

Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software *Prototype of Software Quality Systemic Model (SQSM)*

Luis E. Mendoza, María A. Pérez y Anna C. Grimán

Universidad Simón Bolívar
Departamento de Procesos y Sistemas
Apto. Postal 89000, Caracas, 1080 A
Caracas, Venezuela
e-mail: lmendoza, movalles, agriman {@usb.ve}

Artículo recibido en junio 06 de 2002; aceptado en diciembre 16, 2004

Resumen

A nivel mundial se está generando un conjunto de modelos para medir la calidad de los Sistemas de Software (SS), los cuales representan las características para que estos sean productos/servicios de calidad que compitan en un mercado abierto y global. Los modelos se deben formular con base a las características competitivas para cada tipo de SS, considerando la alta participación humana en el proceso de desarrollo de software. Partiendo de estas premisas, esta investigación propone el prototipo de MOdelo Sistémico de CALidad (MOSCA) para evaluar la calidad de los SS, integrando el modelo de Calidad del Producto y el modelo de Calidad del Proceso de Desarrollo, soportado en los conceptos de la Calidad Total Sistémica. MOSCA fue probado en dos empresas venezolanas desarrolladoras de SS, utilizando el Método "Análisis de Características por Caso de Estudio", indicado por la metodología DESMET.

Palabras clave: Modelo de calidad, Evaluación de producto de software, Evaluación de proceso de software, Calidad de software, Calidad sistémica, Caso de estudio.

Abstract

In the world-wide context a set of models to assess Software Systems quality are been generated, these models represent the characteristics to be met by products/services in order to compete in a global and open market. The models must be formulated based on the competitive characteristics for every kind of Software System, considering the high human participation in software development process. Starting from these premises, this research proposes the Prototype of Software Quality Systemic Model (SQSM) –in Spanish: MOdelo Sistémico de CALidad (MOSCA) – to evaluate Software Systems quality, it integrates the Product Quality model and the Development Process Quality model, supported by the concepts of the Systemic Total Quality. MOSCA was proven in two Venezuelan software systems developing companies, using the Feature Analysis Case Study Method, indicated by DESMET methodology.

Keywords: Quality model, Software product evaluation, Software process evaluation, Software quality, Systemic quality, Case study.

1 Introducción

La calidad es un término que ha adquirido gran relevancia con el paso del tiempo, ya que es considerada como uno de los principales activos con los que cuenta un país para mejorar su posición competitiva global (Ivanisevich et al., 1997). Para conseguir una buena calidad del software es esencial establecer un programa de medidas a tomar con respecto a los suministradores. Es también importante utilizar los modelos y métodos apropiados para controlar el proceso de desarrollo del mismo.

A la hora de definir la calidad de los SS se debe diferenciar entre la calidad del producto software y la calidad del proceso de desarrollo de éste (calidad de diseño y fabricación). No obstante, las metas que se establezcan para la calidad del producto van a determinar los objetivos del proceso de desarrollo, ya que la calidad del primero va a depender, entre otros aspectos, de éstos. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto.

Según Pressman (2002) la calidad del software es "*la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente*".

La ausencia de defectos, la aptitud para el uso, la seguridad, la confiabilidad y la reunión de especificaciones son elementos que están involucrados en el concepto de calidad del software. Sin embargo, la calidad del software debe ser construida desde el comienzo, no es algo que puede ser añadido después (Humphrey, 1997).

El presente artículo tiene por objetivo mostrar la integración del modelo de Calidad del Producto (Ortega et al., 2001) y el modelo de Calidad del Proceso de Desarrollo (Pérez et al., 2001), ambos desarrollados en el Laboratorio de Investigación

en Sistemas de Información (LISI) de la Universidad Simón Bolívar de Venezuela, en un prototipo de Modelo Sistémico de Calidad (MOSCA) para la Medición de la Calidad de los SS, soportado por los conceptos de la Calidad Total Sistémica (Callaos y Callaos, 1996; Rojas y Pérez, 1995), los cuales establecen una relación sistémica entre el producto y el proceso.

Además de esta Introducción, el artículo está conformado por otras 7 secciones. La sección 2 presenta el concepto de calidad sistémica, ya que el mismo constituye el concepto más importante en el que se soporta el prototipo de **Modelo Sistémico de Calidad (MOSCA)** para la Medición de la Calidad de los Sistemas de Software. Posteriormente, en las secciones 3 y 4, se presentan los modelos de calidad del producto y del proceso, respectivamente, que constituyen las principales partes constitutivas del Modelo. En la sección 5 se presenta MOSCA con todos sus niveles, así como el análisis de la interrelación entre proceso y producto, y el algoritmo que permite su aplicación. Seguidamente, en las secciones 6 y 7 se especifican los pasos seguidos en la evaluación de MOSCA, según el método sugerido por DESMET (Kitchenham et al, 1997) *Análisis de las Características por Caso de Estudio* y se muestra el análisis de los resultados obtenidos de la evaluación efectuada. Finalmente, en la sección 8, se enumeran las conclusiones alcanzadas y los trabajos futuros.

2 Matriz de Calidad Global Sistémica

Callaos y Callaos (1996) proponen un concepto de calidad del software en el cual están involucrados tanto características internas como el contexto organizacional, lo que genera un enfoque sistémico del concepto de Calidad del Software. Este enfoque es considerado también por Dromey (1996) y, particularmente reforzado por Voas (1999), cuando se refiere al Triángulo de la Certificación de la Calidad del Software.

En todos los enfoques de calidad, incluyendo el de la calidad del software, la organización necesita ser considerada como organización orgánica más bien que mecánica. Usa sus recursos, las ideas y los esfuerzos para transformar las entradas en mercancías y servicios de la calidad. Así pues, las características y las tecnologías necesarias para lograr este tipo de acercamiento deben ser determinadas. Esta es la razón por la cual el proyecto del desarrollo del software se considera el elemento de la organización a través de el cual se maneja el desarrollo del software (Jacobson et al., 1999). De acuerdo con los autores Yourdon (1992), Jacobson et al. (1999), Bass et al. (2003) y Pressman (2002), un proyecto de software consiste en los siguientes elementos:

- Entradas: requisitos (necesidades, calidad), recursos (tiempo, costo, gente), objetivos, reglas de negocio;
- Proceso: workflow, paradigma de la tecnología de dotación lógica, proceso cultural de la determinación, proceso de aprendizaje de organización;
- Salidas: Producto, metas satisfechas;

Esto incluye el proceso y el producto que tienen un impacto en la organización así como el ambiente de negocio (Yourdon, 1992).

Así pues, el proyecto se puede considerar como el sistema que define el contexto que modela la calidad del proceso y calidad del producto respectivamente. Tal como indica Humphrey (1997), *la calidad del proceso garantiza la calidad del producto y consecuentemente no se pueden desligar estas dos calidades*.

Este es el concepto considerado en el enfoque sistémico propuesto por Callaos y Callaos (1996). La definición de la calidad sistémica de los sistemas se basa en a Matriz de Calidad Global mostrada en la Figura 1, la cual consta de cuatro (4) tipos de calidades y se basa en las dos perspectivas: Proceso y Producto. Estos cuatro (4) tipos de calidades son las consideraciones de los Aspectos Internos y Contextuales de ambas perspectivas. Es decir, Aspectos Internos y Contextuales del Producto y Aspectos Internos y Contextuales del Proceso, considerando además, los puntos de vista del Cliente y del Usuario.

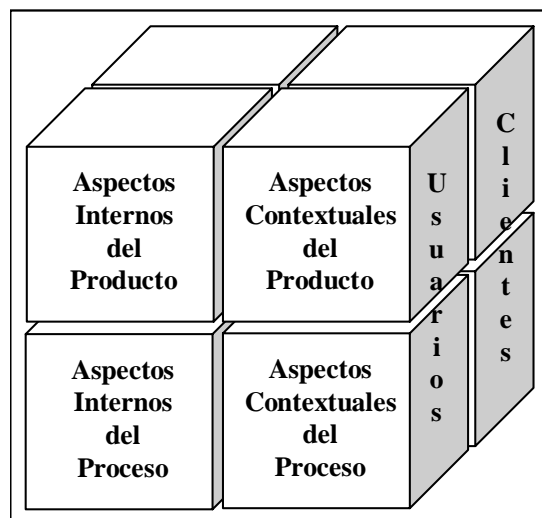


Fig. 1. Matriz de Calidad Global Sistémica. Adaptada de (Callaos y Callaos, 1996)

Esta división se justifica en un sentido, porque un proyecto incluye tanto los aspectos contextuales como los aspectos internos, y en el otro, porque el Sistema concebido (el producto) es diferente al Sistema de las actividades humanas (el proceso) mediante el cual el Sistema-Producto es diseñado (Callaos y Callaos, 1996). El aporte del enfoque de calidad sistémica es que permite balancear las diferentes perspectivas de la calidad del software (Proceso y Producto).

Rojas y Pérez (1995) definen cada uno de los componentes de la matriz de la siguiente manera:

- **Aspectos Internos del Producto:** Son determinados por actividades de diseño interno y programación, ya que un producto eficiente es conseguido cuando se aplican las prácticas correctas de diseño físico y programación.
- **Aspectos Contextuales del Producto:** Son determinados por las actividades de identificación de requerimientos, diseño de interfaces y diseño general de la red (ubicación de puntos), debido a que la misma está relacionada *con la adecuación y confort del usuario*.
- **Aspectos Internos del Proceso:** Están asociados con las actividades de *gerencia de proyectos*, las cuales incluyen el cumplimiento de fechas de entrega, aumento de la productividad y ahorro de recursos.
- **Aspectos Contextuales del Proceso:** Se relacionan con las actividades generales de gerencia, tales como liderazgo, administración de cambio, relaciones humanas y grupales, ya que las mismas conducen a establecer buenas relaciones entre los integrantes del equipo responsable del desarrollo de SS.

Según Callaos y Callaos (1996), la calidad global no es la suma de las calidades parciales, sino el compromiso entre todo el conjunto de calidades que conlleve a un óptimo global con cierto sacrificio de los óptimos parciales. Resumiendo, se tiene que la calidad del software no es algo que depende de una sola característica en particular, sino que obedece al compromiso de todas sus partes. A su vez, permite englobar las dos tendencias actuales de modelos de calidad; estas son calidad del producto (software) y calidad del proceso, con un enfoque sistémico. Esta visión de la calidad es la que se toma como eje de la presente investigación.

3 Modelo de Calidad del Producto de Software con un Enfoque Sistémico

El modelo planteado para la calidad del producto (Ortega et al., 2000) se basa en el Modelo de Calidad Sistémica de Callaos y Callaos. Los componentes que son tomados en cuenta en el modelo de calidad del producto son los siguientes:

- Los aspectos internos y contextuales del producto como calidad parcial del modelo de Calidad Sistémica de Callaos.
- Las características de calidad del modelo de Dromey (1996) y el estándar internacional ISO/IEC 9126 (JTC 1/SC 7, 1991): Eficiencia, Fiabilidad, Funcionalidad, Mantenibilidad, Portabilidad y Usabilidad.
- La relación utilizada en el modelo de McCall (Gillies, 1977) entre los atributos y calidad de las métricas.

Este modelo es mostrado en la Figura 2 y plantea, sobre la base de las 6 características de calidad del estándar internacional

ISO/IEC 9126 (JTC 1/SC 7, 1991), un conjunto de sub-características y métricas asociadas que miden la calidad de un producto de software con un enfoque sistémico. Estas métricas (249 en total) propuestas (Ortega et al., 2000), son el resultado del análisis de las cuatro dimensiones de la Matriz de Calidad Sistémica de Callaos que están asociadas a cada característica. Aunque el modelo propuesto (Ortega et al., 2000) soporta el enfoque sistémico propuesto por Callaos y Callaos (1996), el modelo se orienta únicamente a la medición de la calidad del producto de software omitiendo la calidad del proceso. A pesar de ello, el modelo menciona algunas sub-características de cada métrica asociada al proceso de desarrollo, más no trata esta dimensión en su totalidad.

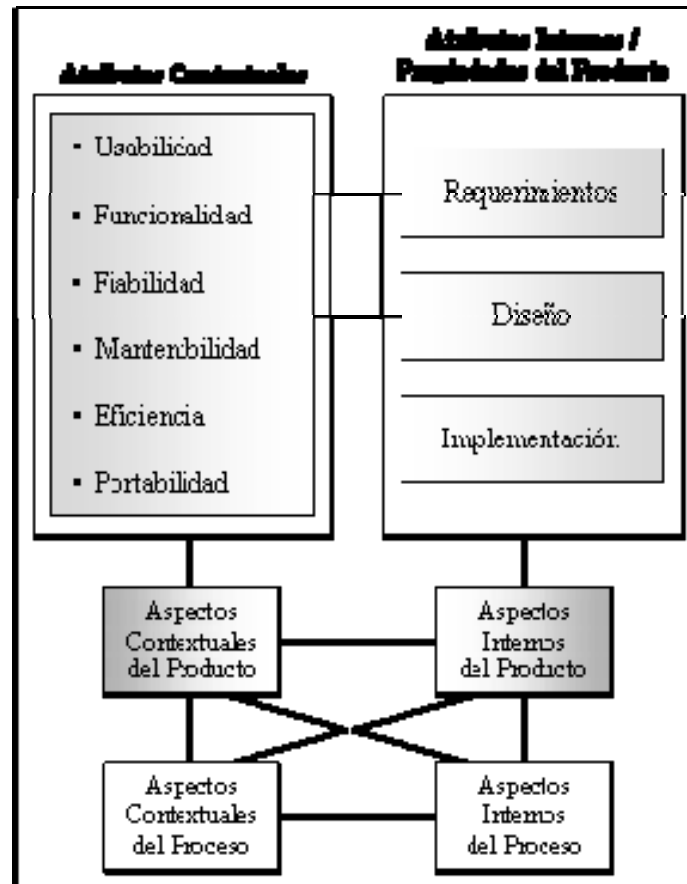


Fig. 2. Modelo de Calidad del Producto de Software (Ortega et al., 2000)

Las 6 características del modelo, aunados a cada una de las sub-características, hacen del modelo un instrumento de medición de gran valor ya que cubre todos los aspectos necesarios e imprescindibles para medir directamente la calidad del producto de software; esto se debe a su similitud con el estándar internacional ISO/IEC 9126. Cabe acotar que, según el método propuesto por Ortega y sus colaboradores para aplicar el modelo, no todas las métricas se usan en todos los casos, sino que el modelo debe ser instanciado dependiendo de las características del producto deseado.

4 Modelo de Calidad del Proceso de Software con un Enfoque Sistémico

Al igual que en el modelo de calidad del producto explicado anteriormente, éste también está relacionado con el modelo de Calidad Sistémica de Callaos, con el objetivo específico de soportar el enfoque de Calidad Global Sistémica para las dos dimensiones asociadas al proceso –interna y contextual- (Pérez et al., 2001). Este modelo y sus características están basados en la adaptación del modelo del proceso SPICE –conocido también como ISO/IEC 15504- (JTC 1/SC 7, 1999) al modelo de calidad sistémica (Pérez et al., 2001).

En la Figura 3 se ilustra el modelo del proceso, el cual presenta una estructura compleja que está definida por niveles, donde cada nivel superior esta conformado por elementos del nivel inferior (Pérez et al., 2001).

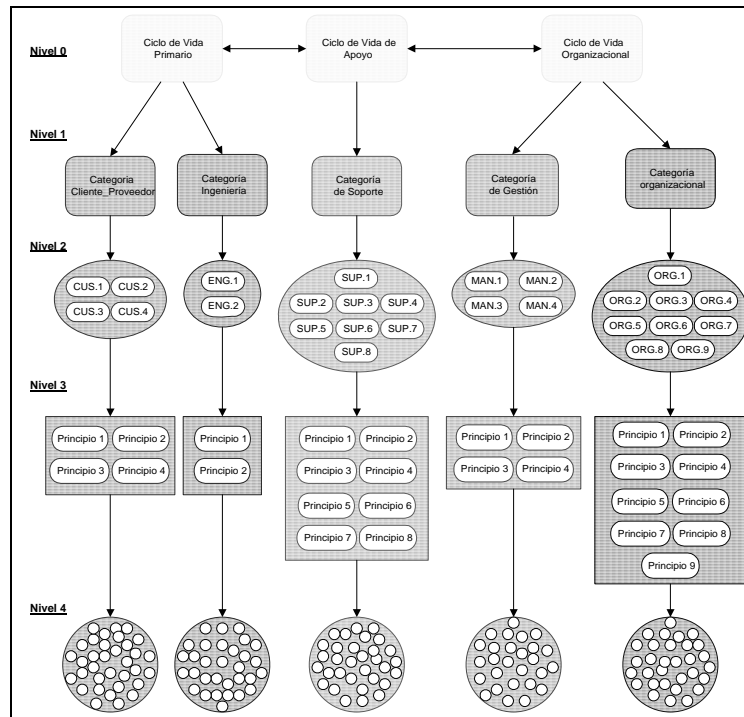


Fig. 3. Estructura del Modelo de la Calidad del Proceso de Desarrollo de Software (Pérez et al., 2001)

Este modelo consta de 4 niveles (ver Figura 3):

- **Nivel 0: Ciclos de Vida:** Ciclo de Vida Primario, Ciclo de Vida de Apoyo y Ciclo de Vida Organizacional.
- **Nivel 1: Categorías de Procesos:** Categoría Cliente-Proveedor -CUS-, Categoría Ingeniería -ENG-, Categoría de Soporte -SUP-, Categoría de Gestión -MAN- y Categoría Organizacional -ORG-.
- **Nivel 2: Procesos.** Cada categoría contiene un conjunto de procesos característicos, los cuales definen las áreas claves a satisfacer para lograr, asegurar, mantener y controlar la calidad.
- **Nivel 3: Principios.** Cada proceso tiene asociado un Principio (P), el cual se define como característica abstracta y genérica de la organización que sirve de indicador para determinar los niveles de calidad en el desarrollo de los Sistemas.
- **Nivel 4: Prácticas Bases.** Las Prácticas Bases son un conjunto de directrices a ser ejecutadas por la organización para lograr alcanzar un principio; donde cada una de estas Prácticas Bases apoya a una o a las dos dimensiones de la Matriz de Calidad Sistémica. Cabe destacar, que con respecto al modelo de proceso SPICE (JTC 1/SC 7, 1999), se aumentaron notablemente el número de prácticas bases (PB) con la finalidad de mantener un equilibrio entre lo interno y o contextual del proceso (Pérez et al., 2001). Cada una de las Prácticas Bases que propone el modelo (430 en total), pueden ser medidas en las organizaciones a través de un cuestionario.

El modelo de calidad del proceso (Pérez et al., 2001), aunque soporta el enfoque sistémico propuesto por Callaos y Callaos, está orientado únicamente a la medición de la calidad del proceso de software, dejando de un lado la calidad del producto.

Hasta ahora se han descrito los modelos que permiten evaluar la calidad del producto de software y la calidad del proceso de desarrollo de manera aislada; es decir, mediante la implementación de modelos distintos que miden cada una de las calidades por separado aunque inspirados ambos en el modelo de Calidad Sistémica. Por ello, surge la necesidad de crear un modelo que integre ambos, ya que como indica Humphrey (1997), la calidad del proceso garantiza la calidad del producto y consecuentemente no se pueden desligar estas dos calidades. Tener modelos separados capaces de medir individualmente la calidad de un producto o de un proceso de software no garantiza la calidad total sistémica de un SS. Esto se debe a que la naturaleza de los sistemas no puede ser dividida en partes, sino que debe existir una interdependencia y colaboración entre las partes para que el mismo sea visto como un todo; se requiere entonces un modelo capaz de unificar (integrar) la manera de medir y adaptar las métricas de ambos.

Por lo anterior, surge la motivación principal del presente estudio, en el cual se propone integrar ambos modelos para poder garantizar la medición de la calidad sistémica global en las organizaciones.

5 Formulación del Modelo Sistémico de Calidad (MOSCA)

Partiendo de los dos modelos descritos anteriormente, se formula el prototipo de MOSCA (ver Figura 4), constituido por cuatro (4) niveles, los cuales son explicados a continuación:

Nivel 0: Dimensiones. Aspectos Internos del proceso, Aspectos Contextuales del proceso, Aspectos Internos del producto y Aspectos Contextuales del producto son las cuatro dimensiones propuestas en el prototipo de modelo. Sólo un balance y una buena interrelación entre ellas garantizan la calidad Sistémica global de una organización.

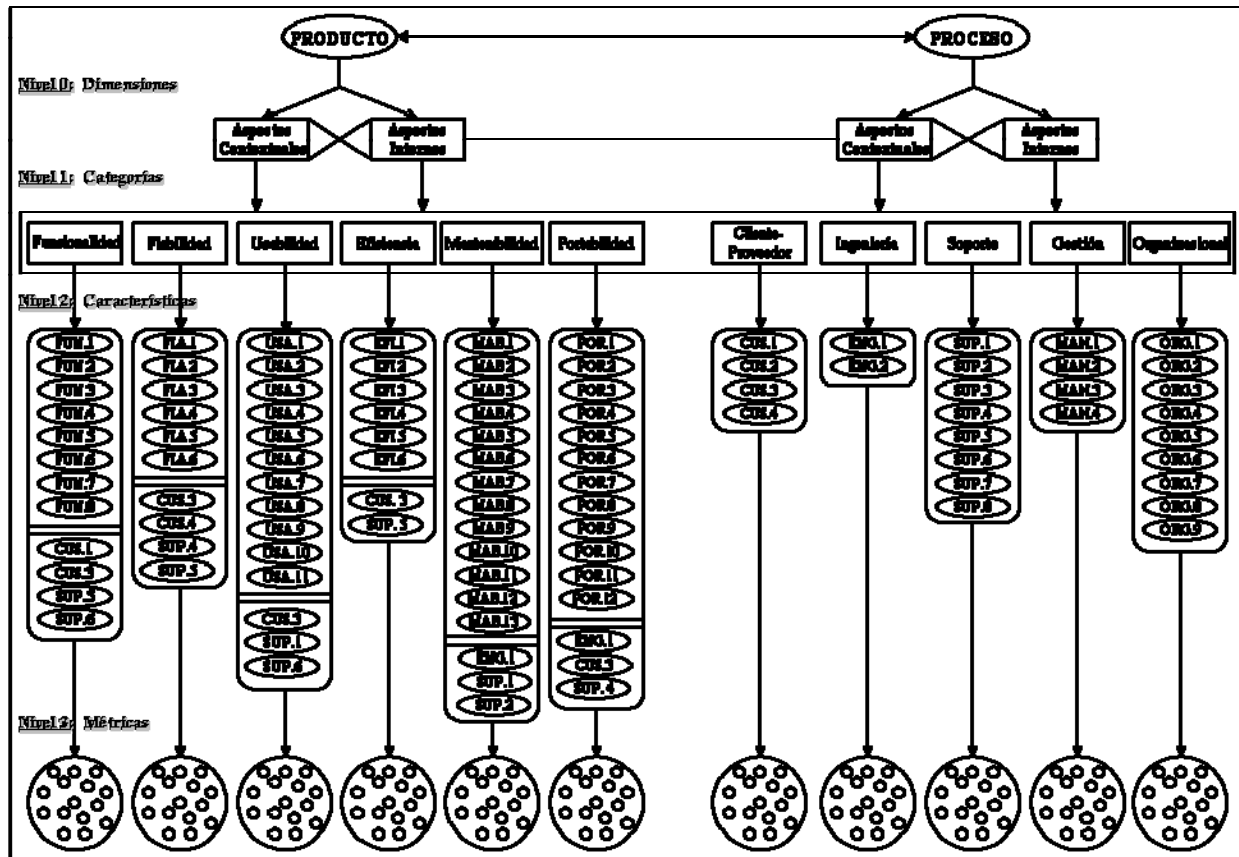


Fig. 4. Diagrama del prototipo final de MOSCA

Nivel 1: Categorías. Se contemplan once (11) categorías: seis (6) pertenecientes al producto (ver Tabla 3) y las otras cinco (5) al proceso de desarrollo (ver Tabla 4). Esta división no implica un desligamiento entre ellas, simplemente se realiza para identificar a que sector o sub-modelo pertenecen.

Categoría del Producto	Definición
Funcionalidad (FUN)	Capacidad del producto del software para proveer funciones que cumplan con necesidades específicas o implícitas, cuando el software es utilizado bajo ciertas condiciones.
Fiabilidad (FIA)	Capacidad del producto de software para mantener un nivel especificado de rendimiento cuando es utilizado bajo condiciones especificadas.
Usabilidad (USA)	Capacidad del producto de software para ser atractivo, entendido, aprendido y utilizado por el usuario bajo condiciones específicas.
Eficiencia (EFI)	Capacidad del producto de software para proveer un rendimiento apropiado, relativo a la cantidad de recursos utilizados, bajo condiciones específicas.
Mantenibilidad (MAB)	Capacidad del producto para ser modificado.
Portabilidad (POR)	Capacidad del producto de software para ser transferido de un ambiente a otro.

Tabla 3. Categorías del sub-modelo del producto

Nivel 2: Características. Cada categoría tiene asociado un conjunto de características, las cuales definen las áreas claves a satisfacer para lograr, asegurar y controlar la calidad tanto en el producto como en el proceso. Entre las características asociadas a cada categoría del producto, se proponen una serie de características del proceso (ver Figura 4). Esto se debe, a que algunas características de la calidad del proceso, impactan directamente en las categorías del producto al igual que ciertas características de la calidad del producto definen categorías del proceso. Esto ayuda a precisar que si una vez medidas las características asociadas a una categoría en particular del producto, arroja resultados no deseados, se pueden analizar las características de la calidad del proceso asociadas a esa categoría del producto para encontrar las posibles causas.

Categoría del Proceso	Definición
Cliente – Proveedor (CUS)	Esta conformada por 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)	Consiste en procesos que directamente especifican, implementan o mantienen el producto de software, su relación con el Sistema y su documentación.
Soporte (SUP)	Consta de procesos que pueden ser empleados por cualquiera de los procesos (incluyendo a los de soporte) en varios niveles del ciclo de vida de adquisición.
Gestión (MAN)	Abarca los procesos que contienen prácticas genéricas, que pueden ser utilizadas por cualquier personal que dirija algún tipo de proyecto o proceso.
Organizacional (ORG)	Agrupar los 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.

Tabla 4. Categorías del sub-modelo del proceso.

En las Tablas 3 y 4 se muestran todas las características del modelo, agrupadas por cada una de las dimensiones de la Matriz Global Sistémica (Callaos y Callaos, 1993) y acompañadas con la clave única que las identifica.

Categoría	Características	
	Aspectos Contextuales del Producto	Aspectos Internos del Producto
Funcionalidad (FUN) Total de métricas: 46	FUN 1. Ajuste a los propósitos (16) FUN 2. Precisión (10) FUN 3. Interoperabilidad (7) FUN 4. Seguridad (2)	FUN 5. Correctitud (8) FUN 6. Estructurado (1) FUN 7. Encapsulado (1) FUN 8. Especificado (1)
	Sub-total de métricas: 35	Sub-total de métricas: 11
Fiabilidad (FIA) Total de métricas: 32	FIA 1. Madurez (17) FIA 2. Tolerancia a fallas (1) FIA 3. Recuperación (4)	FIA 4. Correctitud (8) FIA 5. Estructurado (1) FIA 6. Encapsulado (1)
	Sub-total de métricas: 22	Sub-total de métricas: 10
Usabilidad (USA) Total de métricas: 38	USA 1. Facilidad de comprensión (5) USA 2. Capacidad de Aprendizaje (9) USA 3. Interfaz Gráfica (5) USA 4. Operabilidad (13) USA 5. Conformidad con los estándares	USA 6. Completo (1) USA 7. Consistente (1) USA 8. Efectivo (1) USA 9. Especificado (1) USA 10. Documentado (1) USA 11. Auto-descriptivo (1)
	Sub-total de métricas: 32	Sub-total de métricas: 6
Eficiencia (EFI) Total de métricas: 10	EFI 1. Comportamiento del tiempo (2) EFI 2. Utilización de recursos (4)	EFI 3. Efectivo (1) EFI 4. No redundante (1) EFI 5. Directo (1) EFI 6. Utilizado (1)
	Sub-total de métricas: 6	Sub-total de métricas: 4
Mantenibilidad (MAB) Total de métricas: 79	MAB 1. Capacidad de análisis (2) MAB 2. Facilidad de Cambio (7) MAB 3. Estabilidad (4) MAB 4. Capacidad de prueba (3)	MAB 5. Acoplamiento (1) MAB 6. Cohesión (1) MAB 7. Encapsulado (1) MAB 8. Madurez del Software (17) MAB 9. Estructura de Control (4) MAB 10. Estructura de Información (9) MAB 11. Descriptivo (14) MAB 12. Correctitud (8) MAB 13. Estructural (5) MAB 14. Modularidad (3)

	Sub-total de métricas: 16	Sub-total de métricas: 63
Portabilidad (POR) Total de métricas: 44	POR 1. Adaptabilidad (9) POR 2. Capacidad de Instalación (4) POR 3. Co-existencia (2) POR 4. Capacidad de reemplazo (2)	POR 5. Consistente (1) POR 6. Parametrizado (3) POR 7. Encapsulado (1) POR 8. Cohesivo (1) POR 9. Especificado (1) POR 10. Documentado (1) POR 11. Auto-descriptivo (1) POR 12. No redundante (1) POR 13. Auditoria (6) POR 14. Manejo de la Calidad (3)
	Sub-total de métricas: 17	Sub-total de métricas: 19
	Calidad de los Datos (8) -abarca las dos dimensiones-	

Tabla 5. Distribución de las Características y métricas para medir la Calidad Sistémica del Producto de Software

Categoría	Características	
	Aspectos Contextuales del Proceso	Aspectos Internos del Proceso
Cliente – Proveedor (CUS) Total de métricas: 57	CUS 1. Adquisición del Sistema o producto de Software (24) CUS 3. Determinación de Requerimientos (20)	CUS 2. Suministro (8) CUS 4. Operación (5)
	Sub-total de métricas: 44	Sub-total de métricas: 13
Ingeniería (ING) Total de métricas: 29	ENG 1. Desarrollo (12)	ENG 2. Mantenimiento de Software y Sistemas (17)
	Sub-total de métricas: 12	Sub-total de métricas: 17
Soporte (SUP) Total de métricas: 130	SUP 3. Aseguramiento de la Calidad (17) SUP 6. Revisión Conjunta (14) SUP 7. Auditoria (15) SUP 8. Resolución de Problemas (11)	SUP 1. Documentación (9) SUP 2. Gestión de Configuración (12) SUP 4. Verificación (6) SUP 5. Validación (6) SUP 6. Revisión Conjunta (14) SUP 7. Auditoria (15) SUP 8. Resolución de Problemas (11)
	Sub-total de métricas: 57	Sub-total de métricas: 73
Gestión (MAN) Total de métricas: 91	MAN 1. Gestión (14) MAN 3. Gestión de Calidad (10) MAN 4. Gestión del Riesgo (12)	MAN 1. Gestión (14) MAN 2. Gestión de Proyecto (19) MAN 3. Gestión de Calidad (10) MAN 4. Gestión del Riesgo (12)
	Sub-total de métricas: 36	Sub-total de métricas: 55
Organizacional (ORG) Total de métricas: 123	ORG 1. Lineam. Organizacionales (14) ORG 2. Gestión de Cambio (10) ORG 5. Mejoramiento del Proceso (16) ORG 8. Medición (11) ORG 9. Reuso (12)	ORG 3. Establecimiento del Proceso (11) ORG 4. Evaluación del Proceso (9) ORG 5. Mejoramiento del Proceso (16) ORG 6. Gestión de RRHH (16) ORG 7. Infraestructura (8)
	Sub-total de métricas: 63	Sub-total de métricas: 60

Tabla 6. Distribución de las Características y métricas para medir la Calidad Sistémica de Proceso de Desarrollo

Nivel 3: Métricas

Para cada característica se propone una serie de métricas utilizadas para medir la calidad sistémica. Dada la cantidad de métricas asociadas a cada una de las características que conforman MOSCA (ver Tablas 5 y 6) -679 en total-, éstas no serán presentada en su totalidad en el presente artículo; sin embargo, en la Tabla 7 se presenta un ejemplo de las métricas formuladas para el sub-modelo de producto y para el sub-modelo de proceso de desarrollo (N/A = No Aplica; N/S = No Sabe).

Métrica	Número de componentes de acceso a base de datos.		
Rangos para la métrica	≥ 8 → 5	5 – 7 → 4	3 - 4 → 3
	1 - 2 → 2	0 → 1	

Tabla 7. Ejemplo de las métricas que conforman el sub-modelo del producto del modelo MOSCA

Métrica	Generación de documentación en concordancia con los estándares y políticas establecidos.			
Rangos para la métrica	SI <input type="checkbox"/>	NO <input type="checkbox"/>	N/A <input type="checkbox"/>	N/S <input type="checkbox"/>

Tabla 8. Ejemplos de las métricas que conforman el sub-modelo del proceso del modelo MOSCA

5.1 Características del Proceso de Desarrollo de Software que Impactan Directamente en las Categorías del Producto

Según Humphrey (1997), la calidad del producto final está fuertemente influenciada por la calidad del proceso de desarrollo del mismo. Sin embargo, cada categoría del modelo del producto pareciera tener una dependencia específica de ciertas características del modelo del proceso, lo que implica que si estas características se cumplieron a cabalidad durante el proceso de desarrollo entonces la categoría del producto debiera poseer los requerimientos mínimos de calidad.

En esta sección se analizan la influencia que tienen las características del proceso en las categorías del sub-modelo del producto de software y, a su vez, se justifica el impacto que tienen sobre cada una de ellas. En las Tablas 9 a 14, presentadas a continuación, se explican cada una de las características del proceso de desarrollo que influyen sobre cada una de las categorías del producto.

1) Características del proceso que influyen en la Funcionalidad del producto de software:

Características del Proceso	Descripción
CUS 1 Proceso de adquisición de sistema o producto de software	Este proceso influye en la Funcionalidad del producto de software, ya que es necesario para obtener un producto que satisfaga las necesidades del cliente, garantizando la aceptación del mismo. En caso de que este proceso no se realice adecuadamente, no se contaría con una especificación que permita evaluar el cumplimiento de las necesidades del cliente.
CUS 3 Proceso de Determinación de Requerimientos	Este proceso también influye en la Funcionalidad, ya que se encarga de velar porque las necesidades y los requerimientos del cliente sean satisfechos a través del proceso de desarrollo.
SUP 5 Proceso de Validación	En el proceso de Validación se debe confirmar que los requerimientos para un uso específico del producto del sistema (software) sean satisfechos; es decir, este proceso garantiza que la Funcionalidad del producto de software se cumpla.
SUP 6 Proceso de Revisión Conjunta	Este proceso afecta directamente a la Funcionalidad del producto, ya que se refiere al mantenimiento de un común entendimiento con el cliente sobre el progreso del proceso o del proyecto, en contraste con los objetivos del contrato. De esa manera, si se realiza correctamente dicha característica, el producto resultante se aproximará bastante a los requerimientos planteados por el cliente.

Tabla 9. Características del proceso que impactan sobre la categoría Funcionalidad

2) Características del proceso que influyen en la Fiabilidad del producto de software:

Características del Proceso	Descripción
CUS 4 Proceso de Operación	Este proceso influye en la Fiabilidad del producto de software, ya que se desea obtener un producto que cumpla con un nivel especificado de rendimiento cuando se utiliza bajo condiciones específicas. Este proceso asegura la correctitud de la operación del software durante el uso al que fue destinado y en el ambiente donde fue instalado; es decir, especifica y hace seguimiento a las condiciones de operación del producto. Asegura completitud y consistencia.
SUP 4 Proceso de Verificación	Este proceso afecta a la Fiabilidad del producto ya que se desea confirmar que cada producto trabajado y/o servicio de un proceso o proyecto refleje apropiadamente los requerimientos especificados. El proceso de Verificación asegura que el producto de software tenga todos los componentes necesarios para cumplir con los requerimientos para los cuales fue diseñado.
SUP 5 Proceso de Validación	Para que un producto sea fiable, se deben confirmar que los requerimientos para el uso específico del producto del sistema (software) sean satisfechos. Si este proceso se realiza adecuadamente, se contará con un punto de control que constate que el producto de software funcione correctamente para lo que fue diseñado, ya que se confirma la satisfacción de los requerimientos en el proceso de desarrollo.
CUS 3 Proceso de Determinación de Requerimientos	Este proceso también influye en la Fiabilidad, ya que se encarga de velar la satisfacción de las necesidades y los requerimientos que debe tener el producto. Gracias a éste, se cuenta con una especificación de los requerimientos que el producto de software debe cumplir.

Tabla 10. Características del proceso que impactan sobre la categoría Fiabilidad

3) Características del proceso que influyen en la Usabilidad del producto de software:

Características del Proceso	Descripción
SUP 1 Proceso de Documentación	Para que un producto de software cumpla con la característica de Usabilidad, debería estar correctamente documentado ya que facilita de esa manera el uso y el aprendizaje del mismo. En especial, se refiere a la creación de manuales y ayuda en línea que reflejen de la manera más entendible posible, todas las funciones del producto software
CUS 3 Proceso de Determinación de Requerimientos	Este proceso también influye en la Usabilidad, ya que se encarga de velar la satisfacción de las necesidades y requerimientos del cliente a través del proceso de desarrollo. Con esto se tiene que el producto de software realizará exactamente lo que el cliente desee y por ello el mismo será atractivo y fácil de usar para el usuario final.
SUP 6 Proceso de Revisión Conjunta	Esta característica se refiere al mantenimiento de un común entendimiento con el cliente sobre el progreso del proceso o proyecto, en contraste con los objetivos del contrato. Influye en la Usabilidad, ya que el cliente estará siempre al tanto del cómo será la interacción con el producto.

Tabla 11. Características del proceso que impactan sobre la categoría Usabilidad

4) Características del proceso que influyen en la Eficiencia del producto de software:

Características del Proceso	Descripción
CUS 3 Proceso de Determinación de Requerimientos	Este proceso influye en la Eficiencia del producto de software, ya que a través de él se podrán identificar las limitaciones en el uso de los recursos del SS para que funcione bajo condiciones específicas. De esta manera, el producto usará apropiadamente los recursos.
SUP 5 Proceso de Validación	Para que un producto sea Eficiente, se deben confirmar los requerimientos específicos que debería tener el producto. Si este proceso se realiza adecuadamente, el producto de software no utilizará más recursos de los necesarios, ya que los mismos fueron confirmados en el proceso de desarrollo.

Tabla 12. Características del proceso que impactan sobre la categoría Eficiencia

5) Características del proceso que influyen en la Mantenibilidad del producto de software:

Características del Proceso	Descripción
ENG 1 Proceso de Desarrollo	Este proceso se refiere al conjunto de requerimientos de un producto de software funcional o SS que deben ser transformados para que se satisfagan las necesidades expresadas por el cliente. En caso de que este proceso no se cumpliera siguiendo los estándares de documentación prefijados en la organización desarrolladora, el software final sería muy difícil de modificar, corregir o mejorar ante cualquier cambio en el ambiente o petición expresa del cliente y por ello el sistema no cumpliría con la característica de Mantenibilidad.
SUP 1 Proceso de Documentación	Este proceso influye en la Mantenibilidad de un producto de software ya que tiene como fin desarrollar, mantener y actualizar, durante todo el desarrollo, los documentos que registran información sobre su evolución.
SUP 2 Proceso de Gestión de Configuración	Esta característica establece y mantiene la integridad de todos los productos elaborados durante el proceso. Influye en la Mantenibilidad del producto de software, ya que en la medida que el producto mantenga su integridad, estará en capacidad de ser modificado con mayor facilidad.

Tabla 13. Características del proceso que impactan sobre la categoría Mantenibilidad

6) Características del proceso que influyen en la Portabilidad del producto de software:

Características del Proceso	Descripción
CUS 3 Proceso de Determinación de Requerimientos	Para que un producto de software sea Portable debe ser capaz de ser transferido de un ambiente a otro. El proceso de determinación de requerimientos influye en esta característica, ya que tiene como fin reunir, procesar y monitorear la evolución de las necesidades y requerimientos de Portabilidad que debe ofrecer el producto de software a través del ciclo de vida.
ENG 1 Proceso de Desarrollo	Esta característica del proceso se refiere al conjunto de requerimientos del producto de software que deben ser transformados para que se satisfagan las necesidades expresadas por el cliente. Influye notablemente en la portabilidad de un producto de software, ya que en éste se define el grado o la capacidad del mismo para ser transferido de un ambiente a otro, según los requerimientos del cliente.
SUP 4 Proceso de Verificación	Esta característica afecta la Portabilidad del producto de software ya que la misma debe confirmar que el producto refleje apropiadamente los requerimientos especificados, en particular que tenga la capacidad de ser transferido de un ambiente al otro.

Tabla 14. Características del proceso que impactan sobre la categoría Portabilidad

5.2 Características del Proceso que Influyen en Todas las Categorías del Producto de Software

En la Tabla 15 se describen tres (3) características del proceso que impactan sobre todas las categorías del producto de software; es decir, características del proceso que son comunes a todas las categorías del producto.

Características del Proceso	Descripción
MAN 3 Proceso de Gestión de Calidad	Esta característica del proceso influye en todas las categorías del producto ya que tiene como finalidad monitorear la calidad de los productos y/o servicios a lo largo del proceso de desarrollo. Si no se cumple esta característica, el producto resultante no garantizará un nivel de calidad adecuado y por ende el mismo no se adaptará a las condiciones para las cuales fue diseñado.
SUP 3 Proceso de Aseguramiento de Calidad	Esta característica también afecta a todas las categorías del producto porque proporciona seguridad con el objeto de que los procesos de un proyecto cumplan con los requerimientos específicos y se adhiera a los planes establecidos. Si no se cumple adecuadamente esta característica del proceso, no se garantizará la calidad en el producto de software debido a que el mismo no cumplirá con los requerimientos necesarios para funcionar bajo condiciones específicas.
SUP 7 Proceso de Auditoria	Esta característica del proceso también influye en todas las categorías del producto ya que tiene como finalidad determinar independientemente la complacencia de los productos seleccionados y los procesos con requerimientos, planes y contratos pertinentes. Esta característica afecta a cada una de las categorías ya que garantiza que el producto cumpla con los requerimientos establecidos, tomando en cuenta las especificaciones del cliente.

Tabla 15. Características del proceso que impactan sobre todas las categoría del producto de software

Las características del proceso descritas en la Tabla 14 afectan a todas las categorías del producto porque garantizan la calidad del proceso de desarrollo y por ende calidad del producto final. Además, estas características del proceso aseguran que el producto final cumpla con todos los requerimientos establecidos a lo largo del proceso de desarrollo. Por ello, si en la evaluación del producto de software falla alguna de las categorías, entre las causas probables se encuentran esas tres características.

Una vez explicados todos los elementos que conforman MOSCA y las características del proceso que impactan a las diferentes categorías del producto, se presenta a continuación el algoritmo de aplicación del modelo aquí propuesto.

5.3 Algoritmo de Aplicación del Modelo MOSCA

La Figura 5 explica gráficamente los pasos a seguir (algoritmo) para medir la calidad sistémica dentro de una organización a través de la aplicación del modelo MOSCA. Para ello, se deberá medir primero la calidad del producto de software y luego la calidad del proceso de desarrollo del mismo.

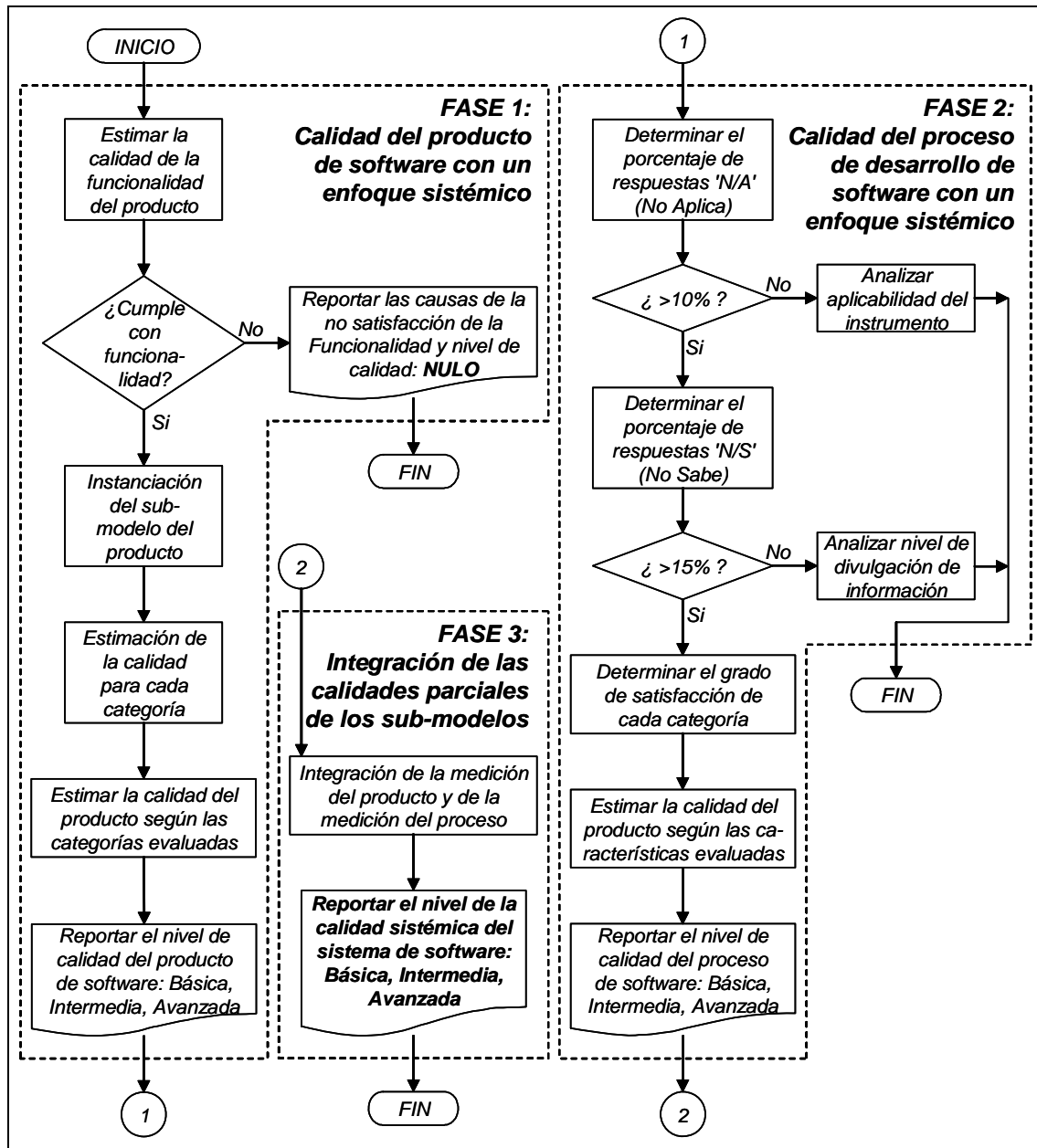


Fig. 5. Algoritmo de aplicación de MOSCA

Según la Figura 5, siempre y en todos los casos se debe medir primero la categoría Funcionalidad del producto. Si cumple con todas las características necesarias que se proponen para esta categoría, entonces se deberá proceder a adaptar el sub-modelo del producto según las especificaciones del cliente. Si el producto no cumple con la categoría Funcionalidad, la evaluación finaliza; es decir, el resto del sub-modelo de producto y el sub-modelo del proceso no deberá ser evaluado. Esto se debe a que la categoría Funcionalidad es la más importante dentro de la medición de la calidad, ya que identifica la capacidad del mismo para cumplir las funciones para las que fue fabricado. Además, como aporte importante, se brinda al cliente las causas del por qué la Funcionalidad no pudo ser satisfecha y el nivel de calidad resultó ser nulo.

Seguidamente se hace la *instanciación* del sub-modelo del producto. Para ello el cliente debe seleccionar dos (2) categorías de las cinco (5) restantes del modelo del producto; aquellas que considera que su producto de software debe cumplir y que desea que sean evaluadas. Una vez adaptado el sub-modelo del producto, se deberán evaluar cada una de las categorías seleccionadas por el cliente.

Finalmente, para poder medir la calidad del producto de software se presenta la Tabla 16, en la cual se relacionan el nivel de calidad con las categorías satisfechas. 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.

Funcionalidad	Segunda categoría evaluada	Tercera categoría evaluada	Nivel de calidad del producto de software
<i>Satisfecha</i>	No satisfecha	No satisfecha	Básico
<i>Satisfecha</i>	<i>Satisfecha</i>	No satisfecha	Intermedio
<i>Satisfecha</i>	No satisfecha	<i>Satisfecha</i>	Intermedio
<i>Satisfecha</i>	<i>Satisfecha</i>	<i>Satisfecha</i>	Avanzado

Tabla 16. Nivel de calidad del producto con respecto a las categorías satisfechas para el producto

Una vez terminada la evaluación del producto y sólo en caso de que este obtenga al menos un nivel de calidad básico, se procederá a medir la calidad del proceso a través del sub-modelo del mismo. Partiendo de las categorías evaluadas en el sub-modelo del Proceso, se estima la calidad de este según las categorías satisfechas:

- *Calidad básica.* Es la mínima calidad requerida. Se satisfacen las características: Cliente-Proveedor e Ingeniería.
- *Calidad intermedia.* Esta no sólo satisface las características de Calidad básica, sino que, además, satisface las características de Soporte y Gestión.
- *Calidad avanzada.* Satisface todas las características.

Por último, se debe realizar una “integración” de la medición del producto y de la medición del proceso para obtener la medición de la calidad sistémica. Los niveles de calidad sistémica se proponen en la Tabla 17.

Nivel de Calidad Producto	Nivel de Calidad Proceso	Calidad Sistémica
Básico	-	Nulo
Básico	Básico	Básico
Intermedio	-	Nulo
Intermedio	Básico	Básico
Avanzado	-	Nulo
Avanzado	Básico	Intermedio
Básico	Intermedio	Básico
Intermedio	Intermedio	Intermedio
Avanzado	Intermedio	Intermedio
Básico	Avanzado	Intermedio
Intermedio	Avanzado	Intermedio
Avanzado	Avanzado	Avanzado

Tabla 17. Nivel de Calidad Sistémica Global a partir del nivel de Calidad del Producto y el nivel de Calidad del Proceso de Desarrollo

Como puede observarse en la Tabla 16, esta propuesta obedece a la necesidad de mantener un equilibrio entre las distintas dimensiones de la Calidad de los SS; es por ello que, la Calidad del Producto de Software tiene igual peso que la Calidad del Proceso de Desarrollo de Software. Se considera que la aplicación del modelo permitirá ajustar con mayor precisión este “equilibrio”. Finalmente, se tiene que la integración de las medidas de calidad de los sub-modelos mide la calidad sistémica como una balanza; es decir, si el nivel de calidad de uno de los sub-modelos es menor que el nivel del otro sub-modelo, entonces la balanza no estará estable y por ello se inclinará hacia el nivel de menor calidad. Esto se debe a la sencilla razón de que **si la Calidad del Producto de Software o la Calidad del Proceso de Desarrollo no cumplen con las características necesarias para tener un nivel más alto de calidad, implicará directamente que la calidad sistémica tampoco cumpla con las características necesarias para tener un nivel de calidad superior.**

Cabe destacar que se realizó un prototipo automatizado del algoritmo de aplicación de MOSCA. Tres de sus interfaces se presentan en las Figuras 6, 7 y 8.

Presentado MOSCA, se hace necesario evaluarlo para determinar su aplicabilidad, esto permitirá analizarlo y ajustarlo según sea su comportamiento. Es por ello que en la siguiente sección se describe y aplica el método de evaluación que DESMET (Kitchenham, 1997) propone, sobre la base de las características de MOSCA y del uso de criterios técnicos y prácticos.



Fig. 6. Interfaz para la selección de las Categorías del Producto de Software

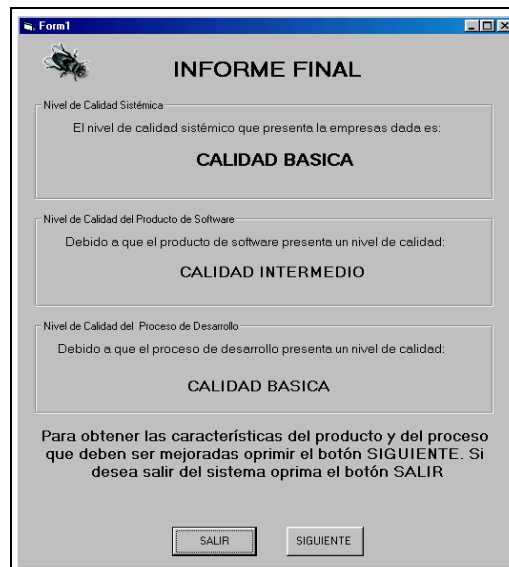


Fig. 7. Interfaz que muestran el resumen del resultado final de la ejecución de MOSCA

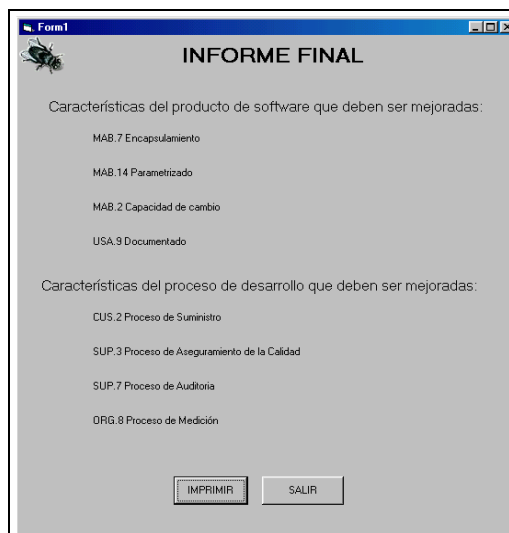


Fig. 8. Interfaz que muestran los detalles del resultado final de la ejecución de MOSCA

6 Análisis de las Características por Caso de Estudio

El método de evaluación “Análisis de las Características por Caso de Estudio”, pautado por DESMET (Kitchenham et al, 1997), consiste en la evaluación del modelo propuesto una vez que este haya sido aplicado en un proyecto de software “real”; es decir, la aplicación del modelo en una organización que permita verificar que los efectos o respuestas del mismo sean los debidos.

El procedimiento que lleva a cabo el método seleccionado para la evaluación comprende, a grandes rasgos, la identificación de las herramientas a ser evaluadas, la definición de un conjunto de características a evaluar, la identificación de los criterios de aceptación o rechazo de las herramientas evaluadas, la selección de un proyecto de reciente data y la definición de la forma en que se presentarán los resultados.

Específicamente, las actividades que se deben seguir para evaluar el prototipo de modelo MOSCA, según el método seleccionado, son las siguientes (Kitchenham, 1997):

1. *Identificar el modelo a evaluar.* La herramienta a ser evaluada es el Modelo Sistémico de Calidad MOSCA, que pretende medir la calidad sistémica en organizaciones desarrolladoras de software. El modelo MOSCA propuesto se describió detalladamente en la sección 3.
2. *Identificar un conjunto de características a evaluar.* Las características que deben ser evaluadas son las categorías del producto y del proceso del modelo MOSCA, que representan el objetivo principal de esta investigación.
3. *Evaluar el modelo contra las características identificadas.* Para evaluar las características previamente mencionadas, se realizaron dos cuestionarios. El primero enfocado al producto y dirigido a tres grupos de evaluación: el Líder del Proyecto, Desarrolladores-Analistas y Usuarios. El segundo cuestionario está enfocado al proceso de desarrollo y va dirigido a dos grupos de evaluación: el Líder del Proyecto y los Desarrolladores-Analistas. Ambos cuestionarios presentan un alto grado de confiabilidad. Específicamente el cuestionario del producto fue validado utilizando la medida de estabilidad (confiabilidad por Test-retest) y arrojó resultados positivos (0.7 y 0.68), lo que indica que el cuestionario es altamente confiable. Para el caso del cuestionario del proceso, este fue validado utilizando la prueba de confiabilidad del coeficiente Alfa-Cronbrach, resultando también ser positivo (0.659), demostrando un alto grado de confiabilidad.
4. *Seleccionar un proyecto piloto.* En dos (2) empresas desarrolladoras de SS se seleccionó un producto de software emblemático y que, además, fuese desarrollado recientemente. Para el caso de la Empresa A se seleccionó un website de una empresa nacional, a través del cual se pueden realizar operaciones de compra con tarjeta de crédito, consultar precios y obtener información de la misma empresa. Para el caso de la Empresa B se seleccionó una herramienta que permite la administración y control de la ejecución de los procesos funcionales de una organización.
5. *Probar el modelo en el proyecto piloto.* Se aplicaron los cuestionarios (ver paso 3) en las dos empresas. Específicamente en la Empresa A, se le aplicó un cuestionario de producto y proceso al líder de proyecto, otro cuestionario de producto y proceso a dos (2) desarrolladores y otro cuestionario del producto a un (1) usuario. Para el caso de la Empresa B, se aplicó un cuestionario de producto y proceso al líder de proyecto, otro cuestionario de producto y proceso a un (1) desarrollador y otro cuestionario del producto a un (1) usuario. Se debe acotar en este punto que el grado de confiabilidad de las respuestas no fue el mismo en ambas empresas, ya que en la Empresa A, el evaluador aplicó el cuestionario personalmente, evitando subjetividades y posibles confusiones del encuestado. Mientras que para el caso de la Empresa B, los cuestionarios fueron respondidos directamente por los encuestados en ausencia del evaluador. Las diferencias en la aplicación de los cuestionarios se debieron a exigencias de las mismas empresas.
6. *Asignar valor a cada característica del modelo.* El algoritmo de aplicación del modelo MOSCA detalla la asignación de valores a cada característica del mismo.
7. *Analizar los valores resultantes y realizar un reporte de evaluación.* Los resultados de la evaluación, su respectivo análisis y reporte de evaluación se presentan en la siguiente sección.

7 Análisis de los Resultados Obtenidos

Según el método seleccionado “Análisis de las Características por Caso de Estudio”, luego de aplicar el modelo en el proyecto piloto de la empresa, se debe analizar los valores resultantes y realizar un reporte de la evaluación. Con este análisis se puede determinar el grado de satisfacción con el que el modelo refleja la calidad sistémica de una organización y, además, se puede refinar el mismo. Los datos de las empresas fueron analizados tomando en cuenta: (1) las categorías del producto seleccionadas por la empresa junto con el evaluador, (2) las categorías del proceso, y (3) la relación entre las características del producto y del proceso.

En primer lugar se analizan los datos referentes al producto. Las categorías seleccionadas por ambas empresas fueron (aparte de *Funcionalidad*): *Mantenibilidad* y *Usabilidad*. Se debe recordar que según el algoritmo del modelo MOSCA, la

empresa debe seleccionar exactamente 3 categorías que identifiquen a su producto de software. La Empresa A seleccionó la categoría *Usabilidad*, ya que el Web site debe ser un producto atractivo, entendible y fácil de utilizar para los usuarios del mismo. Lo más importante de esta aplicación es su front-end, por lo cual el mismo debe cumplir los requerimientos de la categoría *Usabilidad*. La otra categoría seleccionada fue *Mantenibilidad*, ya que el Web site es un producto que debe ser actualizado constantemente y por ello debe tener la facilidad de ser modificado sin ningún problema. La Empresa B seleccionó la categoría *Usabilidad*, ya que su producto está destinado a diferentes tipos de usuarios y la dificultad en el uso del mismo debe ser mínima. Además, esta aplicación debe ser atractiva, ya que el éxito de la misma, dependerá del grado de satisfacción de los usuarios. La otra categoría seleccionada fue *Mantenibilidad*, ya que el producto de software está en constante desarrollo y debe ser capaz de aceptar cualquier tipo de modificaciones. Además, la aplicación debe ser actualizada y aceptar la inclusión de nuevos módulos sin necesidad de cambiar estructuralmente la misma. A continuación se presenta el análisis de los datos obtenidos de las dos empresas.

7.1 Resultado Parcial: Nivel de Calidad del Producto de Software.

- a) Funcionalidad: En la Figura 9 se muestran los porcentajes alcanzados por el producto de software para ambas empresas, en cuanto a los requerimientos de calidad asociados a la Funcionalidad.

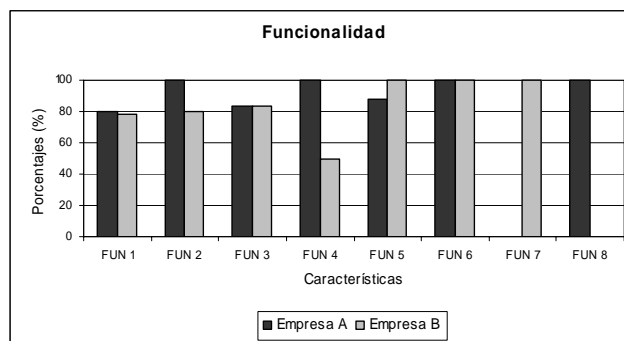


Fig. 9. Porcentajes de satisfacción de los productos frente a las características de la funcionalidad

Como se puede observar en la Figura 9, para la Empresa A siete (7) de las ocho (8) características de la categoría funcionalidad son altamente satisfechas; es decir, cumplen al menos un 75% de las métricas asociadas. Sólo la característica FUN 7 (Encapsulado) no fue satisfecha. Se debe recordar que si seis (6) de las características del producto de software son altamente satisfechas, entonces la categoría Funcionalidad está presente en el producto de software evaluado. En conclusión, como el número de características altamente satisfechas es siete (7), entonces el producto de la Empresa A **cumple** con la categoría Funcionalidad. Paralelamente, para la Empresa B, sólo la característica FUN 4 (Seguridad) y la característica FUN 8 (Especificado) no son altamente satisfechas (ver Figura 9); es decir, no cumplen el 75% de las métricas asociadas. Se concluye entonces que el producto de software de la Empresa B **cumple** con esta categoría, ya que seis (6) de sus características son altamente satisfechas.

- b) Usabilidad: La Figura 10 muestra los porcentajes alcanzados por las características asociadas a la categoría Usabilidad de los productos de software evaluados.

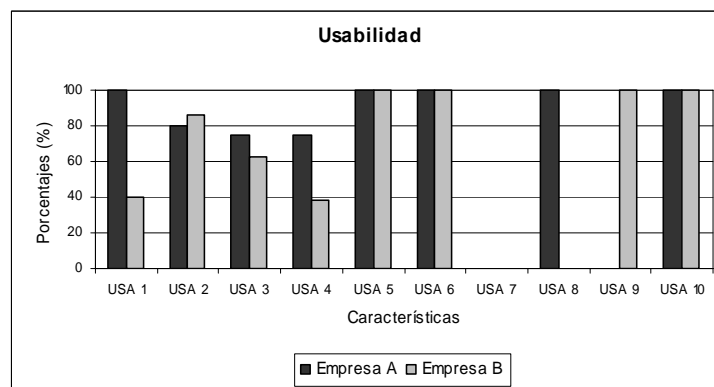


Fig. 10. Porcentajes de satisfacción de los productos frente a las características de la Usabilidad

Según la Figura 10, el producto de software evaluado de la Empresa A **cumple** con la categoría Usabilidad, ya que ocho (8) de las características son altamente satisfechas y ese es el número mínimo que propone el algoritmo de aplicación del modelo MOSCA para que la categoría Usabilidad se cumpla en un producto de software. Únicamente las características USA 7 (Efectivo) y USA 9 (Documentado) no fueron altamente satisfechas, de hecho están ausentes en la categoría del producto. Para el caso del producto de la Empresa B, las características USA 2 (Capacidad de aprendizaje), USA 5 (Completo), USA 6 (Consistente), USA 9 (Documentado) y USA 10 (Auto-Descriptivo), según la Figura 10, son altamente satisfechas, mientras que las demás características no cumplen el 75% de las métricas asociadas. Esto indica que sólo 5 de las 10 características son altamente satisfechas, lo que implica que el producto de software de la Empresa B **no cumple** con la categoría Usabilidad.

- c) Mantenibilidad: Para analizar la Mantenibilidad que presenta el producto de software de ambas empresas, se tiene la Figura 11, en la cual se muestra los porcentajes alcanzados por las características asociadas a la categoría Mantenibilidad.

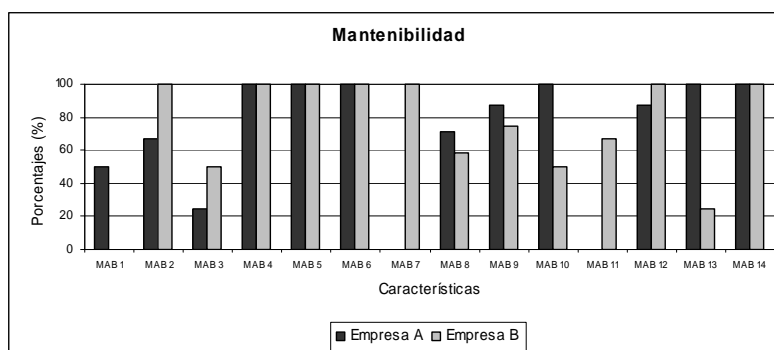


Fig. 11. Porcentajes de satisfacción de los productos frente a las características de la Mantenibilidad

Para el caso de la categoría Mantenibilidad, se puede observar en la Figura 11 que sólo ocho (8) de las características el producto de la Empresa A asociadas a dicha categoría son altamente satisfechas, lo que implica que el producto de software de la Empresa A **no cumple** con esta categoría. Esto se debe a que según el algoritmo de aplicación de modelo MOSCA, para que un producto de software cumpla con la categoría Mantenibilidad debe tener al menos 11 de las 14 características altamente satisfechas (75%). Nótese que la característica MAB 11 (Descriptivo) está ausente en el producto de software. Por otra parte (ver Figura 11), las características MAB 1 (Capacidad de análisis), MAB 3 (Estabilidad), MAB 8 (Atributos de Madurez del Software), MAB 10 (Sistema de Estructura de Información), MAB 11 (Descriptivo), MAB 13 (Estructural), no son altamente satisfechas por el producto de software de la Empresa B. En conclusión, el producto de la Empresa B no es un producto mantenible; es decir, **no cumple** con la categoría Mantenibilidad, ya que sólo siete (7) características son altamente satisfechas.

Siguiendo con los estándares del algoritmo de aplicación del modelo MOSCA, se concluye que **el producto de software de la Empresa A presenta un nivel de calidad Intermedio**, ya que cumple con dos (2) de las tres (3) categorías instanciadas, específicamente con la categoría Funcionalidad y Usabilidad, presentando deficiencias en la categoría Mantenibilidad. Por su parte, **el producto de software de la Empresa B presenta un nivel de calidad Básico**, ya que únicamente cumple con la categoría Funcionalidad.

7.2 Resultado Parcial: Nivel de Calidad del Proceso de Desarrollo.

Una vez obtenido el nivel de calidad del producto de software de ambas empresas, se debe evaluar el proceso de desarrollo de ambos productos para poder medir el nivel de calidad sistémico de las dos (2) empresas. Por ello, se presenta a continuación el análisis de los datos referentes al proceso de desarrollo de las empresas, concretamente las categorías Cliente-Proveedor, Ingeniería, Soporte, Gestión y Organizacional.

- a) Cliente-Proveedor: La Figura 12 muestra los porcentajes de las características asociadas a la categoría Cliente-Proveedor que presentan los procesos de desarrollo evaluados.

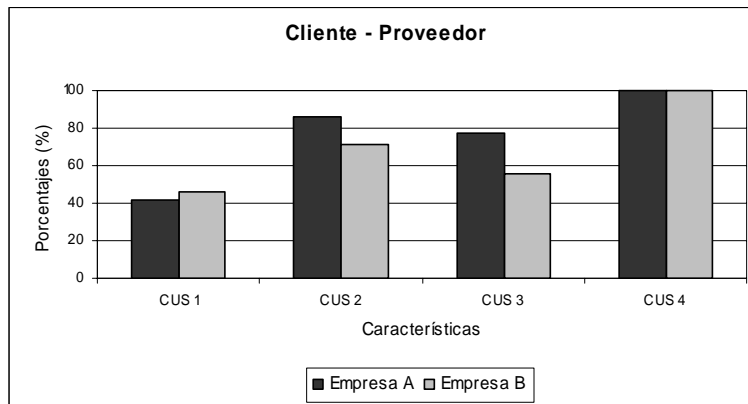


Fig. 12. Porcentajes de satisfacción de las empresas frente a las características de Cliente-Proveedor

Como se puede observar en la Figura 12, tres (3) de las cuatro (4) características de la categoría Cliente-Proveedor de la Empresa A, son altamente satisfechas. Esto indica que dicha categoría **se cumple** en el proceso de desarrollo de la Empresa A, debido a que el algoritmo de aplicación del modelo MOSCA establece que si tres (3) de las cuatro (4) características de la categoría Cliente-Proveedor son altamente satisfechas, entonces dicha categoría esta presente en el proceso de desarrollo evaluado. Únicamente la característica CUS 1 (Proceso de Adquisición del Sistema o producto de Software) no fue altamente satisfecha (40 %) por la Empresa A. Paralelamente, como se puede observar en la Figura 12, la única característica que es altamente satisfecha por la Empresa B en esta categoría, es CUS 4 (Proceso de Operación). Esto indica que la categoría Cliente-Proveedor **no se cumple** en el proceso de desarrollo de la Empresa B.

- b) Ingeniería: Para analizar la categoría Ingeniería del proceso de desarrollo de ambas empresas, se presenta la Figura 13, donde se pueden observar el porcentaje obtenido por las características de dicha categoría.

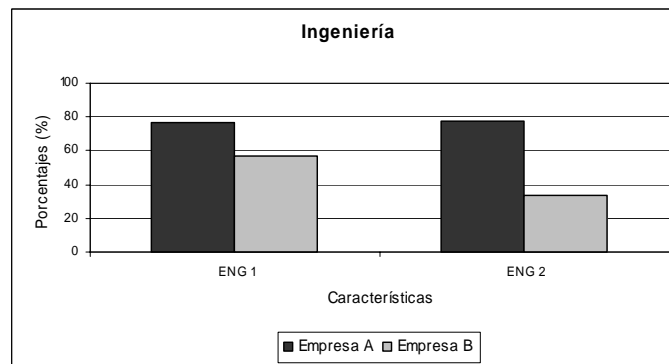


Fig. 13. Porcentajes de satisfacción de las empresas frente a las características de Ingeniería

Según la Figura 13, las dos características de la categoría Ingeniería ENG 1 (Proceso de Desarrollo) y ENG 2 (Proceso de Mantenimiento de software y Sistemas), son altamente satisfechas en la Empresa A; es decir, ambas características cumplen con al menos un 75% de las métricas asociadas. Esto indica que esta categoría **se cumple** en el proceso de desarrollo de la Empresa A. Para el caso de la categoría Ingeniería del proceso de desarrollo de la Empresa B, se puede observar en la Figura 13 que ninguna de las dos características son altamente satisfechas; es decir, ni ENG 1 (Proceso de Desarrollo) ni ENG 2 (Proceso de Mantenimiento de software y Sistemas) fueron cumplidas. En ese sentido, se puede afirmar que la categoría Ingeniería **no está presente** en el proceso de desarrollo de la Empresa B.

- c) Soporte: La Figura 14 muestra el porcentaje obtenido por las características de la categoría Soporte asociados a los procesos de desarrollo evaluados.

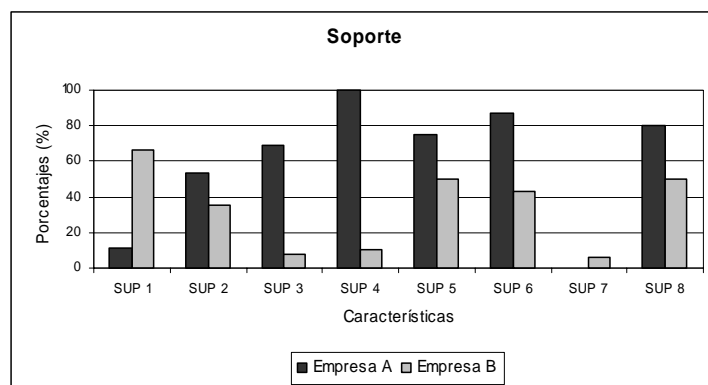


Fig. 14. Porcentajes de satisfacción de las empresas frente a las características de Soporte

Según el algoritmo de aplicación del modelo MOSCA, para que se cumpla la categoría Soporte en un proceso de desarrollo, se debe tener al menos seis (6) características altamente satisfechas. En el caso de la Empresa A, como se puede observar en la Figura 14, sólo cuatro (4) características son altamente satisfechas: SUP 4 (Proceso de Verificación), SUP 5 (Proceso de Validación), SUP 6 (Proceso de Revisión Conjunta) y SUP 8 (Proceso de Resolución de Problemas). Por esta razón, el proceso de desarrollo de la Empresa A **no cumple** con la categoría Soporte en su proceso de desarrollo. Además se puede observar que la característica SUP 7 (Proceso de Auditoría) no es satisfecha. Paralelamente, se puede observar en la Figura 14 que la categoría Soporte **no está presente** en el proceso de desarrollo de la Empresa B, ya que ninguna de sus características son altamente satisfechas.

- d) Gestión: Para poder analizar el grado de satisfacción de la categoría Gestión en el proceso de desarrollo de ambas empresas, se presenta la Figura 15, la cual muestra el porcentaje obtenido por las características de dicha categoría.

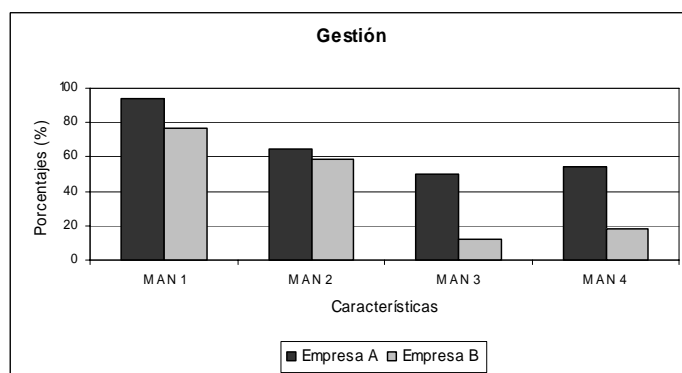


Fig. 15. Porcentajes de satisfacción de las empresas frente a las características de Gestión

La Figura 15 muestra que la categoría Gestión en el proceso de desarrollo de la Empresa A **no se cumple**, ya que para que dicha categoría esté presente en un proceso de desarrollo (según el algoritmo de aplicación de MOSCA), al menos tres (3) de las cuatro (4) características deben estar altamente satisfechas; para el caso de la Empresa A, sólo una (1) característica es altamente satisfecha: MAN 1 (Proceso de Gestión). De igual manera, para el caso de la Empresa B (ver Figura 15), la única característica altamente satisfecha es MAN 1 (Proceso de Gestión). Por esta razón, esta categoría también **está ausente** en el proceso de desarrollo de la Empresa B.

- e) Organizacional: La Figura 16 muestra el porcentaje obtenido por las características de la categoría Organizacional asociada al proceso de desarrollo de ambas empresas.

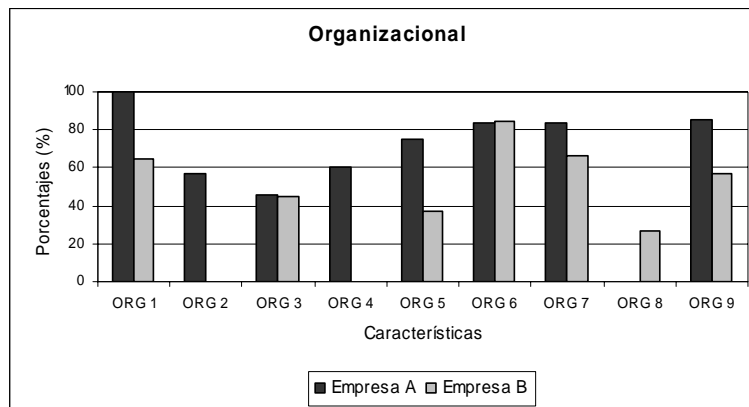


Fig. 16. Porcentajes de satisfacción de las empresas frente a las características de Organizacional

Como se puede observar en la Figura 16, sólo cinco (5) características de la categoría Organizacional son altamente satisfechas por la Empresa A, lo que indica que el proceso de desarrollo de la Empresa A **no cumple** con esta categoría. Para que un proceso de desarrollo cumpla con la categoría Organizacional, debe tener al menos siete (7) características altamente satisfechas. Por otro lado, como se puede observar en la Figura 16, para el caso del proceso de desarrollo de la Empresa B, tan sólo una (1) característica es altamente satisfecha: ORG 7 (Proceso de Infraestructura); por esa razón, la categoría Organizacional **no está presente** en el proceso de desarrollo de la dicha empresa.

Analizando los resultados anteriores se tiene que **el proceso de desarrollo de la Empresa A tiene un nivel de calidad Básica**, ya que cumple únicamente con las categorías Cliente-Proveedor e Ingeniería, siendo éstas las necesarias para obtener este nivel de calidad. Por otra parte, se concluye que **el proceso de desarrollo de la Empresa B tiene un nivel de calidad Nulo**, ya que no cumple con la categoría Cliente-Proveedor ni con la categoría Ingeniería, que son las mínimas para optar por un nivel de calidad básico.

7.3 Resultado Final: Nivel de Calidad Sistémico.

Tomando en cuenta que el producto de software de la Empresa A obtuvo un nivel de calidad Intermedio y sabiendo, además, que el nivel de calidad del proceso de desarrollo fue Básico, entonces se puede concluir, siguiendo los estándares que describe el algoritmo del modelo MOSCA, que **el nivel de calidad sistémico que presenta la Empresa A es Básico**. Cabe acotar que los procesos que según el modelo MOSCA impactan en la categoría Mantenibilidad no son satisfechos, con excepción de ENG 1 (Proceso de Desarrollo) que apenas satisface el mínimo requerido por el modelo. Estos procesos son SUP 1 (Proceso de Documentación), SUP 2 (Proceso de Gestión de Configuración), MAN 3 (Proceso de Gestión de Calidad), SUP 3 (Proceso de Aseguramiento de la Calidad) y SUP 7 (Proceso de Auditoría). Esto explica el nivel de calidad expuesto por la categoría Mantenibilidad.

Por su parte, **el nivel de calidad sistémico que presenta la Empresa B es Nulo**. Esto se debe a que tanto el producto de software como el proceso de desarrollo del mismo, presentan grandes deficiencias. Esencialmente se pudo observar que el proceso de desarrollo no sigue ningún tipo de metodología ni de controles, tanto internos como externos. Se debe destacar que los procesos que impactan sobre las categorías del producto que no fueron satisfechas, Mantenibilidad y Usabilidad, obtuvieron niveles de calidad muy bajos, lo cual explica el nivel de calidad final del producto de software de la Empresa B.

Los procesos que impactan sobre la Mantenibilidad del producto son: ENG 1 (Proceso de Desarrollo), SUP 1 (Proceso de Documentación) y SUP 2 (Proceso de Gestión de Configuración); nótese que ninguno de estos procesos es altamente satisfecho. Además, los procesos que impactan sobre la categoría Usabilidad son: CUS 3 (Proceso de determinación de requerimientos), SUP 1 (Proceso de documentación), SUP 6 (Proceso de revisión conjunta). Como se pudo observar, ninguno de estos procesos es altamente satisfecho. Por último, se puede observar que ninguno de los procesos que impactan sobre todas las categorías del producto son altamente satisfechas en el proceso de desarrollo de la Empresa B. Estos procesos son: MAN 3 (Proceso de Gestión de Calidad), SUP 3 (Proceso de aseguramiento de la calidad), SUP 7 (Proceso de Auditoría).

8 Conclusiones y Trabajos Futuros

La Calidad Sistémica de una organización es el reflejo de la calidad de sus productos de software y la calidad del proceso de desarrollo de los mismos (calidad de diseño y fabricación). En ese sentido, las metas que se establezcan para la calidad del producto van a determinar los objetivos del proceso de desarrollo, porque la calidad del primero va a depender, entre otros aspectos, de éstos.

El **MO**delo **S**istémico de **C**alidad (MOSCA) mide la Calidad Sistémica de una organización desarrolladora de Software partiendo de la Calidad de su producto y de la Calidad del proceso de desarrollo del mismo. Este modelo arroja un nivel de Calidad que puede variar entre Nulo, Básico, Intermedio y Avanzado.

El modelo MOSCA especifica los procesos que se deben mejorar en la empresa y las características que no son satisfechas por el producto de software desarrollado.

Luego de haber evaluado MOSCA a través de la aplicación del método *Análisis de Características por Caso de Estudio* (Kitchenham, 1997) en dos (2) oportunidades, se constató que el modelo constituye una herramienta efectiva de análisis y medición de la calidad Sistémica Global, ya que se analizan aspectos del producto, del proceso y su relación con el medio ambiente. Las empresas evaluadas mostraron una alta aceptación cuando se aplicó el modelo en cada una de ellas y sobre todo cuando se les presentaron los resultados obtenidos, sus respectivas explicaciones y las acciones que debían emprender para mejorar su Calidad Sistémica.

Para futuras investigaciones se recomienda instanciar MOSCA para diferentes tipos de SS. Además, se sugiere la creación de un método de medición de la calidad basado en MOSCA ya que no puede ser la aplicación de cuestionarios a la empresa de forma directa o indirecta, el único medio de recolección de información, dado que esta no es información suficiente.

Agradecimientos

Esta investigación fue financiada por el Fondo Nacional de Ciencia, Tecnología e Innovación (FONACIT) de la República Bolivariana de Venezuela, a través del proyecto S1-2000000437. Los autores desean dar las gracias a J. Martínez por su valiosa colaboración en la culminación de esta investigación.

Referencias

1. **Bass, L., P. Clements, R. Kazman**, *Software Architectur e in Practice*, Second Edition. SEI Series in Software Engineering, Addison-Wesley, New York, 2003
2. **Callaos, N. and B. Callaos**, "Designing with Systemic Total Quality", in *International Conference on Information Systems*, Orlando, Florida, July, 1996, pp. 548-560.
3. **Dromey, R.**, "Cornering the Chimera", *IEEE Software*, Vol. 13, No. 1, 1996, pp. 33-43.
4. **Gillies, A.C.**, *Software Quality: Theory and Management*, Second Edition, Thomson Computer Press, New York, 1998.
5. **Humphrey, W.**, *Introduction to the Personal Software Process*, Addison Wesley Longman, Inc., Massachusetts, 1997.
6. **Ivanisevich, J., P. Lorenzi, S. Skinner, and P. Crosby**, *Management Quality and Competitiveness*, Second Edition, IrWin/McGraw-Hill, New York, 1997.
7. **Jacobson, I., M. Christerson, P. Jonsson, G. Overgaard**, *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1992.
8. **JTC 1/SC 7. ISO/IEC 9126:1991 Information technology - Software product evaluation - Quality characteristics and guidelines for their use**, JTC 1 Organization, Montréal, Québec, 1991.
9. **JTC 1/SC 7. Software Process Assessment, TR 15504, WG 10: Software Process Assessment**, JTC 1 Organization, Montréal, Québec, 1999, <http://www.sqi.gu.edu.au/spice/>.
10. **Kitchenham, B., S. Linkman, and D. Law**, "DESMET: A methodology for evaluating software engineering methods and tools," *IEEE Computing & Control Engineering Journal*, Vol. 8, No. 3, June 1997, pp.120-126.
11. **Ortega, M., M. Pérez, and T. Rojas**, "A Model for Software Product Quality with a Systemic Focus," in *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, 2000, pp. 395-401, <http://www.lisi.usb.ve/publicaciones>.
12. **Pérez, M., T. Rojas, L. Mendoza, and A. Grimán**, "Systemic Quality Model for System Development Process: Case Study", in *Seventh Americas Conference on Information Systems – AMCIS 2001*, Boston, Massachusetts, August, 2001, pp. 1297-1304, <http://www.lisi.usb.ve/publicaciones>.
13. **Pressman, R.** *Ingeniería del Software. Un enfoque práctico*, 5ta edición, McGraw-Hill Interamericana, España, 2002.
14. **Rojas, T. and M. Pérez**, "Improvement in the Development of Information Systems by increasing its Process Effectiveness", in *InterSymp '95 - 5th International Symposium on Systems Research, Informatics and Cybernetics*, Baden-Baden, Germany, July, 1995, pp. 36-42, <http://www.lisi.usb.ve/publicaciones>.
15. **Voas, J.**, "Software Quality's Eight Greatest Myths", *IEEE Software*, Vol. 16, No. 5, pp. 740-745.
16. **Yourdon, E.**, *Análisis Estructurado Moderno*, Primera Edición Prentice Hall, 1992.



Luis Eduardo Mendoza Morales. Miembro de la Association of Information Systems (AIS). Profesor Agregado de la Universidad Simón Bolívar. MSc. en Ingeniería de Sistemas (1999), Especialista en Gestión de Servicios de Información (1993) y Lic. en Matemáticas (1988). Líneas de investigación: Integración de Sistemas, Calidad de Software, Ingeniería del Software y Herramientas CASE. Áreas de interés: Impacto de los Sistemas de Información en las organizaciones. Áreas de experticia: Sistemas de Información, Ingeniería de Software y Metodologías de Desarrollo. Áreas de consultoría: Mejoramiento del Proceso de Desarrollo, Adopción de Metodologías de Desarrollo, Evaluación de la Calidad del Software y Selección de Herramientas CASE.



Maria Angélica Pérez de Ovalles. Miembro de Association of Information Systems (AIS). Profesora Titular de la Universidad Simón Bolívar. PhD en Ciencias de la Computación (1999) UCV. MSc en Sistemas de Información (1993) UCAB, Ingeniero Electricista (1975) UniMet. Líneas de investigación: Mejoramiento del Proceso de Desarrollo, Metodologías, Tecnologías de la Información, Calidad del Software, Evaluación de Arquitecturas de Software. Herramientas CASE, Consultora en: Mejoramiento del Proceso de Desarrollo y Metodologías de Desarrollo en Systar, Banco Mercantil, Banco Banesco, PDVSA.



Anna Cecilia Grimán Padua. Profesor Agregado de la Universidad Simón Bolívar. MSc. en Ingeniería de Sistemas (2000), Universidad Simón Bolívar. Ingeniero en Informática (1995), Universidad Lisandro Alvarado. Líneas de Investigación: Sistemas Colaborativos, Arquitecturas de Software, Evaluación Arquitectónica del Software. Áreas de Experticia: Sistemas de Información, Metodologías de Desarrollo, Gerencia del Conocimiento, Calidad del Software, Ingeniería del Software. Áreas de transferencia de conocimiento: Consultora en Mejoramiento del Proceso de Desarrollo, Evaluación de la Calidad del Software, Selección de Herramientas CASE y Adopción de Metodologías de Desarrollo.