of the results obtained; preparation of the final report and recommendations.
- Users of the model: Organizations that are selecting a GIS through the identification of its weaknesses and strengths.

The project included the evaluation of three FS-GIS (Table 5).

GIS	DESCRIPTION
GRASS	Initially developed by the USA Army Corps of Engineers and subsequently liberated. It is based on UNIX and supports practically all platforms and operative systems. Version 6.0, 2005; language C++; http://grass.itc.it/
Qgis	This software supports the approach with shape files and the storage of spatial data with PostgreSQL/PostGIS. It has a large number of characteristics via a plug-in interface. Version 0.7, 2005, C++; http://qgis.org/
Thuban	Interactive geographic data viewer. It is extensible and multi-platform (GNU/Linux, Windows). Version 1.0, 2005, C++; http://thuban.intevation.org/

Table 5. Description of the GIS evaluated

It is worth mentioning that only three (3) FS-GIS were evaluated, because these are the ones that most precisely meet the four kinds of freedom associated to a FS.

RESULTS ANALYSIS

The results obtained from the measurements based on functionality, maintainability and portability for the three (3) FS-GIS were as follows:

Functionality

Figure 2 shows that Grass meets 7 of the 8 characteristics of the functionality category; this means that this software has at least 75% of the associated metrics; FUN3 (Interoperability) and FUN4 (Safety) were not satisfied. It is worth mentioning that if 6 of the characteristics of the software product are satisfied, then the category of functionality is present in the software product evaluated. However, the other two software products do not reach 75% of the associated metrics; Qgis satisfies only 2 of the 8 functionality characteristics: FUN3 (Interoperability) and FUN8 (Specified). Although Grass satisfies the characteristics of the functionality category, it is weak regarding FUN3 (Interoperability) in comparison with Qgis. Thuban, in turn, fulfills 2 of the functionality characteristics: FUN5 (Correctness) and FUN8 (Specified).

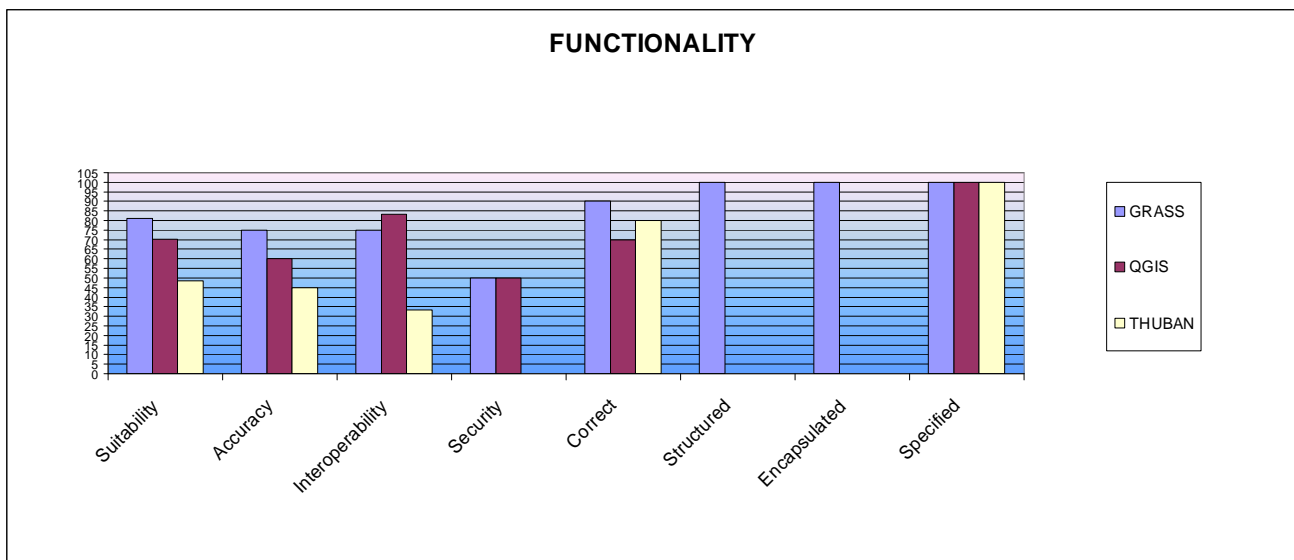


Figure 2. Comparison of FS-GIS in the Functionality category

Maintainability

According to Figure 3, Grass, Qgis and Thuban meet 12 of the 14 characteristics regarding maintainability. The three software products are weak as to MAB4 (Testability) and MAB7 (Encapsulation).

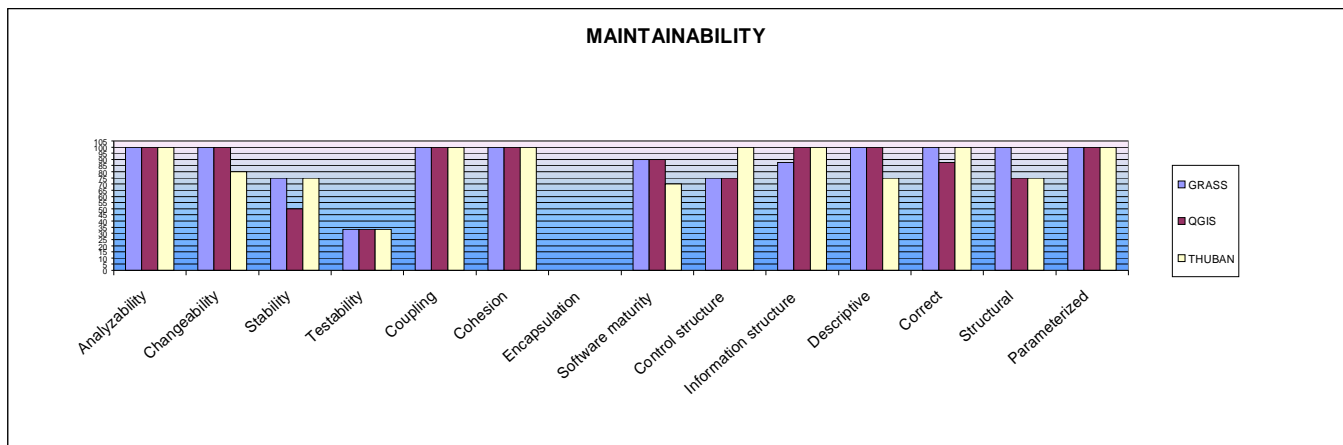


Figure 3. Comparison of the different FS-GISs in the Maintainability category

Portability

Figure 4 shows that Qgis and Thuban comply with 11 of the 12 characteristics of portability, whereas Grass meets 9 of them. It is worth mentioning that Grass is weak in POR2 (Installability), POR4 (Replaceability) and has no POR9 (Specified), being thus less competitive than Thuban and Qgis, which do have POR9 in this category.

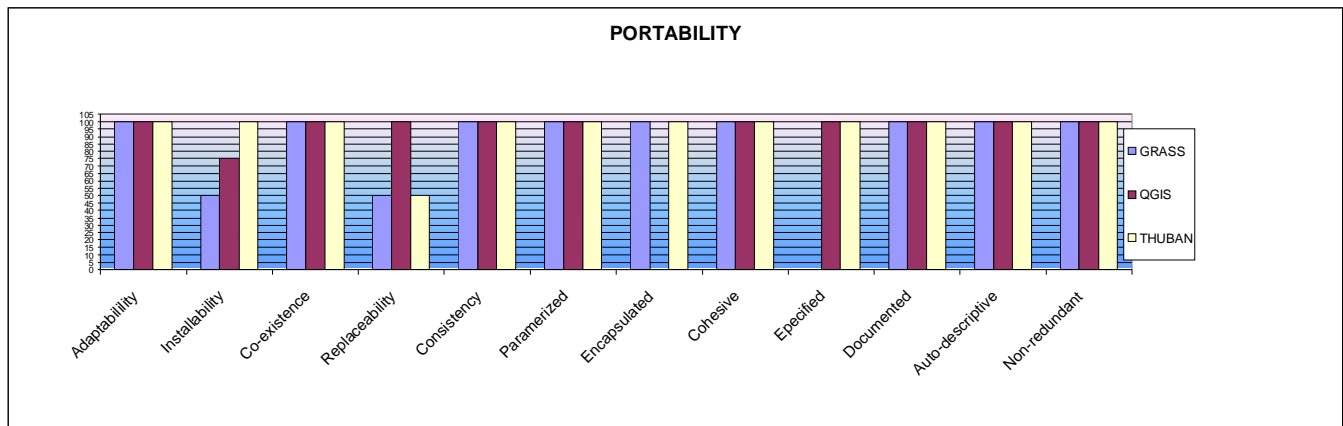


Figure 4. Comparison of the different FS-GIS evaluated in the Portability category

The BAs in Geography considered that the metrics proposed for Functionality were *relevant*. They highlighted the completeness of the classification of the characteristics of this kind of software. They also referred to the significance of developing metrics to evaluate friendliness of the interfaces of the different GISs, which is a key factor for the users of these systems, since it determines how easy the GIS to be selected can be operated. In addition, they pointed out how suitable the model was for the organization. The System Engineer who took part in the evaluation process expressed a similar opinion, and stated that the metrics established for maintainability are relevant, because the software source code is evaluated from instruction lines up to a higher level of modules and dependencies among them. He added that this is a factor to be considered in the development based on the source code of the GIS to be selected, because it reflects the GIS maintainability level. As regards portability, the installability characteristic was also considered relevant. Based on these arguments, the evaluator could conclude that MOSCA is convenient for the organization.

CONCLUSIONS AND RECOMMENDATIONS

This work makes an essential contribution to those organizations that employ GIS and are interested in migrating to FS, because no similar works were found in the consulted literature. Furthermore, it provides a solid evaluation basis, since it can be adapted to the Venezuelan reality. In this work, a quality model for GIS was specified, which is necessary for governmental or other entities that require this type of model to support/justify the selection of the most suitable FS-GIS. The development of this research also helped to improve the Systemic Quality Model (MOSCA). The FS-GIS evaluated were Grass, QGSI, and THUBAN. Grass was the one that reported the best levels of Functionality and Maintainability; with respect to Portability, the three GIS exhibited approximately the same quality levels. It is recommended for future works the evaluation of the process and human perspective dimension in FS-GIS.

REFERENCES

1. Basili, V. R., Caldiera, G., Rombach, H. D. (1994) *Goal Question Metric Paradigm*. In J. J. Marciniak (ed.), *Encyclopedia of Software Engineering*, John Wiley & Sons
2. Baskerville, R. (1999) Investigating Information Systems with Action Research. *Communications of the Association for Information Systems*, Octubre, 2, 19, 1-32.
3. Callaos, N. and Callaos, B. (1996) Designing with Systemic Total Quality, *International Conference on Information Systems*, Orlando, Florida, July, 548-560.
4. Checkland, P. (1993). *Pensamiento de Sistemas, Práctica de Sistemas*. Grupo Noriega Editores.
5. Chorley, R. (1987) *Handling Geographic Information*. HMSO
6. Clarke, M. (1990) *Geographical Information Systems and Model Based Analysis* in Scholten, H.K. and Stillwell, C.H. (eds) *Geographical Information Systems for Urban and Regional Planning*. Kluwer Academic Publishers
7. Cowen, D.J. (1990) *GIS versus CAD versus DBMS: what are the differences?* in Perquet, D. & Marble, D. (eds)
8. Demers, M (2000). *Fundamentals of Geographic Information Systems*. John Wiley & Sons.
9. Dromey, G. (1995) A Model for Software Product Quality, *IEEE Transactions on Software Engineering*, February, Vol 21, Nº 2, 146-162.
10. Dromey, G. (1996) Cornering the Chimera, *IEEE Software*, January, 1996, 33-43.
11. GNU (2005) Sistema operativo GNU - Fundación para el software libre, <http://www.gnu.org/philosophy/free-sw.es.html>
12. Grass (2005) Geographic Resources Analysis Support System, <http://grass.itc.it/>
13. ISO 9126 (2001) ISO/IEC 9126-1:2001(E) Software engineering — Product quality —Part 1: Quality model. First edition,
14. Jiménez, J. (2005) Alternativas de Software Libre a los Sistemas Información Geográfica Comerciales. [Homepage]. Consultado el día 4 de enero del 2006 de la World Wide Web: <http://adm.ing.unibo.it/ADM%20Ingegraf%202005/pdf/COMUNICACIONES%20ACEPTADAS/GIS10.pdf>
15. Kitchenham, B. (1996). Evaluating Software Engineering Methods and Tools. Part 1: The Evaluation Context and Evaluation Methods. *ACM SIGSOFT - Software Engineering Notes*, 21, 1, 11-14.
16. Mendoza, L., Pérez, M., Grimán, A. and Rojas, T. (2002) Algoritmo para la Evaluación de la Calidad Sistémica del Software *Memorias de las 2das. Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC)* 2002) Salvador, Brasil.
17. Mendoza, L; Pérez, M. and Grimán, A. (2005). Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software: *Computación y Sistemas*, 8, 3, 196-217.
18. MCT, Ministerio de Ciencia y Tecnología (2005) *Plan nacional de ciencia y tecnología*. <http://www.mct.gov.ve> Caracas Venezuela
19. Ortega, M., Pérez, M. and Rojas, T. (2000) A Model for Software Product Quality with a Systemic Focus, *4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2000 and The 6th International Conference on Information Systems, Analysis and Synthesis ISAS 2000*, Orlando, Florida, July, 395-401, <http://www.lisi.usb.ve/publicaciones>.

20. Ortega, M.; Pérez, M. and Rojas, T. (2002) A Systemic Quality Model for Evaluating Software Products, *6th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, Florida, Vol. I, 371-376.
21. Ortega, M., Pérez, M. and Rojas, T. (2003) Construction of a Systemic Quality Model for evaluating a Software Product, *Software Quality Journal*, Kluwer Academic Publishers, Julio, 11:3, 219-242
22. Pérez, M. Grimán, A., Mendoza, L. and Rojas, T. (2004) A Systemic Methodological Framework for IS Research. *Proceedings of the Tenth Americas Conference on Information Systems(AMCIS 2004)*, August, USA, 4374-4383.
23. Pérez, M., Rojas T., Mendoza, L. and Grimán, A. (2001) Systemic Quality Model for System Development Process: Case Study, *Seventh Americas Conference on Information Systems – AMCIS*, Boston, Massachusetts, August, 1297-1304, <http://www.lisi.usb.ve/publicaciones>.
24. Pérez, M., Domínguez, K., Mendoza, L. and Grimán, A. (2006) Human Perspective in System Development Quality. *12th Americas Conference on Information Systems (AMCIS)*. Acapulco, México. Agosto.
25. Quantum (2005) Quantum GIS (QGIS), <http://qgis.org/>
26. Sánchez, E. (1999) Una metodología sistémica para la implantación de SIG. *ESRI Latin America User Conference Proceedings*. April 12–16, Venezuela, <http://gis.esri.com/library/userconf/latinproc99/ponencias/ponencia13.html>.
27. Thuban (2005) Interactive Geographic Data Viewer, <http://thuban.intevation.org/>