



**Departamento de Tecnologías y Sistemas de Información
Universidad de Castilla–La Mancha**

Trabajo para la asignatura
“Calidad y Medición de los Sistemas de Información”

**Análisis comparativo del estándar
ISO 9000-3 con las subcaracterísticas
de calidad de la ISO 9126**

*Escuela Superior de Informática
Ciudad Real, Enero de 2009*

Estudiante: César Jesús Pardo Calvache
Tutor: Dr. Manuel Serrano

Doctorado en “Tecnologías Informáticas Avanzadas”

Índice de Contenido

Índice de tablas	3
Índice de figuras	3
1. Introducción.....	4
2. Estado del arte. Marcos de calidad y madurez de procesos.....	5
2.1. Familia de Normas ISO 9000:2000.....	6
2.2. La norma ISO 90003.....	6
2.3. El estándar ISO 9126.....	7
3. Características de calidad en los marcos de madurez de procesos software. ...	8
4. Identificación de las características de calidad a comparar.....	10
5. Comparación de la ISO 9000-3 con las características de calidad de la ISO 9126.....	16
6. Discusión de resultados.....	23
7. Conclusiones y futuros trabajos.....	24
Referencias	25

“Finalmente todo se conecta: personas, ideas, objetos.
La calidad de las conexiones es la clave para la calidad en sí”.

Charles Eames.
1907 - 1978

Índice de tablas

Tabla 1. Características y subcaracterísticas de calidad del estándar ISO 9126.	8
Tabla 2. Factores de calidad empleados por Ashrafi.....	10
Tabla 3. Mapeo de la ISO 9126 con los factores de calidad definidos en el Manual de SQA.	12
Tabla 4. Mapeo de las subcaracterísticas de Funcionalidad.....	17
Tabla 5. Mapeo de las subcaracterísticas de Fiabilidad.	17
Tabla 6. Mapeo de las subcaracterísticas de Usabilidad.	18
Tabla 7. Mapeo de las subcaracterísticas de Eficiencia.....	19
Tabla 8. Mapeo de las subcaracterísticas de Mantenibilidad.	19
Tabla 9. Mapeo de las subcaracterísticas de Portabilidad.	21

Índice de figuras

Figura 1. Características de calidad de ISO 9126 presentes en ISO 9000-3.	22
Figura 2. Subcaracterísticas de calidad de ISO 9126 presentes en ISO 9000-3.....	23

1. Introducción.

El enfoque tradicional de la ingeniería del software ha enfocado sus esfuerzos en la definición de metodologías, lenguajes de programación, modelos de desarrollo y herramientas [1]. Sin embargo, actualmente, la ingeniería del software, en su enfoque orientado a procesos, recomienda prestar más atención a la forma de realizar los productos, sin dejar a un lado su documentación. Además, argumenta que la calidad del producto depende de la calidad del proceso que se sigue para obtenerlo [5]. Es decir, que depende fuertemente de las personas, la organización y los procedimientos utilizados para crearlo, entregarlo y mantenerlo [2]. Por lo tanto, la calidad de un producto no puede ser asegurada simplemente con la simple inspección, llevando a cabo controles estadísticos o centrando sus programas de calidad únicamente al producto [3].

Una prueba del especial interés en los procesos se ha visto materializada desde los años 60 con la definición de varios ciclos de vida, como ejemplo, se encuentran los modelos en cascada, evolutivo y en espiral [4], que aunque no definían y documentaban los procesos a llevar a cabo, permitieron en su momento comprender y determinar mejor el proceso y actividades inherentes al proceso de desarrollo en la ingeniería de software.

Desde los años noventa hasta entonces [5], la mejora de procesos también conocida por sus siglas en inglés SPI (Software Process Improvement) aparece como una línea de la ingeniería de software con el objetivo de facilitar el desarrollo y crecimiento de las organizaciones de software. Ha esta línea hacen parte investigadores, universidades y organismos como la ISO (International Standard Organisation) y el SEI (Software Engineering Institute, Universidad de Carnegie Mellon, USA), los cuales han enfocado sus esfuerzos en la optimización de tecnologías que buscan mejorar los procesos de una organización, creando las herramientas necesarias para que la industria cuente con estándares y guías¹, modelos de mejora de procesos, métodos internos de evaluación, modelos del ciclo de vida del software, modelos de ingeniería de sistemas, entornos de ingeniería orientados al proceso (PSEE), cuerpos de conocimiento, gobierno de las TSI, entre otros.

Sin embargo, y a pesar de todos los avances obtenidos, los organismos de estandarización y especialmente la industria del software, sigue luchando para lograr

¹ En adelante para unificar los términos usaremos el término genérico “marco” para hacer referencia a: modelos, modelos de proceso, modelos de calidad, estándar de calidad, de proceso, etc.

aplicaciones de software que cumplan con las normas de calidad, la presión del tiempo y limitaciones de presupuesto [6, 7]. El costo y el tiempo son elementos cuantitativos a diferencia de la calidad, el cual, por ser un elemento dimensional, es difícil no solo de definir, sino también de medir [8].

Dado que en el pasado, las organizaciones software no lograron resultados positivos al utilizar enfoques especiales para la obtención rápida de productos de software de calidad dentro del presupuesto, el SPI un enfoque más lento, costoso y metódico comenzó entonces a tomar fuerza [9]. Sin embargo, por las condiciones de tiempo, costo y recursos que involucran la implementación de estos marcos en las organizaciones, es necesario identificar y establecer las características basadas en la mejora de los procesos y calidad de producto que motiven a las empresas a adoptar dichos enfoques.

Si bien, en [10] es posible observar la gran proliferación de marcos existentes en la actualidad. En este trabajo analizaremos uno de ellos, ISO 9000-3, por ser un marco basado en ISO 9001, además de estar orientado al desarrollo de software de las organizaciones. El objetivo de este trabajo será el de examinar el estándar ISO 9000-3 para identificar en que medida se tratan las características y subcaracterísticas de calidad de acuerdo al estándar ISO 9126. Además, el trabajo aquí desarrollado, habilita una línea futura de análisis para determinar que marco de madurez y/o capacidad fomenta la calidad en el proceso de desarrollo de software y ayuda a las organizaciones a seleccionar el marco más pertinente según sus necesidades.

2. Estado del arte. Marcos de calidad y madurez de procesos.

La evaluación y mejora de los procesos software es una estrategia mediante la cual se puede juzgar y decidir sobre la capacidad y calidad de los procesos que están sujetos a análisis, con el objetivo de poder establecer una maniobra para su mejora. Como resultado de los esfuerzos de la comunidad científica, se han propuesto toda una serie de marcos y estándares para promover la aplicación de marcos de capacidad, evaluación y calidad de proceso en las organizaciones.

A continuación en los apartados siguientes se presenta un resumen de los marcos de mayor relevancia e importancia para el estudio y análisis de este trabajo. Entre ellos se encuentra: ISO 9001:2000, ISO/IEC 90003 e ISO 9126.

2.1. Familia de Normas ISO 9000:2000.

La familia ISO 9000:2000 son un conjunto de normas y pautas internacionales diseñadas para apoyar la gestión de la calidad en las organizaciones. Desde su primera publicación en 1987, han conseguido convertirse como una de las normas base para el establecimiento de los sistemas de gestión de la calidad.

ISO 9000:2000 agrupa las normas ISO 9001, 9002, 9003 y 9004. ISO 9001, 9002 y 9003 son modelos de sistemas de calidad para el aseguramiento de la calidad externa. Actualmente estos modelos se han convertido en subgrupos exitosos de otros, entre ellos el ISO 9001, catalogado como el más comprensible, abarcando etapas de diseño, manufactura, instalación y sistemas de servicio. Por su parte la norma ISO 9002 se especializa en producción e instalación. La ISO 9003 enfoca su documentación en la inspección y exámenes de productos finales, y la ISO 9004 provee una guía para uso interno, permitiendo el desarrollo de sistemas de calidad propios en los negocios. La aparición en el mercado de estos modelos de procesos (centrados en la mejora) y la aplicación y adaptación de la normativa ISO 9000 como ISO 9003:2004 [11], están ofreciendo a las empresas y departamentos de desarrollo informático, la posibilidad de adaptarse a una nueva forma de trabajo caracterizada principalmente por buscar la satisfacción de los clientes y disponer de una mejor visibilidad y control de la calidad de los procesos y de los productos software finales.

2.2. La norma ISO 9003.

La norma ISO/IEC 9003 también conocida como ISO/IEC 9000-3 [11] fue preparada por el comité técnico conjunto ISO/IEC JTC1 (tecnología de la información) subcomité SC7 (software e ingeniería de sistemas). La primera edición de ISO/IEC 9003 canceló y sustituyó a la norma ISO/IEC 9000-3: 1997, que se ha actualizado en conformidad con ISO 9001:2000.

Mientras que el objetivo de la norma ISO 9001 es la de establecer un sistema de calidad, la ISO 9003:2004 provee las especificaciones de cómo aplicar la ISO 9001 a los procesos de software, entre ellos los procesos de adquisición, provisión, desarrollo, operación y mantenimiento y servicios de ayuda relacionados. Además, partiendo del hecho de que es muy poco probable obtener buenos resultados en un proyecto de desarrollo de software en organizaciones de tamaño mediano, algunos temas como la Administración de la Configuración o Planeación de Proyectos que no están presentes

en las normas de la familia ISO 9000, son incluidos en la ISO 90003. Esta característica implica que para ciertos productos y/o servicios, la especificación de requerimientos contenida en las normas de ISO 9001 no sea la suficiente para asegurar la calidad, justificando la necesidad de otras normas o guías más específicas como complemento.

ISO 9000-3 no impone un modelo de ciclo de vida específico. Asimismo, no provee métodos específicos para evaluar la capacidad de aseguramiento de la calidad de una organización [12]. Por lo tanto, puede ser combinado con otros enfoques más específicos, como por ejemplo el modelo Espiral de Boehm [13].

En resumen, esta norma supone sin lugar a dudas un cambio cualitativo para que los ambientes tecnológicos más especializados avancen, especialmente los enfocados al diseño y desarrollo de software.

2.3. El estándar ISO 9126.

El ISO/IEC 9126 es un estándar internacional que permite evaluar la calidad del producto software. Este estándar actualmente se encuentra siendo guiado y supervisado por el ISO/IEC 25000:2005 y una serie de estándares agrupadas dentro del proyecto SQuaRE (Software product Quality Requirement and Evaluation). El ISO/IEC 9126 contiene un modelo de medición y un modelo calidad que permiten llevar a cabo la evaluación de la calidad de los productos software.

Actualmente el estándar está conformado de 4 partes que dirigen: un modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso. La primera parte agrupa y define un conjunto de características y subcaracterísticas para determinar la calidad de un producto. Ellas son:

Tabla 1. Características y subcaracterísticas de calidad del estándar ISO 9126.

<p>a. Funcionalidad. Capacidad del producto software para proporcionar funciones que satisfagan las necesidades especificadas e implícitas.</p> <ul style="list-style-type: none"> ➤ Adecuación. ➤ Exactitud. ➤ Interoperabilidad. ➤ Seguridad. ➤ Conformidad. 	<p>d. Eficiencia. La capacidad del producto software para proporcionar el rendimiento apropiado, relativo a la cantidad de recursos utilizados.</p> <ul style="list-style-type: none"> ➤ Tiempo de Respuesta. ➤ Utilización de recursos. ➤ Conformidad.
<p>b. Fiabilidad. Capacidad del producto software para mantener un nivel especificado de rendimiento.</p> <ul style="list-style-type: none"> ➤ Madurez. ➤ Tolerancia a fallos. ➤ Recuperabilidad. ➤ Conformidad. 	<p>e. Mantenibilidad. La capacidad del producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en los requisitos o en las especificaciones funcionales.</p> <ul style="list-style-type: none"> ➤ Analizabilidad. ➤ Cambiabilidad. ➤ Estabilidad. ➤ Facilidad de prueba. ➤ Conformidad
<p>c. Usabilidad. La capacidad del producto software de ser entendido, aprendido, utilizado y atractivo al usuario.</p> <ul style="list-style-type: none"> ➤ Comprensibilidad. ➤ Facilidad de aprendizaje. ➤ Operabilidad. ➤ Atracción. ➤ Conformidad. 	<p>f. Portabilidad. La capacidad del producto software de ser transferido de un entorno a otro.</p> <ul style="list-style-type: none"> ➤ Adaptabilidad. ➤ Facilidad de instalación. ➤ Coexistencia. ➤ Reemplazabilidad. ➤ Conformidad.

La segunda parte del estándar define las métricas externas necesarias para medir con respecto a cada una de las características y subcaracterísticas agrupadas en la primera parte.

La tercera parte del estándar define las métricas internas necesarias para estimar las características de calidad de un software que se encuentre en ejecución o desarrollo.

Por último, la cuarta parte, define las métricas para establecer la calidad en uso de un producto software.

3. Características de calidad en los marcos de madurez de procesos software.

Como ya hemos visto, el estándar ISO 9126, agrupa en 6 características, 27 subcaracterísticas de calidad. El estándar provee un buen modelo de referencia para medir la calidad del software. Por esta razón, se decidió usar este marco de calidad a lo largo del trabajo.

Encontrar un modelo de calidad lo suficientemente completo y que satisfaga las necesidades de todas las organizaciones que desarrollan software es imposible. Ashrafi

en [8] plantea un ejemplo muy sencillo para comprender mejor esto; “una organización que produce software de misión crítica, considera la *fiabilidad* la característica más importante, mientras que la *portabilidad* puede ser una necesidad para la organización que produce un producto de software para una variedad de plataformas”. Por tanto, “lo que se considera un objetivo importante para una organización puede no ser importante para otra”. Es por esto que es primordial establecer una solución a la pregunta que inquieta tanto a las organizaciones que se encuentran en la selección del marco adecuado que solucione sus necesidades empresariales o requisitos de calidad: ¿Alguno de los marcos existentes satisface mis requisitos de calidad? [14].

Alrededor de la anterior pregunta, se han realizado algunos estudios empíricos que demuestran que marcos de calidad como CMM e ISO mejoran la calidad de los productos software con base a un conjunto de características de calidad [15, 16, 17]. Sin embargo, aun se sigue sin resolver la forma como las organizaciones puedan seleccionar un marco de calidad basados en la calidad del producto software que necesitan desarrollar. Es por esto que para la elaboración de este trabajo nos preguntamos: ¿Podría la ISO 9126 utilizarse como base para examinar y elegir el marco de SPI² más adecuado a las necesidades y requisitos de calidad de una organización?

Es importante armonizar tanto el punto de vista del desarrollador que proveen los marcos de SPI, como los del usuario que proveen los marcos de calidad como el ISO 9126. Se facilitaría la mejora de los procesos si se estableciera un marco en el que la mejora de procesos de desarrollo de software pudiera adaptarse para satisfacer las necesidades empresariales así como los requisitos de calidad de la organización [8]. Esto facilitaría la selección de marcos adecuados, la integración de prácticas necesarias y ajustadas a los requisitos de calidad del producto, además de acentuar más la importancia de la satisfacción del usuario.

En el siguiente apartado, se define el proceso realizado para examinar los marcos de SPI y su correspondencia con respecto a las características de calidad de ISO 9126.

² En adelante, se llamará marco de SPI al conjunto de modelos, metodologías, tecnologías, etc, que sirvan de referente para la mejora de procesos de desarrollo de software.

4. Identificación de las características de calidad a comparar.

En [8], Ashrafi compara CMM e ISO 9000-3 con los factores de calidad definidos en el Manual de SQA³ desarrollado por el grupo Software Quality Assurance (SQA). Este trabajo analiza la relación e impacto de la familia de marcos ISO (9001 Y 9000-3) y CMM respecto a un conjunto de factores de calidad enfocados en el diseño, realización y adaptación de software. A diferencia de este trabajo, nuestro objetivo principal se enfocó en examinar la relación de las subcaracterísticas de calidad del estándar ISO 9126 con el estándar para el desarrollo de software ISO 9000-3. Para esto se analizaron las cláusulas del estándar ISO 9000-3 que se relacionaban con cada uno de las subcaracterísticas del marco de calidad que no habían sido analizadas en el trabajo de Ashrafi [8].

En la Tabla 2 se presentan un resumen de los factores de calidad usados en el trabajo de Ashrafi, en la cual se incluyen 14 factores de calidad agrupados en tres categorías.

Tabla 2. Factores de calidad empleados por Ashrafi.

Categorías	Factores de calidad empleados por
Calidad de diseño-DI.	DI.1. Exactitud. <i>Medida en que el software cumple con sus especificaciones y se ajusta a sus objetivos declarados</i>
	DI.2. Mantenibilidad. <i>Esfuerzo para facilitar la localización y la fijación de un fallo de software dentro de un plazo determinado.</i>
	DI.3. Verificabilidad. <i>Facilidad de esfuerzo para comprobar el rendimiento y funciones de software sobre la base de los objetivos declarados.</i>
Calidad de desempeño-DE.	DE.1. Eficiencia. <i>Medida en que el software es capaz de hacer más con menos recursos del sistema (hardware, sistema operativo, comunicaciones, etc.) los recursos</i>
	DE.2. Integridad. <i>Medida en que el software es capaz de resistir la intrusión de usuarios no autorizados o software dentro de un plazo determinado.</i>
	DE.3. Fiabilidad. <i>Medida en que el programa se llevará a cabo (de acuerdo a sus objetivos) dentro de un plazo determinado.</i>
	DE.4. Usabilidad. <i>Relativo a la facilidad de aprendizaje y el funcionamiento del software.</i>
	DE.5. Comprobabilidad. <i>Facilidad de programa para la realización de pruebas para verificar que se realiza una determinada función.</i>
Calidad de adaptación-A.	A.1. Expansibilidad. <i>Relativo al esfuerzo necesario para ampliar capacidades de software y/o el rendimiento mediante la mejora de las funciones actuales o nuevas.</i>
	A.2. Flexibilidad. <i>La facilidad de los esfuerzos para cambiar el software de la misión, las funciones o los datos para satisfacer las cambiantes necesidades y requisitos.</i>
	A.3. Portabilidad. <i>Facilidad de esfuerzo para el transporte a otro entorno de software y/o de la plataforma.</i>
	A.4. Reusabilidad. <i>Facilidad de esfuerzo para el uso del software (o de sus componentes) en otro software de sistemas y aplicaciones.</i>
	A.5. Interoperabilidad. <i>Relativo al esfuerzo necesario para acoplar el software en una plataforma a otro software y /o otra plataforma.</i>
	A.6. Intra-operabilidad. <i>Esfuerzo necesario para las comunicaciones entre los componentes del mismo sistema software.</i>

³ Handbook of Software Quality Assurance, Prentice Hall, 1998.

Aunque visualmente ISO 9126 parece ser más detallada que el manual de SQA, se decidió realizar un mapeo basándonos en una adaptación de la tabla de mapeo de marcos de SPI definida por Mutafelija y Stromberg [18]. El objetivo era establecer el nivel de relación de los factores de calidad usados por Ashrafi y las subcaracterísticas de calidad definidas en la ISO 9126, y a partir de los resultados obtenidos, evaluar las subcaracterísticas de calidad que aun no habían sido examinadas. Dicha relación entre factores y subcaracterísticas estuvo comprendida dentro de tres valoraciones: fuerte (F), mediana (M) y débil (D).

A continuación en la Tabla 3 se presenta el mapeo de subcaracterísticas del estándar ISO 9126 y los factores de calidad definidos en el manual de calidad SQA. La Tabla 3 presenta un resumen de las subcaracterísticas de calidad de ISO 9126 agrupadas de acuerdo a su característica correspondiente. Asimismo, presenta 4 columnas adicionales: la primera que relaciona la categoría a la que se relaciona, la segunda presenta el factor de calidad específico, la tercera presenta el nivel de correspondencia de las subcaracterísticas respecto a los factores de calidad analizados por Ashrafi. La quinta columna es de comentarios, y permite documentar cuestiones de similitudes y diferencias. Los acrónimos utilizados son los siguientes:

- Nivel de relación: (F)-Relación fuerte, (M)-Relación media, (D)-Relación débil.
- -----: No Aplica.
- Categorías del Manual de calidad SQA: ver Tabla 2.
- Factor de calidad: ver Tabla 2.

Tabla 3. Mapeo de la ISO 9126 con los factores de calidad definidos en el Manual de SQA.

Subcaracterísticas ISO 9126		Categoría Manual SQA	Factor de calidad	Nivel de relación	Comentarios
Funcionalidad.	Adecuación. La capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.	DI	DI.1.	F	-----
	Exactitud. La capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.	DI	DI.1.	F	-----
	Interoperabilidad. La capacidad del producto software para interactuar con uno o más sistemas especificados.	A	A.5	F	-----
	Seguridad. Referido a la capacidad del producto software para proteger la información y los datos.	DE	DE.2	D	DE.2 Tiene en cuenta el periodo de tiempo en el que el software debe mantenerse íntegro ante la intrusión de usuarios no deseados, no detalla la característica de protección de la información y datos.
Fiabilidad.	Madurez. La capacidad del producto software para evitar fallos provocados por errores en el software.	DE	DE.3	D	DE.3 Es muy subjetivo, no da mayores detalles acerca de las subcaracterísticas asociadas a la fiabilidad, solo se refiere al cumplimiento de los objetivos dentro de un periodo de tiempo determinado.
	Tolerancia a fallos. La capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.				
	Recuperabilidad. La capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.				
Usabilidad.	Comprensibilidad. La capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.	-----	-----	-----	-----
	Facilidad de aprendizaje. La capacidad del producto software para permitir al usuario aprender su aplicación.	DE	DE.4	F	DE.4 Relativo a la facilidad de aprendizaje.
	Operabilidad. La capacidad del producto software para permitir que el usuario lo opere y lo controle.	-----	-----	-----	-----
	Atracción. La capacidad del producto software para atraer al usuario.	-----	-----	-----	-----

Eficiencia.	Comportamiento temporal. <i>La capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.</i>	-----	-----	-----	-----
	Utilización de recursos. <i>La capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones.</i>	DE	DE.1	M	DE.1 Hace énfasis en el incremento del resultado esperado con el uso mínimo de recursos del sistema.
Mantenibilidad.	Analizabilidad. <i>Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.</i>	DI	DI.2	M	DI.2 Solo tiene en cuenta la mantenibilidad del software a nivel de fallos y no de cambios o modificaciones. Sin embargo, es importante el plazo de tiempo en el que son localizados los fallos.
	Cambiabilidad. <i>Capacidad del producto software de permitir implementar una modificación especificada. La implementación incluye los cambios en el diseño, el código y la documentación.</i>	A	A.1	M	A.1 Especifica la ampliación de nuevas mejoras y no solo modificaciones. Sin embargo, no especifica que los cambios también deban hacerse en el diseño, el código y la documentación.
		A	A.2	M	A.2 Tiene en cuenta la cambiabilidad de la misión del software, funciones o datos.
	Estabilidad. <i>Capacidad del producto software de evitar los efectos inesperados de las modificaciones.</i>	-----	-----	-----	-----
Facilidad de prueba. <i>Capacidad del producto software de permitir validar las partes modificadas.</i>	DI	DI.3	D	DI.3 Verifica el rendimiento de funciones sobre la base de los objetivos declarados sin limitarse solo a cambios.	
	DE	DE.5	D	DE.5 Verifica la realización de una determinada función. No se limita solo a los cambios.	

Portabilidad.	Adaptabilidad. <i>La capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado.</i>	A	A.3	D	A.3 No es muy detallado con las subcaracterísticas inherentes a la portabilidad.
	Facilidad de instalación. <i>La capacidad del producto software para ser instalado en un ambiente determinado.</i>				
	Coexistencia. <i>La capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.</i>				
	Reemplazabilidad. <i>La capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.</i>				

Como factor de inclusión o identificación de las subcaracterísticas de calidad a comparar, se decidió seleccionar sólo las subcaracterísticas bajo el criterio de que presentaran ninguna, débil o mediana relación. Esto significa que se examinaron las subcaracterísticas que no fueron analizadas inicialmente en el trabajo de Ashrafi, ya sea por; no estar presentes en el marco de calidad SQA utilizado, o por no presentar descripciones mínimamente similares a las definidas en la ISO 9126.

Por otra parte, de las 27 subcaracterísticas definidas en la ISO 9126, excluimos la subcaracterística de conformidad la cual está incluida en cada característica, quedando así un total de 21 subcaracterísticas. De las cuales, tras ejecutar la comparación se obtuvo que, 4 se analizaron como fuertemente relacionadas, 3 medianamente relacionadas, 9 débilmente relacionadas y 5 que no aplicaron ninguna relación. Resultando así un total de 14 subcaracterísticas a examinar en ISO 9000-3, entre ellas y sin orden de característica se encuentran: seguridad, madurez, tolerancia a fallos, recuperabilidad, comprensibilidad, operabilidad, atracción, comportamiento temporal, estabilidad, facilidad de prueba, adaptabilidad, facilidad de instalación, coexistencia y reemplazabilidad.

Si bien, el mapeo es subjetivo y depende o está influenciado de acuerdo a las interpretaciones individuales de los subcaracterísticas y factores de calidad de la ISO 9126 y el Manual de SQA correspondientemente, la identificación de las relaciones entre factores y subcaracterísticas de calidad permitirá filtrar y trabajar sólo con las subcaracterísticas de calidad que no se han examinado. Excluyendo así las subcaracterísticas de adecuación, exactitud, interoperabilidad, facilidad de aprendizaje, utilización de recursos, analizabilidad y cambiabilidad, no fueron examinadas puesto que ya habían sido tratadas en [8]. Por lo tanto, cuando se analizaron los resultados obtenidos en este trabajo, se agregaron los porcentajes de dichas subcaracterísticas previamente hallados por Ashrafi.

Si bien la ISO 9126 es aplicable a cualquier tipo de software, encontramos que de los 14 factores de calidad que Ashrafi tomó como referencia para evaluar los marcos de CMM e ISO 9000-3, dos factores no pudieron ser correspondidos, ellos son: la reusabilidad y la intra-operabilidad. Estos factores o subcaracterísticas de calidad, pueden ser adaptados e integrados de otros marcos de calidad al estándar ISO 9000-3, pero por el objetivo de este trabajo, “la integración de factores o subcaracterísticas de otros marcos

en el marco de calidad de la ISO 9126”, lo dejamos como un tema propuesto para el desarrollo de futuros trabajos.

5. Comparación de la ISO 9000-3 con las características de calidad de la ISO 9126.

Es importante tener en cuenta que al igual que los marcos de SPI que están definidos según el punto de vista de los desarrolladores, las subcaracterísticas de la ISO 9126 están definidas desde el punto de vista del usuario. Por tanto, las subcaracterísticas seleccionadas para la comparación fueron adaptadas de acuerdo al contexto de los procesos. Por ejemplo, la ISO 9000-3 cubre la eficiencia con respecto al proceso y la facilidad de uso a través de la formación en vez de la facilidad de uso del producto [8]. Asimismo ocurre con la subcaracterística referente al comportamiento temporal, el cual es adaptado con relación al comportamiento del producto software en condiciones especiales y similares al ambiente final de ejecución. Este comportamiento es analizado antes de ofrecer el producto para que sea aceptado por el cliente.

Para llevar a cabo la comparación de los marcos SPI y la ISO 9126 analizamos las cláusulas del marco ISO 9000-3.

A continuación en las Tablas 4, 5, 6, 7, 8 y 9, se presenta la correspondencia de las cláusulas de ISO 9000-3 con las subcaracterísticas de calidad seleccionadas.

Tabla 4. Mapeo de las subcaracterísticas de Funcionalidad.

	Subcaracterísticas ISO 9126.	Cláusulas ISO 9000-3.	Comentarios
Funcionalidad.	Seguridad. <i>Referido a la capacidad del producto software para proteger la información y los datos.</i>	6.3 Infraestructura. Esta cláusula se refiere a la determinación, suministro y mantenimiento de una infraestructura para lograr la conformidad de los requisitos del producto. Asimismo se refiere la necesidad de seguridad, backup, firewall y protección de virus, control de acceso, herramientas necesarias, entre otros aspectos también relacionados con la seguridad.	La seguridad es una característica opcional de soporte a la infraestructura para el funcionamiento del producto software. La mención de firewall, antivirus, y demás herramientas de seguridad son aspectos de infraestructura que no se tratan como elementos inherentes al producto.

Tabla 5. Mapeo de las subcaracterísticas de Fiabilidad.

	Subcaracterísticas ISO 9126.	Cláusulas ISO 9000-3.	Comentarios
Fiabilidad.	Madurez. <i>La capacidad del producto software para evitar fallos provocados por errores en el software.</i>	7.3.6.1 Validación. Proporcionar la confianza razonable de que el software cumplirá sus requisitos de operación.....Identificación de fallos de diseño y desarrollo. 7.5.1.7 Mantenimiento. Literal d)...seguimiento del sistema para detectar fallos. 7.5.2 Validación de los procesos de la producción y de la prestación del servicio. Literal b) Historial de fallos de diseño y desarrollo y como se puede evitar.	Se puede pensar que la madurez en el proceso estaría orientada en la mejora del mismo. La cláusula relacionada sería la 8, relacionada con la medición, análisis y mejora. Sin embargo, se ha decidido adaptar la subcaracterística de Madurez, como la capacidad del proceso para el control de los fallos de diseño y desarrollo.
	Tolerancia a fallos. <i>La capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.</i>	8.3 Control del producto no conforme.En el caso del proveedor....Cuando un elemento software manifieste un defecto durante el desarrollo o el mantenimiento, la investigación y resolución de tales defectos debería controlarse y registrarse. En el caso del software...literal a) reparación o reelaboración para arreglar defectos. 7.3.6.2 Pruebas.Conviene realizar pruebas de regresión en beneficio del cliente.....La aceptación puede ser con o sin defectos o desviaciones de los requisitos...	Define características a tener en cuenta durante el proceso de desarrollo para solucionar fallos o defectos detectados en el software.

	Recuperabilidad. <i>La capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.</i>	6.3 Infraestructura. Esta cláusula se refiere a la determinación, suministro y mantenimiento de una infraestructura para lograr la conformidad de los requisitos del producto. Entre las características menciona el backup como herramienta software que puede ser incluida como elemento de soporte a la infraestructura.	La recuperabilidad es una característica opcional de soporte a la infraestructura para el funcionamiento del producto software. No se trata como una característica inherente al producto.
--	--	---	--

Tabla 6. Mapeo de las subcaracterísticas de Usabilidad.

	Subcaracterísticas ISO 9126.	Cláusulas ISO 9000-3.	Comentarios
Usabilidad.	Comprensibilidad. <i>La capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.</i>	7.1.2 Planificación de la calidad. Literal k) hace referencia a la formación en el uso de las herramientas y técnicas, y planificación de la formación antes de ser necesaria la habilidad. 7.3.1.1 Planificación del diseño y desarrollo. Literal a), numeral 6) hace referencia a la formación requerida del equipo como la cláusula 7.1.2. 7.4.1.1 Proceso de compras. En el desarrollo, suministro, instalación y mantenimiento de producto software.....puede incluir....literal i) cursos y materiales de formación. 7.4.2 Información de las compras. Literal g) Requisitos sobre el personal (...prerrequisitos de formación, conocimiento del producto). 7.5.1.5 Instalación. Literal j) se debería definir la necesidad de proporcionar la formación asociada con el uso querido del producto....	La comprensibilidad es una subcaracterística que en la ISO 9000-3 es llevada a cabo desde el punto de vista de la formación, tanto del desarrollador para llevar a cabo la realización del producto software, como del usuario luego del proceso de compras. Esta fuertemente relacionada a la subcaracterística de Facilidad de Aprendizaje.
	Operabilidad. <i>La capacidad del producto software para permitir que el usuario lo opere y lo controle.</i>	NA. -----	-----
	Atracción. <i>La capacidad del producto software para atraer al usuario.</i>	NA. -----	-----

Tabla 7. Mapeo de las subcaracterísticas de Eficiencia.

	Subcaracterísticas ISO 9126.	Cláusulas ISO 9000-3.	Comentarios
Eficiencia.	Comportamiento temporal. <i>La capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.</i>	7.3.6.1 Validación. Antes de ofrecer el producto para la aceptación del cliente, conviene que la organización valide la operación del producto de acuerdo con su uso específico que se pretenda, bajo condiciones similares al entorno de aplicación.....	El comportamiento temporal es adaptado con relación al comportamiento del producto software en condiciones especiales y similares al ambiente final de ejecución. Este comportamiento es analizado antes de ofrecer el producto para que sea aceptado por el cliente.

Tabla 8. Mapeo de las subcaracterísticas de Mantenibilidad.

	Subcaracterísticas ISO 9126.	Cláusulas ISO 9000-3.	Comentarios
Mantenibilidad.	Estabilidad. <i>Capacidad del producto software de evitar los efectos inesperados de las modificaciones.</i>	NA. -----	-----
	Facilidad de prueba. <i>Capacidad del producto software de permitir validar las partes modificadas.</i>	7.3.7 Control de los cambios del diseño y desarrollo. Los cambios del diseño y desarrollo deben identificarse y deben mantenerse registros. Los cambios deben revisarse, verificarse y validarse, según sea apropiado, y aprobarse antes de su implementación...Conviene que los cambios a una especificación software o componente mantenga la adecuada consistencia entre los requisitos, especificaciones de pruebas.... 7.4.1.1 Productos comprados. En el desarrollo, suministro, instalación y mantenimiento de productos software, los tipos de productos comprados pueden incluir: Literales d) actividades contratadas a terceros, entre ellos las pruebas...., e) herramientas destinadas a la ayuda en el desarrollo software, entre ellas analizadores de pruebas... 7.5.1.7 Mantenimiento. Se debería estipular en el contrato el mantenimiento del producto software...., éste debería incluir: Literal f) las	Las pruebas no sólo se limitan a ser realizadas en las partes modificadas antes de su implementación, sino también durante y después, entre ellas: 4.2.1 Generalidades. La documentación del sistema de gestión de la calidad debe incluir: Numeral 5) temas técnicos como documentos de normas o documentos de guía para la codificación, diseño y pruebas. 4.2.4.1. Evidencias de conformidad con los requisitos. La evidencia de conformidad con los requisitos puede incluir: Literal a) resultado de las pruebas documentados. 7.1 Planificación de la realización del producto. Durante la planificación de la realización del producto, la organización debe determinar cuando sea apropiado ... Literal c) Las actividades requeridas de verificación, validación, seguimiento, inspección y ensayo o prueba específicas así como los criterios para la aceptación del mismo. 7.1.2 Planificación de la calidad. La planificación de la calidad del software a nivel de proyecto debería tener los siguientes objetivos: Literal d) procedimientos e instrucciones de proyectos específicos, tales como planes detallados de especificaciones de pruebas del software, diseños, casos de prueba y procedimientos para las pruebas

		<p>actividades de gestión de la configuración, pruebas y aseguramiento de la calidad.</p>	<p>unitarias, de integración, del sistema y de aceptación.</p> <p>7.2.3.2 Comunicación con el cliente durante el desarrollo. Las revisiones conjuntas.....pueden cubrir los aspectos siguientes como: Literal a) información del producto, incluyendo: numeral 4) resultados de las pruebas de aceptación.</p> <p>7.3.1.1 Planificación del diseño y desarrollo. La planificación del diseño y desarrollo debería tratar lo siguiente, donde sea apropiado: Literal a) las actividades de análisis de requisitos,pruebas....., Literal h) la identificación de la planificación relacionada tratando asuntos tales comopruebas.....</p> <p>7.3.3 Resultados del diseño y desarrollo. Los resultados de diseño y desarrollo pueden expresarse en forma de texto, mediante diagramas....., puede incluir: a) especificaciones de diseño, desarrollo y prueba.</p> <p>7.3.4 Revisión del diseño y desarrollo. La revisión del diseño y desarrollo debería llevarse a cabo de acuerdo con las disposiciones planificadas... Literal a) lo que se va a revisar, cuándo y el tipo de revisión, tales como.... Pruebas formales de corrección, ...</p> <p>7.3.5 Verificación del diseño y desarrollo. La verificación del software tiene la finalidad de asegurar que la salida de una actividad de diseño y desarrollo es conforme a los requisitos de entrada... La verificación puede comprendersimulaciones o pruebas....</p> <p>7.3.6.1 Validación. Las auditorías de configuración o evaluaciones confirman, por medio del examen.....y de las pruebas...que el producto software cumple con sus requisitos contractuales o especificados... Literal b) que tipos de pruebas o análisis pueden realizarse para aumentar la confianza en que el producto funcionará correctamente...</p> <p>7.3.6.2 Pruebas. La validación a menudo puede realizarse mediante pruebas. Pueden requerirse pruebas desde diferentes niveles...Existen diferentes enfoques de las pruebas,..... (TODA LA CLÁUSULA ESTÁ DEDICADA A LAS PRUEBAS)</p> <p>7.4.3 Verificación de los productos comprados. Esta verificación puede aplicarse a la aceptación del software comprado...</p>
--	--	---	--

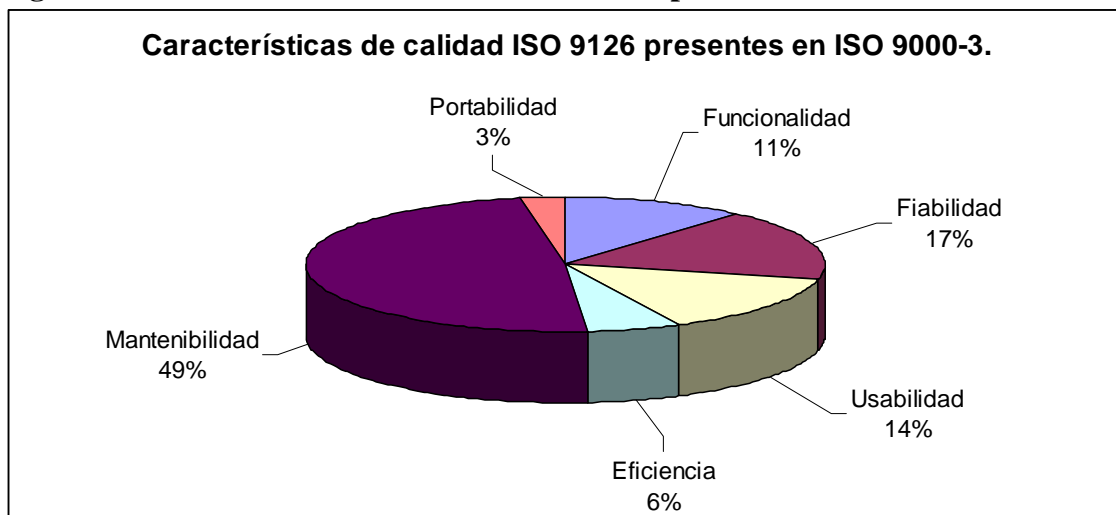
Tabla 9. Mapeo de las subcaracterísticas de Portabilidad.

	Subcaracterísticas ISO 9126.	Cláusulas ISO 9000-3.	Comentarios
Portabilidad.	Adaptabilidad. <i>La capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado.</i>	NA. -----	-----
	Facilidad de instalación. <i>La capacidad del producto software para ser instalado en un ambiente determinado.</i>	7.5.1.5 Instalación. A veces, los clientes o terceras partes llevan a cabo la instalación. En este caso el papel de la organización es descubrir los pasos que el cliente o la tercera parte necesita tomar para realizar la instalación. A veces la instalación se lleva a cabo por la organización. Para el último caso se aplica: Literales a) a k).	-----
	Coexistencia. <i>La capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.</i>	NA. -----	NA. -----
	Reemplazabilidad. <i>La capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.</i>	NA. -----	NA. -----

En primer lugar cabe destacar que de acuerdo al análisis realizado, es posible determinar la distribución de cada una de las características de calidad presentes en el estándar ISO 9000-3. Como se puede apreciar en la Figura 1, podemos ver en orden creciente que: la portabilidad posee una distribución del 3%, la eficiencia del 6%, la funcionalidad un 11%. Seguido se encuentra la usabilidad con un 14% y la fiabilidad con un 17%. Por último, se encuentra la mantenibilidad con un 49%. Esto quiere decir que ISO 9000-3 hace mayor énfasis en las subcaracterísticas comprendidas en las características de funcionalidad, usabilidad, fiabilidad y mantenibilidad. Siendo la última en la que más se especializa o que ofrece un mayor grado de detalle.

El nivel de distribución fue hallado de acuerdo a la cantidad de cláusulas definidas por cada subcaracterística. La suma total de las cláusulas correspondidas en cada subcaracterística, ayudó a determinar el porcentaje por cada característica.

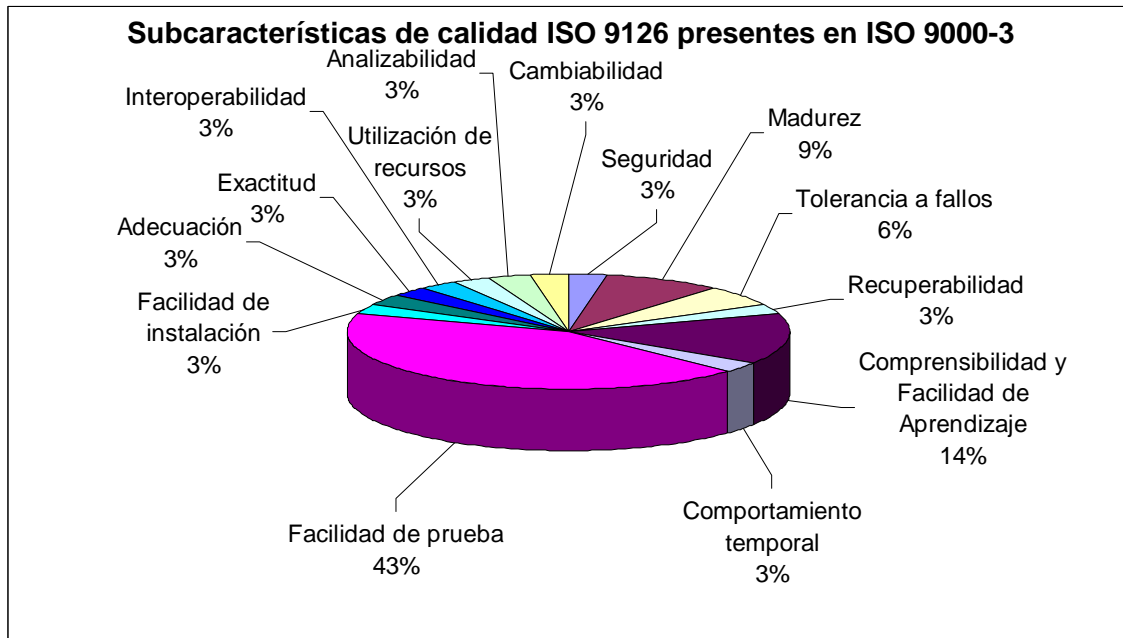
Figura 1. Características de calidad de ISO 9126 presentes en ISO 9000-3.



En la Figura 2 se puede apreciar con más detalle las distribuciones de cada una de las subcaracterísticas analizadas. En orden creciente podemos ver que la operabilidad, atracción, adaptabilidad, coexistencia y reemplazabilidad, son subcaracterísticas en las que ISO 9000-3 no se especializa o no define cláusulas específicas, por lo tanto no aparecen en la figura. Asimismo, se puede apreciar que la facilidad de instalación, adecuación, exactitud, interoperabilidad, utilización de recursos, analizabilidad, cambiabilidad, seguridad, recuperabilidad y comportamiento temporal, presentan una distribución del 3%. La tolerancia a fallos presenta una distribución del 6%. La madurez con un 9%, mientras que la comprensibilidad y facilidad de aprendizaje comprenden un 14%. Por último, podemos ver que la facilidad de prueba con un 43%, es la

subcaracterística con mayor detalle en ISO 9000-3. Las subcaracterísticas de comprensibilidad y facilidad de aprendizaje se agruparon como una sola subcaracterística de calidad, debido a que están fuertemente relacionadas.

Figura 2. Subcaracterísticas de calidad de ISO 9126 presentes en ISO 9000-3



Si bien, fueron analizadas 14 subcaracterísticas de calidad, fue necesario también tener en cuenta los resultados obtenidos en el trabajo Ashrafi. De esta manera, se presentó un resultado general, el cual incorpora los datos obtenidos con respecto a todas las subcaracterísticas de calidad definidas en la ISO 9126 y no solamente el conjunto de subcaracterísticas no tratadas previamente.

6. Discusión de resultados.

- Es posible que la mejora de los procesos software mejore la calidad de las subcaracterísticas del producto software. Sin embargo, el nivel de detalle y especialización de las subcaracterísticas utilizadas depende del marco de SPI que se elija. Por ejemplo ISO 9000-3 presenta un alto porcentaje de especialización en las subcaracterísticas referentes a la mantenibilidad.
- La ISO 9126 agrupa en dos dimensiones diferentes las características de usabilidad y funcionamiento, mientras tanto el Manual de SQA, no hace una diferencia marcada de las dos, y define el funcionamiento dentro del factor de calidad de la usabilidad.

- ISO 9000-3 en la cláusula 8.2.4 Seguimiento y medición del producto. Trata las características de los productos software (funcionalidad, mantenimiento, eficiencia, portabilidad, usabilidad y fiabilidad) por medio de recomendaciones de seguimiento que permitan medir la conformidad con los requisitos de calidad.
- ISO 9000-3 no cubre todas las características de calidad de la ISO 9126. Sin embargo, y teniendo en cuenta el contexto para el cual se aplica normalmente (los productos software), es obvio que algunas de las subcaracterísticas deban adaptarse en función al marco o metodología que se use.
- Si la portabilidad y eficiencia son aspectos cruciales para una organización, esta deberá tener en cuenta que ISO 9000-3 no recomienda prácticas detalladas para llevar a cabo la mejora de estas características de calidad en el producto. Sería importante en el futuro examinar o comparar la distribución de estas características en otros marcos.
- Si la mantenibilidad, fiabilidad, usabilidad y funcionalidad son características de calidad de principal preocupación para una organización que desarrolla productos software donde se implementan dichas características, sería aconsejable usar la ISO 9000-3. Sin embargo, la fiabilidad, usabilidad y funcionalidad podrían ser complementadas con otros marcos.
- Las distribuciones con porcentajes muy bajos encontrados en varias de las subcaracterísticas de calidad, pueden llegar a ser complementadas por la integración de otros marcos, de esta manera, se podrían crear apalancamientos clave para la solución de necesidades específicas.

7. Conclusiones y futuros trabajos.

El desarrollo de este trabajo ha permitido aplicar conceptos relacionados pero muy poco trabajados de manera integrada: el producto software y los procesos utilizados para desarrollarlo. La calidad por ser un concepto multidimensional, posee diferentes definiciones dependiendo del contexto desde donde se examine. Es por esto que al principio no fue muy clara aplicación de los factores o características de calidad para examinar el nivel de correspondencia con los marcos para la mejora de procesos software. Implementar los conceptos definidos para la evaluación de la calidad de los productos software en los marcos de SPI, es una tarea que permite evaluar la selección

de dichas metodologías desde otra perspectiva, no sólo desde los procesos software de la organización sino también desde el usuario.

Las organizaciones basan la elección de su marco de SPI no solo por la tendencia definida en nichos de mercado internacionales, también se encuentran influenciados por la búsqueda de soluciones a sus necesidades organizacionales y requisitos de calidad. El trabajo aquí desarrollado presenta un análisis del marco ISO 9000-3, el cual puede ser extendido para analizar y elegir cualquier marco de SPI adecuado para una organización. Sin embargo, es necesario tener en cuenta que los niveles de relación de las características y subcaracterísticas del estándar de calidad ISO 9126, difieren dependiendo del marco a examinar. Asimismo, el estándar ISO 9126 no provee todas las características de calidad que pueda necesitar evaluar una organización. Por lo tanto, la integración de características y subcaracterísticas, además de la evaluación y comparación de otros marcos de SPI habilita una línea futura de análisis para determinar que marco de madurez y/o capacidad fomenta la calidad en el proceso de desarrollo de software, los resultados podrían ayudar a las organizaciones a seleccionar el marco más pertinente según sus necesidades.

Por otra parte, se han considerado y adaptado técnicas de comparación de marcos de SPI. Sin embargo, dicho mapeo aplicado al análisis de la correspondencia de las subcaracterísticas de calidad con ISO 9000-3 sigue siendo subjetivo y está determinado e influenciado de acuerdo a las interpretaciones individuales. En el futuro, sería importante llevar a cabo un estudio empírico que permita comparar las subcaracterísticas de calidad de la ISO 9126 y la ISO 9000-3 desde la experiencia de las personas involucradas con el uso de los marcos de SPI. Esta validación permitirá comprobar su correspondencia desde un punto no solo teórico, sino también empírico-práctico.

Referencias

1. Piattini, M., F. García, and I.C. Muñoz-Reja, *Calidad de Sistemas Informáticos*. 2006, Madrid, España: Ra-Ma. 388.
2. Ruiz, F., "*MANTIS: Definición de un Entorno para la Gestión del Mantenimiento de Software*". in *Departamento de Informática*. 2003, Universidad de Castilla-La Mancha: Ciudad Real.
3. Satpathy, M. and R. Harrison. *A typed generic process model for product focused process improvement*. in *Proceedings - IEEE Computer Society's International Computer Software and Applications Conference*. 2002.

4. Piattini, M.G., J.A.C. Manzano, J. Cervera, and L. Fernández, *Análisis y diseño de aplicaciones informáticas de gestión - Una perspectiva de Ingeniería del Software*. 2003: Madrid, Ra-Ma. 736.
5. Pino, F., F. García, and M. Piattini, *Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas*. Revista Española de Innovación, Calidad e Ingeniería del Software (REICIS), 2006. **2**(1): p. 6-23.
6. Pearson, J.M., C.S. McCahon, and R.T. Hightower, *Total quality management: are information systems managers ready?* [http://dx.doi.org/10.1016/0378-7206\(95\)00028-0](http://dx.doi.org/10.1016/0378-7206(95)00028-0) Inf. Manage. , 1995 **29** (5): p. 251-263
7. Perry, W.E., *Quality assurance for information systems: methods, tools, and techniques*. 1991: QED Information Sciences, Inc. 814.
8. Ashrafi, N., *The impact of software process improvement on quality: in theory and practice*. [http://dx.doi.org/10.1016/S0378-7206\(02\)00096-4](http://dx.doi.org/10.1016/S0378-7206(02)00096-4) Inf. Manage. , 2003 **40** (7): p. 677-690
9. Herbsleb, J., D. Zubrow, D. Goldenson, W. Hayes, and M. Paulk, *Software quality and the Capability Maturity Model*. <http://doi.acm.org/10.1145/255656.255692> Commun. ACM 1997 **40** (6): p. 30-40
10. Sheard, S.A., *The Frameworks Quagmire*. Crosstalk: The Journal of Defense Software Engineering, 1997. **10**(9).
11. ISO, *ISO/IEC 90003:2004. Software engineering - Guidelines for the application of ISO 9001:2000 to computer software*. 2004, International Organization for Standardization: Ginebra.
12. Derniame, J.-C., B. Kaba, and D. Wastell, *Software Process: Principles, Methodology and Technology*. Lecture Notes in Computer Science 1500, 1999: p. 307.
13. Boehm, B., *A spiral model of software development and enhancement*. IEEE Computer, 1988. **21**(5): p. 61-72.
14. Fazlollahi, B. and M.R. Tanniru, *Selecting a requirement determination methodology-contingency approach revisited* [http://dx.doi.org/10.1016/0378-7206\(91\)90005-M](http://dx.doi.org/10.1016/0378-7206(91)90005-M) Inf. Manage. , 1991 **21** (5): p. 291-303
15. Harter, D.E., M.S. Krishnan, and S.A. Slaughter, *Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development* <http://dx.doi.org/10.1287/mnsc.46.4.451.12056> Manage. Sci. , 2000 **46** (4): p. 451-466
16. Krishnan, M.S., C.H. Kriebel, S. Kekre, and T. Mukhopadhyay, *An Empirical Analysis of Productivity and Quality in Software Products* <http://dx.doi.org/10.1287/mnsc.46.6.745.11941> Manage. Sci. , 2000 **46** (6): p. 745-759
17. Krishnan, M.S. and M.I. Kellner, *Measuring Process Consistency: Implications for Reducing Software Defects*. <http://dx.doi.org/10.1109/32.824401> IEEE Trans. Softw. Eng. , 1999 **25** (6): p. 800-815
18. Mutafelija, B. and H. Stromber, *ISO 9001:2000 - CMMI V1.1 Mappings*. 2003, Software Engineering Institute. p. 31.