

Calidad de Software en el Módulo de Talento Humano del Sistema Informático de la Universidad Técnica del Norte

Tulia Nohemí Vaca Sierra¹, Lenin Omar Lara Castro¹

¹ Programa de Maestría en Ingeniería del Software
Instituto de Posgrado
Universidad Técnica del Norte
Ibarra, Ecuador
tnvaca@utn.edu.ec; olara@utn.edu.ec

Resumen. El Standish Group en sus Reportes CHAOS 2011-2015, promedia el 19% de proyectos de software como *Fallidos*, el 52% como *Discutidos* y el 29% como *Exitosos*. Esta estadística reafirma la idea de que el aseguramiento de calidad de proyectos de software debe fundamentarse en aplicar políticas entorno a la calidad de software. Además, según la definición de calidad de software de IEEE, Std. 610-1990, este trabajo se orienta a determinar y aplicar un Modelo / Estándar de Calidad de Software para el Módulo de Talento Humano (MTH) del Sistema Informático Integrado Universitario (SIU) de la Universidad Técnica del Norte (UTN). Para efectos de esta investigación y al constituir el SIU un producto de software en producción, este trabajo se centra en: revisar estudios comparativos, analizar modelos y estándares de calidad de software, determinar necesidades institucionales mediante la aplicación de entrevistas dirigidas a usuarios del módulo y desarrolladores de software de la UTN. con la finalidad de evaluar el MTH y aplicar una propuesta de mejora. **Resultados:** Se ha aportado a la obtención de información de calidad mediante la ejecución de reportes proporcionados por el MTH, particularidad que se ha logrado al mantener un plan de aseguramiento de calidad del producto de software y que, al proyectar sus resultados, ha demostrado ser una opción viable para implantar en el desarrollo de futuros módulos y aplicar en aquellos que están en producción. **Conclusión:** La implantación del concepto de calidad de software a través de modelos o estándares, constituye la base para garantizar la calidad de todo el SIU y lo que organizacionalmente a la UTN le interesa, contar con información de calidad.

Palabras Clave: modelo, calidad de software, métricas, estándares

Eje temático: Soluciones TIC para la Gestión

1 Introducción

La realidad en que las organizaciones se desenvuelven desde hace un par de décadas atrás, no concibe la ejecución de procesos sin apoyo de un producto de software. Por esta razón la importancia de la calidad de software se ha convertido en un tema de relevancia e interés, en especial, para aquellas empresas y departamentos dedicados al desarrollo.

En la última década, en el Ecuador, las Instituciones de Educación Superior (IES), han sido sujetas a grandes y dinámicos cambios; por lo que es primordial disponer de

Sistemas de Información útiles que garanticen la calidad de sus datos y suministren a la alta dirección, elementos confiables para la toma de decisiones.

La UTN dispone de un Sistema de Información conocido como SIIU, mismo que fue construido desde el 2006 como una plataforma informática integral; se orientó fundamentalmente a sistematizar procesos académicos. Con el pasar del tiempo se incorporaron módulos de apoyo a procesos administrativos, financieros y otros, que constituyen parte esencial de la gestión de una IES. Dentro de estos procesos la UTN se encuentra alerta de que los datos proporcionados por módulos del SIIU, garanticen su calidad. Sin embargo, en más de una ocasión la información relacionada con talento humano ha demostrado inconsistencia.

Se seleccionó el MTH en vista que sus datos son primordiales dentro de procesos de evaluación y acreditación tanto institucional como de carreras de grado y programas de posgrado, asignación de recursos fiscales y, en definitiva, toda su gestión.

2 Calidad de Software

Pressman define la calidad del software como *“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”*.

El uso de Modelos y Estándares de Calidad del Software ayuda a lograr una mejor Gestión de la Calidad a la cual define como *“conjunto de actividades y medios necesarios para definir e implantar un sistema de la calidad, por una parte, y responsabilizarse de su control, aseguramiento y mejora continua, por otra”* [1]. En este sentido, la gestión de la calidad en cualquier organización se centra en los siguientes niveles de trabajo:

- (1) **Nivel de Organización:** Consiste en crear una estructura organizativa apropiada para fomentar el trabajo de calidad de toda la organización. Se mide a través de ISO 9001, ISO 90003 y modelo CMMi (Capability Maturity Model Integration).
- (2) **Nivel de Proyecto:** Depende de cómo se lleve a cabo todo el proceso y subprocesos del proyecto. Una alteración en cualquier fase, restará calidad tanto al proyecto cuanto a la satisfacción del usuario. Para garantizar la calidad a nivel de proyecto se puede realizar Revisiones Técnicas Formales (RTF) actividad ejecutada por profesionales de Ingeniería de Software que amplía la visión sobre lo que se revisa.
- (3) **Nivel de Producto de Software:** En este caso la calidad se convierte en algo concreto, posible de definir, medir y, sobre todo, de planificar. Para establecer la calidad a nivel de producto se aplican modelos como McCall, Boehm y otros.

Como se indicó en la sección anterior, esta investigación se orienta al MTH del SIIU, módulo que constituye un producto de software en producción, razón por la que, a partir de este punto, el estudio se enfoca a los modelos y estándares de calidad a *nivel de producto de software*.

2.1 Modelos y Estándares de Calidad de Software a Nivel Producto

Los modelos y estándares de calidad de software han sido tema de discusión para gran cantidad de autores, dada la trascendencia del tema y el gran despliegue que ha tenido a lo largo de los años [2].

2.1.1 Modelos de Calidad de Software

Los modelos de calidad son: “*documentos que constituyen la mayor parte de principales destrezas; además, constituyen un apoyo en la práctica del concepto general de calidad, ofreciendo una definición más operacional*” [5]. A continuación, se muestra la estructura de un modelo de calidad.

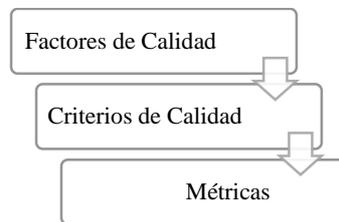


Fig. 1: Estructura de un Modelo de Calidad

- Los *factores de calidad* o *atributos externos*, son características que componen la calidad, representándola desde el punto de vista de *usuario*.
- Los *criterios de calidad* o *atributos internos*, son aquellos en los que se desagregan los diferentes factores, representan la calidad desde el punto de vista del *producto* asociados a cada factor.
- Las *métricas* se definen para cada criterio de calidad, son medidas cuantitativas que indican el grado de presencia de un atributo en el producto.

Para evaluar la calidad de un producto de software, han surgido distintos modelos, formados por factores y criterios. Al evaluar estos factores de calidad en diferentes jerarquías, se puede determinar la calidad del software.

Tipos de Modelos de Calidad de Software:

- Modelos Fijos:** Disponen de un catálogo de factores de calidad de partida que se usa como base para evaluar y se usará para cada proyecto concreto. Siguen este enfoque: McCall, Boehm, Keller y FURPS. Proporcionan una vista común y comparable que se reutiliza en cada proyecto, ya que el conjunto de factores de calidad siempre es el mismo. Sin embargo, ofrece poca flexibilidad debido a que asumen que siempre bastará con un subconjunto de sus factores para evaluar la calidad en cualquier proyecto.
- Modelos a Medida:** Este tipo de modelos no existe ningún catálogo de factores de partida, además, que dichos factores deben ser identificados para cada proyecto. Así, los modelos son creados desde cero para todo nuevo proyecto. Ofrecen total

adaptabilidad; sin embargo, el coste de construcción es muy alto comparado con modelos fijos, y la reutilización de modelos de un proyecto a otro, es difícil.

- c) **Modelos Mixtos:** Intentan combinar las ventajas de los *fijos* y *a medida*. La idea es que exista un conjunto de factores de calidad más abstractos que sean reutilizados en prácticamente todos los proyectos posibles, y que puedan ser refinados y operacionalizados para un proyecto particular. Se destaca como propuestas de este tipo de modelos: ADEQUATE, Gilb e ISO/IEC 9126-1:2001. [6].

2.1.2 Estándares de Calidad de Software

Según Piattini (2003) citado por Scalone, F. (2006), los Estándares de Calidad *permiten definir un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software. Suministran medios para que todos los procesos se realicen de la misma forma y son una guía para lograr la productividad y calidad.* [1]

2.2 Comparativa de Modelos y Estándares de Calidad de Software a Nivel Producto

Revisando literatura especializada, esta investigación se ajusta al análisis de 3 estudios comparativos con distinto enfoque, mismos que contribuyen a determinar un grupo de *modelos y estándares de calidad de software a nivel producto*, de los cuales se realiza un detalle de sus principales características.

Estudio 1: Se determina cinco criterios con su respectiva escala de valoración; se evalúa 27 entre modelos y estándares de calidad obteniendo 8 con mayor calificación, a estos se contrasta con 12 atributos de calidad. El resultado de este estudio señala que: los modelos y estándares más relevantes y recomendados son: McCall y Quint2 seguidos de Boehm, Dromey e ISO 9126, no muy lejos del segundo grupo se encuentra FURPS. [2]

Estudio 2: Los autores establecen 9 criterios y 18 características de calidad, evalúan con esos criterios y características a 10 modelos y estándares, identificados como los más referenciados y utilizados. Se obtuvo como resultado 6 características y 22 subcaracterísticas como las más usadas y relevantes en esos 10 modelos y estándares. Se concluye indicando que los estándares más completos son ISO/IEC 9126 y 25010, pues son estándares mixtos, con un propósito general y reutilizable, con aplicabilidad en casi todo tipo de proyecto, capaces de evaluar la calidad interna, externa y en uso. [7]

Estudio 3: Realiza un estudio y comparativa descriptiva de cinco modelos y estándares de calidad de software entre los que resalta aspectos como: nombre, tipo, características, año, en qué se enfoca o centraliza, factores y métricas. [8]

2.2.1 Conclusión de la Comparativa

Una vez establecidos los *modelos y estándares de calidad de software a nivel producto* con mayor contemplación de atributos de calidad y mayor uso y referencia; para este trabajo se estudiará: *McCall, FURPS, Boehm, ISO/IEC 9126, ISO/IEC 25010*.

2.3 Estudio de Modelos y Estándares de Calidad de Software a Nivel Producto

A. Modelo McCall

Este modelo fue creado por Jim McCall en 1977. Establece 3 perspectivas para el análisis de la calidad de software, define 11 factores y 23 criterios relacionados a estos. Las métricas que propone son preguntas que ponderan numéricamente un determinado atributo del producto de software. Después de obtener los valores para todas las métricas de un criterio específico, el promedio de todas ellas es el valor para ese criterio [9]. La estructura del modelo se detalla en la tabla 1.

Tabla 1: Estructura Modelo McCall

Perspectivas	Factores	Criterios
Operatividad del Producto: Factores de calidad que influyen en el grado en que el software cumple con su especificación.	Usabilidad: Facilidad de uso del software.	Operatividad Entrenamiento Comunicación
	Integridad: Protección del programa al acceso no autorizado.	Control de Acceso Auditoría de Acceso
	Corrección: Grado en que una funcionalidad coincide con su especificación.	Rastreabilidad Complejidad Consistencia
	Fiabilidad – Confiabilidad: Capacidad de los sistemas de no fallar / medida en que falla el sistema.	Consistencia Exactitud Tolerancia a fallos
	Eficiencia: Clasificado en eficiencia de ejecución y eficiencia de almacenamiento ejemplo: tiempo de procesador, memoria.	Eficiencia en ejecución Eficiencia en almacenamiento
Revisión del Producto: Factores de calidad que influyen en la	Mantenibilidad: Esfuerzo requerido para localizar y arreglar un fallo en el programa dentro de su entorno operativo.	Simplicidad Concreción

capacidad de cambiar el producto de software.	<p>Facilidad de Prueba: Facilidad del programa de realizar pruebas para asegurarse de que está libre de errores y cumple con su especificación.</p>	<p>Simplicidad Instrumentación Auto-descripción Modularidad</p>
	<p>Flexibilidad: Facilidad de hacer cambios necesarios según lo solicitado en el entorno operativo.</p>	<p>Auto-descripción Capacidad de expansión Generalidad Modularidad</p>
<p>Transición del Producto: Factores de calidad que influyen en la capacidad de adaptar el software a nuevos entornos.</p>	<p>Reusabilidad: Facilidad de reutilización de software en un contexto diferente.</p>	<p>Auto-descripción Generalidad Modularidad</p>
	<p>Interoperabilidad: Esfuerzo requerido para acoplar el sistema a otro sistema.</p>	<p>Modularidad Similitud de comunicación Similitud de datos Independencia del sistema Independencia de la máquina</p>
	<p>Portabilidad: Esfuerzo requerido para transferir un programa desde un entorno a otro.</p>	<p>Auto-descripción Independencia del sistema Independencia de la máquina</p>

B. Modelo FURPS

Este modelo fue desarrollado por Hewlett-Packard en el año 1987. En él se desarrollan un conjunto de factores de calidad de software, bajo el acrónimo de FURPS: funcionalidad (Functionality), usabilidad (Usability), confiabilidad (Reliability), desempeño (Performance) y capacidad de soporte (Supportability), así como un subconjunto de criterios por cada factor [10]. La estructura del modelo se detalla en la tabla 2.

Tabla 2: Estructura Modelo FURPS

Factores	Criterios
Funcionalidad	Características y capacidades del programa. Generalidad de las funciones Seguridad del Sistema.
Usabilidad	Factores humanos Factores estéticos Consistencia de la interfaz Documentación
Confiabilidad	Frecuencia y severidad de fallos Exactitud de las salidas Tiempo medio de fallos Capacidad de recuperación ante fallos Capacidad de predicción
Rendimiento	Velocidad de procesamiento Tiempo de respuesta Consumo de recursos Rendimiento efectivo total Eficacia
Capacidad de Soporte	Extensibilidad Adaptabilidad Capacidad de Prueba Capacidad de configuración Compatibilidad Requisitos de instalación

Plus (+)	<p>Restricciones de diseño: Limitan las posibilidades para diseñar un sistema.</p> <p>Restricciones de implementación: Se refieren a las reglas para la programación, como la utilización específica de un lenguaje, o apegarse a ciertos estándares.</p> <p>Restricciones de interface: Indican elementos externos con los que el sistema debe interactuar.</p> <p>Restricciones físicas: Se refieren a indicaciones para el hardware.</p>
-----------------	---

C. Modelo Boehm

Este modelo propone una jerarquía de niveles, en forma de árbol con tres ramas principales, que permiten que el software sea de utilidad: Portabilidad, Facilidad de Uso y Facilidad de Mantenimiento. Se estructura en tres niveles: Aplicaciones primarias, Construcciones Intermedias (factores) y Construcciones Primitivas, y finalmente las Métricas que determinan los valores para los criterios (construcciones primitivas) [11]. La tabla 3 contiene la estructura del modelo.

Tabla 3: Factores y criterios – Modelo Boehm

Factores	Criterios
Portabilidad	Independencia dispositivos Compleitud
Fiabilidad	Compleitud Exactitud Consistencia
Eficiencia	Eficiencia dispositivo Accesibilidad
Ingeniería humana	Accesibilidad Comunicatividad Estructuración Auto-descripción
Comprensibilidad	Consistencia Estructuración Auto-descripción Concisión Legibilidad Expansibilidad
Modificabilidad	Estructuración

D. Estándar ISO/IEC 9126

El estándar ISO/IEC 9126 fue formulado inicialmente en 1991 estableciendo un modelo de calidad y su uso como marco para la evaluación de software. En esta norma se distingue entre calidad interna y calidad externa, y se introduce también el concepto de calidad en uso; esta norma es una de las normas ISO que goza de más reconocimiento dentro de la comunidad y tiene como fundamento modelos de calidad aportados por diversas investigaciones realizadas en los últimos 30 años para la caracterización de la calidad del producto software.

Es aplicable a todo tipo de software, se basa en un modelo jerárquico de tres niveles: *características*, *subcaracterísticas* y *métricas*. En el primer nivel tiene seis características principales: [5] Funcionalidad, Fiabilidad, Eficiencia, Facilidad de mantenimiento, Portabilidad y Facilidad de Uso.

Estas características (factores) están compuestas a su vez por 27 subcaracterísticas (subfactores) relacionadas con la calidad externa, y 21 subcaracterísticas relacionadas con la calidad interna [12]. A continuación, la tabla 4 describe la composición del estándar.

Tabla 4: Criterios asociados a factores de calidad – ISO/IEC 9126

Factores	Criterios
Funcionalidad	Adaptabilidad Exactitud Interoperabilidad Seguridad
Usabilidad	Comprensibilidad Aprendizaje Operatividad Atractivo
Mantenibilidad	Análisis Cambio Estabilidad Prueba
Fiabilidad	Madurez Tolerancia a fallos Recuperabilidad
Eficiencia	Comportamiento del tiempo Uso de los recursos
Portabilidad	Adaptabilidad Instalación Coexistencia Reemplazo

E. Estándar ISO/IEC 25010

ISO/IEC 25000, conocida como SQuaRE (*System and Software Quality Requirements and Evaluation*), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software. Resulta de la evolución de normas anteriores, como ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. [3]

ISO/IEC 25010 forma parte de ISO/IEC 2501n – *División de Modelo de Calidad*. Constituye una de las normas que presentan modelos de calidad detallados incluyendo características para calidad interna, externa y en uso del producto software; específicamente describe el modelo de calidad para el producto software y para la calidad en uso. Esta norma presenta las características y subcaracterísticas de calidad frente a las cuales evaluar el producto software. [4]. Este modelo de calidad del producto definido por la ISO/IEC 25010 se compone de 8 características de calidad.

Tabla 5: Características y Subcaracterísticas del Modelo ISO/IEC 25010

Características	Subcaracterísticas
Adecuación funcional	Complejidad funcional Corrección funcional Pertinencia funcional
Eficiencia de desempeño	Comportamiento temporal Utilización de recursos Capacidad
Compatibilidad	Coexistencia Interoperabilidad
Usabilidad	Inteligibilidad Aprendizaje Operabilidad Protección frente a errores de usuario Estética Accesibilidad
Fiabilidad	Madurez Disponibilidad Tolerancia a fallos Capacidad de recuperación
Seguridad	Confidencialidad Integridad No repudio Autenticidad Responsabilidad
Mantenibilidad	Modularidad Reusabilidad Analizabilidad Capacidad de ser modificado Capacidad de ser probado
Portabilidad	Adaptabilidad

Características	Subcaracterísticas
	Facilidad de instalación Capacidad de ser reemplazado

2.4 Requerimientos y necesidades organizacionales sobre la calidad de la información del MTH

Con la finalidad de recabar técnicamente las necesidades que la UTN respecto a la calidad de información, se aplicó entrevistas a 6 usuarios del MTH que principalmente ejecutan acciones de creación, registro, actualización y reportería del personal docente y administrativo de la institución; se entrevistó también al director del Departamento de Desarrollo Tecnológico e Informático (DDTI) y al administrador del MTH.

Las entrevistas se enfocaron, en primera instancia, a detectar el nivel de satisfacción de usuarios, posteriormente se brindó una breve explicación de factores o características de calidad que reincidieron en los 5 modelos y estándares analizados en la sección anterior. Para el director y programador del DDTI se orientó la entrevista de manera más técnica debido a su experiencia y formación en el área.

De los dos grupos de entrevistados se obtuvo las siguientes características o factores como necesidades organizacionales para garantizar la calidad de información del MTH.

Tabla 6: Determinación de Factores / Características reincidentes de los 5 modelos analizados

McCall	FURPS	BOEHM	ISO/IEC 9126	ISO/IEC 25010
Corrección	Capacidad de Soporte	Comprensibilidad	Eficiencia	Adecuación funcional
Eficiencia	Confiabilidad	Eficiencia	Fiabilidad	Compatibilidad
Facilidad de Prueba	Funcionalidad	Fiabilidad	Funcionalidad	Eficiencia de desempeño
Fiabilidad – Confiabilidad	Plus (+) - Restricciones	Ingeniería humana	Mantenibilidad	Fiabilidad
Flexibilidad	Rendimiento	Modificabilidad	Portabilidad	Mantenibilidad
Integridad	Usabilidad	Portabilidad	Usabilidad	Portabilidad
Interoperabilidad				Seguridad
Mantenibilidad				Usabilidad
Portabilidad				
Reusabilidad				
Usabilidad				

A continuación, los 6 factores / características definidas por su reincidencia y consideradas en las entrevistas: Eficiencia, Confiabilidad, Mantenibilidad, Portabilidad, Usabilidad, Funcionalidad.

Cabe mencionar que de las entrevistas aplicadas se obtuvo como resultado la asignación de prioridades a los seis factores / características según los entrevistados, los cuales se presentan en la tabla 7.

Tabla 7: Factores / Características de Calidad de los modelos analizados por prioridad según entrevistados

Factores / Características	Prioridad
Confiabilidad	1
Eficiencia	3
Funcionalidad	2
Mantenibilidad	6
Portabilidad	5
Usabilidad	4

3 Resultados

3.1 Selección del Modelo de Calidad de Software

Se contabilizó los criterios o subcaracterísticas de cada uno de los cinco modelos analizados, obteniéndose los siguientes datos:

Tabla 8: Criterios / Subcaracterísticas de Calidad de los modelos analizados

McCall	FURPS	BOEHM	ISO/IEC 9126	ISO/IEC 25010
23	27	12	20	31

Se distingue que el modelo definido por ISO/IEC 25010, abarca el mayor número de subcaracterísticas; además, según la minuciosa conclusión del *Estudio 2*, citado en la sección de comparativa de modelos de este trabajo, y tomando en cuenta que es una norma producto de la evolución de normas anteriores; se determina la adopción para su aplicación en la evaluación del MTH del SIIU.

3.2 Evaluación del MTH aplicando el Modelo de Calidad ISO/IEC 25010

Para efectos de este trabajo y tomando en cuenta los seis factores priorizados por los entrevistados, se elaboró seis instrumentos de evaluación en base a la norma ISO/IEC 25010, uno para cada característica y se ejecutó la evaluación en común acuerdo con el Director DDTI y administrador del MTH.

Las características Funcionalidad y Usabilidad fueron factibles de evaluar de manera externa, es decir, sin revisar el código fuente del MTH.

A continuación, una breve descripción del MTH en la figura 2.

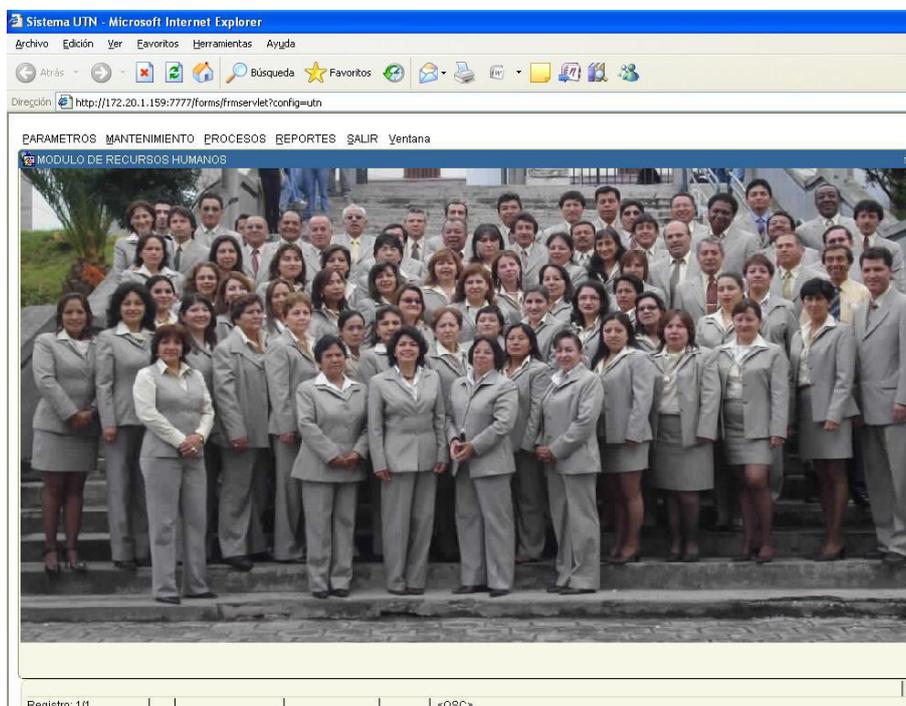


Fig. 2 Pantalla principal del MTH – SIIU

Menú “Parámetros”: Despliega listados iniciales necesarios como parámetros de otros menús y submenús del MTH, como: tipos de escalafones, tipos de categorías, abreviaturas de títulos, tipos de servicios, tipos de fondos, cargos, tipos de autoridades, autoridades.

Menú “Mantenimiento”: Despliega pantallas que maneja la Secretaria del Departamento de Talento Humano, para ingreso de información de todo el personal docente y administrativo de la UTN.

Menú “Procesos”: Despliega pantallas que maneja el Analista de Talento Humano, para contratación y acciones de personal.

Menú “Reportes”: Despliega pantallas de reportes que maneja tanto la/el Director como la Secretaria de Talento Humano.

Como parte fundamental de este trabajo se presenta en la tabla 9, un resumen ejecutivo de los siguientes resultados:

Tabla 9: Resultados del MTH evaluado en base al modelo ISO/IEC 25010

Característica	Hallazgos		Propuesta de mejora
	Total	Cumplidos	
Confiabilidad	12	9	Elaborar y cumplir un plan de contingencia para prever la disponibilidad de hardware y software adecuado y suficiente que abastezca y garantice el correcto y oportuno funcionamiento, recuperación, madurez y tolerancia a fallos del SIIU.
Funcionalidad	10	10	Se sugiere realizar sondeos anuales o semestrales a fin de determinar modificación de opciones del MTH en base a usuarios.
Eficiencia	10	7	El diseño de la base de datos requiere ser revisado y orientado a la no redundancia de datos. Los reportes del MTH deberán ser ampliados y orientados a las crecientes necesidades de la institución, autoridades y usuarios internos y externos.
Usabilidad	8	6	Incluir validaciones en campos de ingreso de datos con la finalidad de restringir el registro incorrecto. Garantizar el acceso al MTH 24/7 mediante el uso de herramientas de recuperación ante desastres u otros eventos.
Portabilidad	5	4	El uso de la máquina virtual de java dificulta la portabilidad de todo el SIIU, en vista que debido al uso e inversión de la UTN en tecnología Oracle, no es posible realizar una sugerencia de mejora viable a pesar de esta limitante.

Característica	Hallazgos		Propuesta de mejora
	Total	Cumplidos	
Mantenibilidad	4	4	Fomentar un plan de pruebas periódicas de los módulos integrantes del SIIU.

3.3 Aplicación de la Propuesta de Mejora

Una vez establecida la propuesta de mejora e informado al DDTI los resultados del proceso de evaluación del MTH, se procedió por parte del administrador del módulo en coordinación con el Director DDTI a la ejecución de sugerencias y recomendaciones alineadas con el modelo ISO/IEC 25010.

Esta aplicación tuvo una duración de 4 semanas; para evidenciar la viabilidad funcional de esta implantación de presenta en la tabla 10, los resultados obtenidos de los entrevistados en lo referente a su satisfacción inicial y actual como usuarios del MTH.

Tabla 10: Nivel de satisfacción inicial y actual de usuarios del MTH en referencia a características del modelo ISO/IEC 25010

Característica	Nivel de satisfacción inicial	Nivel de satisfacción actual
Confiabilidad	Medianamente satisfactorio	Satisfactorio
Funcionalidad	Satisfactorio	Satisfactorio
Eficiencia	Medianamente satisfactorio	Satisfactorio
Usabilidad	Medianamente satisfactorio	Satisfactorio
Portabilidad	Medianamente Satisfactorio	Medianamente Satisfactorio
Mantenibilidad	Satisfactorio	Satisfactorio

4 Conclusiones

La implantación del concepto de calidad de software a través de modelos o estándares, constituye la base para garantizar la calidad de todo el SIIU y lo que organizacionalmente a la UTN le interesa, contar con información de calidad.

Se logró evidenciar, que una adecuada revisión y análisis de modelos de calidad de software en base a los requerimientos organizacionales de la UTN, aportó al incremento del nivel de satisfacción de usuarios del módulo y por lo tanto de la alta dirección.

La UTN garantiza la calidad de información del MTH para los distintos procesos en los que incurra en lo posterior.

La aplicabilidad del modelo de calidad de software ISO/IEC 25010 está plenamente comprobada y se estima conveniente su seguimiento a través de un plan de aseguramiento de calidad.

Agradecimientos

El agradecimiento a la Universidad Técnica del Norte, institución que ha brindado su apoyo y apertura para realizar esta investigación, a los Departamentos de Talento Humano y de Desarrollo Tecnológico e Informático, así como al Instituto de Posgrado.

Referencias

1. Scalone, F.: “Estudio comparativo de los modelos y estándares de calidad del software”. Director: R. García. Tesis de maestría en ingeniería en calidad, Universidad Tecnológica Nacional, Facultad Regional de Buenos Aires, Buenos Aires, Argentina, (2006).
2. Moreno, J.; Bolaños, L.; Navia, M.: “Exploración de Modelos y Estándares de calidad para el producto software”. Universidad Industrial de Santander. (2010).
3. Portal ISO 25000. “ISO 25000 Calidad del Producto de Software”. Sección: La familia de normas ISO/IEC 25000. <http://iso25000.com/index.php/normas-iso-25000>
4. Portal ISO 25000. “ISO 25000 Calidad del Producto de Software”. Sección: ISO/IEC 25010. <http://iso25000.com/index.php/normas-iso-25000/iso-25010>
5. Constanzo, M.: “Comparación de Modelos de Calidad, factores y metricas en el ámbito de la Ingeniería del Software”. Universidad Nacional de la Patagonia Austral - Unidad Académica Río Gallegos, España, (2014).
6. Calero, C.; Piattini, M.; Moraga, M.: “Calidad del producto y proceso software”. Madrid: Editorial Ra-Ma., (2010).
7. González A., André M., Hernández A.: “Análisis comparativo de modelos y estándares para evaluar la calidad del producto de software”. Revista Cubana de Ingeniería, (2015).
8. Lozano, E.: “Cuadro comparativo de los modelos de calidad”. Bucaramanga, Santander, Colombia: Universidad de Santander, (2015).
9. McCall, J.A., Cavano, J.P.: “A Framework for the Measurement of Software Quality”. ACM Software Quality Assurance Workshop, (1978).
10. Olsina, L.: “Ingeniería Web; Marco de medición y evaluación de calidad.” Departamento de informática. Universidad Nacional de San Luis - La Rioja – Catamarca, (2007).
11. Boehm, B.: “Characteristics of Software Quality”. North-Holland Publishing, (1981).
12. ISO/IEC 9126: “Software Engineering - Product quality.” International Organization for Standardization, (2000).