



FACULTAD DE INGENIERÍA Y ARQUITECTURA
SECCIÓN DE POSGRADO

**MÉTODO PARA LA EVALUACIÓN DE CALIDAD DE SOFTWARE
BASADO EN ISO/IEC 25000**

PRESENTADA POR
EDÚ JAMES BALDEÓN VILLANES

**TESIS
PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO EN COMPUTACIÓN
Y SISTEMAS CON MENCIÓN EN GESTIÓN DE
TECNOLOGÍAS DE INFORMACIÓN**

LIMA – PERÚ

2015



**Reconocimiento - No comercial - Compartir igual
CC BY-NC-SA**

El autor permite transformar (traducir, adaptar o compilar) a partir de esta obra con fines no comerciales, siempre y cuando se reconozca la autoría y las nuevas creaciones estén bajo una licencia con los mismos términos.

<http://creativecommons.org/licenses/by-nc-sa/4.0/>



USMP
UNIVERSIDAD DE
SAN MARTIN DE PORRES

**FACULTAD DE
INGENIERÍA Y ARQUITECTURA**

**ESCUELA PROFESIONAL DE INGENIERÍA DE COMPUTACIÓN Y
SISTEMAS**

**MÉTODO PARA LA EVALUACIÓN DE CALIDAD DE
SOFTWARE BASADO EN ISO/IEC 25000**

TESIS

**PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO EN
COMPUTACIÓN Y SISTEMAS CON MENCIÓN EN GESTIÓN DE
TECNOLOGÍAS DE INFORMACIÓN**

PRESENTADO POR

BALDEÓN VILLANES, EDÚ JAMES

LIMA - PERÚ

2015

Dedico a Dios por ser mi fortaleza y a mis padres por ser el ejemplo a seguir en cada paso que emprendo y así lograr mis metas.

Agradezco a mis asesores, la Dra. Sussy Bayona Oré y el Mg. Luis Palacios Quichiz por su apoyo y guía constante en la elaboración del presente trabajo de investigación. Sus conocimientos, sus orientaciones, su paciencia y motivación han sido fundamentales para guiarme en el camino de la investigación.

A los profesionales que me apoyaron con la revisión del método propuesto en este trabajo de investigación.

A Moises Rodriguez Monje, Director de Alarcos Quality Center – España, quien a pesar de su recargada agenda tuvo la gentileza de revisar el presente trabajo y completar el cuestionario de validación del método por expertos en calidad de software.

ÍNDICE

	Página
RESUMEN	xi
ABSTRACT	xii
INTRODUCCIÓN	xiii
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	
1.1 Determinación del Problema	1
1.2 Formulación del Problema	4
1.3 Objetivos	4
1.4 Justificación del Problema	4
1.5 Limitaciones	6
CAPÍTULO II: MARCO TEÓRICO	
2.1 Antecedentes	7
2.2 Bases Teóricas	25
2.3 ¿Porqué un método para la evaluación de calidad del software?	35
2.4 Definiciones de Términos básicos	39
2.5 Hipótesis y Variables	40
CAPÍTULO III. METODOLOGÍA	
3.1 Diseño Metodológico	42
3.2 Población y muestra	43
3.3 Operacionalización de variables	45
3.4 Técnicas de recolección de datos	46
3.5 Técnicas para el procesamiento de la información	46

CAPÍTULO IV: MÉTODO PARA LA EVALUACIÓN DE CALIDAD DEL PRODUCTO SOFTWARE

4.1	Fase 1: Establecer los requisitos de la evaluación	53
4.2	Fase 2. Especificación de la evaluación	61
4.3	Fase 3. Diseño de la evaluación	68
4.4	Fase 4. Ejecución de la evaluación	72
4.5	Fase 5. Conclusión de la evaluación	79
4.6	Resumen de las Actividades	83

CAPÍTULO V: PRUEBAS Y RESULTADOS

5.1	Resumen Descriptivo	84
5.2	Evaluación de la Normalidad de Datos	90
5.3	Proceso de prueba de hipótesis	94
5.4	Resultados de la validación del método por expertos en calidad de software	98

CAPÍTULO VI: DISCUSIÓN Y APLICACIONES

6.1	Discusión de los resultados	99
-----	-----------------------------	----

CONCLUSIONES 102

RECOMENDACIONES 104

FUENTES DE INFORMACIÓN 106

ANEXOS 112

ÍNDICE DE TABLAS

	Página
Tabla 1 Breves detalles de los modelos estudiados	20
Tabla 2 Problemas en los modelos de calidad	24
Tabla 3 Vacíos encontrados en el campo de calidad del software	36
Tabla 4 Matriz de consistencia	41
Tabla 5 Variable independiente	45
Tabla 6 Métricas de las variables dependientes	45
Tabla 7 Entregables que serán evaluados por el método de calidad	48
Tabla 8 Entregables y los objetivos de la evaluación de calidad	50
Tabla 9 Roles que participan en la evaluación de calidad	51
Tabla 10 Necesidades de la evaluación de calidad de cada entregable	53
Tabla 11 Ejemplos de dominios de aplicación de software	55
Tabla 12 Entregables a evaluar en cada iteración del método	57
Tabla 13 Características y subcaracterísticas de calidad	57
Tabla 14 Niveles de importancia	60
Tabla 15 Referencia cruzada entre la información necesaria y los productos a evaluar	64
Tabla 16 Métrica de calidad para la subcaracterística Complejidad Funcional	66
Tabla 17 Métricas de calidad para la subcaracterística Idoneidad Funcional	66

Tabla 18 Métricas de calidad para la subcaracterística Madurez	67
Tabla 19 Ejemplos de métodos de evaluación según la característica de calidad seleccionada	67
Tabla 20 Ejemplo de detalle del componente Documento de Análisis	76
Tabla 21 Ejemplo de detalle del componente Código fuente del software	76
Tabla 22 Ejemplo de detalle del software utilizado en la evaluación de calidad	77
Tabla 23 Ejemplo de detalle del software utilizado en la evaluación de calidad	78
Tabla 24 Ejemplo de detalle del software utilizado en la evaluación de calidad	79
Tabla 25 Resumen de las entradas y salidas del procedimiento de evaluación	83
Tabla 26 Escala de Likert de cinco niveles utilizada en el cuestionario	98
Tabla 27 Muestra de proyectos de desarrollo de software	113
Tabla 28 Respuestas de los expertos en calidad de software	120

ÍNDICE DE FIGURAS

	Página
Figura 1 El proceso “W-process”	9
Figura 2 Relación entre la serie de normas ISO/IEC 9126 y la ISO/IEC 14598	14
Figura 3 El proceso SREP	16
Figura 4 Calidad del producto software y los estándares relacionados a través del ciclo de vida de desarrollo	17
Figura 5 Concepto de requisito de calidad y evaluación de calidad	19
Figura 6 Proceso general de la evaluación de calidad del producto	19
Figura 7 Modelos de calidad propuestos hasta el año 2011	20
Figura 8 Esquema general de un modelo de la calidad del producto	26
Figura 9 Calidad en el ciclo de vida	27
Figura 10 Modelo del ciclo de vida de calidad del producto software	27
Figura 11 Modelo de calidad del producto	29
Figura 12 Modelo de calidad en uso del producto software	30
Figura 13 Modelo de referencia general SQuaRE	31
Figura 14 Relación entre las propiedades para cuantificar, Método de medición y Elementos de medida de calidad (QME)	33
Figura 15 Diseño experimental de la investigación	43
Figura 16 Entregables a evaluar según el modelo de desarrollo en cascada	49
Figura 17 Fases de la evaluación de calidad	52

Figura 18 Plantilla del documento de Requisitos de Evaluación de Calidad	61
Figura 19 Plantilla del documento de Especificaciones de la Evaluación de Calidad	68
Figura 20 Plantilla de plan de trabajo de la evaluación de calidad	71
Figura 21 Plantilla del documento de Diseño de Evaluación de Calidad	72
Figura 22 Plantilla de reporte de evaluación	82
Figura 23 Gráfico descriptivo de la duración de los proyectos	84
Figura 24 Histograma de frecuencias para la duración de los proyectos	85
Figura 25 Frecuencia de la duración de los proyectos	85
Figura 26 Gráfico descriptivo de la cantidad de observaciones en los proyectos	86
Figura 27 Distribución de la cantidad de observaciones en los dos grupos de proyectos	86
Figura 28 Gráfico descriptivo de la cantidad de errores en los proyectos	87
Figura 29 Distribución de la cantidad de errores en los dos grupos de proyectos	88
Figura 30 Gráfico descriptivo de la cantidad de reprocesos en los proyectos	89
Figura 31 Distribución de la cantidad de reprocesos en los dos grupos de proyectos	89
Figura 32 Número de observaciones con la metodología normal de desarrollo, sin el método de evaluación de calidad	90
Figura 33 Número de observaciones con la aplicación del método de evaluación de calidad basado en ISO/IEC 25000	91
Figura 34 Número de errores sin el método de evaluación de calidad	91
Figura 35 Número de errores con el método de evaluación de calidad	92
Figura 36 Número de reprocesos sin el método de evaluación de calidad	93

Figura 37 Número de reprocesos con el método de evaluación de calidad	93
Figura 38 Resultados de la prueba t-Student para la hipótesis general	94
Figura 39 Resultados de la prueba Mann-Whitney para la hipótesis específica 1	96
Figura 40 Resultados de la prueba Mann-Whitney para la hipótesis específica 2	97
Figura 41 Resultados de la pregunta alineada al objetivo general	98
Figura 42 Evidencia de la bitácora de pases al ambiente de aceptación por el usuario (UAT)	114
Figura 43 Evidencia del reporte de errores encontrados en el ambiente de producción	115
Figura 44 Formato de la solicitud de validación dirigida a expertos en calidad de software	116
Figura 45 Cuestionario de validación del método dirigido a expertos en calidad de software	119

RESUMEN

La gestión de la calidad del producto es un factor crítico de éxito en los proyectos de desarrollo de software. En este sentido, el presente trabajo de investigación propone un método basado en ISO/IEC 25000 (2005) para evaluar la calidad de los entregables de un proyecto. Este método proporciona los lineamientos necesarios para contribuir al incremento de la calidad del producto final y asegurar el cumplimiento de los requisitos del usuario.

En esta investigación, se revisa la literatura relacionada al estudio, se muestra el análisis de la norma ISO/IEC 25000 (2005) y sus principales divisiones; luego se detalla el método propuesto para evaluar la calidad del producto software considerando los entregables desde la etapa de análisis. Finalmente, el método se aplica en una muestra representativa de proyectos, llegando a demostrar que su aplicación durante el ciclo de vida del software mejora la calidad del producto final, facilita la conformidad por parte del usuario y disminuye los errores después de su puesta en producción.

Palabras Claves:

Calidad de software, Evaluación de calidad, ISO/IEC 25000.

ABSTRACT

The management of the quality product is a critical factor of the success in software development projects. In this regard, this research paper proposes a method based on ISO / IEC 25000 (2005) in order to evaluate the quality of the project deliverables. This method provides the necessary guidelines to contribute the increase of the quality of the final product and ensures the fulfillment of the user requirements.

In this research, firstly, the literature related to the project is reviewed as well as is shown the analysis of ISO / IEC 25000 (2005) standard and its major divisions; then it is detailed the proposed method to evaluate the quality of software product considering the deliverables from its analysis stage. Finally, this method is applied on a representative sample of projects; showing that its application during the software lifecycle improves the quality of the final product, facilitates the user acceptance and reduces the errors after being set in production.

Keywords

Software Quality, Quality Assessment, ISO/IEC 25000.

INTRODUCCIÓN

Actualmente existen estándares relacionados a temas de calidad, sin embargo no se utilizan en todo su potencial; los trabajos revisados se limitan a aplicar algunas métricas y procedimientos a casos particulares y no cubren todo el ciclo de vida del producto software. La calidad de este debe ser vista de forma integral, desde el análisis hasta su implementación y debe gestionarse en función a los requisitos de calidad definidos por los interesados, ya que el cumplimiento de estos indicará el nivel de calidad del producto.

La encuesta de la Pontificia Universidad Católica del Perú - PUCP (2005) encontró que en el 40% de los proyectos de software se descubren defectos en sus etapas; asimismo, el reporte CHAOS (2013) muestra que un 43% de proyectos de software tuvo problemas de tiempos, costos y/o no completaron la funcionalidad que originalmente se especificó. Ello evidencia problemas de calidad de software no solamente en el producto final del proyecto sino en los entregables de las etapas tempranas del ciclo de vida del producto.

En las empresas, la calidad generalmente está ligada a las pruebas en la etapa de testing, lo cual es insuficiente para considerar esta actividad como una evaluación completa. La calidad del producto software debe ser vista de una forma integral, y debe validar los entregables de cada etapa del ciclo de vida, para identificar en forma temprana problemas de calidad; todo ello con

participación de los interesados, que es el principal factor de éxito en un proyecto de software según lo indica el reporte CHAOS (2013).

Este trabajo de investigación propone desarrollar **un método que permite guiar la evaluación de calidad de software basado en la ISO/IEC 25000, para mejorar su calidad y asegurar el cumplimiento de los requisitos del usuario y del negocio.**

El presente trabajo ha sido dividido en seis capítulos: en el Capítulo I se plantea el problema, los objetivos, la justificación y las limitaciones de la investigación. En el Capítulo II se presenta el Marco Teórico, que considera una revisión de los antecedentes y las bases teóricas utilizadas en el desarrollo del proyecto de investigación. En el Capítulo III se explica el modelo metodológico utilizado para validar las hipótesis. En el Capítulo IV se detalla el método propuesto de evaluación de calidad del producto, considerando sus fases y los componentes del producto software que se consideran en el proceso de evaluación. En el Capítulo V se muestran los resultados de la investigación. En el Capítulo VI se presenta la discusión de los resultados. Finalmente se explican las conclusiones y recomendaciones.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Determinación del Problema

Uno de los problemas en la industria del software, se da cuando las empresas culminan un proyecto de software y publican un software nuevo o actualizan la versión del producto; ya que una vez iniciada la etapa de operación en un ambiente real se detectan errores, errores que muchas veces son encontrados por los usuarios finales ocasionando una percepción negativa de la calidad del software.

En un estudio PUCP (2005) se encontró que en 40% de los proyectos de software se descubren defectos en todas sus etapas (análisis de requisitos, diseño de alto nivel, diseño detallado, codificación, pruebas, instalación y operación). La mayoría de estos se encuentran: (1) en la parte lógica de negocios y (2) la interfaz de usuarios. Asimismo, el estudio refleja que las organizaciones que desarrollan software tienen un alto uso de las pruebas de tipo aleatorias, lo cual indica que no se sigue un método formal para las pruebas del producto software y a su vez no se garantiza la mejora de la calidad del producto final.

Nasir y Sahibuddin (2011), en su investigación sobre los factores críticos de éxito en los proyectos de software, encontraron que solamente 32% de proyectos termina satisfactoriamente. Entre los problemas identificados se encuentran: (1) la inadecuada especificación de requerimientos en la etapa de

análisis y (2) la falta de claridad en los objetivos del proyecto. Esto conlleva a que el producto final (software) tenga deficiencias de calidad. Asimismo, según el estándar ISO/IEC 25000 (2005), la calidad del producto no solamente debe ser vista como una actividad exclusiva de la fase de pruebas de software, sino también debe considerar la evaluación de los entregables del producto durante su ciclo de vida.

El reporte CHAOS (2013) muestra que solo el 39% de los proyectos de software terminan satisfactoriamente dentro de los tiempos, costos y con la funcionalidad inicialmente definida. El 43% tuvo problemas de tiempos, costos y/o no completaron el alcance definido; en tanto que, el 18% de proyectos fueron cancelados. De la misma forma, el reporte explica que el principal factor para que un proyecto se cancele son los requisitos incompletos, y también es el segundo factor que influye en los proyectos con problemas en los tiempos, costos y alcance. También se menciona que dos de los principales factores de éxito en los proyectos de software son: (1) el soporte del sponsor del proyecto y (2) el involucramiento del usuario.

Si se considera esta realidad y el hecho de que los usuarios finales y el sponsor son los que definen los requisitos del producto, el no contar con su participación y soporte impacta directamente en un nivel inadecuado de la calidad de los requisitos y del producto software final. Consecuentemente, un nivel de calidad inadecuado en el producto software se verá reflejado en errores, y para corregir estos, se debe realizar correcciones en el producto, lo cual conlleva a invertir en recursos y tiempo, ocasionado sobrecostos a la empresa que desarrolló el software. Según Seaman y Guo (2011), estos sobrecostos se originan por la “deuda técnica” asumida en el desarrollo del producto (artefactos inmaduros, incompletos o inadecuados en el ciclo de vida del desarrollo del software).

Finalmente, los errores del software no solo afectan a quien lo desarrolló, sino principalmente al cliente, ya que su servicio podría estar comprometido, así como la integridad de su información y su reputación.

La problemática se resume en lo siguiente:

- Calidad insuficiente en los productos de software terminados, CHAOS (2013), Kaur y Sengupta (2011).
- Inadecuada gestión de las restricciones de los proyectos de software, se prioriza tiempos, alcance y costos de desarrollo en lugar de calidad de software. Según Zazworka, Shaw, Shull, Seaman (2011) este tipo de priorización siempre tienen un impacto negativo en la calidad del producto y por lo tanto debe ser gestionado y manejado durante el desarrollo del producto. Según Kaur y Sengupta (2011), en la etapa de pruebas de aceptación (etapa de testing) y pruebas de conformidad por el usuario, no se capturan todos los errores debido a: (1) que las pruebas no se planifican, (2) el personal no está capacitado o (3) el tiempo asignado para pruebas no es el adecuado debido a que el proyecto está retrasado (se prioriza tiempo en lugar de calidad).
- Sobrecostos debido a asignación de recursos para solucionar errores de software y retrasos en los proyectos de software. Según Seaman y Guo (2011), el hecho de sacrificar una dimensión del desarrollo, como la calidad, ocasiona sobrecostos en el mantenimiento del producto.
- Deficiencia en la evaluación de calidad del producto software en los entregables de las fases tempranas del ciclo de vida del software. El reporte CHAOS (2013) ubica a los requerimientos del producto como el tercer principal factor crítico de éxito, Nasir y Sahibuddin (2011) identifican como principal factor crítico de éxito la claridad de los requisitos y especificaciones, adicionalmente Kaur y Sengupta (2011) encuentran que uno de los factores vitales que causan que un proyecto de software falle es una pobre gestión de calidad de los entregables del producto software, algunas actividades que no se realizan son: (1) revisión

de los requisitos, (2) revisión del diseño, (3) revisión del código fuente.

1.2 Formulación del Problema

1.2.1 Problema general

¿De qué manera un método para la evaluación de calidad mejorará la calidad del software?

1.2.2 Problemas específicos

- ¿Cómo el método de evaluación de calidad disminuirá los errores del software después de la puesta en producción?
- ¿De qué manera el método de evaluación de calidad facilita la conformidad del software por parte del usuario?

1.3 Objetivos

1.3.1 Objetivo General

Mejorar la calidad del software a través de la aplicación de un método para la evaluación de calidad basado en ISO/IEC 25000.

1.3.2 Objetivos Específicos

- Disminuir los errores del software después de su puesta en producción, a través de la aplicación de un método para la evaluación de calidad basado en ISO/IEC 25000.
- Facilitar la conformidad del software por parte del usuario, mediante la aplicación de un método para la evaluación de calidad basado en ISO/IEC 25000.

1.4 Justificación del Problema

1.4.1 Justificación Teórica

La serie ISO 25000 es la última actualización de los estándares para la evaluación de calidad de los productos de software; sin embargo, estos no son muy utilizados en nuestro medio y generalmente las

empresas optan por desarrollar sus propios procedimientos basados en la experiencia, la necesidad y centrados en la evaluación de calidad en la etapa de testing.

Hosni y Kirinic (2013) mencionan que ISO/IEC 9126 (2001), ISO/IEC 14598-1 (1999) y relacionados, como ISO/IEC 25000 (2005), no son fáciles de adaptar y utilizar, ya que requieren de cierto grado de experiencia y conocimiento para poder utilizarlas correctamente. Asimismo, indican que hay muchas relaciones y referencias cruzadas, así como hacen mención a distintos ciclos de vida de desarrollo de software, lo cual dificulta su adopción. Como resultado tenemos metodologías de evaluación de calidad propias que no son efectivas; las consecuencias de esta realidad se reflejan, según Seaman y Guo (2011), en sobrecostos que se presentan generalmente después de la puesta en producción del producto software e insatisfacción del usuario.

La investigación presentada en este trabajo propone un método que aborda estas debilidades en el conocimiento teórico actual y facilita el proceso de evaluación de calidad. Del mismo modo, este aporte mejorará las ausencias en el campo de la calidad de software, coincidiendo con lo mencionado por Hwang (2014), que al notar ausencias en la ingeniería del software, propone contenidos esenciales para la educación del proceso de desarrollo de software y calidad de software. Hwang (2014) agrega que hay muchos problemas en los proyectos de software, como sobrecostos y degradación de la calidad; por ello considera importante para la industria del software y la educación mejorar el conocimiento y habilidades de la mano de obra de ingeniería del software, puntos importantes para mejorar la calidad y el desarrollo de la industria del software.

El presente trabajo **mejora el conocimiento teórico existente, ya que desarrolla un método que detalla las fases de evaluación de calidad en los proyectos de software, consecuentemente facilita la evaluación de la calidad** del software no solamente cuando el producto está terminado (software en etapa de testing), sino también incluye a los entregables del ciclo de vida del software (desde la etapa de análisis

hasta la implementación en un ambiente de producción). Este método estará soportado por ISO/IEC 25000 (2005) como estándar aceptado en la industria.

1.4.2 Justificación Práctica

Según los estudios realizados por Seaman, Guo (2011) y Guo et al. (2011) sobre los proyectos de software, es un problema no considerar desde el principio el costo elevado en el que se puede incurrir a largo plazo por los arreglos y modificaciones sobre productos de baja calidad (deuda técnica). Los costos que conlleva **corregir** estos **errores** siempre lo paga el cliente o la empresa que desarrolla el software; generalmente los costos no solamente son monetarios sino también se ve reflejado en la insatisfacción de los clientes y la degradación de la imagen (riesgo reputacional) de la empresa o equipo que desarrolla el software. Asimismo, en el caso de empresas que están bajo el dominio de normas locales, el riesgo de sufrir sanciones por incumplimiento es elevado.

Por lo tanto, se debe considerar **la calidad** como un factor importante que **debe ser gestionado durante el ciclo de vida del proyecto de software y desde su concepción**. Para ello es necesario contar con un método que ayude a realizar esta gestión. El desarrollo y aplicación del mismo será abordado en el presente trabajo.

1.5 Limitaciones

En este estudio no se evaluará las herramientas ni las metodologías utilizadas para el desarrollo del software. El método propuesto estará enfocado únicamente a la evaluación de la calidad del producto software (entregables desde la etapa de análisis hasta la puesta en producción). En concordancia con esta limitación, durante la recopilación de la información no se evaluarán datos relacionados a la calidad de los procesos de desarrollo, ya que partiremos del supuesto que existe un procedimiento de desarrollo que cumple niveles de calidad adecuados, como RUP.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

2.1.1 Modelos de calidad personalizados basados en normas ISO

Los estándares de calidad que publica la Organización Internacional de Normalización (ISO) han sido mejorados en el tiempo por diversos autores; en algunos casos inclusive, han desarrollado extensiones para ser aplicadas a un sector específico. Uno de los primeros estudios en discutir las deficiencias del estándar de calidad ISO/IEC 14598 lo realizaron Kusters, Trienekens, Bemelmans y Brombacher (2004), donde proponen un método orientado a objetivos basados en la norma ISO/IEC 14598, ello debido a que encontraron que este estándar, uno de los pilares de ISO/IEC 25000 (2005), tenía problemas en llevar a la práctica el proceso de evaluación principalmente debido a una insuficiente atención a la definición de los objetivos y a las relaciones implícitas entre actividades.

Algunas deficiencias encontradas por Kusters et al. (2004) aún pueden encontrarse en la nueva serie ISO/IEC 25000.

Ante esta problemática Kusters et al. (2004) propone un proceso llamado “K-process”, que extiende el estándar a través de un procedimiento mejorado y directrices adicionales. El estudio puntualiza que solamente poner foco en las medidas de calidad del producto es insuficiente

para mejorar las evaluaciones del producto software, ya que también es necesario manejar las expectativas de los interesados durante la etapa de formulación de los objetivos, por ello el enfoque debe estar no solamente en la medición de la calidad sino en el **proceso de evaluación de calidad del producto**.

Los problemas identificados por Kusters et al. (2004) en la serie ISO/IEC 14598 son cuatro:

a) Formulación insuficiente del objetivo

Definen una actividad genérica que no respalda la forma de definir los objetivos de la evaluación de calidad de tal forma que sea ejecutado con claridad. Tampoco provee soporte para definir las formas de involucrar a todos los interesados.

b) Relaciones implícitas entre actividades

El estándar provee una clara definición de las actividades y explica qué se debe abordar en cada una de ellas; sin embargo no especifica claramente las relaciones entre las actividades. Esta pérdida de relación hace que las actividades se realicen de forma aislada sin tomar los resultados de las actividades anteriores.

c) No atención al balanceo entre objetivos y recursos

El estándar no provee forma de balancear los recursos con los objetivos de la evaluación de calidad. Si bien el estándar hace mención a módulos de evaluación, esto será insuficiente para soportar un objetivo de la evaluación (por ejemplo un módulo de evaluación puede hacer referencia a la mantenibilidad del software, pero esto será insuficiente para considerar a la mantenibilidad como el factor más importante). Por lo tanto, una relación explícita con el objetivo de la evaluación es necesaria para abordar el costo, tiempo y esfuerzo de los módulos de evaluación.

d) Atención insuficiente en la retroalimentación

Estos estándares no abordan explícitamente el monitoreo y feedback en su definición de proceso de evaluación.

Ante estas falencias Kusters et al. (2004) definen el proceso “K-Process” (Figura 1) que cubre los vacíos encontrados en la serie ISO/IEC 14598, posteriormente aplica este proceso y logra la satisfacción de los participantes.

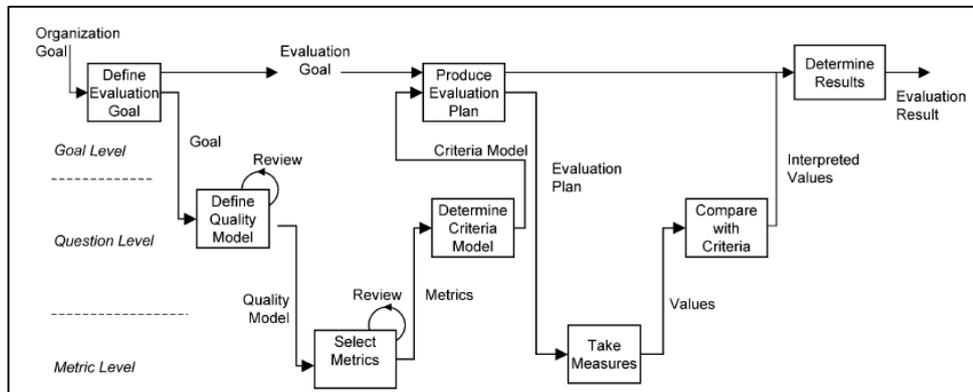


Figura 1 El proceso “W-process”
Fuente: Kusters et al. (2004)

Mellado, Rodríguez, Verdugo, Piattini y Fernández-Medina (2010) presentan el entorno MEDUSAS como un esfuerzo de la cooperación público-privada en España, este marco permitiría ofrecer a las empresas y organismos públicos un conjunto de servicios de evaluación y control de la calidad del software de forma independiente y basada en la serie ISO/IEC 25000. Mellado et al. (2010) justificaron el desarrollo de este marco debido a que la calidad ha sido tratada con más amplitud a nivel de calidad del proceso que a nivel de la calidad del producto, y si bien las áreas de testing (sobre todo funcionalidad) son un campo bien trabajado, todavía no se han desarrollado las técnicas necesarias para evaluar de forma efectiva la calidad y la seguridad de un producto software.

El entorno MEDUSAS desarrollado por Mellado et al. (2010) permite: (1) No solo evaluar la calidad y seguridad del código (software), sino también la calidad y seguridad de su diseño y (2) llevar a cabo la evaluación de la calidad en los entregables del producto durante las fases del proceso de desarrollo de software: análisis (diagramas de casos de uso, estados, etc.), diseño (diagrama de clases, diseño de arquitecturas, etc.), desarrollo (código fuente, documentación, casos de prueba implementados, etc.).

Asimismo, las principales características de este marco son: está formado por un conjunto estructurado de procesos (con entradas y salidas). Está orientado a la relación con el cliente y a la externalización de la evaluación de calidad. Está pensado para ser una metodología adaptativa y está respaldada por un conjunto de modelos y herramientas de medición.

Podemos notar similitud entre los estudios de Kusters et al. (2006) y Mellado et al. (2010), ambos consideran necesario establecer un proceso explícito en el que cada actividad tiene entradas y salidas los cuales deben estar claramente definidos, también destacan la importancia de evaluar la calidad desde etapas previas al desarrollo, y la orientación a los objetivos con participación de los involucrados.

Pasrija, Kumar y Srivastava (2012) propusieron un modelo para cuantificar diversos parámetros de calidad de software basado en diferentes modelos como el Modelo de Boehm, Modelo de McCall y la serie ISO/IEC 9126. El modelo de Pasrija, Kumar y Srivastava (2012) soporta cinco vistas diferentes de la calidad del software y aplica la teoría difusa para medir la calidad del producto de software mediante la cuantificación de los parámetros de calidad que pueden reducir la ambigüedad y la incertidumbre de los atributos de calidad de software. Ellos concluyen que basándose en los resultados del caso de estudio, los tomadores de decisiones pueden entender las cualidades y limitaciones de los productos de software antes de ser desarrollados, adicionalmente pueden mejorar la calidad del producto en general. Se destaca la importancia del concepto y efecto de interacción entre los criterios sobre los resultados, de ahí que estos últimos pueden variar con el grado de interacción.

Las cinco vistas de calidad de software propuestas por Pasrija, Kumar y Srivastava (2012) son: vista de usuario, vista de fabricación, vista del producto, vista basado en valor, vista trascendental. La “vista del usuario” es la percepción del usuario a partir de su interacción con el software, esto permitirá obtener los parámetros de calidad fiabilidad y usabilidad. La “vista de fabricación” es la calidad de producto durante la fase de fabricación y después de la fase de entrega del producto, ello permitirá obtener los

parámetros: calidad de arquitectura, flexibilidad, mantenibilidad, modificabilidad, comprobabilidad y reutilización. Desde la “vista del producto” se identifican la calidad inherente al producto, con ello se obtiene los parámetros de calidad: corrección, eficiencia, funcionalidad, interoperabilidad, portabilidad, seguridad, sistema. La “vista basado en valor” cubre los aspectos económicos y costo, con ello se obtiene parámetros de calidad relacionados al costo y tiempo. La “vista trascendental” afirma que la calidad del producto es aún abstracta y universalmente identificable, y está relacionado al parámetro valor de marca.

2.1.2 Estándares de certificación de calidad de producto software

El campo de la calidad del producto software presenta menor desarrollo en cuanto a certificaciones bajo normas comparado con las certificaciones a nivel de proceso; así lo menciona Rodríguez y Piattini (2012), este estudio coincide con el estudio de Mellado et al. (2010) que indica que la calidad ha estado centrado principalmente en los procesos que se siguen para el desarrollo del software, surgiendo certificaciones bajo normas como CMMI o ISO/IEC 15504 dejando de lado el foco en el producto.

Rodríguez y Piattini (2012) menciona la preocupación por la calidad del propio producto software y no solo por la calidad de los procesos que se utilizan para su desarrollo. Si bien han surgido normas como la familia ISO/IEC 25000, que definen un modelo de calidad y un proceso de evaluación para el producto, son menos extendidas las certificaciones que tomen como objeto el propio producto software y permitan asegurar el cumplimiento por parte de este de un conjunto de requisitos.

El trabajo de Rodríguez y Piattini (2012) identifica 77 estudios relevantes, de estos tan solo 10 se consideraron primarios para ser analizados, de estos estudios se puede extraer que a pesar del interés que existe por la calidad del producto software, el proceso para llevar a cabo su certificación es un campo todavía bastante incipiente en la Ingeniería del Software. Las pocas propuestas de certificación halladas en este estudio

utilizan las series ISO/IEC 9126 e ISO/IEC 14598, sin embargo ninguna de las propuestas ha adoptado todavía la nueva serie de normas ISO/IEC 25000.

2.1.3 Experiencia de evaluación y control de calidad en áreas específicas

Ahamed, Sundaraj, Ahmad, Rahman y Ali (2012), describen un marco de trabajo para la evaluación y control de calidad en el software utilizado en equipos de rehabilitación médica. Los autores consideran que la calidad es un parámetro muy importante y que implica muchos atributos, como la auditoría, la evaluación comparativa, la mejora continua, la satisfacción del cliente, el diseño y desarrollo, el control de calidad y la trazabilidad. En esencia, sostienen que los sistemas utilizados en equipos de rehabilitación médica deben ser en gran medida libre de errores.

Ahamed et al. (2012) consideran que algunas de las principales ventajas de la evaluación del software en los sistemas de rehabilitación médica son:

- Mejoras en la usabilidad
- Satisfacción de los usuarios
- Aporte en la gestión del tiempo
- Software libre de errores

Aunque los autores listaron las principales variables para la evaluación de la calidad de software, se debe considerar que alguna de ellas podría ser más importante que otra, esto depende del objetivo que se quiere lograr y del caso en estudio.

Kuwata, Taketa y Miura (2014) proponen un método de evaluación de calidad para software libre basado en el modelo de madurez de la comunidad de desarrollo de software libre. El objetivo del estudio es estimar la calidad de los productos desarrollados por las comunidades de software libre, el resultado se utiliza para tomar la decisión si adoptan o no el software. El proceso de evaluación inicia cuando se define un conjunto de **indicadores**, seguida de un **marco** para la evaluación de la calidad. El estudio evalúa

únicamente los requisitos no funcionales del software, debido a que la funcionalidad varía de acuerdo al dominio de la aplicación.

El método propuesto por Kuwata, Taketa y Miura (2014) evalúa lo siguiente: (1) La calidad del software, (2) la continuidad del desarrollo y mantenimiento del software y (3) la facilidad de agregar funcionalidad al software, que es medido por el nivel de modularidad de los componentes del software facilidad del mantenimiento. Los autores llegaron a definir niveles de madurez para la comunidad de software libre. Finalmente, como trabajo futuro plantean desarrollar un método para la evaluación de calidad para cada nivel de madurez.

Li y Fan (2012), realizaron un análisis de un sistema de pruebas de software en el campo de la aviación civil. Ellos efectuaron un análisis de cómo se está desarrollando las pruebas de software en los últimos años en la aviación de China, y mencionan que no se le da el tiempo suficiente para garantizar el 100% de la calidad del producto. Los autores también describen las principales características de las pruebas de software para la aviación de China.

Li y Fan (2012), consideran que las principales variables a tomar en cuenta en un sistema de pruebas de software son:

- Tener un sistema estándar de pruebas
- Tener un sistema de documentación de pruebas
- Tener un sistema de aseguramiento de la calidad
- Tener un sistema de gestión de rendimiento
- Tener un sistema de mejora de procesos
- Tener una metodología de pruebas

2.1.4 Normas de calidad del producto software

En el año 1987 la Oficina Internacional para la Estandarización (ISO) y la Comisión Electrotécnica Internacional (IEC) constituyeron un comité técnico conjunto (JTC 1) con la finalidad de desarrollar normas en el campo de las tecnologías de la información.

En el año 1991 se publica la primera versión de la norma internacional ISO/IEC 9126, que constituyó el primer esfuerzo para unificar los términos de calidad referidos al producto software; asimismo, adoptó y mejoró los trabajos de MC Call y Boehm entre otros.

En 1994, se determina la revisión de la norma ISO/IEC 9126, resultado de la revisión, se producen dos series de normas: (1) La serie ISO/IEC 9126 referida al modelo de calidad del producto software y (2) la serie ISO/IEC 14598 referida a la evaluación de la calidad del producto. La publicación completa de ambas series, se iniciaron en julio de 1998 y concluyeron en abril del 2004, habiéndose elaborado cuatro normas en las serie 9126 y seis normas en la serie 14598 que se detallan a continuación:

- 9126-1 Modelo de calidad
- 9126-2 Métricas externas
- 9126-3 Métricas internas
- 9123-4 Métricas de calidad en uso
- 14598-1 Visión General
- 14598-2 Planeamiento y gestión
- 14598-3 Proceso para desarrolladores
- 14598-4 Proceso para compradores
- 14598-5 Proceso para evaluadores
- 14598-6 Documentación de los módulos de evaluación

La relación entre la serie 9126 y la 14598 se muestra en la Figura 2.

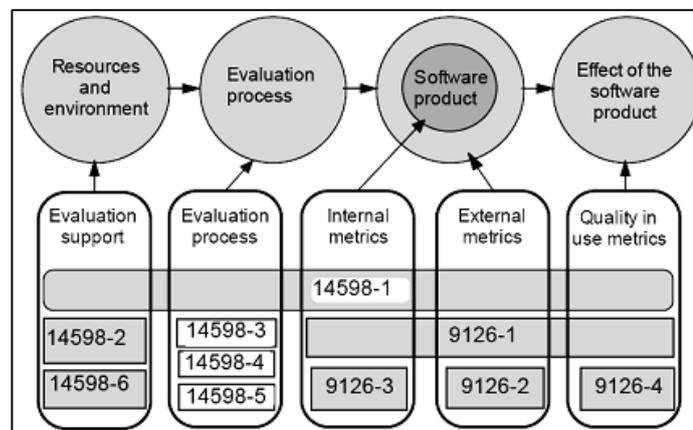


Figura 2 Relación entre la serie de normas ISO/IEC 9126 y la ISO/IEC 14598
Fuente: ISO/IEC 25000 (2005)

Una nueva propuesta de calidad de producto se plantea en 1999 y se aprueba en el 2000. La propuesta se denomina proyecto SQuaRE (es la abreviatura en inglés de Software producto Quality REquirements) con la idea de proponer un nuevo marco de referencia para el tema de calidad de producto software. Las normas SQuaRE están identificadas con la serie 25000 e incluyen completamente a sus predecesoras la serie ISO/IEC 9126 e ISO/IEC 14598.

2.1.5 Calidad de producto software en el ciclo de vida del desarrollo

Marulanda y Ceballos (2012), destacan la importancia de implementar metodologías de desarrollo seguro que sean aplicadas en cada fase del ciclo de vida del software: análisis, diseño, desarrollo y pruebas. Consideran importante evaluar la seguridad desde etapas tempranas del ciclo de vida del software, ello contribuye no solo a generar software de calidad sino de alta seguridad. Los autores recopilan una serie de metodologías y herramientas existentes que manejan la seguridad como parte de la metodología de desarrollo.

Entre estas metodologías revisadas está el proceso SREP (Proceso de Ingeniería de Requisitos de Seguridad) que es una serie de actividades que ayudan a fortalecer y mantener actualizados los requisitos de seguridad durante el ciclo de vida del desarrollo de software y permite mitigar efectivamente los riesgos asociados, la Figura 3 muestra el proceso SREP.

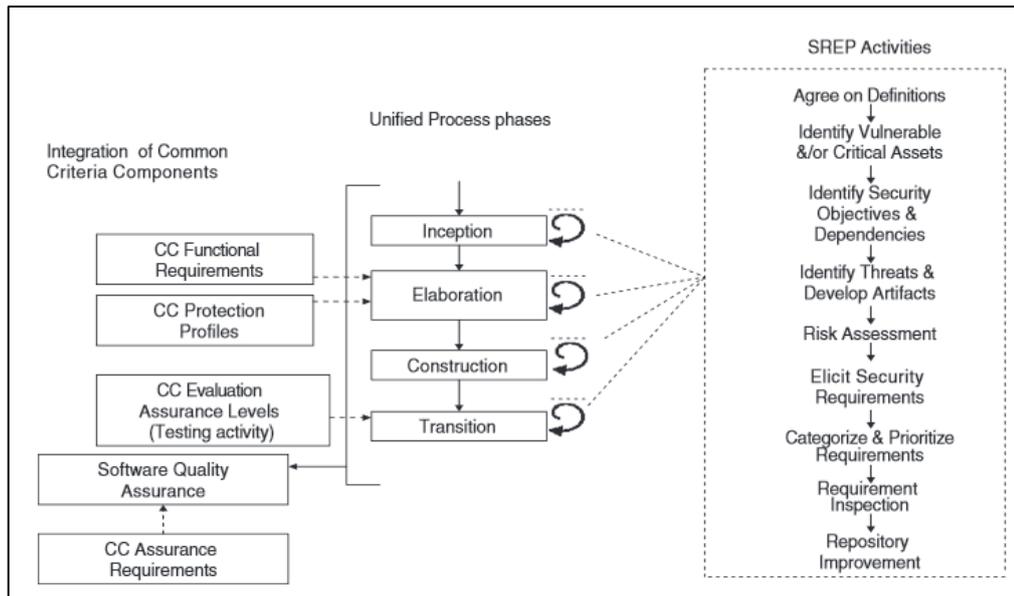


Figura 3 El proceso SREP
Fuente: Ceballos y Marulanda (2012)

El proceso SREP define varios pasos para construir modelos de seguridad desde las etapas tempranas del ciclo de vida del software. Los pasos que considera son:

- Acuerdo en las definiciones
- Identificar metas de seguridad
- Desarrollar artefactos
- Evaluación de los riesgos
- Definición de los requisitos de seguridad
- Clasificar los requisitos
- Priorizar los requisitos
- Revisión de los requisitos

Si bien el estudio de Ceballos y Marulanda (2012) solamente se enfoca en abordar la seguridad desde etapas tempranas del ciclo de vida, podemos encontrar otros estudios como el de Hosni y Kirinic (2013) que describe cómo varios estándares podrían ser usados a través del ciclo de vida del software con la finalidad de planear, revisar, y mejorar su calidad de forma integral. Hosni y Kirinic (2013) parten del estándar ISO/IEC 12207 para identificar los entregables en cada etapa de desarrollo sobre los que se debe evaluar la calidad de software. ISO/IEC 12207 se usa en forma conjunta con

ISO/IEC 15288 para identificar documentos que son necesarios para definir y realizar una evaluación exitosa. Luego, se aplica en forma conjunta la serie ISO/IEC 9126 e ISO/IEC 14598 para realizar la evaluación de calidad de dichos entregables (Figura 4).

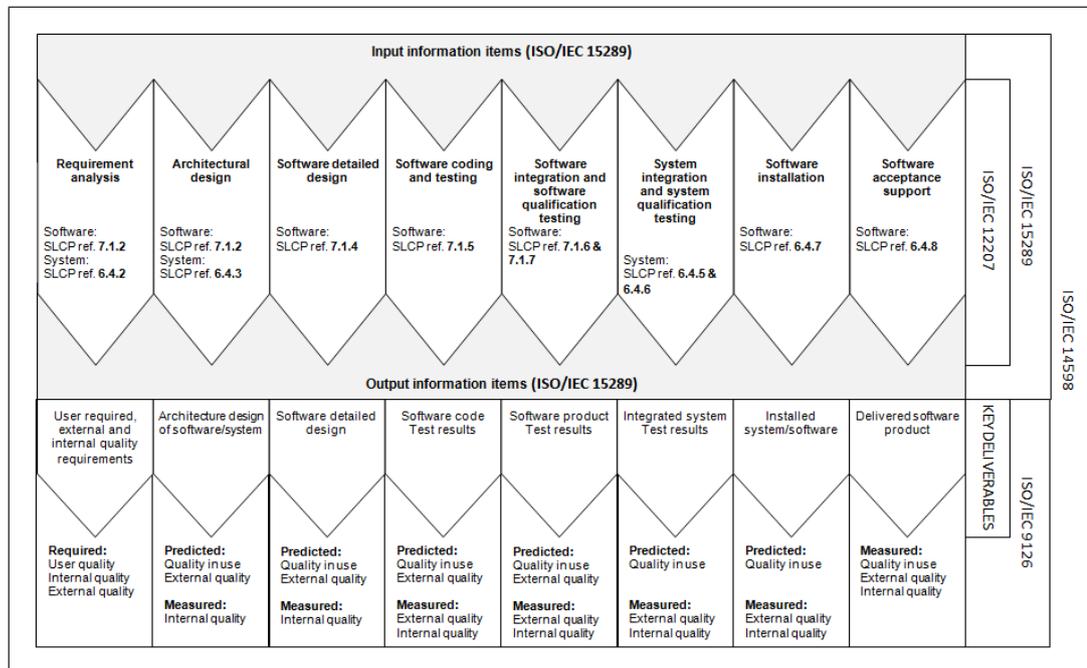


Figura 4 Calidad del producto software y los estándares relacionados a través del ciclo de vida de desarrollo
Fuente: Hosni y Kirinic (2013)

Hosni y Kirinic (2013) concluyen que el uso de ISO/IEC 9126 e ISO/IEC 14598, así como los estándares relacionados, no son fáciles de usar porque hay muchas relaciones entre ellos, muchas referencias cruzadas, distintos tipos de ciclos de vida de software. Puntualiza que proyectos como la serie de normas ISO/IEC 25000 que plantean sincronizar estos estándares son alternativas de solución a este problema. Finalizan indicando que el propósito de un estándar es aclarar las confusiones, dar la dirección correcta, y no complicar las cosas.

En el 2013, Esaki destacó la importancia de los requisitos de calidad y su evaluación al utilizar ISO/IEC 25000.

Esaki (2013) menciona que para el propósito del desarrollo o la adquisición de software con éxito es muy importante especificar los requisitos de calidad y verificar que el producto software corresponde a las

necesidades reales del cliente durante una etapa temprana de desarrollo; esto garantizará la calidad del software, y el cumplimiento de los requisitos de calidad, así como guiará la evaluación del producto software. Este estudio agrega que si tomamos un enfoque equivocado de exigencia de calidad de acuerdo con las necesidades reales del cliente, esto puede causar pérdida económicas, incluso si el proyecto se ha completado sin problemas relevantes.

Esaki (2013) reconoce haber trabajado mucho tiempo con un equipo, en la evolución de ISO/IEC 9126 y en el desarrollo de ISO/IEC25000. Indica que el primero no es útil para el apoyo en la especificación de requisitos durante etapas tempranas del desarrollo, pues no tiene norma correspondiente al análisis de los requisitos de calidad.

Adicionalmente, Esaki (2013) indicó que si se omite definir claramente los requisitos de calidad antes del desarrollo, no se podrá realizar una implementación adecuada que contemple las necesidades reales de los clientes. Esta omisión no podrá cubrirse dentro de las actividades de desarrollo, diseño o evaluación de calidad del software. Por ello con la finalidad de dar soporte a los requisitos de calidad de sistemas y software, se desarrolló el estándar ISO/IEC 25000, que es básicamente la revisión de las series ISO/IEC 9126 e ISO/IEC 14598.

Esaki (2013) agrega que la norma ISO/IEC 25030 (2007), es útil para la especificación de requisitos de calidad en etapas tempranas del desarrollo, pues ayuda a detallarlos en base al modelo de calidad que se describe en la norma ISO/IEC 25010 (2010). Posteriormente la evaluación de calidad puede ser ejecutada utilizando el proceso de evaluación sugerido por ISO/IEC 25040 (2010) e ISO/IEC 25041 (2012). La Figura 5 muestra cómo interactúan el concepto de requisito de calidad y la evaluación de calidad bajo el marco de la serie ISO/IEC 25000.

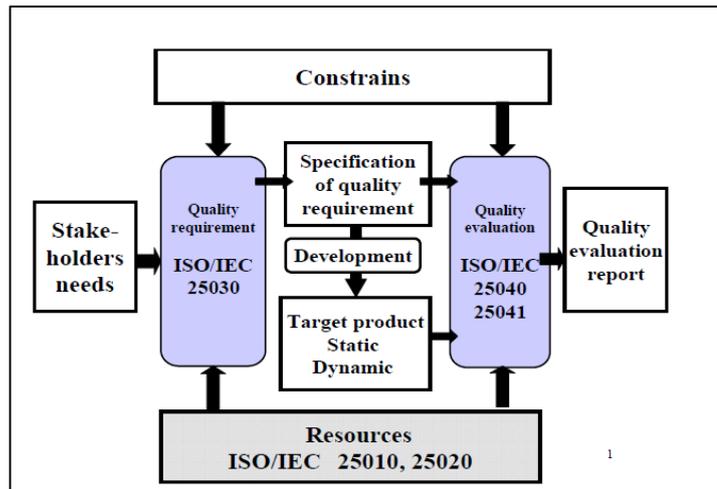


Figura 5 Concepto de requisito de calidad y evaluación de calidad
Fuente: Esaki (2003)

Asimismo, Esaki (2013) resalta la importancia de ISO/IEC 25040 (2010) e ISO/IEC 25041 (2012) para realizar el proceso de evaluación de calidad, reemplazando completamente a su antecesor, la serie ISO/IEC 14598. Este proceso contempla la medición, evaluación y valoración total de la calidad del producto, tal como se muestra en la Figura 6.

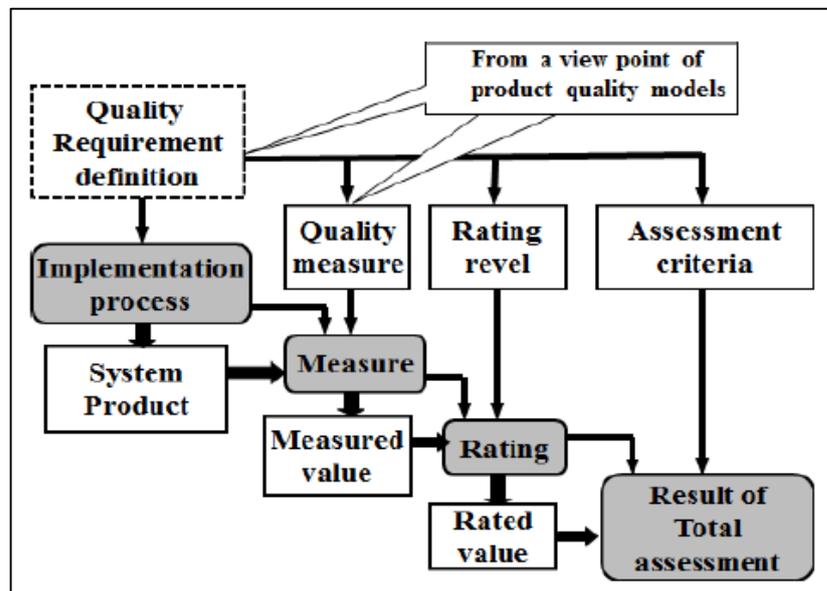


Figura 6 Proceso general de la evaluación de calidad del producto
Fuente: Esaki (2013)

2.1.6 Desafíos para el desarrollo de un modelo estándar de calidad de software

Thapar, Singh y Rani (2012) realizan una investigación sobre los desafíos para el desarrollo de un modelo estándar de calidad de software;

para ello, utilizan como universo de estudio a los modelos de calidad propuestos hasta el año 2011 (Figura 7 y Tabla 1).

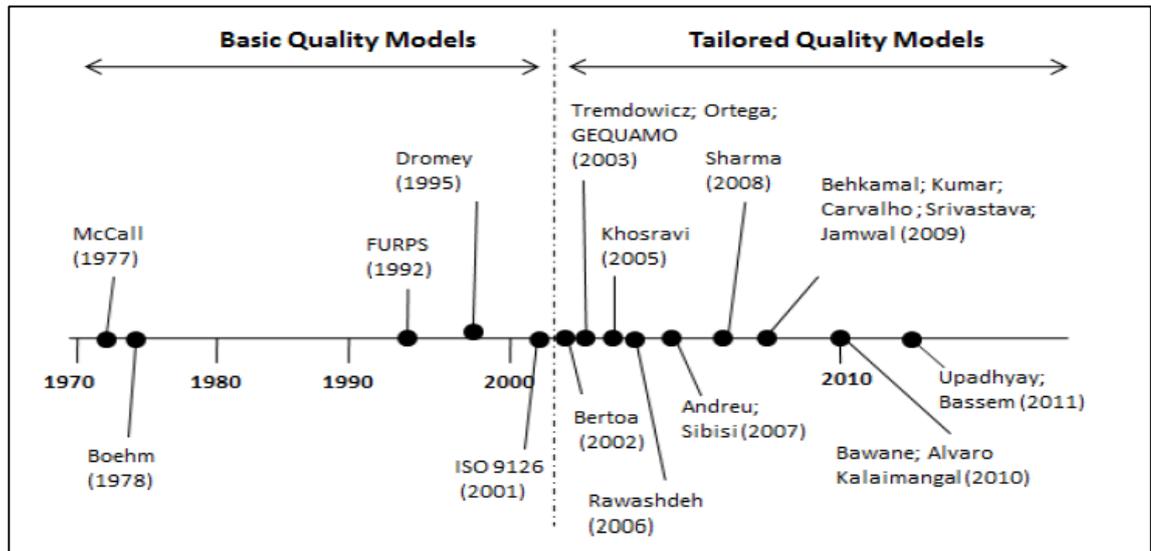


Figura 7 Modelos de calidad propuestos hasta el año 2011
Fuente: Thapar et al. (2012)

Tabla 1 Breves detalles de los modelos estudiados

Models	Year	Thrust area(s)	Funded research (R) / General Research (G)	Original idea (O) / Derived (D)
McCall	1977	<ul style="list-style-type: none"> Quality factors and their relationships Metrics User and developer's view 	G	O
Boehm	1978	<ul style="list-style-type: none"> Characteristics and Sub-characteristics Quantitative measurement 	G	O
FURPS	1992	<ul style="list-style-type: none"> Reusability Quantitative measurement 	R	O
Dromey	1995	<ul style="list-style-type: none"> Product based model Relationship among characteristics 	G	O
ISO 9126	2001	<ul style="list-style-type: none"> Coverage of overall quality Standardization of quality model Metrics 	R	D
Bertoa	2002	<ul style="list-style-type: none"> Quality model for components Evaluation of components Metrics for components 	G	D
Tremdowicz; Ortega	2003	<ul style="list-style-type: none"> Quality modeling in software product lines 	G	O
GEQUAMO	2003	<ul style="list-style-type: none"> Systemic quality model for quality 	R	O
Rawashdeh	2006	<ul style="list-style-type: none"> Multilayered Customizing and dynamic model 	G	O
Khosravi	2005	<ul style="list-style-type: none"> Quality model Quality issues Metrics and Quality evaluation using 	G	D
Rawashdeh	2006	<ul style="list-style-type: none"> Software quality model for evaluation of components 	G	D

Andreu	2007	• Quality framework for developing and evaluating	G	D
Sibisi	2007	• Process framework for customizing software quality models	G	D
Sharma	2008	• Quality evaluation using analytic	G	D
Behkamal	2009	• Quality model for evaluation of B2B	G	D
Kumar	2009	• Aspect oriented software quality	G	D
Carvalho	2009	• Quality verification framework to evaluate	G	D
Srivastava	2009	• Metrics and measurement approaches • Statistical measurement of software quality	G	D
Jamwal	2009	• Software quality evaluation using	G	O
Bawane	2010	• Quantitative assessment of quality • Correspondence with software development process	G	D
Alvaro	2010	• Component quality framework	R	D
Kalaiman	2010	• Evaluation of quality of COTS	G	D
Upadhyay	2011	• Quality model to build component quality assurance framework	G	D
Bassem	2011	• Quality evaluation model based on	G	D

Fuente: Thapar et al. (2012)

Thapar et al. (2012) puntualiza que un modelo de calidad debe satisfacer las necesidades de los interesados así como servir para la evaluación del producto software. Asimismo, al analizar la literatura descrita en el la Tabla 1, Thapar et al. (2012) establece que estos modelos no son integrales y completos, y todavía hay problemas que plantean desafíos para el desarrollo de modelos de calidad estándar. Estos problemas son:

a) Ausencia de asociación entre modelos de calidad y el proceso de desarrollo del software

Las relaciones entre el modelo de calidad y el proceso de desarrollo deben ser establecidos. En la fase de requerimientos, se definen los requisitos de calidad. En la fase de diseño, los requisitos se descomponen en niveles refinados de características, subcaracterísticas e indicadores. En fase de ejecución, se eligen aquellas características o subcaracterísticas que son apropiados para su aplicación. En fase de prueba, se debe medir las características utilizando métricas. En la fase de mantenimiento, se evalúa la fiabilidad de modelo de calidad y en base a los resultados puede ser modificado o mejorado.

b) Modelos de calidad que no evolucionan

El modelo de calidad debe evolucionar desde detalles abstractos hasta detalles más finos. El nivel de detalle del modelo está en función al avance del desarrollo del software, y siempre debe ser simple de entender y fácil de utilizar.

c) Modelos de calidad no mantenibles

Los modelos fijos son de poca utilidad en la mayoría de las aplicaciones debido a su falta de flexibilidad a los cambios inminentes. Los modelos de calidad deben ser de naturaleza iterativa para superar los problemas residuales. El mantenimiento de los mismos es necesario para eliminar las deficiencias (relacionadas con la calidad) detectadas, y también para hacer frente a los requisitos de las partes interesadas, la competitividad empresarial y el medio ambiente en constante cambio. Por lo tanto, un mecanismo debe estar integrado en el marco de calidad para que sea susceptible a los cambios potenciales.

d) Generalidad de los modelos de calidad

Modelos separados deben ser desarrollados para dominios y áreas de aplicación como los componentes.

e) Negligencia del aspecto de control de riesgo en el modelo de calidad

No se han seguido enfoques de gestión del riesgo en los modelos de calidad. Se sabe que, si no se mitigan los riesgos entonces todo proyecto puede terminar en peligro, por lo tanto en los modelos de calidad debe ser incorporado. De la misma forma, debe darse especial atención a las características que puedan causar riesgo de costo, calidad, y tiempo. Por ejemplo, si la funcionalidad no está implementada correctamente en el software, esto puede causar riesgos para la calidad que afecta indirectamente riesgos de costo y tiempo.

f) La falta de participación de los interesados en el marco de calidad

La participación de los interesados es muy importante en el marco de calidad ya que son ellos quienes definen los requisitos de calidad. Adicionalmente, se sabe que la calidad es mejor percibida por la satisfacción de los usuarios y el cumplimiento de los requerimientos. Por lo tanto, los interesados deben ser parte de su evaluación y el

marco de la calidad, ellos deben participar en la estructura del modelo para modificarlo (si es necesario), sus valiosos comentarios alinean o ajustan el proceso de ingeniería de calidad. Finalmente, son ellos quienes evalúan la calidad con respecto a los requisitos definidos.

g) Subjetividad en la evaluación de calidad

La calidad debe ser evaluada objetiva y estadísticamente usando herramientas que proveen resultados precisos de calidad. En este punto las métricas deben ser parte obligatoria de los modelos de calidad y su uso debe ser descrito en detalle suficiente.

h) La falta de equidad en la validación de la calidad

La validación del modelo de calidad siempre debe ser realizada de forma independiente o por académicos expertos o personal externo a la organización para obtener estimaciones fiables. Esto demostrará el valor exacto de manera imparcial.

Un modelo de calidad debe ser validado por dos o más evaluadores; puede incluir académicos expertos, organizaciones de terceros. Para considerar un modelo de calidad como exitoso, este se debe administrar en varias aplicaciones de software obteniendo resultados idénticos en todas ellas.

i) La ausencia de directrices o documentos necesarios para el modelo de calidad

Las directrices o la documentación del modelo de calidad ayudan a los desarrolladores y gestores a entender su uso. Debe describir cómo utilizarlo y aplicarlo, y en qué entorno y condiciones se puede utilizar.

Finalmente, Thapar et al. (2012) al comparar estos problemas versus los modelos de calidad, encuentra que la norma ISO/IEC 9126 está entre los que mejor aborda estos problemas (Tabla 2), también concluye que un modelo de calidad debe contemplar estos nueve aspectos para ser un estándar aceptado ampliamente.

Tabla 2 Problemas en los modelos de calidad

Models	Issues								
	a (association)	b (evolution)	c (maintainable)	d (general)	e (risk - driven)	f (stakeholders)	g (subjective evaluation)	h (validation fairness)	i (guidelines)
McCall	√	√	√	√	√		√	√	√
Boehm	√	√	√	√	√		√	√	√
FURPS	√	√	√		√	√	√	√	√
Dromey	√	√	√		√			√	√
ISO 9126	√				√			√	
Bertoa	√	√	√		√			√	
Tremdowicz			√		√	√	√	√	√
Ortega	√	√		√	√			√	√
GEQUAMO	√						√	√	
Khosravi	√	√	√	√	√			√	√
Rawashdeh	√	√	√	√	√		√	√	√
Andreu	√				√			√	√
Sibisi	√	√			√	√		√	√
Sharma	√	√	√		√			√	√
Behkamal	√	√	√		√				√
Kumar	√	√	√		√			√	√
Carvalho	√	√	√		√			√	√
Srivastava	√	√	√		√			√	√
Jamwal	√			√	√			√	√
Bawane					√			√	√
Alvaro	√			√	√				
Kalaimangal	√				√				√
Upadhyay	√	√	√		√	√		√	
Bassem	√	√	√	√	√		√	√	√

Fuente: Thapar et al. (2012)

2.1.7 Difusión de los estándares de calidad de software

Iyidogan (2014), realizó un estudio sobre la difusión de los estándares de calidad de software, Este estudio tiene como objetivo contribuir a la comprensión de la estructura y los determinantes de la difusión de las normas de calidad del software a través del análisis de sector de industria de software de Turquía. El estudio concluye que en el 60% del sector servicios y pequeñas empresas de software, la certificación de calidad del software se ha mantenido baja desde mediados de la década de 2000. Su adopción por las empresas no está evolucionando pero, por el contrario, presenta una situación estacionaria. El autor considera ciertas determinantes para la difusión de Estándares de Calidad de Software, para ello identificó las fuerzas motrices y las barreras para su adopción.

Las fuerzas motrices son: (1) la construcción de una buena reputación constituye uno de los más importantes factores, la cual influye en

los mercados externos e internos si una empresa quiere exportar software, y (2) la aplicación de las regulaciones obligatorias, por parte del Estado o del mercado.

Las barreras para la difusión de estándares de calidad son: (1) la falta de conocimiento y conciencia de los estándares de calidad y (2) los recursos limitados en términos relacionados a financiamiento y empleados.

El estudio también precisa que uno de los principales puntos para que Turquía sea competitivo es que fortalece la conciencia sobre los estándares de calidad de software, no solo para mejorar la competitividad de las empresas, sino para que aprovechen los beneficios que estos estándares puedan ofrecer de manera interna.

Aunque es un artículo orientado a la realidad de Turquía, es importante destacar que la adopción de estándares de calidad en una empresa de software influyen directamente en la competitividad de las empresas en el mercado; sin embargo, no se debe desvirtuar la importancia de la aplicación de los estándares de calidad para aprovechar los beneficios que esta ofrece, para ello se debe tener una profunda conciencia sobre ello.

2.2 Bases Teóricas

2.2.1 Calidad del producto

El software es un elemento presente en una importante cantidad de actividades cotidianas y, con frecuencia, su correcta operación es crítica para el éxito del negocio y/o la seguridad de las personas; por lo tanto, el desarrollo de productos software de calidad es de suma importancia.

Una evaluación de la calidad del producto software es un factor clave para asegurar la calidad adecuada, ello se puede lograr al definir de manera apropiada un proceso para su evaluación, que debe considerar en su contenido la verificación de las características relevantes de la calidad del producto utilizando métricas validadas o de amplia aceptación.

Para poder comprender la calidad del producto software, es necesario recurrir a un modelo de calidad. La norma ISO/IEC 25040 (2010) lo define como un conjunto de características y la relación entre las mismas, que conforman la base para la evaluación de calidad. La Figura 8 representa un modelo de calidad de dos niveles para las características y subcaracterísticas y en el tercer nivel presenta los atributos de calidad; estas últimas se pueden obtener de la medición de los diversos atributos que tiene el producto y que influyen en cada subcaracterística.

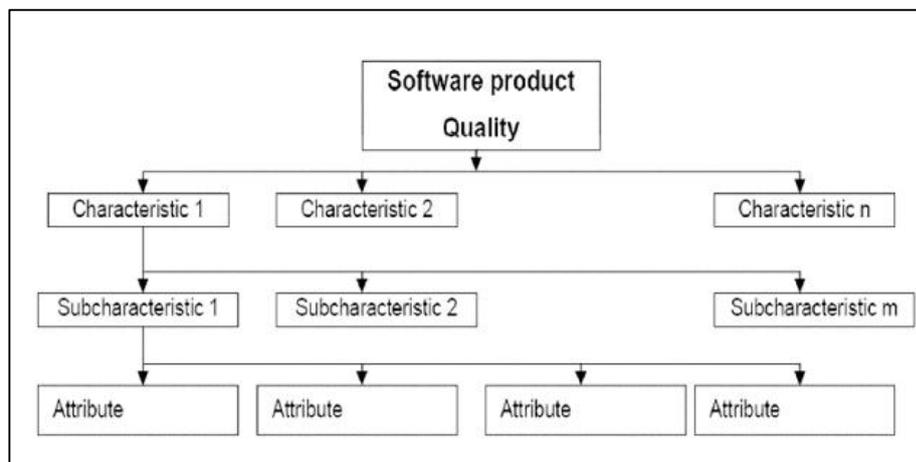


Figura 8 Esquema general de un modelo de la calidad del producto
Fuente: ISO/IEC 25000 (2005)

El modelo de calidad de la serie ISO/IEC 25000 presenta el concepto de calidad en uso, calidad externa y calidad interna. También señala que “la calidad del proceso contribuye a mejorar la calidad del producto, y la calidad del producto contribuye a mejorar la calidad en uso. Por lo tanto, evaluar y mejorar un proceso es una manera de mejorar la calidad del producto, y evaluar y mejorar la calidad del producto es una manera de mejorar la calidad en uso. De igual manera, evaluar la calidad en uso puede proporcionar una retroalimentación para mejorar el producto, y evaluando un producto puede proporcionar una retroalimentación para mejorar un proceso”. (ISO25000, 2005).

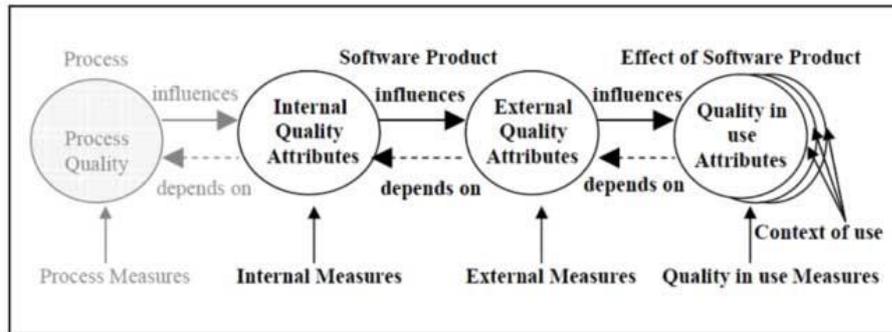


Figura 9 Calidad en el ciclo de vida
Fuente: ISO/IEC 25010 (2010)

La Figura 9 representa el ciclo de vida de la calidad que muestra la influencia o dependencia entre los distintos enfoques de calidad (interna, externa y en uso) y en la Figura 10 se puede apreciar que las necesidades de calidad en uso contribuyen a especificar los requerimientos de calidad externa y estos a su vez los requerimientos de calidad interna. El cumplimiento de los requisitos de calidad interna se comprobarán en un proceso de verificación que permitirá medirlo, el cumplimiento de los requisitos de calidad externa se comprobarán en un proceso de validación que permitirá medirlo y finalmente la satisfacción de las necesidades de la calidad del producto se comprobarán en un proceso de evaluación que permitirá medir la calidad en uso.

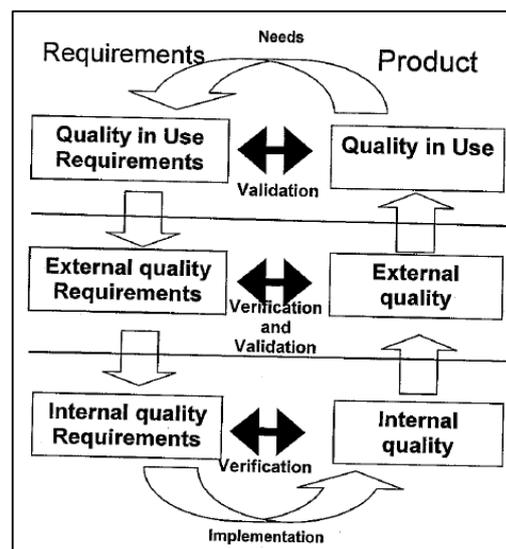


Figura 10 Modelo del ciclo de vida de calidad del producto software
Fuente: ISO/IEC 25000(2005)

2.2.2 Calidad del producto software – modelos y definiciones

La serie ISO/IEC 25000 presenta dos modelos de calidad, la primera referida a la calidad interna y externa y el segundo modelo referido a la calidad en uso. En las secciones siguientes se describirá cada uno de ellos.

2.2.2.1 Calidad externa e interna

La norma ISO/IEC 25010 (2010) define la calidad interna como: “la totalidad de las características del producto software desde una perspectiva interna. La calidad interna es medida y evaluada en base a los requerimientos de calidad interna. Los detalles de la calidad del producto software pueden ser mejorados durante la implementación, revisión y prueba del código software, pero la naturaleza fundamental de la calidad del producto software representada por la calidad interna permanece sin cambios a menos que sea rediseñado”.

Esta norma también define a la calidad externa como: “la totalidad de las características del producto software desde una perspectiva externa. Es la calidad cuando el software es ejecutado, la cual es típicamente medida y evaluada mientras se prueba en un ambiente simulado con datos simulados y usando métricas externas. Durante las pruebas, muchas fallas serán descubiertas y eliminadas. Sin embargo, algunas fallas todavía pueden permanecer después de las pruebas. Como es difícil corregir la arquitectura de software u otros aspectos fundamentales del diseño del software, el diseño fundamental permanece sin cambios a través de las pruebas”.

La Figura 11 representa el modelo de calidad interna o externa, que muestra un conjunto de 8 características: funcionalidad, eficiencia en el rendimiento, compatibilidad, usabilidad, confiabilidad, seguridad, facilidad de mantenimiento y portabilidad.

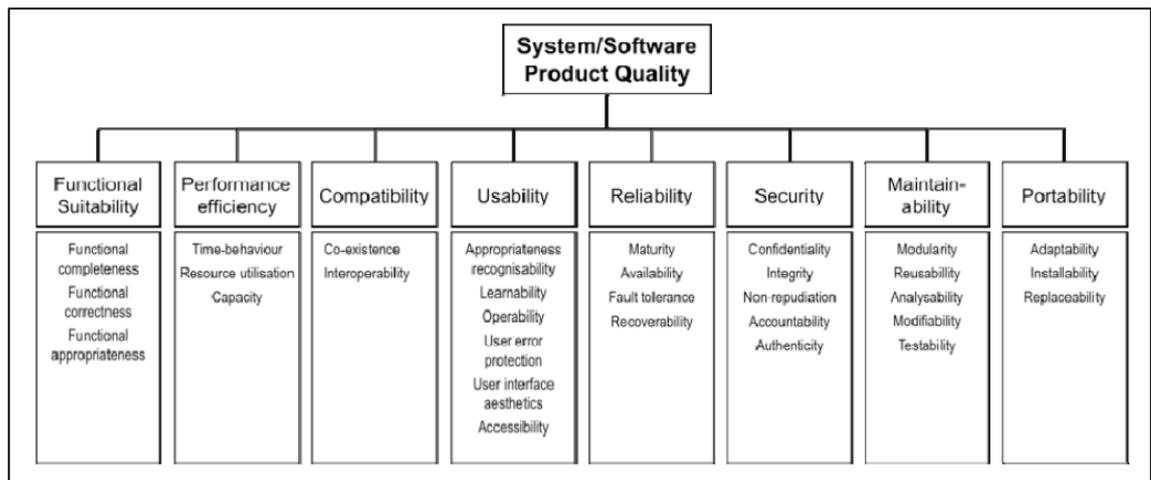


Figura 11 Modelo de calidad del producto
Fuente: ISO/IEC 25010 (2010)

2.2.2.2 Calidad en uso

La norma ISO/IEC 25010 (2010) define la calidad en uso como la perspectiva del usuario de la calidad del producto software cuando este es usado en un ambiente específico y un contexto de uso específico. Esta mide la extensión para la cual los usuarios pueden conseguir sus metas en un ambiente particular, en vez de medir las propiedades del software en sí mismo.

Un usuario es cualquier tipo de posible usuario y cuyos requerimientos pueden ser diferentes; por ejemplo un operador del software tiene un requerimiento diferente que un responsable del mantenimiento del software.

En la Figura 12 se presenta el modelo de calidad en uso que muestra un conjunto de cinco características: efectividad, eficiencia, satisfacción, libertad de riesgo, cobertura de contexto.

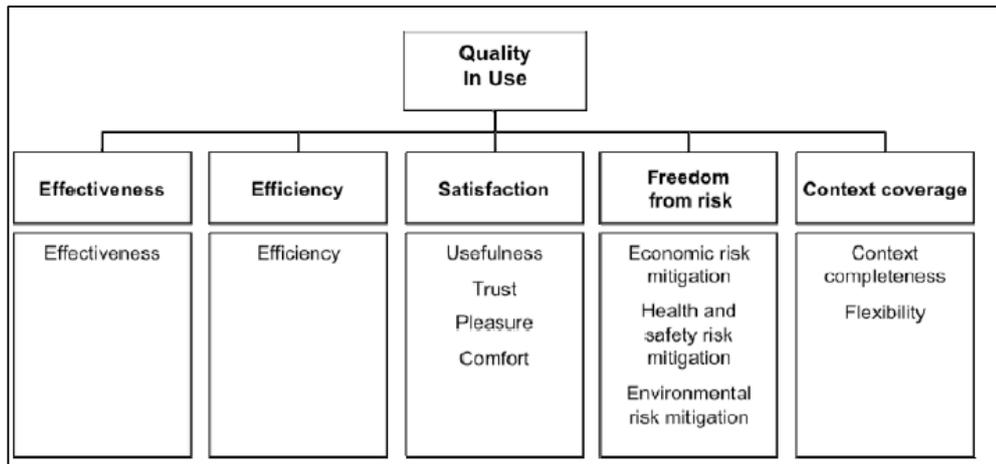


Figura 12 Modelo de calidad en uso del producto software
Fuente ISO/IEC 25010 (2010)

2.2.3 Serie ISO/IEC 25000

Es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.

La serie ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de la serie de normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, y la serie ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones.

- ISO/IEC 2500n - División de Gestión de Calidad
- ISO/IEC 2501n - División de Modelo de Calidad
- ISO/IEC 2502n - División de Medición de Calidad
- ISO/IEC 2503n - División de Requisitos de Calidad
- ISO/IEC 2504n - División de Evaluación de Calidad

2.2.3.1 ISO/IEC 2500n - División de Gestión de Calidad

Las normas que forman este apartado definen todos los modelos, términos y definiciones comunes referenciados por todas las otras normas de la familia 25000. Actualmente, esta división se encuentra formada por:

- ISO/IEC 25000 (2005) – **Guía para SQuaRE**: contiene el modelo de la arquitectura de SQuaRE, la terminología de la familia, un resumen de las partes, los usuarios previstos y las partes asociadas, así como los modelos de referencia. La Figura 13 muestra el modelo de referencia SQuaRE.

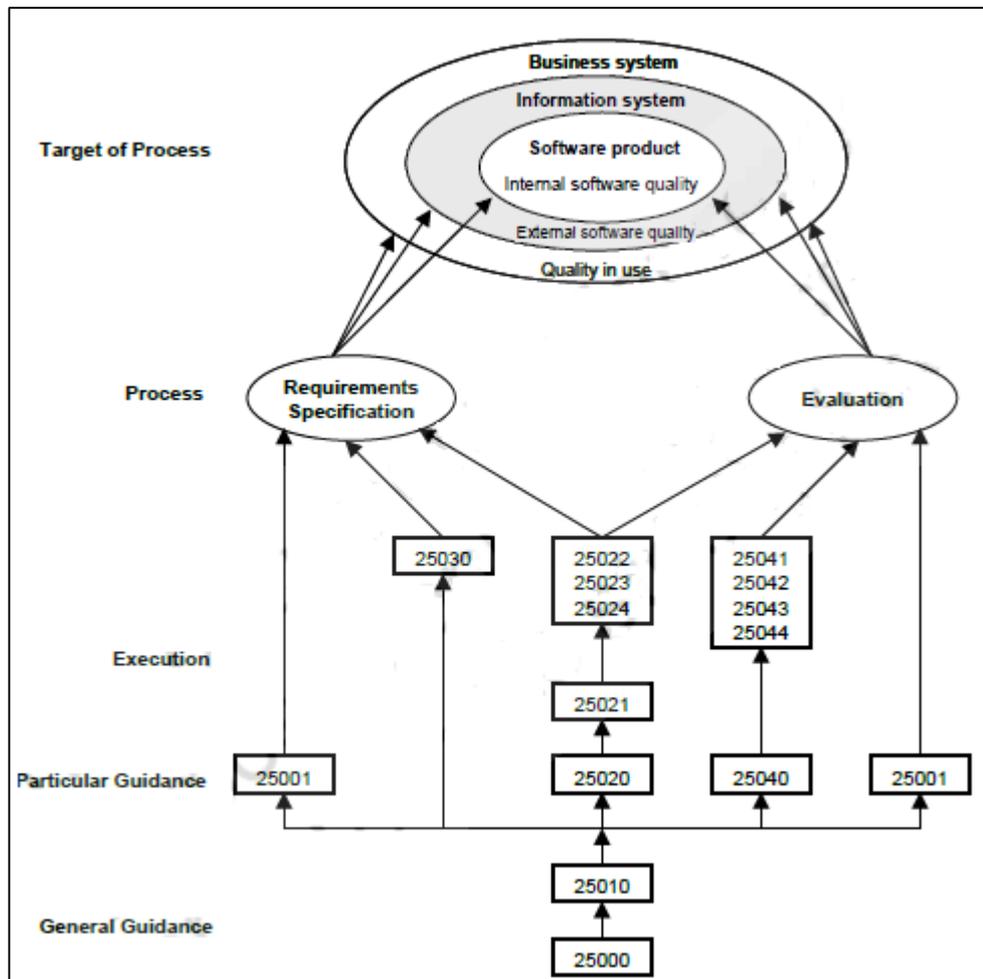


Figura 13 Modelo de referencia general SQuaRE
Fuente: ISO/IEC 25000 (2005)

- ISO/IEC 25001 – **Planeamiento y Gestión**. Establece los requisitos y orientaciones para gestionar la evaluación y especificación de los requisitos del producto software. Proporciona requisitos y recomendaciones para una organización responsable de la implementación y administración de las especificaciones de los requisitos de calidad de sistemas y productos de software, y las actividades de evaluación a través de la provisión de tecnología, herramientas, experiencias y habilidades de gestión.

El papel del grupo de evaluación incluye motivar a los empleados y capacitarlos para la especificación de los requisitos y las actividades de evaluación, preparación de documentos apropiados, la identificación o el desarrollo de los métodos requeridos, y responder a las preguntas sobre las tecnologías relevantes.

La gestión de la tecnología está relacionada con la especificación de requerimientos de calidad de sistemas y software, así como de los procesos, mediciones y herramientas de evaluación. Esto incluye la gestión del desarrollo, la adquisición, la normalización, el control, la transferencia y la retroalimentación de las experiencias de especificación de requisitos y tecnología de evaluación dentro de la organización.

2.2.3.2 ISO/IEC 2501n - División de Modelo de Calidad

Las normas de este apartado presentan modelos de calidad detallados incluyendo características para calidad interna, externa y en uso del producto software. Actualmente, esta división se encuentra formada por:

- **ISO/IEC 25010 (2010) –Modelos de Calidad de Software y Sistemas.** 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 se debe evaluar el producto software.
- **ISO/IEC 25012 – Modelo de Calidad de Datos.** Define un modelo general para la calidad de los datos, aplicable a aquellos que se encuentran almacenados de manera estructurada y forman parte de un Sistema de Información.

2.2.3.3 ISO/IEC 2502n - División de Medición de Calidad

Estas normas incluyen un modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad

(interna, externa y en uso) y guías prácticas para su aplicación. Actualmente esta división se encuentra formada por:

- ISO/IEC 25020 - **Guía y Modelo de Referencia para la medición.** Presenta una explicación introductoria y un modelo de referencia común a los elementos de medición de la calidad. También proporciona una guía para que los usuarios seleccionen o desarrollen y apliquen medidas propuestas por normas ISO.
- ISO/IEC 25021 (2011) - **Elementos de Medida de Calidad.** Define y especifica un conjunto recomendado de métricas base y derivadas que puedan ser usadas a lo largo de todo el ciclo de vida del desarrollo software. Este conjunto de métricas se utilizará como entrada en el proceso de medida de la calidad interna, externa y en el uso, también especifica la forma de crear nuevas métricas de calidad en el modelo. La Figura 14 muestra la relación existente entre las propiedades a cuantificar y los elementos de medida de calidad QME.

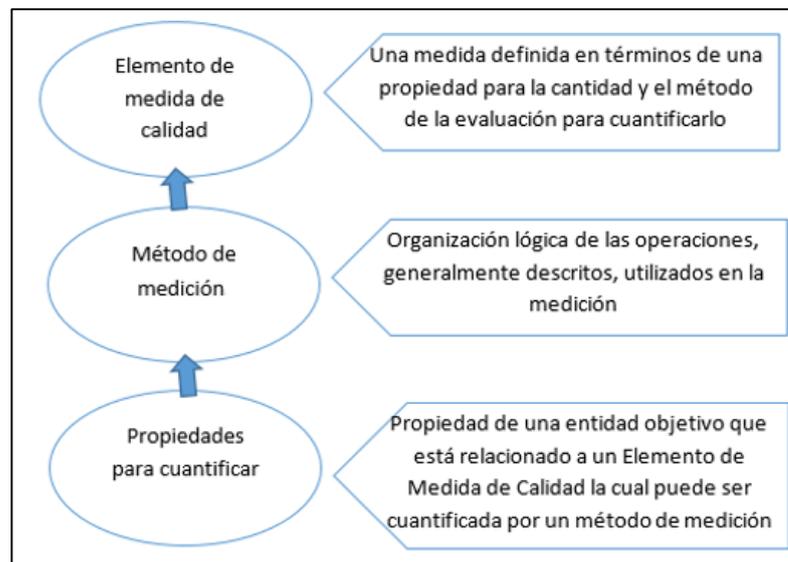


Figura 14 Relación entre las propiedades para cuantificar, Método de medición y Elementos de medida de calidad (QME)

Fuente: ISO/IEC 25021 (2011)

- ISO/IEC 25022 - **Medición de Calidad en Uso.** Define específicamente las métricas para realizar la medición de la calidad en uso del producto.

- ISO/IEC 25023 (2013) - **Medición de la Calidad del Producto Software y Sistemas.** Define específicamente las métricas para realizar la medición de la calidad de productos y sistemas software.
- ISO/IEC 25024 - **Medición de la Calidad de Datos:** define específicamente las métricas para realizar la medición de la calidad de datos.

2.2.3.4 ISO/IEC 2503n - División de Requisitos de Calidad

Las normas que forman este apartado ayudan a especificar requisitos de calidad que pueden ser utilizados como entrada del proceso de evaluación. Para ello, este apartado se compone de:

- ISO/IEC 25030 (2007) – **Requisitos de Calidad:** provee de un conjunto de recomendaciones para realizar la especificación de los requisitos de calidad del producto software.

2.2.3.5 ISO/IEC 2504n - División de Evaluación de Calidad

Este apartado incluye normas que proporcionan requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software. Se encuentra formada por:

- ISO/IEC 25040 (2010) - **Guía y Modelo de Referencia de la Evaluación.** Propone un modelo de referencia general para la evaluación, que considera las entradas al proceso de evaluación, las restricciones y los recursos necesarios para obtener las correspondientes salidas.
- ISO/IEC 25041 (2012) - **Guía de Evaluación para Desarrolladores, Adquirentes y Evaluadores Independientes.** Describe los requisitos y recomendaciones para la implementación práctica de la evaluación del producto software desde el punto de vista de los desarrolladores, de los adquirentes y de los evaluadores independientes.

- ISO/IEC 25042 - **Módulos de Evaluación**. Define lo que la Norma considera un módulo de evaluación y la documentación, estructura y contenido que se debe utilizar a la hora de definir uno de estos módulos.
- ISO/IEC 25045 – **Módulo de Evaluación para Recuperabilidad**: Define un módulo para la evaluación de la subcaracterística “Recuperabilidad” (Recoverability).

2.3 ¿Porqué un método para la evaluación de calidad del software?

La Tabla 3 muestra los principales vacíos y deficiencias encontrados en el campo de la calidad del producto software. El método para la evaluación de la calidad de software presentado en detalle en el capítulo IV aborda este vacío; pues es una herramienta que ayuda a gestionar la calidad durante el ciclo de vida del software desde la etapa de análisis, facilita visibilizar problemas de calidad y tomar acciones correctivas desde etapas tempranas; asimismo, estructura una forma ordenada de ejecutar la evaluación de calidad en los entregables del software.

El método en mención está centrado en la calidad del producto y debe ser complemento de un proceso de desarrollo de software que cumpla estándares de calidad. Si bien el área de calidad de procesos ha sido bastante desarrollada, y su aceptación e implementación es de amplia cobertura, aún existe un alto porcentaje de proyectos con problemas, que inclusive se extienden a entidades con implementación CMMI nivel 5 (Harter, Kemerer, y Slaughter, 2012). El método de calidad propuesto pretende cubrir la brecha que **no** aborda la calidad de procesos y así contribuir con la mejora de la calidad del software y por ende disminuir la cantidad de proyectos de desarrollo con problemas.

Tabla 3 Vacíos encontrados en el campo de calidad del software

Referencia	Vacíos en el campo de calidad del producto software
Mellado, Rodríguez, Verdugo, Piattini y Fernández-Medina (2010)	La calidad ha sido tratada con más amplitud a nivel de calidad del proceso que a nivel de la calidad del producto, y si bien las áreas de testing (sobre todo funcionalidad) es un campo bien trabajado, todavía no se han desarrollado las técnicas necesarias para evaluar de forma efectiva la calidad y la seguridad de un producto software.
Seaman y Guo (2011)	La inadecuada gestión de la calidad y otras restricciones durante el ciclo de vida del desarrollo del software ocasiona sobre costos por en la etapa de operación del software.
Kaur y Sengupta (2011)	La etapa de pruebas (testing) es insuficiente. Existe una pobre gestión calidad de los entregables.
Zazworka, Shaw, Shull, Seaman (2011)	Priorización inadecuada de las restricciones del proyecto de software.
Rodríguez y Piattini (2012)	<p>El campo de la calidad del producto software presenta menor desarrollo en cuanto a certificaciones bajo normas comparado con las certificaciones a nivel de proceso. Se debe gestionar la calidad del propio producto software y no solo por la calidad de los procesos que se utilizan para su desarrollo.</p> <p>El proceso para llevar a cabo la certificación del producto software es un campo todavía bastante incipiente en la Ingeniería del Software. Las pocas propuestas de certificación halladas en este estudio utilizan la serie ISO/IEC 9126 y la serie ISO/IEC 14598, sin embargo ninguna de las propuestas ha adoptado todavía la nueva serie de normas ISO/IEC 25000.</p>

<p>Thapar, Singh y Rani (2012)</p>	<p>Este estudio identifica que el modelo de calidad presentado por ISO/IEC 9126-1 (2001), que es base del modelo de calidad que presenta la serie ISO/IEC 25000, tiene como uno de sus problemas principales la ausencia de asociación entre el modelo de calidad y el proceso de desarrollo del software.</p> <p>Las relaciones entre el modelo de calidad y el proceso de desarrollo deben ser establecidos. En la fase de requerimientos los requisitos de calidad de los interesados son definidos. En la fase de diseño, se descomponen estos requisitos abstractos en niveles refinados de características, subcaracterísticas e indicadores. En fase de ejecución, se debe elegir aquellas características o subcaracterísticas que son apropiados para su aplicación. En fase de pruebas, se debe medir estas características utilizando métricas. Y en la fase de mantenimiento, se evalúa la fiabilidad de modelo de calidad y puede ser modificado o mejorado.</p>
<p>Harter, Kemerer y Slaughter (2012)</p>	<p>Este estudio analiza 7545 errores de software recolectados por 20 años, tomados de proyectos completados.</p> <p>De igual manera demuestra estadísticamente que los Niveles altos de CMMI reduce la probabilidad de que existan defectos de graves. Asimismo, los niveles más altos de la mejora de procesos son beneficiosos en la reducción de defectos graves cuando el sistema desarrollado es grande o complejo, pero son menos beneficiosos cuando los requisitos son ambiguos, poco claros o incompletos.</p>
<p>Hosni y Kirinic (2013)</p>	<p>La serie de normas ISO/IEC 9126 e ISO/IEC 14598 y relacionados como ISO/IEC 25000 no son fáciles de</p>

	<p>adaptar y utilizar, requieren experiencia y conocimiento, hay muchas relaciones y referencias cruzadas, hacen mención a distintos ciclos de vida de desarrollo de software.</p> <p>Describe como varios estándares podrían ser usados a través del ciclo de vida del software con la finalidad de planear, revisar, y mejorar su calidad.</p>
Hwang (2014)	Existen ausencias en el campo de la ingeniería del software, uno de ellos es la calidad del software.
Iyidogan (2014)	<p>Análisis de sector de industria de software de Turquía.</p> <p>Barreras para adopción de estándares de calidad:</p> <p>La falta de conocimiento y conciencia de los estándares de calidad.</p> <p>Recursos limitados en términos relacionados a financiamiento y empleados</p>
CHAOS (2013), Nasir y Sahibuddin (2011)	<p>Solo 39% de los proyectos terminan satisfactoriamente.</p> <p>El principal factor crítico de éxito en un proyecto de software es la claridad de los requisitos y especificaciones.</p>

Elaboración: el autor

2.4 Definiciones de Términos básicos

- **Atributo:** Propiedad inherente de una entidad que puede distinguirse cuantitativa o cualitativamente.
- **Calidad interna:** Capacidad de un conjunto estático de atributos para satisfacer las necesidades declaradas e implícitas de un producto software bajo ciertas condiciones especificadas.
- **Calidad externa:** Capacidad de un producto software para desarrollar el comportamiento de un sistema de forma que satisfaga las necesidades declaradas e implícitas de un sistema utilizado bajo ciertas condiciones especificadas.
- **Calidad en uso:** Grado en que un producto satisface los objetivos del usuario en un contexto específico.
- **Métrica:** Medida, tanto base como derivada, utilizada para medir la calidad del software.
- **Medida base:** Conjunto formado por la medida definida en términos de un atributo más el método para su cuantificación.
- **Medida derivada:** Medida obtenida a partir dos o más medidas base.
- **Módulo de evaluación:** Empaquetado de las métricas de calidad, incluyendo métodos y técnicas de evaluación, entradas a procesar, datos a recoger y medir, herramientas y procedimientos de apoyo.
- **Verificación:** Confirmación por medio de pruebas objetivas de que se satisfacen los requisitos especificados.
- **Conformidad por parte del usuario:** Conformidad otorgada por el usuario en el ambiente de pruebas de aceptación del software, antes del pase a producción del software.
- **Ambiente de pruebas de aceptación del software por el usuario (UAT):** Servidor de prueba con el software integrado, que sirve para ejecutar pruebas de aceptación del software por parte del usuario.
- **Reprocesos para la conformidad por parte del usuario:** Cantidad de veces que se realizan cambios en el software debido

a observaciones realizadas en el ambiente de certificación del software.

2.5 Hipótesis y Variables

2.5.1 Hipótesis

2.5.1.1 Hipótesis General

Contar con un método para la evaluación de calidad basado en ISO/IEC 25000 mejora la calidad del software.

2.5.1.2 Hipótesis Específicos

- El uso de un método para la evaluación de calidad basado en ISO/IEC 25000 disminuye los errores del software después de su puesta en producción.
- El uso de un método para la evaluación de calidad basado en ISO/IEC 25000 facilita la conformidad del software por parte del usuario.

2.5.2 Variables

Variables Independientes:

V.I.1: Método para la evaluación de calidad de software basado en ISO/IEC 25000.

Variables Dependientes:

V.D.1: Calidad del software.

V.D.2: Errores del software relacionados al cumplimiento de los requisitos funcionales en el ambiente de producción.

V.D.3: Conformidad del software por parte de los usuarios.

2.5.3 Matriz de Consistencia

La Tabla 4 muestra la matriz de consistencia de los problemas, objetivos e hipótesis de la presente tesis.

Tabla 4 Matriz de consistencia

Problemas	Objetivos	Hipótesis	Variables
PG: ¿De qué manera un método para la evaluación de calidad mejorará la calidad del software?	OG: Mejorar la calidad del software a través de la aplicación de un método para la evaluación de calidad basado en ISO/IEC 25000.	HG: Contar con un método para la evaluación de calidad basado en ISO/IEC 25000 mejora la calidad del software.	V.I. 1 V.D. 1
PE1: ¿Cómo el método de evaluación de calidad disminuirá los errores del software después de la puesta en producción?	OE1: Disminuir los errores del software después de su puesta en producción, a través de la aplicación de un método para la evaluación de calidad basado en ISO/IEC 25000.	HE1: El uso de un método para la evaluación de calidad basado en ISO/IEC 25000 disminuye los errores del software después de su puesta en producción.	V.I. 1 V.D. 2
PE2: ¿De qué manera el método de evaluación de calidad facilita la conformidad del software por parte del usuario?	OE2: Facilitar la conformidad del software por parte del usuario, mediante la aplicación de un método para la evaluación de calidad basado en ISO/IEC 25000.	HE2: El uso de un método para la evaluación de calidad basado en ISO/IEC 25000 facilita la conformidad del software por parte del usuario.	V.I. 1 V.D.3

Elaboración: el autor

CAPÍTULO III METODOLOGÍA

3.1 Diseño Metodológico

En este capítulo se describe la metodología utilizada para validar el método de calidad desarrollado, el método mencionado es explicado en detalle en el Capítulo IV.

Los pasos metodológicos para validar las hipótesis propuestas en el presente trabajo son las siguientes:

- Selección del diseño experimental a seguir
- Selección de la muestra
- Recolección de los datos
- Análisis de los datos obtenidos

El diseño seleccionado es cuasiexperimental con postprueba únicamente y grupo de control (Hernandez, Fernandez y Baptista, 2010). Los grupos son los siguientes:

- Grupo1: Proyectos sin la aplicación del método de calidad (grupo de control).
- Grupo2: Proyectos en el que se aplicó el método para la evaluación de calidad.

El nivel de manipulación de la variable independiente (método para la evaluación de calidad de software basado en ISO/IEC 25000) es presencia-

ausencia. El Grupo2 se expone al método de evaluación de calidad o el otro no; posteriormente se comparará los resultados para saber si el grupo expuesto (Grupo2) difiere del grupo que no fue expuesto (Grupo1).

El emparejamiento entre los grupos se realizó tomando en cuenta: (1) la duración de los proyectos como criterio que indica su complejidad y (2) son proyectos de aplicaciones sobre plataforma .NET y a nivel de base datos con SQL Server u Oracle. Por este motivo se ha seleccionado un diseño cuasiexperimental, ya que no se ha utilizado una técnica más estricta para lograr la equivalencia entre los grupos como asignación al azar o una técnica de emparejamiento más estricto.

La Figura 15 muestra gráficamente el diseño experimental. Al Grupo1 no se aplica el método de evaluación de calidad y se toma los resultados, al Grupo2 se aplica el método de evaluación de calidad y se toman los resultados.

Grupo1	-	O1
<hr/>		
Grupo2	X	O2

Figura 15 Diseño experimental de la investigación
Elaboración: el autor

3.2 Población y muestra

La población son los proyectos de la Financiera Autos de Perú, con duración de entre tres y nueve semanas cronológicas (desde el análisis hasta la conformidad del usuario) y cuya fecha de inicio es mayor al 01/01/2014.

La muestra para este estudio son 28 proyectos divididos en dos grupos:

- Grupo 1: catorce proyectos de desarrollo de software sin la aplicación del método de calidad de software.
- Grupo 2: catorce proyectos de desarrollo de software en el que se aplicó el método de evaluación de calidad.

3.2.1 Unidad de análisis

Las unidades de análisis serán los proyectos de desarrollo de software. Los proyectos considerados para efectos del presente trabajo tienen las siguientes características:

- Se sigue una metodología de desarrollo basada en PMI y RUP.
- Existe un registro de los pases a los ambientes de certificación usuaria y producción.
- Existe un registro de los errores detectados en el ambiente de producción.

3.2.2 Criterios de inclusión y exclusión

Se **incluirán** los proyectos que cumplan las condiciones descritas a continuación:

- Proyectos de desarrollo de aplicaciones sobre plataforma .NET y a nivel de base datos con SQL Server u Oracle.
- Aquellos cuya duración está entre tres a nueve semanas cronológicas, debido a que estos proyectos originan el 80% de los errores que se presentan en producción.

Serán **excluidos** los proyectos que cumplen las siguientes condiciones:

- Que se hayan iniciado antes del 01/01/2014.
- Aquellos cuya duración exceda nueve semanas, esto en concordancia a los criterios de inclusión y también debido a que las fechas de ejecución del experimento para el Grupo 2 (del 24/03/2015 al 03/06/2015) no permite trabajar con proyectos de mayor duración.

En el Anexo 1 se detallan los proyectos considerados en este estudio.

3.3 Operacionalización de variables

La Tabla 5 detalla la variable independiente y la Tabla 6 detalla las métricas utilizadas las variables dependientes.

Tabla 5 Variable independiente

Variable	Descripción
Método para la evaluación de calidad de software basado en ISO/IEC 25000.	La variable independiente es de tipo cualitativo; es el método propuesto, los valores que toma es: <ul style="list-style-type: none">- En el Grupo 1, ausencia del método- En el Grupo 2, presencia del método

Elaboración: el autor

Tabla 6 Métricas de las variables dependientes

Variable	Métrica
Calidad del software.	<p>Cantidad de observaciones encontradas en el software, que está representado por la cantidad de veces que el software es modificado, debido a (1) reprocesos por errores encontrados en la etapa de conformidad por el usuario y (2) los reprocesos originados por errores encontrados en el ambiente de producción.</p> <p>La métrica por excelencia y de mayor uso para medir la calidad, son los errores encontrados en el software, a continuación se enumeran algunos estudios relacionados:</p> <ul style="list-style-type: none">- Wilson (2013)- Wilkerson, Nunamaker y Mercer (2012)- Rawat y Dubey (2012)- Hansen, Jonasson y Neukirchen (2011)- Kitchenham y Pfleeger (1996)

Conformidad del software por parte de los usuarios.	Cantidad de reprocesos para la conformidad final por parte de los usuarios en el ambiente de aceptación por el usuario (UAT).
Errores del software relacionados al cumplimiento de los requisitos funcionales en el ambiente de producción.	Cantidad de errores relacionados al cumplimiento de los requisitos funcionales en el ambiente de producción.

Elaboración: el autor

3.4 Técnicas de recolección de datos

Para la recolección de los datos se utilizarán las siguientes técnicas con sus respectivas fuentes de información:

- Revisión de la **Bitácora de pases al ambiente de aceptación por el usuario**, esto servirá para hallar la cantidad de reprocesos para la aceptación final por parte de los usuarios. En el Anexo 2 se muestra la Bitácora.
- Revisión del **Reporte de errores identificados**, a partir de los incidentes reportados a mesa de ayuda sobre el software en producción, se debe considerar solamente los errores relacionados al cumplimiento de los requisitos funcionales. En el Anexo 3 se muestra el Reporte de Errores.

3.5 Técnicas para el procesamiento de la información

Se utilizarán gráficos descriptivos para mostrar los datos obtenidos y se verificará la normalidad de su distribución.

Para la verificación de las hipótesis utilizaremos la prueba t-Student cuando los datos se ajusten a una distribución normal y la prueba de Mann-Whitney que es una prueba de tipo no paramétrica cuando los datos no se ajusten a una distribución normal.

El software utilizado para ejecutar las pruebas estadísticas es la versión 17.1 de Minitab.

CAPÍTULO IV

MÉTODO PARA LA EVALUACIÓN DE CALIDAD DEL PRODUCTO SOFTWARE

La evaluación de la calidad del producto no se debe realizar solamente cuando el software está en la etapa de pruebas (testing), sino también a los entregables que se generan en las etapas del ciclo de vida del software, tal como lo indica el estándar ISO/IEC 25041 (2012) e ISO/IEC 14598-5 (1998), esto asegurará la máxima probabilidad de que el producto software satisfaga los requisitos, así como minimizará el riesgo de costos adicionales no esperados.

El método planteado establece las fases secuenciales para evaluar la calidad de los entregables generados en las etapas tempranas del ciclo de vida del software; la finalidad de evaluar dichos entregables es mejorar la calidad del producto software final. Asimismo, como parte del método se han considerado los lineamientos definidos en la serie ISO/IEC 25000.

El método evaluará la calidad de los siguientes entregables: especificación de requerimientos del software, diseño de la arquitectura del software, diseño de la base de datos, código fuente del software y el software integrado. La Tabla 7 muestra estos entregables a evaluar, e indica la etapa del ciclo de vida del software en el que se encuentran.

Tabla 7 Entregables que serán evaluados por el método de calidad

ID	Entregable	Etapa según modelo en cascada	Etapa según ISO/IEC 15288(2008)	Etapa según ISO/IEC 12207(2008)
1	Especificación de requerimientos del software.	Análisis	Proceso de análisis de requerimientos	Análisis de requerimientos del sistema
2	Diseño de la arquitectura del software	Diseño	Proceso de diseño de arquitectura	Diseño de arquitectura del software
3	Diseño de la base de datos.	Diseño	Proceso de diseño de arquitectura	Diseño detallado del software
4	Código fuente del software	Implementación	Proceso de Implementación	Codificación del software y pruebas
5	Software integrado	Implementación	Proceso de Integración	Integración del software.

Elaboración: el autor

En la Figura 16 se presenta los entregables a evaluar según el ciclo de vida del software. En la Tabla 8 se presenta los entregables y los objetivos de evaluación de cada uno.

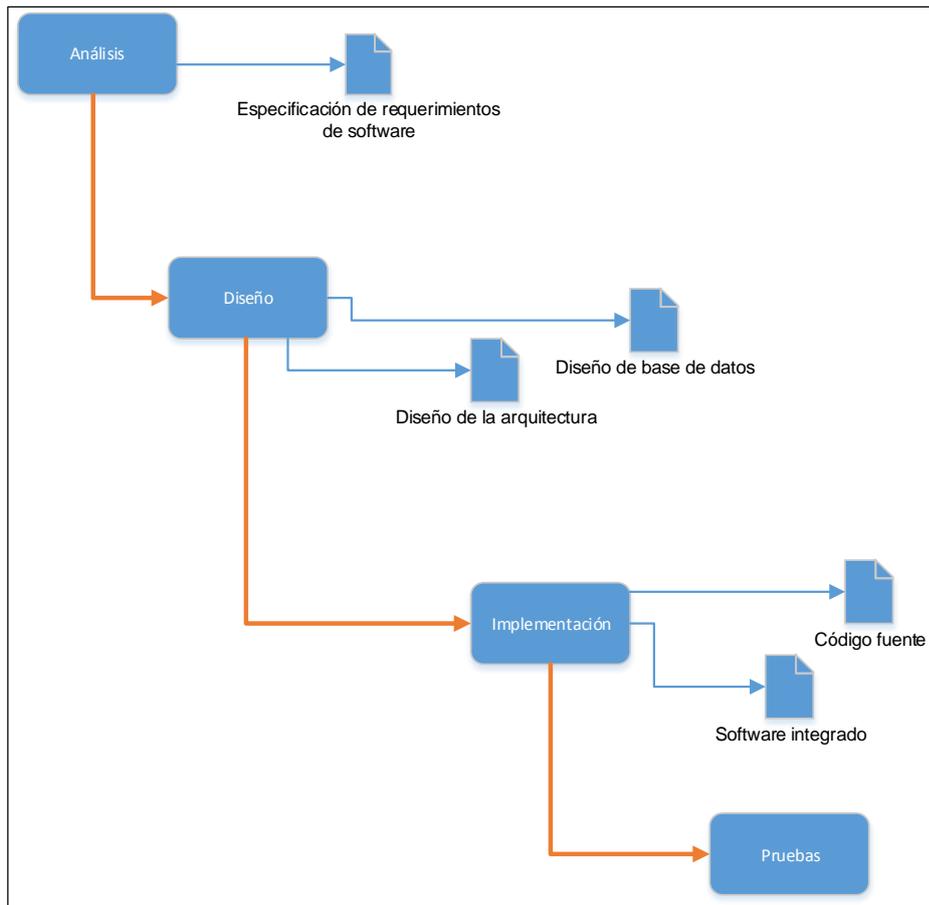


Figura 16 Entregables a evaluar según el modelo de desarrollo en cascada
Elaboración: el autor

Tabla 8 Entregables y los objetivos de la evaluación de calidad

Etapa del ciclo de vida	Entregable	Descripción	Objetivo de la evaluación de calidad
Análisis	Especificación de requerimientos del software	Es el documento que contiene los requisitos funcionales y no funcionales documentados, estos requisitos son el resultado de la etapa de análisis y generalmente se encuentran en un documento de análisis, especificación funcional, especificación de requerimientos u otro equivalente, según el Anexo C de la ISO/IEC 14598-5 (1998) es la Especificación de Requerimientos del Software.	Verificar que los requisitos funcionales y no funcionales cubren las necesidades de los usuarios y stakeholders.
Diseño	Diseño de la arquitectura del software	Es el diagrama de la arquitectura del software, los componentes del diagrama deben estar descritos. Este entregable según el Anexo C de ISO/IEC 14598-5 (1998) está incluido en el documento Diseño y especificación del sistema.	Verificar que el diseño de la arquitectura sea soportado por la arquitectura existente en la empresa; es decir, esta evaluación de calidad debe verificar si la arquitectura diseñada es acorde a la arquitectura de la empresa, tanto en hardware, software, redes y conectividad y estándares de diseño.
Diseño	Diseño de base de datos del software	El diseño de la base de datos contempla el diagrama entidad relación y el diccionario de datos. Este entregable según el Anexo C de ISO/IEC 14598-5 (1998) está incluido en el documento Diseño y especificación del sistema.	Verificar que el diseño de base de datos cumpla estándares de la empresa.
Implementación	Código fuente del software	Son los archivos que contienen el código fuente de la aplicación, según el Anexo C de la ISO/IEC 14598-5 (1998) es el Programa Fuente.	Verificar que el código fuente del producto software cumple los estándares de la empresa, así como los estándares de desarrollo seguro, mejores prácticas de desarrollo y las políticas de seguridad de información.
Implementación	Software integrado	Es el producto terminado publicado en el ambiente de pruebas. Este entregable según el Anexo C de ISO/IEC 14598-5 (1998) es llamado Sistema.	Verificar que el software terminado cumple los requisitos funcionales y no funcionales definidos para el producto final.

Elaboración: el autor

El método considera la participación de dos roles: (1) Solicitante y (2) Evaluador, la Tabla 9 detalla los dos roles con sus respectivas responsabilidades.

Tabla 9 Roles que participan en la evaluación de calidad

	Rol Solicitante	Rol Evaluador
Descripción del Rol	<p>Persona o empresa que solicita la evaluación de calidad.</p> <p>El Solicitante podría ser la:</p> <p>(1) Persona o empresa que está desarrollando el software y desea conocer la calidad del software y sus respectivos entregables.</p> <p>(2) Persona o empresa que es el cliente de un proveedor de desarrollo de software, quien necesita conocer la calidad del software y los entregables involucrados que su proveedor le está entregando.</p>	<p>Persona o empresa que realiza el proceso de evaluación de calidad, a solicitud del “Solicitante”, y debe tener en cuenta las cinco fases del método de evaluación de calidad.</p> <p>Es mandatorio que el “Evaluador” no haya formado parte del equipo que participó en el desarrollo del entregable a evaluar, esto asegurará una evaluación imparcial del producto a evaluar.</p>
Responsabilidades	<ul style="list-style-type: none"> - Realizar la solicitud de evaluación. - Facilitar la información, datos y recursos necesarios al “Evaluador” para realizar sus tareas. - Tomar las acciones correctivas pertinentes, basado en los resultados de la evaluación de calidad. 	<ul style="list-style-type: none"> - Ejecutar las actividades de las cinco fases del método. - El “Evaluador” como responsable de la evaluación de calidad, debe definir si requiere la participación de personas con un perfil especializado según el entregable seleccionado, esto se debe incluir en el plan de evaluación considerado en la Fase 3 del método. - Entregar al “Solicitante”, el resultado final de la aplicación del método, que es el reporte de evaluación de calidad realizado por el “Evaluador”.

Elaboración: el autor

El método propuesto para la evaluación de calidad del software, consta de cinco fases y está basado en los estándares ISO/IEC 25040 (2010) e ISO/IEC 25041 (2012) proceso para evaluadores (anteriormente ISO/IEC 14598-5). Las fases del método son:

- Fase 1: Establecer los requisitos de la evaluación
- Fase 2: Especificación de la evaluación
- Fase 3: Diseño de la evaluación
- Fase 4: Ejecución de la evaluación
- Fase 5: Conclusión de la evaluación

En la Figura 17 se presentan las fases del método. A continuación se detallan cada una de las fases.

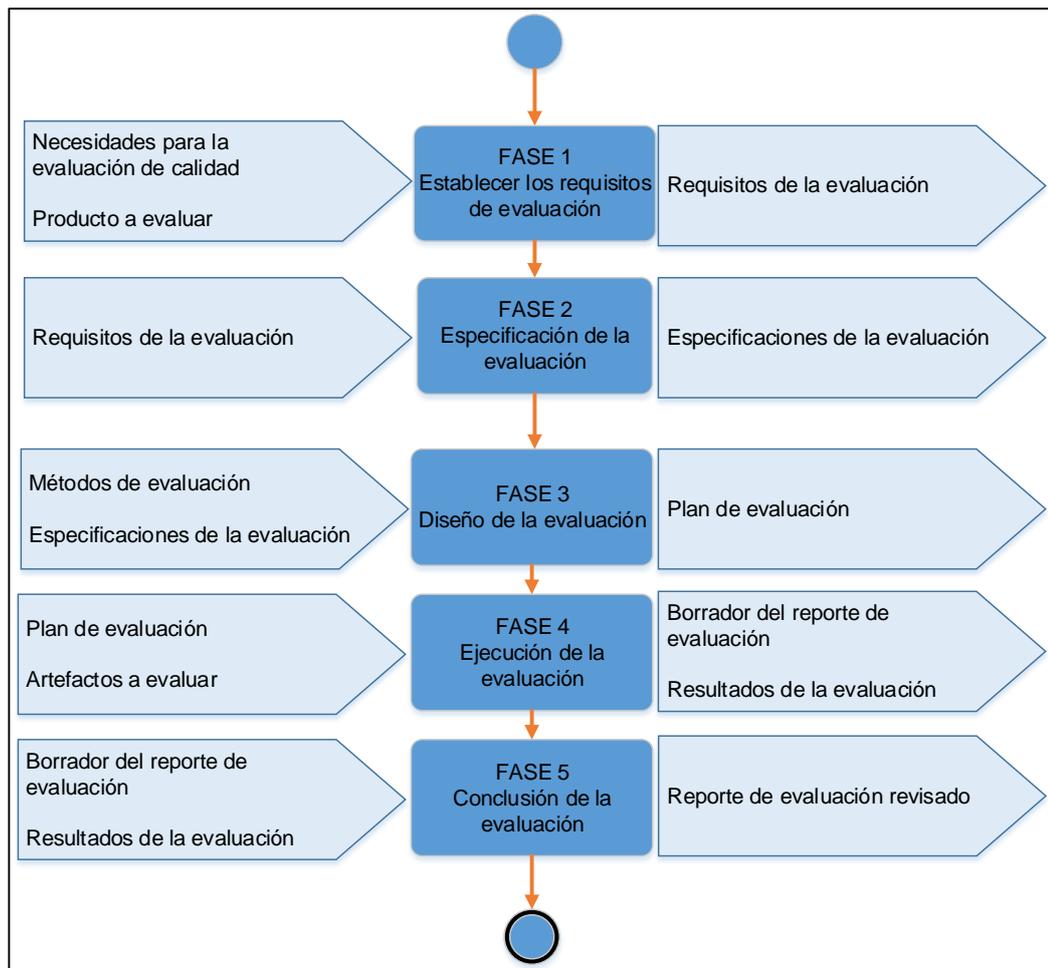


Figura 17 Fases de la evaluación de calidad

Fuente: ISO/IEC 25040 (2010)

Elaboración: el autor

4.1 Fase 1: Establecer los requisitos de la evaluación

4.1.1 Propósito

Definir los requisitos que debe considerar la evaluación. En esta fase se debe definir qué componentes se evaluarán, y las características y subcaracterísticas de calidad que se van a considerar en la evaluación.

4.1.2 Precondiciones

Que exista una solicitud de evaluación de calidad

4.1.3 Entradas

- **Necesidades de la evaluación de calidad**

Se debe especificar claramente las necesidades de evaluación de calidad de los entregables a evaluar.

En la Tabla 10 se describen las necesidades de la evaluación de calidad de cada entregable, definidas por el rol “Solicitante”.

Tabla 10 Necesidades de la evaluación de calidad de cada entregable

Entregable	Necesidades de la evaluación de calidad
Especificación de requerimientos del software	Verificar que el entregable cubra las necesidades de los stakeholders relevantes. Verificar que el entregable cuente con las aprobaciones de los stakeholders relevantes.
Diseño de la arquitectura del software	Verificar el entregable sea soportado por la infraestructura actual: (1) hardware, (2) software, (3) redes, (4) políticas.
Diseño de la base de datos	Verificar que el entregable cumpla las políticas y estándares de diseño de base de datos.
Código fuente del software	Verificar que el entregable cumpla lo siguiente: (1) Autenticación según políticas de seguridad de información, (2) Inyección SQL (OWASP, 2013), (3) Manejo de excepciones y errores (OWASP, 2013), (4) Validación de datos de entrada, (5) Uso de parámetros, (6) Estándares de nomenclatura y arquitectura y (7) Seguridad de componentes y servicios según políticas de seguridad de información.
Software integrado	Verificar que el software implementado cubra los requisitos funcionales y no funcionales (etapa de testing en ambiente de pruebas).

Elaboración: el autor

- **Productos a evaluar**

Los productos que evalúa el método, son los entregables detallados en la Tabla 8.

El criterio de selección de los entregables a considerar en la evaluación se detalla en la Actividad 2 (Productos a ser incluidos en la evaluación).

4.1.4 Actividad 1 - Elaboración de los requisitos de la evaluación

- **Requisitos de la evaluación**

Incluir una descripción de los requisitos de la evaluación definidos por el solicitante de la evaluación de calidad.

- **Cobertura de la evaluación**

Describir la cobertura de la evaluación definida por el solicitante; es decir, se debe definir los productos que se deben evaluar y su descripción.

- **Motivo de la evaluación**

Describir el motivo que tiene el solicitante para solicitar la evaluación, temas críticos como seguridad, costos, aspectos ambientales o leyes deben ser tomados en cuenta.

- **Grado de confianza y rigor de la evaluación**

Se debe definir la rigurosidad de la evaluación, con el objetivo de proporcionar confianza de la calidad del producto software.

4.1.5 Actividad 2 – Documentar los requisitos de la evaluación

El contenido de la documentación de esta fase debe incluir lo siguiente:

- **Descripción del dominio de la aplicación del producto software**

Describir el dominio del producto a evaluar. En la Tabla 11 se muestra un listado de los posibles dominios de aplicación del software.

Tabla 11 Ejemplos de dominios de aplicación de software

Dominio	Descripción
Core Empresa	Aplicación que participa en el core del negocio de crédito vehicular.
Atención al Usuario	Aplicación que implica la atención al cliente.
Soporte Negocio	Aplicaciones que utilizan las áreas de soporte. Incluye Recursos Humanos, Administración, Riesgos y Finanzas
Soporte Sistemas	Aplicaciones para Mesa de Ayuda, Desarrollo y Proyectos de Sistemas.

Elaboración: el autor

- **Descripción del propósito de la evaluación**

El propósito de la evaluación es: evaluar la calidad de los entregables seleccionados. Esta evaluación de calidad debe cubrir las necesidades de evaluación definidos en la Tabla 10.

- **Lista de requerimientos de calidad**

Los requerimientos de calidad deben tener como componentes las características y subcaracterísticas de calidad definidas en ISO/IEC 25030 (2007).

Se debe seleccionar las características y subcaracterísticas de calidad que debe cumplir cada entregable, el listado completo se muestra en la Tabla 13.

- **Nivel de importancia de las características de calidad**

Para expresar esta importancia podemos utilizar el concepto de nivel de evaluación descrito en el anexo B de ISO/IEC 14598-5 (1998). Este nivel de importancia nos ayudará a enfocar los recursos disponibles, en aquellas características y subcaracterísticas de calidad ubicadas en los niveles que están por encima del umbral de riesgo aceptado de la empresa.

En la Tabla 14 se muestran ejemplos de niveles de importancia. Se debe elegir el nivel de importancia de cada característica y subcaracterística de calidad.

- **Productos a ser incluidos en la evaluación**

Se debe seleccionar los entregables de la Tabla 7 que serán considerados en la evaluación de calidad. El criterio para seleccionar los entregables a evaluar, es la etapa del ciclo de vida del software; es decir: (1) al culminar la etapa de análisis se debe evaluar los entregables de esta etapa, (2) al culminar la etapa de diseño se debe evaluar los entregables de esta etapa y (3) al culminar la etapa de implementación se deben evaluar los entregables de esta etapa. Por lo tanto debemos iterar el método tres veces, considerando los entregables indicados en la Tabla 12.

Tabla 12 Entregables a evaluar en cada iteración del método

Número de iteración del Método de Evaluación de calidad	Entregables e Evaluar
Iteración 1, al finalizar la etapa de análisis.	Entregable 1 de la Tabla 7: Documento con la “Especificación de los requerimientos del software”
Iteración 2, al finalizar la etapa de diseño.	Entregables 2 y 3 de la Tabla 7: Documento con el “Diseño de la arquitectura del software” Documento con el “Diseño de base de datos del software”
Iteración 3, al finalizar la etapa de implementación.	Entregables 4 y 5 de la Tabla 7: Archivos que contienen el “Código fuente del software” “Software integrado” y publicado en el ambiente de pruebas

Elaboración: el autor

4.1.6 Actividad 3 - Aprobación

Se requiere aprobación de la documentación generada en la Actividad 2, la aprobación debe ser por parte del solicitante y del evaluador.

4.1.7 Herramientas

- a) ISO/IEC 25030 (2007) Requisitos de calidad.
- b) La Tabla 13 muestra las características y subcaracterísticas de Calidad según ISO/IEC 25010 (2010).

Tabla 13 Características y subcaracterísticas de calidad

Funcionalidad	
Complejidad funcional	Grado en el que el conjunto de funciones cubre todas las tareas y los objetivos especificados por el usuario.
Exactitud funcional	Grado en que un producto o sistema proporciona los resultados correctos con el grado necesario de precisión.

Idoneidad funcional	Grado en el que las funciones facilitan la realización de tareas y objetivos especificados.
Rendimiento	
Comportamiento en el tiempo	Grado en que los tiempos de respuesta y procesamiento y las tasas de rendimiento de un producto o sistema, al realizar sus funciones, cumplen con los requisitos.
Utilización de recursos	Grado en el que las cantidades y tipos de recursos utilizados por un producto o sistema al realizar sus funciones cumple con los requisitos.
Capacidad	Grado en que cumplen los requisitos respecto de los límites máximos de los parámetros de un producto o sistema. Ejemplo: Cantidad de usuarios concurrentes.
Compatibilidad	
Coexistencia	Grado en que un producto puede llevar a cabo sus funciones requeridas de manera eficiente mientras comparten un entorno y recursos con otros productos, sin impacto perjudicial en los otros productos.
Interoperabilidad	Grado en el cual dos o más sistemas, productos o componentes pueden intercambiar información y utilizar la información que se ha intercambiado.
Usabilidad	
Reconocimiento de Idoneidad	Grado en el cual los usuarios pueden reconocer si un producto o sistema es apropiado para sus necesidades.
Facilidad de Aprendizaje	Grado en que un producto o sistema puede ser utilizado para el aprendizaje del uso del producto o sistema con eficacia, eficiencia, ausencia de riesgo y satisfacción en un contexto de uso.
Operabilidad	Grado en que un producto o sistema tiene atributos que lo hacen fácil de operar y controlar.
Protección de errores de usuario	Grado en que el sistema protege a los usuarios de cometer errores.
Atractividad	Grado en el que la interfaz de usuario permite la interacción agradable y satisfactoria para el usuario.
Accesibilidad	Grado en que un producto o sistema pueden ser utilizados por personas con la más amplia gama de características y capacidades para alcanzar un objetivo especificado en un contexto de uso especificado.
Fiabilidad	
Madurez	Grado en que un sistema cumpla con las necesidades de fiabilidad bajo operación normal.

Disponibilidad	Grado en que un sistema, producto o componentes está operativo y accesible cuando se requiere para su uso.
Tolerancia a fallos	Grado en que un sistema, producto o componente funciona como se esperaba a pesar de la presencia de fallos de hardware o software.
Capacidad de recuperación	Grado en el cual, en caso de una interrupción o un fracaso, un producto o sistema puede recuperar los datos directamente afectados y restablecer el estado deseado del sistema.
Seguridad	
Confidencialidad	Grado en que un producto o sistema se asegura de que los datos sean accesibles a las personas autorizadas a tener acceso.
Integridad	Grado en que un sistema, producto o componente impide el acceso o modificación no autorizado de programas o los datos.
No repudio	Grado en el que las acciones o eventos se pueden probar que han ocurrido, por lo que los eventos o acciones no pueden ser repudiados.
Responsabilidad	Grado en el que las acciones de una entidad pueden rastrearse y que corresponden únicamente a dicha entidad.
Autenticidad	Grado en el que la identidad de un sujeto o recurso se puede probar que es quien dice ser.
Mantenibilidad	
Modularidad	Grado en que un sistema o programa se compone de componentes discretos de tal manera que un cambio en uno de los componentes tiene un impacto mínimo en otros componentes.
Reusabilidad	Grado en que un elemento se puede utilizar en más de un sistema, o en la construcción de otros elementos.
Analizabilidad	Grado de eficacia y eficiencia con la que es posible evaluar el impacto sobre un producto o sistema de un cambio previsto a uno o más de sus partes, o para diagnosticar deficiencias o causas de los fallos, o para identificar las partes a ser modificadas.
Modificabilidad	Grado en que un producto o sistema pueden ser modificadas de manera efectiva y eficiente sin introducir defectos o degradar la calidad del producto existente.
Capacidad para ser probado	Grado de eficacia y eficiencia con la que los criterios de prueba se pueden establecer para un sistema, producto o componente y las pruebas se puede realizar para determinar si se han cumplido los criterios.

Portabilidad	
Adaptabilidad	Grado en que un producto o sistema puede eficazmente y eficientemente ser adaptado en distintas plataformas de hardware, software u otros entornos operativos o de uso.
Facilidad de instalación	Grado de eficacia y eficiencia con la que un producto o sistema puede ser instalado y / o desinstalado en un entorno especificado con éxito.
Reemplazabilidad	Grado en que un producto puede ser sustituido por otro producto para el mismo propósito en el mismo entorno.

Fuente: ISO/IEC 25010 (2010)

c) Nivel de importancia

Tabla 14 Niveles de importancia

Nivel de Importancia	Riesgo
Nivel D	Pérdida económica insignificante
Nivel C	Pérdida económica significativa (proyecto afectado)
Nivel B	Pérdida económica grande (proyecto en peligro de cancelación)
Nivel A	El desastre financiero (proyecto se cancelará)

Fuente: Anexo A ISO/IEC 25040 (2010)

Elaboración: el autor

En la Tabla 14 se muestran niveles de importancia basados en el Anexo A de ISO/IEC 25040 (2010) (anteriormente Anexo B de ISO/IEC 14598-5). El Anexo A del estándar ISO/IEC 25040 (2010) también indica algunas técnicas de evaluación de acuerdo a los niveles de evaluación.

d) Los siguientes estándares de la división de medición de la calidad: ISO/IEC 25021 (2011), ISO/IEC 25023 (2013).

4.1.8 Salidas

Requisitos de evaluación de calidad, la Figura 18 muestra la plantilla del documento.

Requisitos de Evaluación de Calidad	
1. Información de Identificación	
1.1 Identificación del Evaluador	Esta subsección contiene información relativa al evaluador: - Nombre de la persona responsable de la evaluación
1.2 Identificación del Reporte de Evaluación	Esta subsección contiene información de la identificación del reporte: - Identificador único del reporte (ej. Correlativo) - Número de páginas del reporte (Debe coincidir con la numeración de página del reporte)
2. Dominio de la aplicación	
3. Propósito de la evaluación	
4. Lista de requerimientos de calidad	
5. Productos a ser incluidos en la evaluación	
6. Control de cambios	Esta sección contiene el control de cambios del documento, con la versión, la persona que creó o modificó el documento y la fecha de creación o modificación.
7. Aprobaciones	Esta sección contiene el detalle de las aprobaciones del documento

Figura 18 Plantilla del documento de Requisitos de Evaluación de Calidad
Elaboración: el autor

4.2 Fase 2. Especificación de la evaluación

4.2.1 Propósito

Especificar las medidas de calidad, los métodos de evaluación a utilizar y los criterios de decisión para los requisitos definidos en la fase 1.

4.2.2 Precondiciones

Que se haya finalizado la fase 1, con la aprobación de los requerimientos de evaluación.

4.2.3 Entradas

- **Requisitos de la evaluación**

Salida de la fase 1.

4.2.4 Actividad 1 - Elaboración de la especificación de la evaluación

- **Analizar la descripción del producto a evaluar**

El objetivo es definir el alcance de la evaluación, debe quedar claro qué componentes del producto se deben evaluar; de la misma forma, el evaluador debe tener claro las relaciones entre los componentes del producto a evaluar.

- **Seleccionar las métricas de calidad (módulos de evaluación)**

El evaluador debe seleccionar las métricas de calidad a utilizar en el producto y sus componentes.

- **Especificar métricas de calidad para cubrir todos los requerimientos de evaluación (características y subcaracterísticas de calidad)**

La norma ISO/IEC 25023 (2013) nos muestra ejemplos de métricas que pueden ser utilizadas como están o pueden ser adaptadas a necesidades específicas.

- **Documentar los métodos de evaluación**

Se deben considerar las acciones a ser realizadas para obtener los resultados de la evaluación, si es necesario un procedimiento para interpretar los resultados de la medición, este también debe estar detallado. La norma ISO/IEC 14598-6 (1999) especifica como agrupar los métodos de evaluación dentro de un módulo de evaluación.

- **Definir los criterios de decisión para las métricas de calidad**

Los criterios de decisión deben ser definidos para cada métrica individual.

- **Verificar la especificación de la evaluación con los requerimientos de evaluación**

El evaluador debe realizar una verificación de la especificación de la evaluación respecto a los requerimientos de la evaluación de calidad, también se debe verificar que se tiene la información necesaria para realizar la evaluación según los requerimientos de calidad definidos.

Las mediciones y verificaciones deben ser suficientes para cubrir los objetivos de la evaluación expresados en los requerimientos de evaluación.

4.2.5 Actividad 2 – Documentar la especificación de la evaluación

El contenido de la documentación debe incluir como mínimo lo siguiente:

- **El alcance de la evaluación, respecto de los componentes del producto a ser evaluados**

El alcance es: realizar la evaluación de calidad de los entregables seleccionados en la Actividad 2 de la fase 1 del método, alineado al propósito también definido en la Actividad 2 de la fase 1.

- **Referencia cruzada entre la información necesaria para realizar la evaluación y los productos a evaluar (Entregables)**

La Tabla 15 detalla la información necesaria por cada entregable susceptible a ser evaluado. Se debe considerar únicamente a los entregables incluidos en la evaluación.

Tabla 15 Referencia cruzada entre la información necesaria y los productos a evaluar

Información Necesaria	Productos a evaluar
Requisitos funcionales y no funcionales de los interesados	Entregable 1: Especificación de requerimientos del software
Evidencia de la conformidad de los requisitos funcionales y no funcionales del sistema	
Diagrama de la arquitectura del software	Entregable 2: Diseño de la arquitectura del software
Descripción de los elementos de la arquitectura del software	
Diagrama Entidad Relación con el modelo de la base de datos	Entregable 3: Diseño de la base de datos
Diccionario de datos del modelo	
Código fuente del producto software	Entregable 4: Código fuente del software
Requisitos funcionales y no funcionales del sistema	Entregable 5: Software integrado
Software integrado y publicado	

Elaboración: el autor

- **Especificación de las métricas de calidad y sus criterios de decisión, incluyendo el detalle de los métodos de evaluación que se utilizarán**

Se deben especificar las métricas de calidad para los requisitos definidos en la fase 1 (características y subcaracterísticas de calidad) con sus respectivos criterios de decisión. En el ítem b) del punto 4.2.7 se referencian los documentos que contienen métricas de calidad que pueden ser utilizadas como están o ser personalizadas según el contexto.

En este punto también debemos incluir los métodos de evaluación que utilizaremos. En el ítem c) del punto 4.2.7 se presenta una lista de métodos de evaluación que podemos utilizar.

Una forma de empaquetar los métodos de evaluación, las métricas de calidad y los criterios de decisión, es utilizando los módulos de evaluación que se describen en detalle en el documento referenciado en el ítem a) del punto 4.2.7.

4.2.6 Actividad 3 - Aprobación

Se requiere aprobación de la documentación generada en la Actividad 2; la aprobación debe ser por parte del solicitante y del evaluador.

4.2.7 Herramientas

- a) ISO/IEC 14598-6 (1999) Módulos de evaluación
- b) ISO/IEC 25023 (2013) Medición de la calidad del producto software. Las Tablas 16, 17, y 18 detallan algunas métricas de calidad específicas.

Tabla 16 Métrica de calidad para la subcaracterística Completitud Funcional

ID	Nombre	Descripción	Función de medición y QME
FCP-G-1	Cobertura de implementación funcional	¿Qué tan completa es la implementación de acuerdo a las especificaciones de los requerimientos?	$X = 1 - A/B$ A = Número de funciones que faltan o están incorrectamente implementadas, que han sido detectadas en la evaluación. B = Número de funciones establecidas en la especificación de los requerimientos.

NOTA: Una función que falta o está incorrectamente implementada puede ser:
 a) Función que no funciona según lo especificado en los manuales de usuario, especificación de requisitos o especificaciones de diseño.
 b) Función que no proporcionan un resultado razonable y aceptable para lograr el objetivo específico previsto de la tarea de usuario.

Fuente: ISO/IEC 25023 (2013)

Tabla 17 Métricas de calidad para la subcaracterística Idoneidad Funcional

ID	Nombre	Descripción	Función de medición y QME
FAP-G-1	Idoneidad funcional	¿Qué parte de las funciones implementadas son percibidas como idóneas?	$X = A / B$ A = Número de funciones realmente útiles para realizar tareas específicas. B = Número de funciones implementadas para la consecución de tareas específicas

NOTA: Un ejemplo de idoneidad funcional, es que el usuario solo cuenta con las funciones necesarias para completar una tarea, con exclusión de cualesquiera funciones innecesarias.

FAP-S-1	Estabilidad de la especificación funcional	¿En qué medida los requisitos funcionales han cambiado después de iniciar el desarrollo?	$X = 1 - A / B$ A = Número de funciones cambiados durante la etapa de desarrollo. B = Número de funciones descritas en la especificación de requisitos.
----------------	--	--	---

Fuente: ISO/IEC 25023 (2013)

Tabla 18 Métricas de calidad para la subcaracterística Madurez

ID	Nombre	Descripción	Función de medición y QME
RMA-G-1	Eliminación de fallos	¿Qué proporción de errores detectados han sido corregidos?	$X = A / B$ A = Número de errores corregidos en la etapas de diseño / desarrollo / pruebas B = Número de errores detectados en las revisiones o pruebas.

Fuente: ISO/IEC 25023 (2013)

- c) El Anexo B de la ISO/IEC 25040 (2010) provee ejemplos de métodos de evaluación para las medidas de calidad. La Tabla 19 lista algunos métodos de evaluación según la característica de calidad que requiere ser evaluada.

Tabla 19 Ejemplos de métodos de evaluación según la característica de calidad seleccionada

Característica de Calidad	Método de Evaluación
Funcionalidad	Revisión de documentación técnica y de usuario Revisión de la especificación de los requerimientos del software Inspección de código Pruebas de caja negra de software
Mantenibilidad	Análisis del diseño de arquitectura del software Análisis de la trazabilidad de los documentos
Fiabilidad	Análisis de riesgos en el diseño del software

Fuente: ISO/IEC 25040 (2010)

4.2.8 Salidas

Especificaciones de la evaluación de calidad, la Figura 19 muestra la plantilla del documento.

Especificaciones de la Evaluación de Calidad

1. Información de Identificación

1.1 Identificación del Evaluador

Esta subsección contiene información relativa al evaluador:
- Nombre de la persona responsable de la evaluación

1.2 Identificación del Reporte de Evaluación

Esta subsección contiene información de la identificación del reporte:
- Identificador único del reporte (ej. Correlativo)
- Número de páginas del reporte (debe coincidir con la numeración de página del reporte)

2. Alcance de la evaluación

3. Referencia cruzada entre la información necesaria para realizar la evaluación y los componentes del producto referenciados en su descripción

4. Especificación de la métricas de calidad y sus criterios de decisión

5. Control de cambios

Esta sección contiene el control de cambios del documento, con la versión, la persona que creó o modificó el documento y la fecha de creación o modificación.

6. Aprobaciones

Esta sección contiene el detalle de las aprobaciones del documento

Figura 19 Plantilla del documento de Especificaciones de la Evaluación de Calidad

Elaboración: el autor

4.3 Fase 3. Diseño de la evaluación

4.3.1 Propósito

Planificar las actividades de la evaluación de calidad.

4.3.2 Precondiciones

Que se haya finalizado la Fase 2, con la aprobación de la especificación de las medidas de calidad y sus criterios de decisión.

4.3.3 Entradas

- **Especificaciones de la evaluación**

Salida de la fase 2.

4.3.4 Actividad 1 - Elaboración del plan de evaluación

- **Optimizar el plan de evaluación**

Tener en cuenta que se aplicarán varios métodos de evaluación sobre el mismo producto o sus componentes, el plan debe ser revisado de tal forma que se eviten acciones duplicadas.

- **Programar acciones de evaluación considerando los recursos disponibles**

Una vez que las acciones duplicadas han sido removidas, se deben programar las acciones planificadas. Para este proceso se debe tener en cuenta la disponibilidad de los recursos (personas, equipos, herramientas).

- **Identificar los riesgos**

Considerar aquellos que podrían afectar a la ejecución del plan de evaluación de calidad.

- **Puntos adicionales a tener en cuenta**

Las reuniones y entrenamiento (capacitación).

4.3.5 Actividad 2 – Documentar el plan de evaluación

El contenido de la documentación de esta fase, debe incluir como mínimo lo siguiente:

- **Detalle del plan de trabajo con las acciones a realizar**

Detallar el plan de trabajo de la evaluación de calidad. Se debe considerar lo siguiente: (1) actividades, (2) personas

con su porcentaje de dedicación y responsabilidades, (3) equipos, (4) capacitación, (5) presupuesto, (6) restricciones, (7) hitos.

En el ítem a) del punto 4.3.7 se detalla una plantilla de plan de trabajo.

- **Listado de riesgos identificados**

Detallar los riesgos que podrían afectar la ejecución del plan de trabajo. A continuación se listan los riesgos que se deben considerar para obtener resultados satisfactorios en la aplicación del método:

- a) Falta de apoyo de la Gerencia o Jefatura para destinar recursos.
- b) No contar con personal capacitado para realizar la evaluación de calidad.
- c) Desinterés de los stakeholders por participar en la evaluación de calidad.

A este listado agregar los riesgos considerados pertinentes de acuerdo al caso particular; adicionalmente se debe mantener actualizado la base de conocimientos de riesgos identificados para facilitar su aplicación en evaluaciones posteriores.

4.3.6 Actividad 3 - Aprobación

Se requiere aprobación del plan de trabajo generado en la Actividad 2; la aprobación debe ser por parte del solicitante y del evaluador.

4.3.7 Herramientas

a) La Figura 20 muestra una plantilla de plan de trabajo de evaluación de calidad.

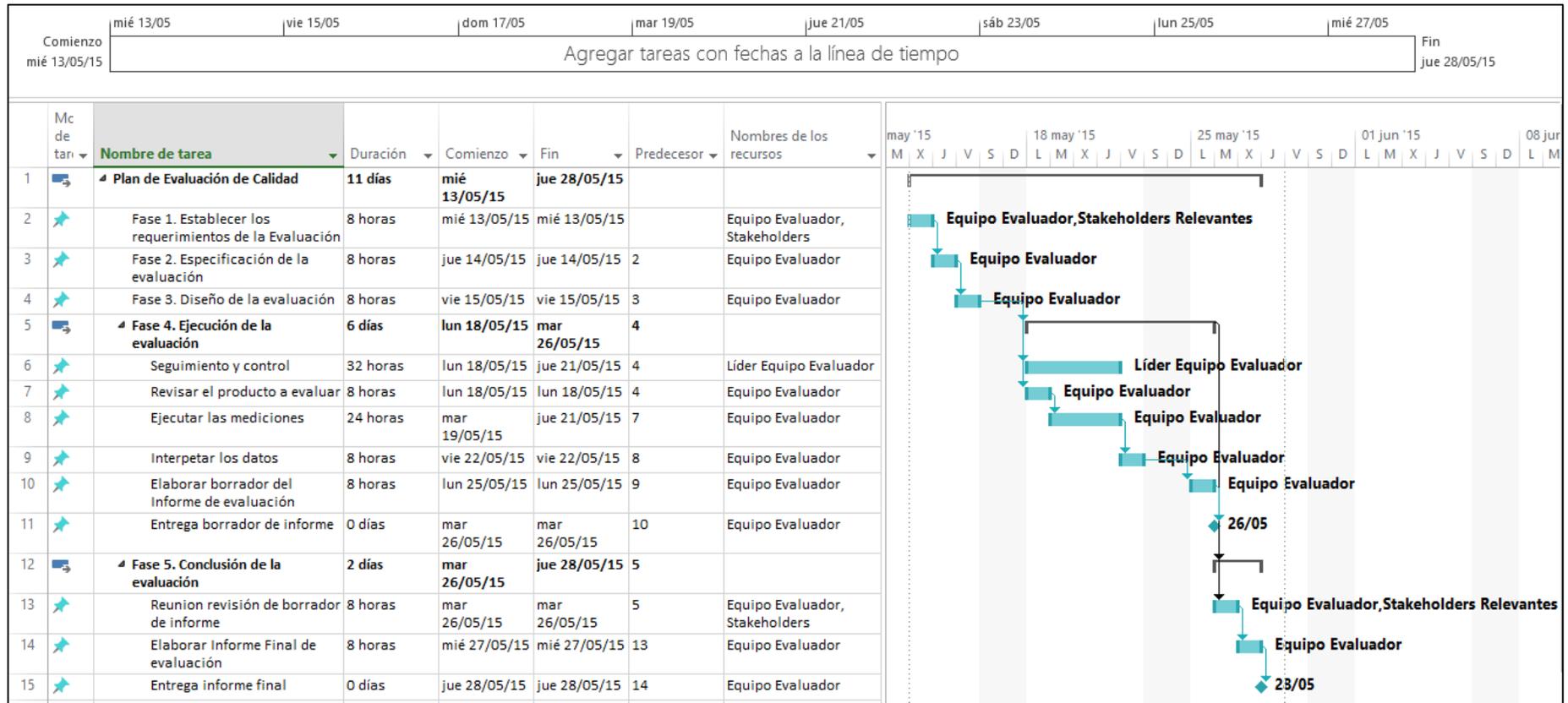


Figura 20 Plantilla de plan de trabajo de la evaluación de calidad
Elaboración: el autor

4.3.8 Salidas

Diseño de la evaluación de calidad, que contiene principalmente el plan de trabajo de la evaluación de calidad. La Figura 21 muestra la plantilla del documento.

Diseño de la Evaluación de Calidad	
1. Información de Identificación	
1.1 Identificación del Evaluador	Esta subsección contiene información relativa al evaluador: - Nombre de la persona responsable de la evaluación
1.2 Identificación del Reporte de Evaluación	Esta subsección contiene información de la identificación del reporte: - Identificador único del reporte (ej. Correlativo) - Número de páginas del reporte (Debe coincidir con la numeración de página del reporte)
2. Personas y Responsabilidades	
3. Restricciones	
4. Riesgos	
5. Plan de trabajo	
6. Control de cambios	Esta sección contiene el control de cambios del documento, con la versión, la persona que creó o modificó el documento y la fecha de creación o modificación.
7. Aprobaciones	Esta sección contiene el detalle de las aprobaciones del documento

Figura 21 Plantilla del documento de Diseño de Evaluación de Calidad
Elaboración: el autor

4.4 Fase 4. Ejecución de la evaluación

4.4.1 Propósito

Ejecutar las actividades de la evaluación de calidad y documentarlas debidamente.

4.4.2 Precondiciones

Que se haya finalizado la fase 3, con la aprobación del plan de trabajo.

4.4.3 Entradas

- **Plan de evaluación**

Salida de la fase 3.

- **Especificación de la evaluación**

Salida de la fase 2.

- **Requerimientos de evaluación**

Salida de la fase 1.

4.4.4 Actividad 1 - Ejecutar las acciones del plan de evaluación

- **Gestionar los componentes del producto entregados por el solicitante**

Según el plan definido, el solicitante debe entregar los productos a evaluar, y el evaluador debe registrar estos productos o sus componentes y los documentos relacionados. La información de registro debe contener los siguientes datos: nombre del componente, condición del componente (anomalías especiales o condiciones físicas), versión y fecha de recepción.

- **Confidencialidad y documentación**

La confidencialidad de los componentes y la documentación relacionada debe ser acordada con el solicitante.

- **Gestionar los datos que son producto de las acciones de evaluación (incluyendo registros y reportes)**

Producto de las acciones de evaluación se obtienen datos para producir los resultados que serán incluidos en el reporte de evaluación, estos datos pueden ser números, gráficos, diagramas entre otros. La confidencialidad de esta información debe ser protegida en la misma forma

que los componentes evaluados. En resumen el evaluador debe registrar todos los datos intermedios necesarios para los resultados del reporte de evaluación.

- **Gestionar las herramientas que se utilizarán para realizar las acciones de evaluación**

Cuando una herramienta de software es necesaria para recolectar datos o interpretarlos, una referencia a esta herramienta debe ser incluida en el reporte de evaluación; la información mínima será: el nombre de la herramienta, el fabricante, y la versión (son ejemplos los analizadores de código y las herramientas CASE para modelamiento de software). Se debe tener en cuenta que el evaluador y su equipo deben estar entrenados apropiadamente en el uso de las herramientas necesarias.

- **Requerimientos sobre técnicas de evaluación específicos**

Se debe incluir las técnicas específicas; por ejemplo, el uso de listas de verificación cuando una acción de evaluación requiere la inspección de un documento.

4.4.5 Actividad 2 – Documentar la ejecución de la evaluación

El contenido de la documentación debe incluir como mínimo lo siguiente:

- **Detalle de los componentes evaluados**

Se debe detallar los componentes evaluados, se sugiere utilizar la plantilla detallada en el ítem a) del punto 4.4.7.

- **Detalle de las herramientas de software utilizadas**

Se debe detallar el software utilizado para la evaluación de calidad y la ejecución del mismo, se sugiere utilizar la plantilla detallada en el ítem b) del punto 4.4.7.

- **Resultados de la aplicación de técnicas de evaluación utilizadas**

Para cada evaluación de calidad de los entregables, se deben guardar los datos de acuerdo a las técnicas específicas utilizadas. Por ejemplo en el entregable 1, utilizaremos la plantilla de lista verificación detallada en el ítem c) del punto 4.4.7.

Estos puntos deben ser incluidos en la sección 5(Resultados de la evaluación) del reporte de evaluación de calidad (Figura 22).

- **Elaborar el borrador del reporte de evaluación**

En esta actividad también se debe **elaborar el borrador completo del reporte de evaluación de calidad**, que debe contener principalmente los resultados obtenidos, como se menciona en el párrafo anterior. La plantilla sugerida es la mostrada en la Figura 22.

4.4.6 Actividad 3 - Revisión e Informe

- **Revisión de la evaluación**

Con la finalidad de lograr la máxima objetividad de la evaluación, es importante que todas acciones de evaluación sean revisadas por una persona diferente a la que realizó la acción. Los resultados de esta revisión deben estar incluidos en los registros de la evaluación.

- **Elaborar el informe de evaluación**

Una vez revisados los resultados de la evaluación, estos deben ser incluidos en el reporte de evaluación de calidad.

4.4.7 Herramientas

- a) Formato para detallar los componentes del producto software evaluados. Si hay más de dos componentes evaluados, se debe crear una tabla por cada componente. Las Tablas 20 y 21 muestran el formato con ejemplos de su aplicación.

Tabla 20 Ejemplo de detalle del componente Documento de Análisis

ID	Componente 1
Nombre del Componente	Documento de Análisis
Descripción	Documento que contiene los requisitos funcionales y no funcionales del software.
Formato de recepción	Documento físico de 12 páginas.
Condición del componente	Documento en buen estado.
Versión del componente	1.3
Fecha de recepción	16/05/2015
Confidencialidad	Documento confidencial de uso únicamente para la evaluación de calidad.

Elaboración: el autor

Tabla 21 Ejemplo de detalle del componente Código fuente del software

ID	Componente 2
Nombre del Componente	Código fuente del software
Descripción	Archivo que contiene el código fuente del software.
Formato de recepción	Carpeta con 114 archivos.
Condición del componente	Integridad de los archivos verificados con algoritmos hash 3DES.
Versión del componente	3.5
Fecha de recepción	16/05/2015
Confidencialidad	Archivos confidenciales.

Elaboración: el autor

- b) Formato para detallar el software utilizado en la evaluación de calidad, la Tabla 22 muestra el formato con un ejemplo de su aplicación.

Tabla 22 Ejemplo de detalle del software utilizado en la evaluación de calidad

ID	Nombre del Software	Descripción	Fabricante	Versión
1	Microsoft Word	Software para redactar el informe de evaluación de calidad y los documentos necesarios para su construcción.	Microsoft	2003
2	Microsoft Excel	Software para elaborar los gráficos y los cuadros para el informe final.	Microsoft	2003
3	Microsoft Project	Software para elaborar el plan de trabajo, seguimiento y control de la ejecución.	Microsoft	2003
4	Visual Studio 2012	Software para abrir y revisar el código fuente de la aplicación.	Microsoft	2012

Elaboración: el autor

c) Plantilla de lista de verificación para la evaluación de calidad del entregable 1; esta plantilla se debe completar con la información de entrevistas.

- En la columna Documento, se detallarán los requisitos funcionales y no funcionales del software.
- En la columna Entrevista Usuario Final, se escribirá:
 - (1) **Conforme**, cuando el requisito funcional o no funcional cubre la necesidad del Usuario Final.
 - (2) **No Conforme**, cuando el requisito funcional o no funcional NO cubre la necesidad del Usuario Final, en este caso adicionalmente se debe detallar el motivo de la no conformidad.
 - (3) **No aplica**, cuando el requisito funcional o no funcional es de dominio de la Jefatura del Usuario Final o de la Jefatura de Proyectos de Sistemas. Ejemplo: Motor de Base de datos a utilizar, en este caso el responsable será la Jefatura de Proyectos de Sistemas.

- En la columna Entrevista Usuario Final, se escribirá:
 - (1) **Conforme**, cuando el requisito funcional o no funcional cubre la necesidad de la Jefatura del Usuario Final.
 - (2) **No Conforme**, cuando el requisito funcional o no funcional NO cubre la necesidad de la Jefatura del Usuario Final, en este caso adicionalmente se debe detallar el motivo de la no conformidad.
 - (3) **No aplica**, cuando el requisito funcional o no funcional no es de dominio de la Jefatura del Usuario Final.
- En la columna Entrevista Jefatura de Proyectos Sistemas, se escribirá:
 - (1) **Conforme**, cuando el requisito funcional o no funcional cubre la necesidad de la Jefatura de Proyectos Sistemas.
 - (2) **No Conforme**, cuando el requisito funcional o no funcional NO cubre la necesidad de la Jefatura de Proyectos Sistemas, en este caso adicionalmente se debe detallar el motivo de la no conformidad.
 - (3) **No aplica**, cuando el requisito funcional o no funcional no es de dominio de la Jefatura de Proyectos Sistemas. Ejemplo: La funcionalidad deseada por el usuario final.
- Si en las entrevistas, algún requisito funcional o no funcional no es encontrado en el documento, se debe incluir el requisito, similar a la fila Otros, y debe ser evaluado de forma similar que el resto de requisitos.

Tabla 23 Ejemplo de detalle del software utilizado en la evaluación de calidad

Requisitos funcionales			
Requisitos	Entrevista Usuario Final	Entrevista Jefatura Usuario Final	Entrevista Jefatura de Proyectos Sistemas
RF1	Conforme	Conforme	Conforme
RF2	No aplica	No Conforme	No Conforme
RF3	Conforme	Conforme	Conforme
RF4	No aplica	No Conforme	No Conforme

RF5	Conforme	Conforme	Conforme
RF6	No aplica	No Conforme	No Conforme
RF7	Conforme	Conforme	Conforme
RF8	No aplica	No Conforme	No Conforme
RF9	Conforme	Conforme	Conforme
RF10	Conforme	Conforme	Conforme
Otros	No aplica	No Conforme	No Conforme

Elaboración: el autor

Tabla 24 Ejemplo de detalle del software utilizado en la evaluación de calidad

Requisitos NO funcionales			
Requisitos	Entrevista Usuario Final	Entrevista Jefatura Usuario Final	Entrevista Jefatura de Proyectos Sistemas
RNF1	Conforme	Conforme	Conforme
RNF2	No aplica	No Conforme	No Conforme
RNF3	Conforme	Conforme	Conforme
RNF4	No aplica	No Conforme	No Conforme
RNF5	Conforme	Conforme	Conforme
RNF6	No aplica	No Conforme	No Conforme
RNF7	Conforme	Conforme	Conforme
RNF8	No aplica	No Conforme	No Conforme
RNF9	Conforme	Conforme	Conforme
RNF10	Conforme	Conforme	Conforme
Otros	No aplica	No Conforme	No Conforme

Elaboración: el autor

4.4.8 Salidas

Borrador del reporte de evaluación de calidad según la plantilla mostrada en la Figura 22.

4.5 Fase 5. Conclusión de la evaluación

4.5.1 Propósito

Revisar y presentar el informe final de la evaluación de calidad.

4.5.2 Precondiciones

Que se haya finalizado la fase 4, con la presentación del borrador del reporte de evaluación de calidad.

4.5.3 Entradas

- **Borrador del reporte de evaluación**

Salida de la fase 4.

4.5.4 Actividad 1 - Revisión conjunta del reporte de evaluación

- **Personalizar el reporte de evaluación**

Es necesario evaluar la personalización del borrador del reporte de evaluación previo a su revisión con el público objetivo. Es decir: (1) cuando este reporte va dirigido a personas con conocimientos técnicos, el nivel de detalle debe ser mayor; y (2) cuando este reporte va dirigido a personas del nivel estratégico, como gerentes o directores, la información debe tener un mayor nivel de abstracción.

- **Revisión conjunta del reporte**

El borrador del reporte de evaluación debe ser entregado al solicitante de la evaluación. Posteriormente, se debe organizar una revisión conjunta entre el solicitante y el evaluador, en esta revisión conjunta el solicitante tiene la oportunidad de realizar comentarios sobre el reporte de evaluación de calidad.

4.5.5 Actividad 2 – Elaborar el reporte de evaluación de calidad final

Se debe modificar el borrador del reporte de evaluación de calidad con la retroalimentación obtenida de la reunión con el solicitante. Estas modificaciones finales dan lugar al reporte final de evaluación de calidad, la plantilla utilizada es la indicada en la Figura 22.

El reporte de evaluación final que se entrega al solicitante debe incluir los comentarios de la revisión conjunta.

4.5.6 Actividad 3 - Determinar el destino de los datos de la evaluación y los documentos

Una vez que el reporte de evaluación fue entregado al solicitante, el evaluador debe determinar el destino de los datos pertenecientes a la evaluación, algunas posibilidades son:

- (1) Entrega al solicitante
- (2) Archivado por un periodo, terminado este debe ser conservado por un tiempo adicional o destruido en forma segura
- (3) Destruído de una forma segura

En caso el evaluador pertenezca a la misma organización, se sugiere crear un repositorio para almacenar las evaluaciones; la política de seguridad de información de la empresa debe incluir este tipo de información en su detalle.

En caso el evaluador pertenezca a una organización externa, se sugiere indicar la entrega de toda la información, y la evidencia de la destrucción segura de la documentación y datos obtenidos para la evaluación de calidad, esto también debe formar parte de la política de seguridad de información de la empresa. Solo con el consentimiento del solicitante, los resultados de la evaluación pueden ser utilizados por el evaluador para estudiar técnicas de evaluación y métricas de software.

4.5.7 Herramientas

- a) La Figura 22 muestra la plantilla del reporte de evaluación de calidad.

Reporte de Evaluación de Calidad	
1. Información de Identificación	
1.1 Identificación del Evaluador	Esta subsección contiene información relativa al evaluador: - Nombre de la persona responsable de la evaluación
1.2 Identificación del Reporte de Evaluación	Esta subsección contiene información de la identificación del reporte: - Identificador único del reporte (ej. Correlativo) - Número de páginas del reporte (debe coincidir con la numeración de página del reporte)
1.3 Identificación del Solicitante	Esta sección contiene información del solicitante de la evaluación: - Nombre de la persona que solicita la evaluación
2. Requerimientos de Evaluación	Esta sección contiene los requerimientos de evaluación según la Fase 1 del método descrito.
3. Especificación de la Evaluación	Esta sección contiene los requerimientos de evaluación según la Fase 2 del método descrito.
4. Diseño de la Evaluación	Esta sección contiene los requerimientos de evaluación según la Fase 3 del método descrito.
5. Resultados de la Evaluación	Esta sección contiene los requerimientos de evaluación según las Fases 4 y 5 del método descrito.
6. Control de cambios	Esta sección contiene el control de cambios del documento, con la versión, la persona que creó o modificó el documento y la fecha de creación o modificación.
7. Aprobaciones	Esta sección contiene el detalle de las aprobaciones del documento

Figura 22 Plantilla de reporte de evaluación
Fuente: Anexo E ISO/IEC 25040 (2010)
Elaboración: el autor

4.5.8 Salidas

- Definición del destino de los datos de la evaluación y los documentos.

- Reporte final de evaluación de calidad del producto, según Figura 22.

4.6 Resumen de las Actividades

La Tabla 25 muestra un resumen de las actividades del proceso de evaluación de calidad de producto.

Tabla 25 Resumen de las entradas y salidas del procedimiento de evaluación

N°	Entrada	Fase	Salida
1	Requerimientos solicitados Producto a evaluar	Establecer los requerimientos de evaluación	Requerimientos de evaluación
2	Requerimientos de evaluación	Especificación de la evaluación	Especificaciones de la evaluación
3	Métodos de evaluación Especificaciones de la evaluación	Diseño de la evaluación	Plan de evaluación
4	Plan de evaluación Artefactos a evaluar	Ejecución de la evaluación	Borrador del reporte de evaluación Resultados de la evaluación
5	Borrador del reporte de evaluación Resultados de la evaluación	Conclusión de la evaluación	Reporte de evaluación revisado

Fuente: ISO/IEC 25040 (2010)

Elaboración: el autor

CAPÍTULO V

PRUEBAS Y RESULTADOS

5.1 Resumen Descriptivo

En este primer apartado realizamos una evaluación descriptiva.

Para realizar la comparación entre los dos grupos de proyectos es importante definir las similitudes entre ambos, para ello se utilizó el grado de complejidad de los proyectos a través de su duración (D), las Figuras 23 y 24 lo muestran:

- D: Duración en proyectos sin el método de evaluación de calidad (Grupo de control).
- D_ISO: Duración en proyectos sobre los que se aplicó el método para la evaluación de calidad de software basado en ISO/IEC 25000.

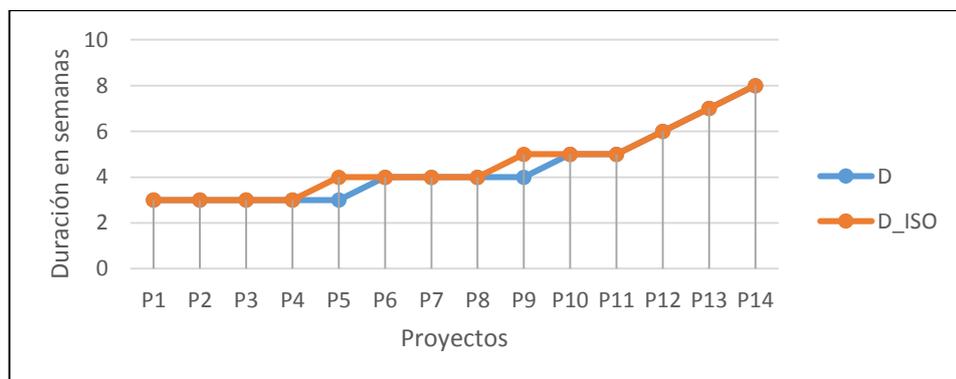


Figura 23 Gráfico descriptivo de la duración de los proyectos
Elaboración: el autor

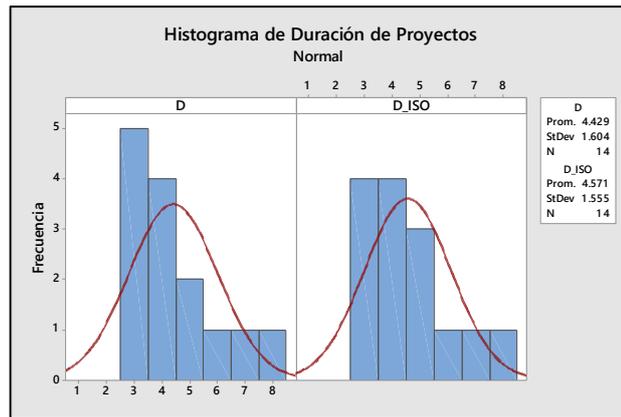


Figura 24 Histograma de frecuencias para la duración de los proyectos
Elaboración: el autor

La Figura 23 muestra que los grupos están distribuidos idénticamente en función a la duración.

En la Figura 24 se observa que la duración promedio del grupo de proyectos sin la aplicación del método para la evaluación de calidad es de 4.4 semanas, mientras que la duración promedio del grupo de proyectos con la aplicación del método para la evaluación de calidad es de 4.5 semanas. Adicionalmente, presentan dispersiones y formas similares; alineado a esto, la Figura 25 muestra las duraciones de los dos grupos de forma superpuesta, y se puede apreciar que los dos grupos tienen una distribución similar.

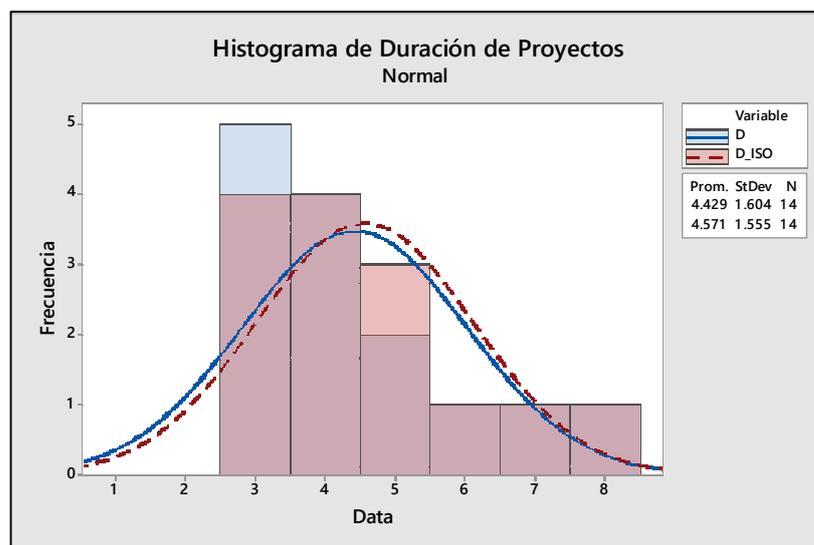


Figura 25 Frecuencia de la duración de los proyectos
Elaboración: el autor

Las Figuras 26 y 27 muestran las observaciones en los dos grupos:

- O: Observaciones en proyectos sin el método de evaluación de calidad (Grupo de control).
- O_ISO: Observaciones en proyectos sobre los que se aplicó el método para la evaluación de calidad de software basado en ISO/IEC 25000.

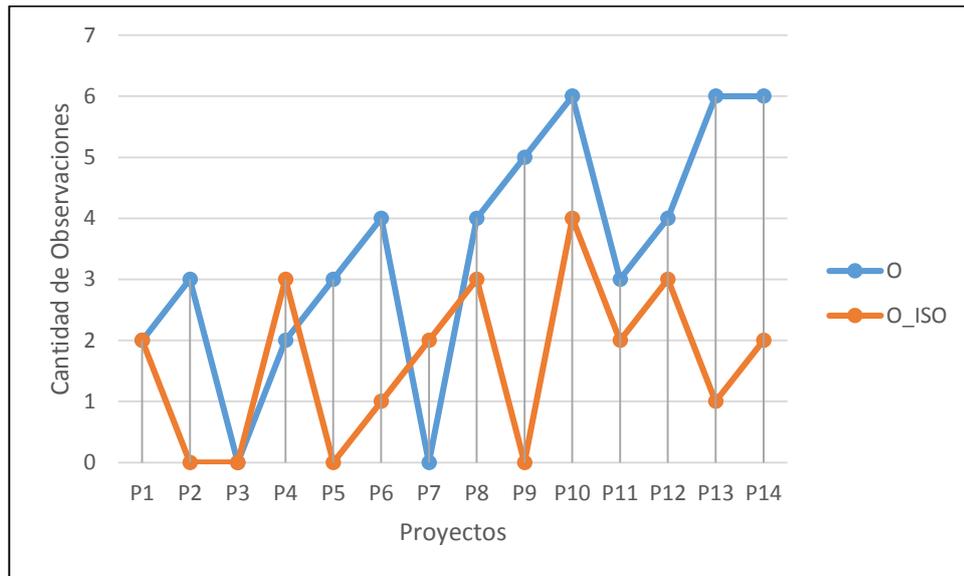


Figura 26 Gráfico descriptivo de la cantidad de observaciones en los proyectos
Elaboración: el autor

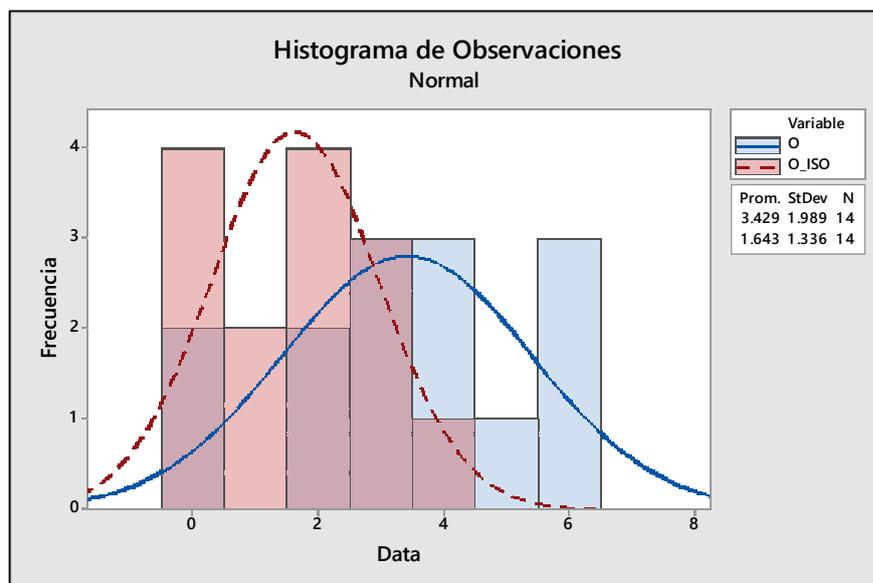


Figura 27 Distribución de la cantidad de observaciones en los dos grupos de proyectos
Elaboración: el autor

En la Figura 26 se puede apreciar una disminución en la cantidad de observaciones; alineado a este resultado, la Figura 27 muestra un desplazamiento de la curva de observaciones hacia la izquierda con la aplicación del método para la evaluación de calidad basado en ISO/IEC 25000, dando los primeros indicios de que la aplicación del método mejora la calidad del software.

Las Figuras 28 y 29 muestran los errores en los dos grupos:

- E: Errores en proyectos sin el método de evaluación de calidad (Grupo de control).
- E_ISO: Errores en proyectos sobre los que se aplicó el método para la evaluación de calidad de software basado en ISO/IEC 25000.

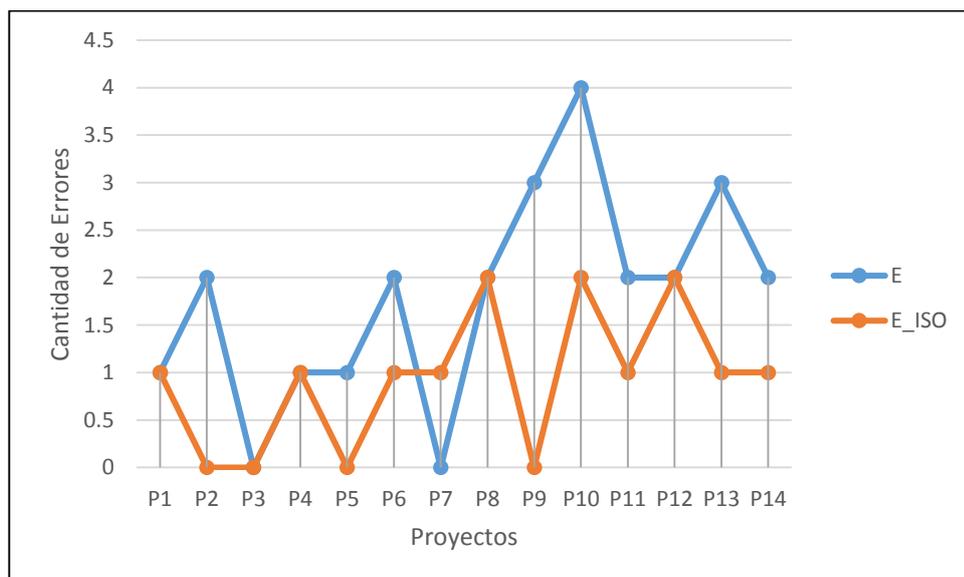


Figura 28 Gráfico descriptivo de la cantidad de errores en los proyectos
Elaboración: el autor

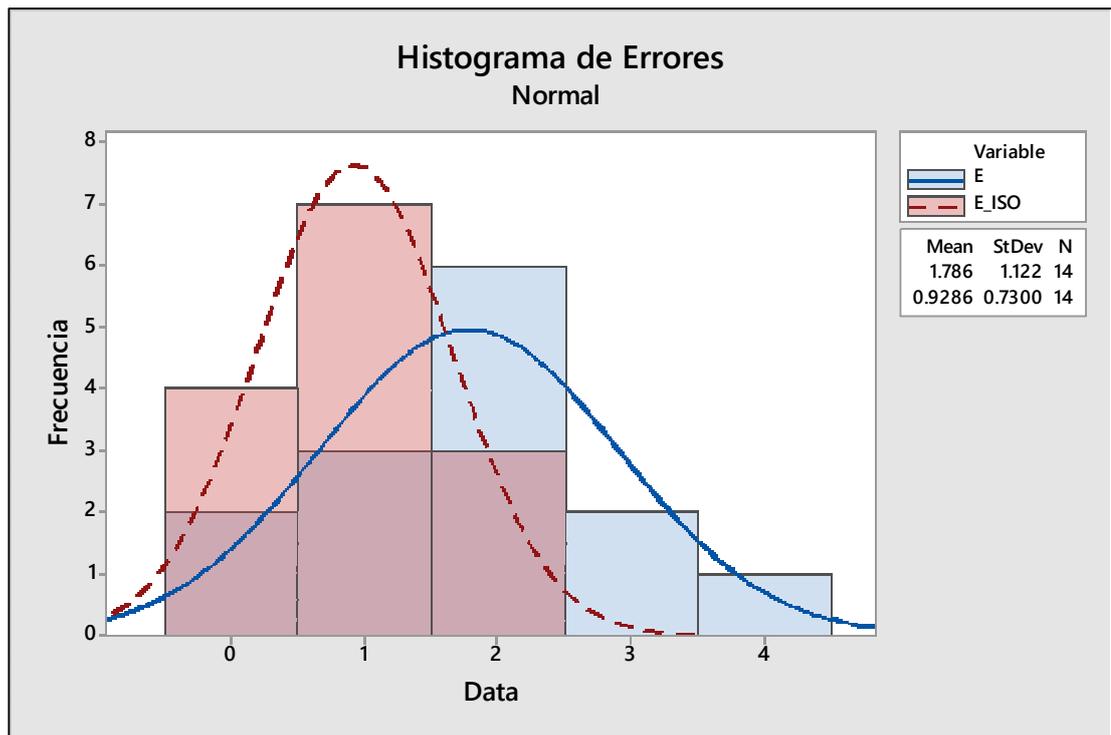


Figura 29 Distribución de la cantidad de errores en los dos grupos de proyectos
Elaboración: el autor

En la Figura 28 se puede apreciar una disminución en la cantidad de errores, alineado a este resultado, la Figura 29 muestra un desplazamiento de la curva de errores hacia la izquierda con la aplicación del método para la evaluación de calidad basado en ISO/IEC 25000, dando los primeros indicios de que la aplicación del método disminuye los errores del software después de su puesta en producción.

Las Figuras 30 y 31 muestran los reprocesos en los dos grupos:

- R: Reprocesos en proyectos sin el método de evaluación de calidad (Grupo de control).
- R_ISO: Reprocesos en proyectos sobre los que se aplicó el método para la evaluación de calidad de software basado en ISO/IEC 25000.

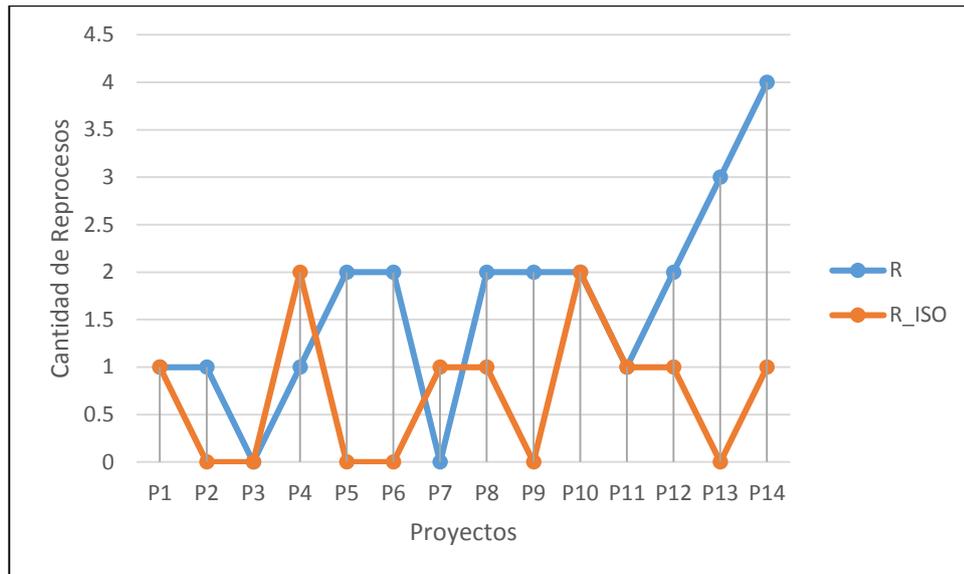


Figura 30 Gráfico descriptivo de la cantidad de reprocesos en los proyectos
Elaboración: el autor

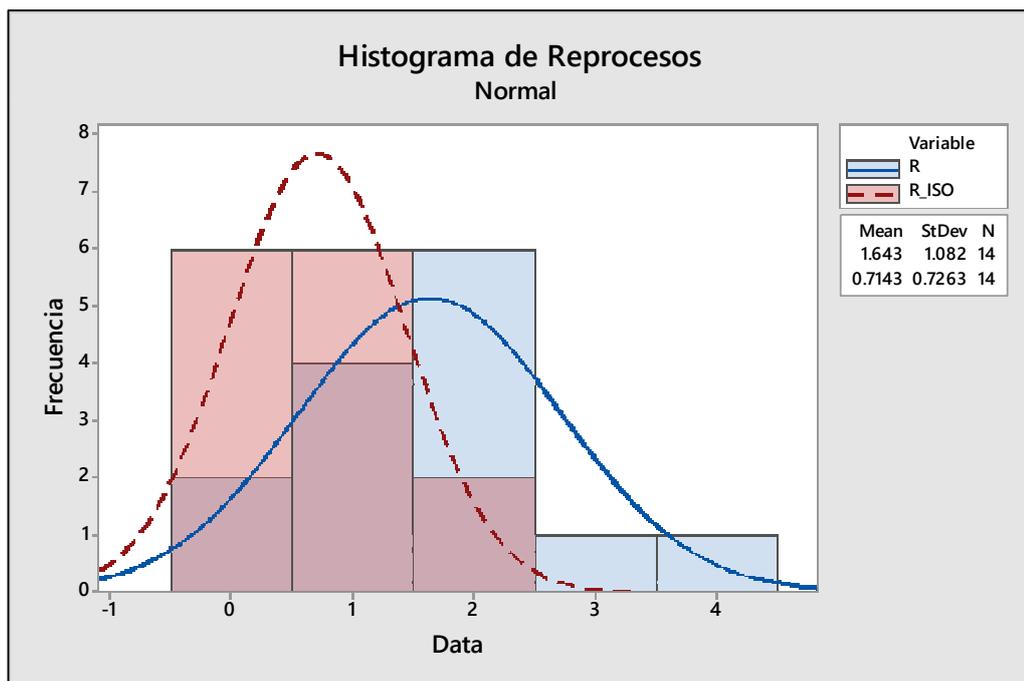


Figura 31 Distribución de la cantidad de reprocesos en los dos grupos de proyectos
Elaboración: el autor

En la Figura 30 se puede apreciar una disminución en la cantidad de reprocesos, alineado a este resultado, la Figura 31 muestra un

desplazamiento de la curva de reprocesos hacia la izquierda con la aplicación del método para la evaluación de calidad basado en ISO/IEC 25000, dando los primeros indicios de que la aplicación del método facilita la conformidad del software por parte del usuario.

5.2 Evaluación de la Normalidad de Datos

5.2.1 Normalidad de las observaciones sin el método de evaluación de calidad

En la Figura 32, el p-valor asociado a la prueba estadística es 0.37 y resulta mayor al nivel de significancia 0.05 (5%), por lo tanto las observaciones ($O=R+E$) se ajustan a una distribución normal.

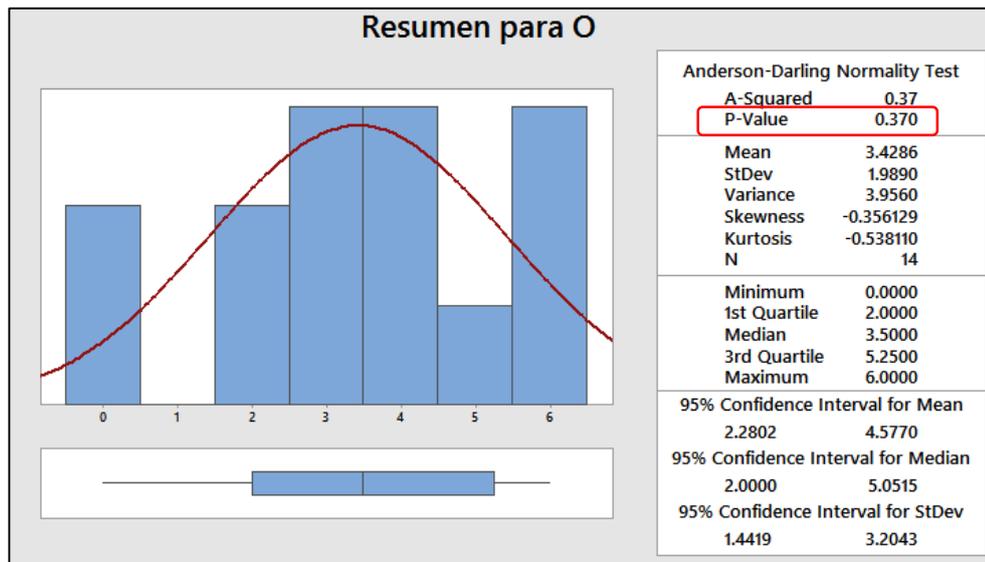


Figura 32 Número de observaciones con la metodología normal de desarrollo, sin el método de evaluación de calidad
Elaboración: el autor

5.2.2 Normalidad de las observaciones con la aplicación del método de evaluación de calidad basado en ISO/IEC 25000

En la Figura 33, el p-valor asociado a la prueba estadística es 0.119 y resulta mayor al nivel de significancia 0.05 (5%), por lo tanto las observaciones ($O_{ISO}=R_{ISO}+E_{ISO}$) se ajustan a una distribución normal.

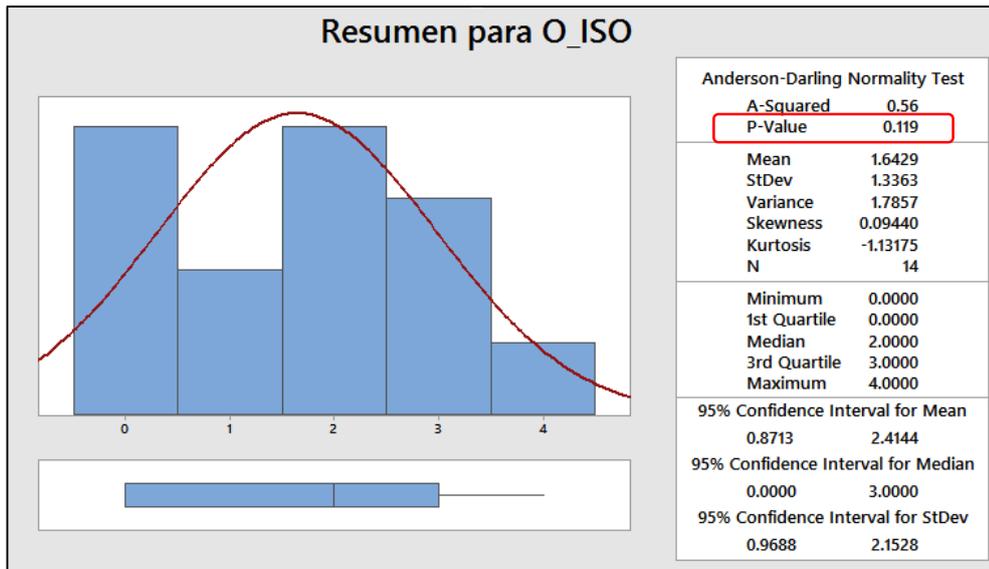


Figura 33 Número de observaciones con la aplicación del método de evaluación de calidad basado en ISO/IEC 25000

Elaboración: el autor

5.2.3 Normalidad de los errores en producción sin el método de evaluación de calidad

En la Figura 34, como el p-valor asociado a la prueba estadística es 0.118, y resulta mayor al nivel de significancia 0.05 (5%), por lo tanto los errores (E) sin el método de evaluación de calidad se ajustan a una distribución normal.

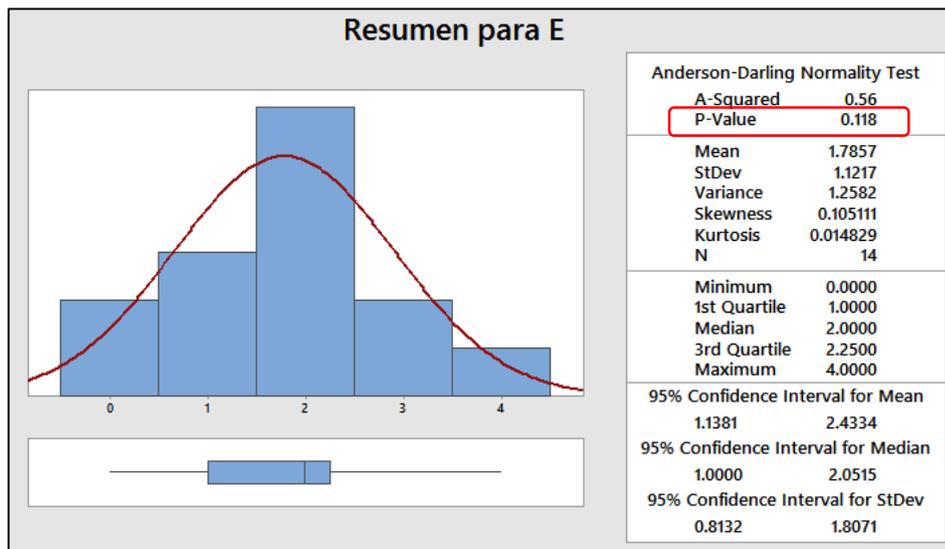


Figura 34 Número de errores sin el método de evaluación de calidad

Elaboración: el autor

5.2.4 Normalidad de los errores con la aplicación del método para la evaluación de calidad basado en ISO/IEC 25000

En la Figura 35, como el p-valor asociado a la prueba estadística resulta menor al nivel de significancia 0.05 (5%), los errores (E_ISO) con la aplicación del método para la evaluación de calidad basada en ISO/IEC 25000 no se ajustan a una distribución normal.

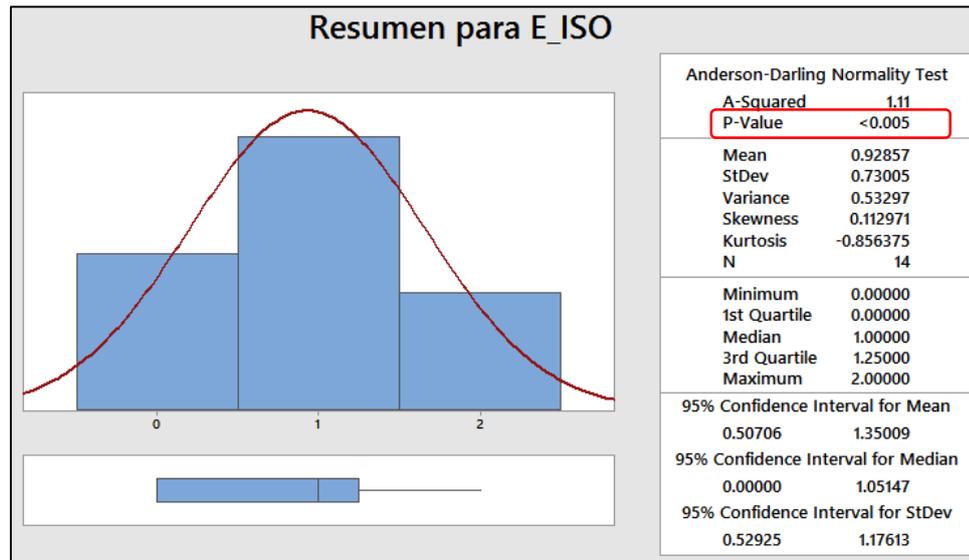


Figura 35 Número de errores con el método de evaluación de calidad
Elaboración: el autor

5.2.5 Normalidad de los reprocesos sin el método de evaluación de calidad

En la Figura 36, como el p-valor asociado a la prueba estadística es 0.068, y resulta mayor al nivel de significancia 0.05 (5%), por lo tanto el número de reprocesos (R) sin el método de calidad se ajustan a una distribución normal.

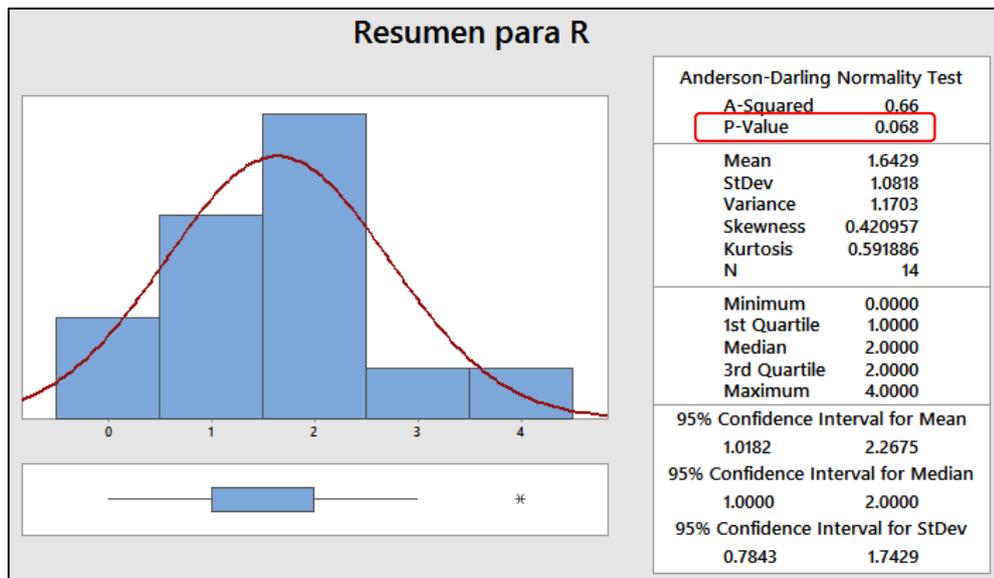


Figura 36 Número de reprocesos sin el método de evaluación de calidad
Elaboración: el autor

5.2.6 Normalidad de los reprocesos con la aplicación del método de evaluación de calidad basado en ISO/IEC 25000

En la Figura 37, el p-valor asociado a la prueba estadística es menor al nivel de significancia 0.05 (5%), por lo tanto el número de reprocesos (R_ISO) con la aplicación del método para la evaluación de calidad basado en ISO/IEC 25000 no se ajustan a una distribución normal.

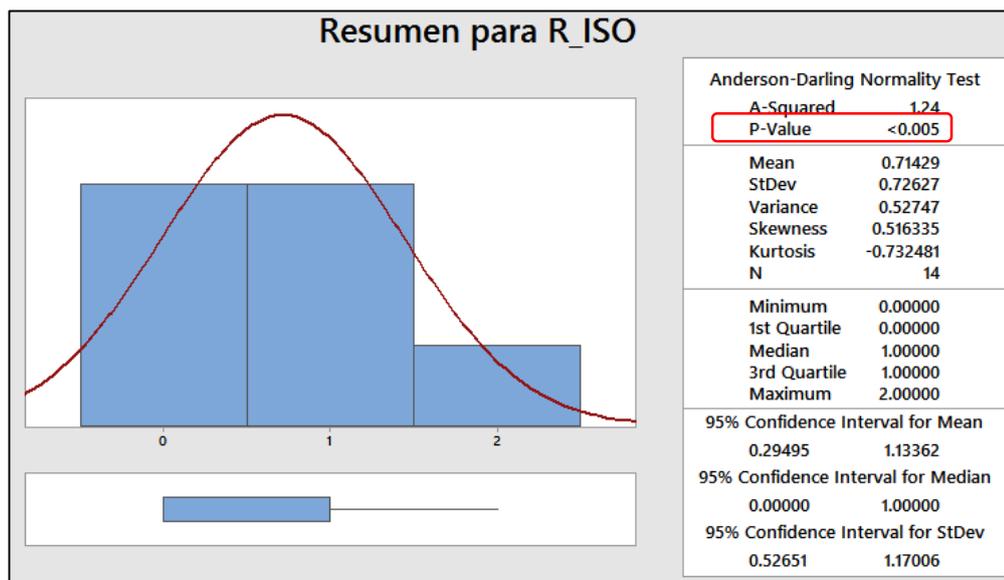


Figura 37 Número de reprocesos con el método de evaluación de calidad
Elaboración: el autor

5.3 Proceso de prueba de hipótesis

El proceso a seguir es el siguiente: (1) Se plantea la hipótesis nula y alterna, (2) se define la regla de decisión, (3) se ejecuta la prueba estadística y finalmente (3) se interpretan los resultados.

5.3.1 Prueba de la Hipótesis General

Ho: Contar con un método para la evaluación de calidad basado en ISO/IEC 25000 no mejora la calidad del software.

Ha: Contar con un método para la evaluación de calidad basado en ISO/IEC 25000 mejora la calidad del software.

5.3.1.1 Regla de decisión

Si el p-valor resultante de la prueba es menor al nivel de significancia 0.05 se acepta la hipótesis alterna, en caso contrario se acepta la hipótesis nula.

5.3.1.2 Estadístico de prueba de hipótesis

Se plantea la prueba de t-Student para dos muestras ya que los grupos presentan distribución normal. En la Figura 38 se muestran los resultados obteniendo un p-valor resultante de 0.005.

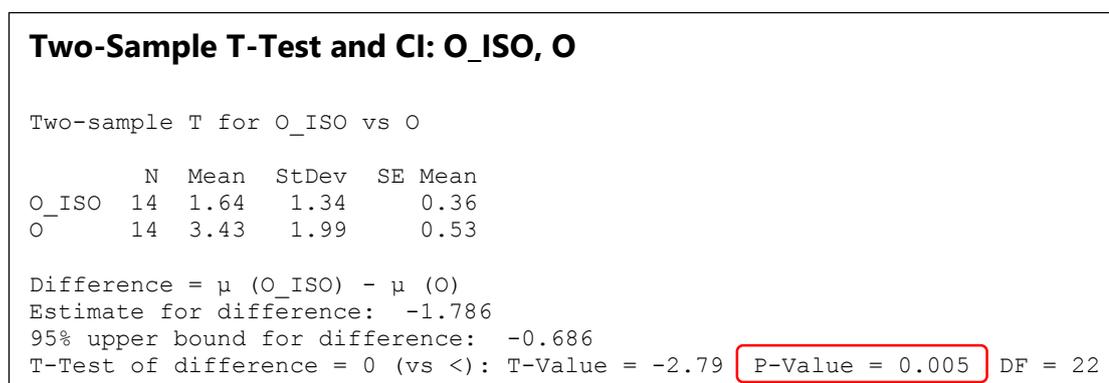


Figura 38 Resultados de la prueba t-Student para la hipótesis general
Elaboración: el autor

5.3.1.3 Interpretación

Como el p-valor obtenido 0.005 es menor al nivel de significancia 0.05 (5%), se acepta la hipótesis alterna y se rechaza la hipótesis nula; por lo tanto, se afirma con un 95% de confianza que la aplicación del método para la evaluación de calidad basado en ISO/IEC 25000 mejora la calidad del software. Asimismo, este resultado demuestra que al contar con un método que describe una secuencia clara de pasos a seguir y los entregables que se debe evaluar, se disminuye la cantidad de observaciones, consecuentemente esto se reflejará en una disminución de los sobrecostos por correcciones; finalmente esto representa una mejora en la calidad.

5.3.2 Prueba de la Hipótesis Específica 1

Ho: El uso de un método para la evaluación de calidad basado en ISO/IEC 25000 no disminuye los errores del software después de su puesta en producción.

Ha: El uso de un método para la evaluación de calidad basado en ISO/IEC 25000 disminuye los errores del software después de su puesta en producción.

5.3.2.1 Regla de decisión

Si el p-valor resultante de la prueba es menor al nivel de significancia 0.05 se acepta la hipótesis alterna, en caso contrario se acepta la hipótesis nula.

5.3.2.2 Estadístico de prueba de hipótesis

Se plantea la prueba de Mann-Whitney para datos no paramétricos. En la Figura 39 se muestran los resultados que muestran un p-valor resultante de 0.0193.

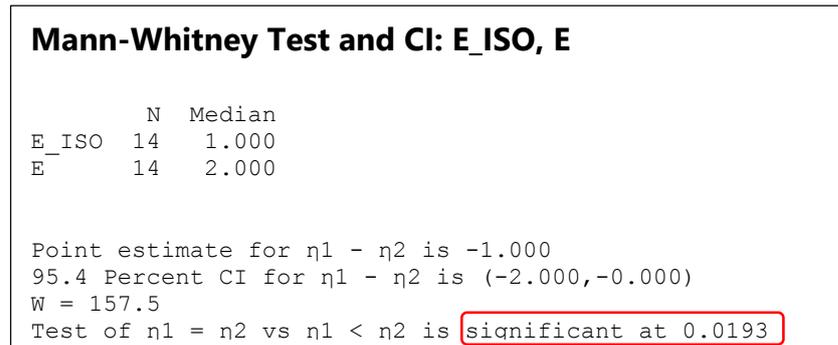


Figura 39 Resultados de la prueba Mann-Whitney para la hipótesis específica 1

Elaboración: el autor

5.3.2.3 Interpretación

Como el p-valor obtenido 0.0193 es menor al nivel de significancia 0.05 (5%), se acepta la hipótesis alterna y se rechaza la hipótesis nula; por lo tanto, se afirma con un 95% de confianza que el método para la evaluación de calidad basado en ISO/IEC 25000 disminuye los errores del software después de su puesta en producción. Asimismo, este resultado demuestra que al aplicar un método para la evaluación de calidad basado en ISO/IEC 25000, que evalúa los entregables durante el ciclo de vida del software, al hacer visible los problemas de calidad en etapas tempranas disminuye la ocurrencia de errores del software en producción.

5.3.3 Prueba de la Hipótesis Específica 2

Ho: El uso de un método para la evaluación de calidad basado en ISO/IEC 25000 no facilita la conformidad del software por parte del usuario.

Ha: El uso de un método para la evaluación de calidad basado en ISO/IEC 25000 facilita la conformidad del software por parte del usuario.

5.3.3.1 Regla de decisión

Si el p-valor resultante de la prueba es menor al nivel de significancia 0.05 se acepta la hipótesis alterna, en caso contrario se acepta la hipótesis nula.

5.3.3.2 Estadístico de prueba de hipótesis

Se plantea la prueba de Mann-Whitney para datos no paramétricos. En la Figura 40 se muestran los resultados que muestran un p-valor resultante de 0.0115.

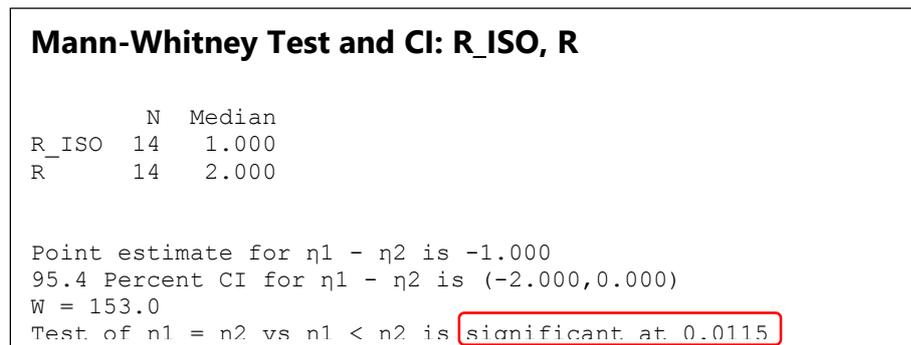


Figura 40 Resultados de la prueba Mann-Whitney para la hipótesis específica 2

Elaboración: el autor

5.3.3.3 Interpretación

Como el p-valor obtenido 0.0115 es menor al nivel de significancia 0.05 (5%), se acepta la hipótesis alterna y se rechaza la hipótesis nula; por lo tanto, se afirma con un 95% de confianza que el método para la evaluación de calidad basado en ISO/IEC 25000 facilita la conformidad del software por parte del usuario. Asimismo, este resultado demuestra que al aplicar un método para la evaluación de calidad basado en ISO/IEC 25000 que evalúa la especificación de requerimientos del software, asegura que la etapa de análisis abstraiga lo que el usuario realmente necesita, ello permite construir el software con los requisitos validados y se refleja finalmente en la disminución de los reprocesos en la etapa de pruebas, facilitando la conformidad del software por parte del usuario.

5.4 Resultados de la validación del método por expertos en calidad de software

Los resultados de la validación del método propuesto, por expertos en calidad de software, se encuentran en el Anexo 6. En resumen, en todas las secciones, el resultado promedio está por encima del valor 4 según la escala Likert de 5 niveles de respuesta mostrada en la Tabla 26.

Tabla 26 Escala de Likert de cinco niveles utilizada en el cuestionario

Respuesta	Valor
Totalmente en desacuerdo	1
En desacuerdo	2
Ni de acuerdo ni en desacuerdo	3
De acuerdo	4
Totalmente de acuerdo	5

Elaboración: el autor

Asimismo, dentro del cuestionario se incluyó una pregunta alineada al objetivo general: “Considera usted que la aplicación del método para la evaluación de calidad mejorará la calidad del producto software final”. Los resultados se muestran en la Figura 41, con valores 4 y 5 en las respuestas de los evaluadores, lo cual está alineado a los resultados estadísticos y fortalece la validez del método.

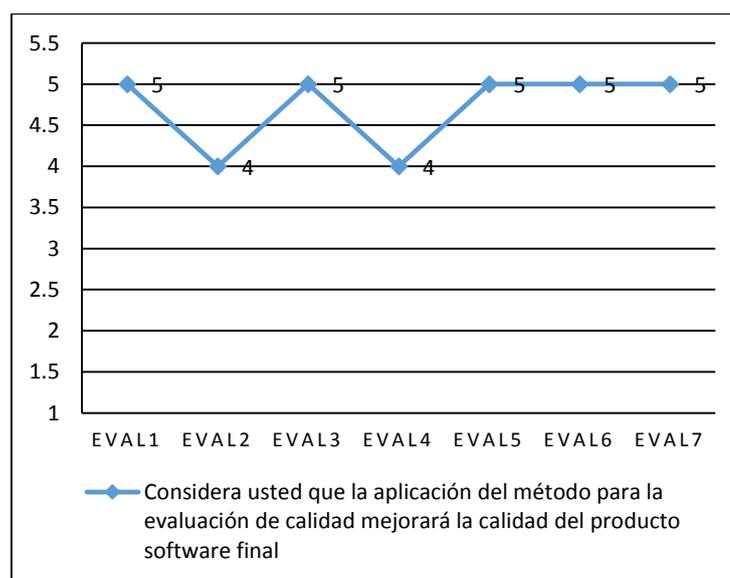


Figura 41 Resultados de la pregunta alineada al objetivo general
Elaboración: el autor

CAPÍTULO VI

DISCUSIÓN Y APLICACIONES

6.1 Discusión de los resultados

La aplicación del método de evaluación de calidad ha permitido mejorar la calidad del software.

Para la **elaboración del método** se ha recogido el aporte de Kusters et al. (2004) y Mellado et al. (2010), con los que se formula claramente los objetivos de la evaluación, y se realiza el balanceo entre objetivos y recursos al considerar para ello los niveles de importancia de las características y subcaracterísticas de calidad. También se han establecido actividades claras con una secuencia bien definida, y se logra finalmente: facilitar la conformidad del software por los usuarios, obtener coincidencias con lo mencionado por Kusters et al. (2004) y mejorar la calidad del software al aplicar la evaluación de calidad no solamente en la etapa de testing sino en otras etapas de su ciclo de vida, lo cual está alineado al aporte Mellado et al. (2010).

Asimismo, siguiendo las recomendaciones de Hosni y Kirinic (2013), se ha **construido un método** fácil de utilizar y se ha homologado las etapas en las que se encuentran los entregables a evaluar (comparación entre el modelo en cascada y los estándares ISO que gobiernan el ciclo de vida de los sistemas); todo ello ha contribuido a lograr el objetivo general. También, **el método propuesto, en concordancia** con Thapar et al. (2012) y Esaki (2013)

considera fundamental la evaluación de los entregables desde etapas tempranas del desarrollo del software, para asegurar que el producto final cubra las necesidades reales del usuario. Adicionalmente, el método utiliza métricas objetivas de calidad, y considera la participación de los stakeholders relevantes; esto se refleja directamente en la mejora conseguida en la conformidad del usuario final.

Por otro lado, se ha conseguido disminuir la cantidad de errores con la aplicación del método de evaluación de calidad, pues al recordar a Ahamed et al. (2012), estos errores miden directamente la calidad, y al disminuir estos en cantidad, la calidad del software mejora.

Durante la aplicación del método se validó, que un factor importante es el tiempo que se debe dedicar a la calidad, pues se recordará que cualquier actividad de calidad que se sacrifica por otras restricciones como funcionalidad o tiempo generan una deuda técnica que provocaran sobrecostos más tarde, tal como lo mencionan Seaman y Guo (2011), y Hwang (2014). La inversión de tiempo y recursos en actividades de calidad se reflejaron en la reducción de horas hombre en correcciones de errores y reprocesos en el ambiente de certificación. Adicionalmente, este estudio muestra evidencia de lo mencionado por Li y Fan (2012), que al dedicar recursos a actividades de calidad con una metodología de pruebas y estándares, se da una mejora en la calidad del producto.

El presente trabajo de investigación ha demostrado la utilidad de tomar como referencia los estándares (en nuestro caso la serie ISO/IEC 25000), al lograr el objetivo de mejorar la calidad del producto con las herramientas que proveen.

Un punto adicional es: que en la aplicación del método de calidad resulta fundamental romper las barreras descritas por Iyidogan (2014). En este sentido; si bien el método propuesto plantea tomar conciencia de los estándares de calidad y planificar los recursos necesarios para ejecutar actividades de calidad, si estas barreras no se superan no será posible lograr

mejoras en la calidad del software ni mejorar la reputación de un área de desarrollo y de la empresa, ya que la calidad del software y la reputación de un área de desarrollo tienen una relación directamente proporcional. Adicionalmente, se debe tener en cuenta que un catalizador para lograr esta mejora en la reputación es la reducción de errores en el software, lo cual se ha logrado.

Después de analizado y contrastado las hipótesis se realiza las discusiones:

- a) Los resultados estadísticos para la hipótesis general confirma la disminución de la cantidad de observaciones del software en el grupo de proyectos donde se aplica el método para la evaluación de calidad; esto evidencia mejoras en la calidad del software.
- b) Los resultados estadísticos confirman la hipótesis específica 1, y se verifica una disminución en los errores del software en los proyectos donde se aplica el método para la evaluación de calidad.
- c) Los resultados estadísticos confirman una reducción en la cantidad de reprocesos para la aceptación final por parte de los usuarios, lo cual indica que la aplicación del método logra facilitar la aceptación del software por parte de los usuarios, y se verifica también la hipótesis específica 2.

Para concluir, los resultados del cuestionario de validación del método dirigido a expertos en calidad de software, muestra que el 100% de encuestados considera que la aplicación del método de calidad mejorará la calidad del software, lo cual coincide y robustece el resultado de las pruebas estadísticas para la hipótesis general.

CONCLUSIONES

1. Se logró mejorar la calidad del producto software como resultado de la aplicación del método de evaluación de calidad de software basado en ISO/IEC 25000, esto se reflejó en una menor cantidad de reprocesos para que el usuario otorgue la conformidad del software y en una menor cantidad de errores luego del pase a producción del software. Asimismo, respecto a la mejora en la calidad del software, se halló un 95% de confianza en que el método para evaluación de calidad basado en ISO/IEC 25000 mejora la calidad del software.
2. La cantidad de errores relacionados a requisitos funcionales disminuyó luego de la aplicación del método de evaluación de calidad. Durante la ejecución de las actividades de evaluación de calidad de los entregables del ciclo de vida del desarrollo se hicieron visible problemas que se atendieron oportunamente, antes de la puesta en producción del software. De la misma forma, respecto a los errores del software, se halló un 95% de confianza en que el método para evaluación de calidad basado en ISO/IEC 25000 disminuye los errores del software después de su puesta en producción.
3. La aplicación del método para la evaluación de calidad permitió asegurar que el equipo de desarrollo plasme adecuadamente lo que el usuario necesita. Como consecuencia, la cantidad de reprocesos para la conformidad del usuario disminuyó luego de la ejecución de dicha

evaluación. De igual manera, respecto a la conformidad del software por parte del usuario, se halló un 95% de confianza en que el método para evaluación de calidad basado en ISO/IEC 25000 facilita la conformidad del software por parte del usuario.

RECOMENDACIONES

1. Realizar seguimiento a una mayor cantidad de proyectos, inclusive a aquellos con duración mayor a nueve semanas, y homologar los grupos del experimento con más variables, como la cantidad de líneas de código u otra variable que permita emparejar los grupos con mayor precisión, esto permitirá robustecer el trabajo de investigación.
2. Considerar recursos adicionales para las actividades de evaluación de calidad, ello permitirá contar con las personas, equipos y presupuesto para ejecutar las actividades con éxito, tener en cuenta que la cantidad de recursos necesarios dependerá de factores como la complejidad del proyecto y el nivel de detalle que se requiere para la evaluación de calidad; ya que por ejemplo, las evaluaciones de alto nivel no requieren mayor inversión.
3. Plantear trabajos futuros de los siguientes temas:
 - a) Evaluar la inclusión en el método de otros entregables como el plan de pruebas, los casos de prueba, el manual de usuario, el manual técnico entre otros. En el caso de los entregables: plan de pruebas y casos de prueba, si bien la etapa de testing es un tema bastante estudiado y desarrollado por las empresas, sería importante investigar si la inclusión de estos entregables en el método de la evaluación de calidad aporta una mejora en la calidad del producto software.

- b) Analizar los resultados de la evaluación de calidad mediante un estudio, para identificar los elementos que pueden manejarse como deuda técnica, este resultado puede ser un criterio para tener alternativas para gestionar las restricciones de un proyecto: tiempos, costos, calidad y funcionalidad.

FUENTES DE INFORMACIÓN

Ahamed, N., Sundaraj, K., Ahmad, R., Rahman, M., y Ali, A. (2012). A framework for the development of measurement and quality assurance in software-based medical rehabilitation systems. *Procedia Engineering*, 41, 53-60.

CHAOS (2013). *The Standish Group: Chaos Manifesto 2013*. The Standish Group International.

Esaki, K. (2013). System Quality Requirement and Evaluation: Importance of application of the ISO/IEC25000 series. *Global Perspectives on Engineering Management*, 2, 52-59.

Guo, Y., Seaman, C., Gomes, R., Cavalcanti, A., Tonin, G., Da Silva y Siebra, C. (2011). Tracking technical debt—An exploratory case study. In *Software Maintenance (ICSM), 27th IEEE International Conference on*, 528-531.

Hansen, K., Jonasson, K. y Neukirchen, H. (2011). An empirical study of software architectures' effect on product quality. *Journal of Systems and Software*, 84(7), 1233-1243.

Harter, D., Kemerer, C. y Slaughter, S. (2012). Does software process improvement reduce the severity of defects? A longitudinal field study. *Software Engineering, IEEE Transactions on*, 38(4), 810-827.

Hernandez, R. Fernandez y C., Baptista, P. (2010) *Metodología de la investigación*. McGraw-Hill.

Hosni, M. y Kirinic, V. (2013). Application of software product quality international standards through software development life cycle. *Central European Conference on Information and Intelligent Systems*, 284-296.

Hwang, S. (2014). Essential contents for software development process and software quality education. *International Journal of Engineering Systems Modelling and Simulation*, 6(1), 44-53.

ISO/IEC 14598-1 (1999). *International Standard, Information technology - Software product evaluation - Part 1: General overview*. International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 14598-5 (1998). *International Standard, Information technology - Software product evaluation – Part 5: Process for evaluators*. International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 14598-6 (1999). *International Standard, Information technology - Software product evaluation – Part 6: Documentation of evaluation modules*. International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 9126 (2001). *International Standard, Information technology – Software product evaluation – Quality characteristics and guidelines for their use*, International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 9126-1 (2001). *International Standard, Software engineering - Product quality - Part 1: Quality model*, International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 25000 (2005). *Systems and software engineering – Systems and software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE*, International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 25010 (2010). *Systems and software engineering – Systems and software product Quality Requirements and Evaluation (SQuaRE) – System and software quality models*, International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 25021 (2011). *Systems and software engineering – Systems and software product Quality Requirements and Evaluation (SQuaRE) – Quality measure elements*, International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 25023 (2013). *Systems and software engineering – Systems and software product Quality Requirements and Evaluation (SQuaRE) – Measurement of system and software product quality*, International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 25030 (2007). *Systems and software engineering - Systems and software product Quality Requirements and Evaluation (SQuaRE) - Quality requirements*, International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 25040 (2010). *Systems and software engineering - Systems and software product Quality Requirements and Evaluation (SQuaRE) –*

Evaluation Process, International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 25041 (2012). *Systems and software engineering - Systems and software product Quality Requirements and Evaluation (SQuaRE) – Evaluation guide for developers, acquirers and evaluators*, International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 15288 (2008). *Systems and software engineering – System life cycle processes*, International Organization for Standardization, Geneva, Switzerland.

ISO/IEC 12207 (2008). *Systems and software engineering – Software life cycle processes*, International Organization for Standardization, Geneva, Switzerland.

Iyidogan, S. (2014). Exploring the Diffusion of Software Quality Standards: Evidence from the Case of Turkey. *Procedia-Social and Behavioral Sciences*, 122, 362-366.

Kaur, R. y Sengupta, J. (2011). Software Process Models and Analysis on Failure of Software Development Projects. *International Journal of Scientific & Engineering Research*, 2.

Kitchenham, B., y Pfleeger, S. (1996). Software quality: The elusive target. *IEEE software*, (1), 12-21.

Kusters, R., Trienekens, J., Bemelmans, T. y Brombacher, A. (2004). The W-Process for Software Product Evaluation: A Method for Goal-Oriented Implementation of the ISO 14598 Standard. *Software Quality Journal*, 12, 137-158.

Kuwata, Y., Takeda, K., y Miura, H. (2014). A study on maturity model of open source software community to estimate the quality of products. *Procedia Computer Science*, 35, 1711-1717.

Li, Y. y Fan, W. (2012). Analysis of Software Testing System in Civil Aviation Field. *International Conference on Medical Physics and Biomedical Engineering 2012. Physics Procedia*, 33, 470 – 475.

OWASP (2013). *Los diez riesgos más críticos en aplicaciones web – OWASP Top 10*. The Open Web Application Security Project.

Marulanda, C., y Ceballos, J. (2012). Una revisión de metodologías seguras en cada fase del ciclo de vida del desarrollo de software. *Revista Ingenierías USBmed*, 3(1), 15-22.

Mellado, D., Rodriguez, M., Verdugo, J., Piattini, M. y Fernández-Medina, E. (2010). Evaluación De La Calidad Y Seguridad En Productos Software. *Tecnimap 2010*.

Nasir, M., y Sahibuddin, S. (2011). Critical success factors for software projects: A comparative study. *Scientific research and essays*, 6(10), 2174-2186.

Pasrija, V., Kumar, S., y Srivastava, P. R. (2012). Assessment of Software Quality: Choquet Integral Approach. *Procedia Technology*, 6, 153-162.

PUCP (2005). Resultado de Encuesta Sobre las Prácticas Actuales en Pruebas, Verificación y Validación en el Perú. *Facultad de Ciencias e Ingeniería Pontificia Universidad Católica del Perú*, Grupo de desarrollo e investigación en Ingeniería del Software.

Rawat, M. y Dubey, S. (2012). Software defect prediction models for quality improvement: a literature study. *International Journal of Computer Science*, 9, 288-296.

Rodriguez, M. y Piatinni M. (2012). Revisión Sistemática sobre la Certificación del Producto Software. *Computer Science and Engineering*, Julio 2012, 16-24.

Seaman, C. y Guo, Y. (2011). Measuring and monitoring technical debt. *Advances in Computers*, 82, 25.

Thapar, S., Singh, P., y Rani, S. (2012). Challenges to the Development of Standard Software Quality Model. *International Journal of Computer Applications*, 49(10), 1-7.

Wilson, G. (2013). An empirical study on software quality: developer perception of quality, metrics, and visualizations (tesis de maestría). Universidad de Austin, Texas, EEUU.

Wilkerson, J., Nunamaker Jr, J., y Mercer, R. (2012). Comparing the defect reduction benefits of code inspection and test-driven development. *Software Engineering, IEEE Transactions on*, 38(3), 547-560.

Zazworka, N., Shaw, M. A., Shull, F. y Seaman, C. (2011). Investigating the impact of design debt on software quality. *Proceedings of the 2nd Workshop on Managing Technical Debt, ACM*, 17-23.

ANEXOS

	Página
Anexo 1. Listado de proyectos	113
Anexo 2. Bitácora de pases al ambiente de aceptación por el usuario	114
Anexo 3. Reporte de errores identificados en el ambiente de producción	115
Anexo 4. Solicitud de validación dirigida a expertos en calidad de software	116
Anexo 5. Cuestionario de validación del método para la evaluación de calidad de software basado en ISO/IEC 25000	117
Anexo 6. Resultados del cuestionario de validación del método por expertos en calidad de software	120

Anexo 1. Listado de proyectos

Tabla 27 Muestra de proyectos de desarrollo de software

Grupo	Id	Proyecto	D (1)	R (2)	E (3)	O (4)
1	1	Comisiones vendedores	6	2	2	4
1	2	Informes entidades supervisoras	4	2	2	4
1	3	Grupos económicos	5	2	4	6
1	4	Reestructuración de créditos vencidos	3	1	1	2
1	5	Fechas de pago fijos según perfil del cliente	4	0	0	0
1	6	Aplicaciones parciales para créditos en estado de recuperación	5	1	2	3
1	7	Módulo de gestión de documentos registrados en registros públicos	3	1	2	3
1	8	Gestión de excepciones de características del crédito	4	2	2	4
1	9	Desembolsos mejoras	4	2	3	5
1	10	Implementación de controles en las aprobaciones	3	0	0	0
1	11	Campañas especiales	7	3	3	6
1	12	Campaña 2014	8	4	2	6
1	13	Actualización de los sistemas de cobranza	3	1	1	2
1	14	Generación automática de documentación	3	2	1	3
2	1	Preaprobaciones de créditos	5	0	0	0
2	2	Producto LSM	3	1	1	2
2	3	Producto LS	8	1	1	2
2	4	Integración con bancos	4	0	0	0
2	5	Intranet	4	0	1	1
2	6	Gestión de revisiones en el file del cliente	5	2	2	4
2	7	Aplicaciones anticipadas	3	0	0	0
2	8	Contabilización de ingresos generados en los créditos	4	1	1	2
2	9	Gestión de llamadas	4	1	2	3
2	10	Cambio en las condiciones del crédito	3	0	0	0
2	11	Control de pagos	3	2	1	3
2	12	Aplicaciones por reconocimiento de pérdida	5	1	1	2
2	13	Gestión de incidencias	7	0	1	1
2	14	Cambio de fecha de las operaciones vigentes	6	1	2	3

Elaboración: el autor

(1) D: Duración del desarrollo (semanas)

(2) R: Cantidad de Reprocesos para la aceptación del software por parte de los usuarios.

(3) E: Cantidad de Errores del software relacionados al cumplimiento de los requisitos funcionales en el ambiente de producción.

(4) O: Cantidad de Observaciones.

Anexo 2. Bitácora de pases al ambiente de aceptación por el usuario

N°	Fecha	Hora	ID Requerimiento	Descripción
1222	18/09/2014	16:00	2813	
1223	22/09/2014	12:30	2980	
1224	22/09/2014	15:30	3514	
1225	23/09/2014	12:00	3438	
1226	25/09/2014	08:45	3475	
1227	25/09/2014	11:00	3249	
1228	25/09/2014	17:30	2862	
1229	25/09/2014	18:45	2212	
1230	26/09/2014	11:00	3489	
1231	26/09/2014	16:30	3444	
1232	29/09/2014	12:30	3253	

Figura 42 Evidencia de la bitácora de pases al ambiente de aceptación por el usuario (UAT)

Elaboración: el autor

Anexo 3. Reporte de errores identificados en el ambiente de producción

ID	Fecha Error	Error Funcionalidad (Si/No)	Detalle del Error	Diagnóstico	Fecha Solución	Solución
E1	29/09/2012	Si	El sistema no permite generar un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación, lo que impide la generación de los datos de la aplicación en el momento de la generación de los datos de la aplicación.	Se generó un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	29/09/2012	Se actualizó la aplicación.
E2	31/10/2012	Si	El sistema no permite generar un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	Se generó un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	31/10/2012	Se actualizó la aplicación.
E3	11/12/2012	No	El sistema no permite generar un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	Se generó un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	13/12/2012	Se actualizó la capa de datos de la aplicación.
E4	08/02/2013	Si	El sistema no permite generar un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	Se generó un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	08/02/2013	Se actualizó la aplicación.
E5	04/03/2013	Si	El sistema no permite generar un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	Se generó un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	04/03/2013	Se aplicó script de roolback y ejecutó script corregido.
E6	22/04/2013	Si	El sistema no permite generar un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	Se generó un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	23/04/2013	Se incluyó la instrucción para manejar transacciones y se paso a pruebas.
E7	17/06/2013	Si	El sistema no permite generar un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	Se generó un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	18/06/2013	Se aplicó validación adicional y se actualizó aplicación.
E8	26/07/2013	Si	El sistema no permite generar un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	Se generó un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	26/07/2013	Se realizó el cambio solo en el reporte.
E9	14/08/2013	Si	El sistema no permite generar un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	Se generó un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	14/08/2013	Se actualizó la clase Sdesembolso.
E10	30/09/2013	No	El sistema no permite generar un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	Se generó un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	21/11/2013	Se realizó la modificación.
E11	23/01/2014	Si	El sistema no permite generar un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	Se generó un reporte de la información AS/400 en el momento de la generación de los datos de la aplicación, ya que se requiere un tiempo de ejecución de 1 hora y 30 minutos para la generación de los datos de la aplicación.	23/01/2013	Se realizó la modificación.

Figura 43 Evidencia del reporte de errores encontrados en el ambiente de producción

Elaboración: el autor

Anexo 4. Solicitud de validación dirigida a expertos en calidad de software

SOLICITUD DE VALIDACION DIRIGIDA A EXPERTOS EN CALIDAD DE SOFTWARE

Lima, .. de de 2015

Estimado Señor:

Presente

Reciba un cordial saludo

Motivado por su reconocida formación en materia de Calidad de Software, me complace dirigirme a usted en solicitud de su valiosa colaboración para la validación del "Método para la evaluación de calidad de software basado en ISO/IEC 25000" que adjunto a la presente; este método forma parte del trabajo de investigación que estoy realizando, y su utilidad radica en evaluar la calidad de los entregables del software desde la etapa de análisis. Los resultados de la evaluación de calidad de cada entregable permitirán tomar acciones correctivas desde etapas iniciales en un proyecto de software, y no esperar hasta la etapa de pruebas. El objetivo final es mejorar la calidad del software con la aplicación del método propuesto.

Agradeciendo anticipadamente su valiosa colaboración en el desarrollo e impulso de la investigación, me suscribo.

Muy Cordialmente,

|
Edú James Baldeón Villanes
DNI 41594705

Figura 44 Formato de la solicitud de validación dirigida a expertos en calidad de software
Elaboración: el autor

Anexo 5. Cuestionario de validación del método para la evaluación de calidad de software basado en ISO/IEC 25000

CUESTIONARIO					
DIRIGIDO A EXPERTOS EN CALIDAD DE SOFTWARE					
INSTRUCCIONES					
Este cuestionario tiene el objetivo de validar el "Método para la evaluación de calidad de software basado en ISO/IEC 25000", cuyo detalle se adjunta al presente.					
Marque con una (X) dentro del recuadro que indica su nivel de conformidad sobre cada ítem.					
I. ENTREGABLES QUE EVALÚA EL MÉTODO					
En esta sección se presentan las preguntas sobre los entregables descritos en la Tabla 4.2 del método.					
Ítem	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
1. Existe claridad en la descripción de los entregables a evaluar					
2. Los objetivos de la evaluación de calidad propuestos para cada entregable, son relevantes en un proyecto de software					
3. Considera usted que la evaluación de calidad del entregable "Requisitos funcionales y no funcionales del software" mejorará la calidad del producto software final					
4. Considera usted que la evaluación de calidad del entregable "Diseño de la arquitectura del producto software" mejorará la calidad del producto software final					
5. Considera usted que la evaluación de calidad del entregable "Diseño de la base de datos" mejorará la calidad del producto software final					
6. Considera usted que la evaluación de calidad del entregable "Código fuente del producto software" mejorará la calidad del producto software final					
7. Considera usted que la evaluación de calidad del entregable "Producto software integrado y publicado (ambiente de pruebas)" mejorará la calidad del producto software final					
Observaciones de la sección I:					

II. ROLES QUE CONSIDERA EL MÉTODO					
En esta sección se presentan las preguntas sobre los roles descritos en la Tabla 4.3 del método.					
Ítem	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
8. Existe claridad en la descripción de los dos roles: (1) Solicitante y (2) Evaluador y sus responsabilidades					
9. Para asegurar una evaluación imparcial de cada entregable, es mandatorio que el "Evaluador" no haya formado parte del equipo que desarrolló en entregable a evaluar					
Observaciones de la sección II:					
III. FASE 1 DEL MÉTODO: ESTABLECER LOS REQUERIMIENTOS DE EVALUACIÓN					
En esta sección se presentan las preguntas sobre la fase 1 del método de evaluación de calidad propuesto.					
Ítem	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
10. Existe claridad en el contenido de la fase 1 del método					
11. Es relevante el uso del modelo de calidad propuesto por el estándar ISO25010, para identificar las características y subcaracterísticas de calidad a evaluar en cada entregable					
12. Es pertinente incluir el nivel de importancia de las características de calidad, para enfocar los recursos en las características más importantes.					
Observaciones de la sección III:					

IV. FASE 2 DEL MÉTODO: ESPECIFICACIÓN DE LA EVALUACIÓN

En esta sección se presentan las preguntas sobre la fase 2 del método de evaluación de calidad propuesto.

Ítem	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
13. Existe claridad en el contenido de la fase 2 del método					
14. Es relevante el uso de las medidas (métricas) de calidad definidos en la serie ISO25000.					
15. Es importante detallar las medidas de calidad y los métodos de evaluación que se utilizarán en la evaluación, antes de ser ejecutada.					
16. Es importante definir los criterios de decisión antes de ejecutar la evaluación de calidad.					

Observaciones de la sección IV:

V. FASE 3 DEL MÉTODO: DISEÑO DE LA EVALUACIÓN

En esta sección se presentan las preguntas sobre la fase 3 del método de evaluación de calidad propuesto.

Ítem	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
17. Existe claridad en el contenido de la fase 3 del método					
18. Es relevante planificar las tareas de la evaluación de calidad antes de su ejecución					
19. Es importante optimizar el plan de trabajo asegurándose de no tener tareas duplicadas					

Observaciones de la sección V:

VI. FASE 4 DEL MÉTODO: EJECUCIÓN DE LA EVALUACIÓN

En esta sección se presentan las preguntas sobre la fase 4 del método de evaluación de calidad propuesto.

Ítem	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
20. Existe claridad en el contenido de la fase 4 del método					
21. Es relevante detallar los componentes evaluados y las herramientas utilizadas en la ejecución de la evaluación					
22. Es importante contar con una estructura del borrador de informe de evaluación de calidad para su redacción					

Observaciones de la sección VI:

VII. FASE 5 DEL MÉTODO. CONCLUSIÓN DE LA EVALUACIÓN

En esta sección se presentan las preguntas sobre la fase 5 del método de evaluación de calidad propuesto.

Ítem	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
23. Existe claridad en el contenido de la fase 5 del método					
24. Es relevante revisar el borrador del informe de evaluación de calidad con el usuario antes de presentar el informe final.					
25. Es importante definir el destino de los datos y documentos generados en la evaluación de calidad, según las políticas de seguridad de información de la empresa					
26. Es relevante incluir los datos de los roles: (1) evaluador y (2) solicitante en el Reporte final de evaluación de calidad					

Observaciones de la sección VII:

VIII. EVALUACIÓN GENERAL					
En esta sección se presentan las preguntas generales sobre el método de evaluación de calidad propuesto.					
Ítem	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
27. Considera usted que la aplicación del método para la evaluación de calidad mejorará la calidad del producto software final					
28. Considera que el método es pertinente para su aplicación en los proyectos de software					
29. Considera relevante el método propuesto en el área de calidad de software					
Observaciones de la sección VIII:					
IX. DATOS DEL EVALUADOR					
Validado por:			Documento de Identidad:		
Profesión:					
Empresa o Institución donde labora:					
Cargo que desempeña:					
Tipo de empresa donde labora: <input type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/> Otros:					
Lugar y fecha de validación:					
Firma:					

Figura 45 Cuestionario de validación del método dirigido a expertos en calidad de software
Elaboración: el autor

Anexo 6. Resultados del cuestionario de validación del método por expertos en calidad de software

Tabla 28 Respuestas de los expertos en calidad de software

	Respuestas	Respuestas	Respuestas	Respuestas	Respuestas	Respuestas	Respuestas
Validado por:	Lizbeth Pino	Milka Pozo	Confidencial	Confidencial	Leyla Collantes	Erika Vega	Moises Rodriguez Monje
Profesión:	Ing. Sistemas	Ing. Informática	Ing. Sistemas	Ing. Sistemas	Ing. Computación y Sistemas	Ing. Computación y Sistemas	Ing. en Informática
Empresa o Institución donde labora:	Tata Consultancy Services	Tata Consultancy Services	Banco de la Nación	Banco de la Nación	Banco Interbank	Banco Continental	Alarcos Quality Center
Cargo que desempeña:	Analista Calidad	Analista Calidad	Analista Calidad	Analista Calidad	Analista de Certificación	Analista Calidad	Diector
Tipo de empresa donde labora: <input type="checkbox"/> Pública <input type="checkbox"/> Privada <input type="checkbox"/>	Privada	Privada	Pública	Pública	Privada	Privada	Privada
Lugar y fecha de validación:	05/06/2015	05/06/2015	05/06/2015	05/06/2015	12/06/015	15/06/2015	05/07/2015
I. ENTREGABLES QUE EVALÚA EL MÉTODO							
Ítem	Respuestas	Respuestas	Respuestas	Respuestas	Respuestas	Respuestas	Respuestas
1. Existe claridad en la descripción de los entregables a evaluar	4	4	4	4	4	5	4
2. Los objetivos de la evaluación de calidad propuestos para cada entregable, son relevantes en un proyecto de software	5	5	5	4	5	4	4
3. Considera usted que la evaluación de calidad del entregable “Requisitos funcionales y no funcionales del software” mejorará la calidad del producto software final	5	5	4	5	5	5	5

4. Considera usted que la evaluación de calidad del entregable “Diseño de la arquitectura del producto software” mejorará la calidad del producto software final	5	4	4	4	4	5	4
5. Considera usted que la evaluación de calidad del entregable “Diseño de la base de datos” mejorará la calidad del producto software final	5	5	5	4	4	5	4
6. Considera usted que la evaluación de calidad del entregable “Código fuente del producto software” mejorará la calidad del producto software final	4	4	4	4	5	5	5
7. Considera usted que la evaluación de calidad del entregable “Producto software integrado y publicado (ambiente de pruebas)” mejorará la calidad del producto software final	5	4	4	4	5	5	5
II. ROLES QUE CONSIDERA EL MÉTODO							
Ítem	Respuestas						
8. Existe claridad en la descripción de los dos roles: (1) Solicitante y (2) Evaluador y sus responsabilidades	4	4	2	2	5	5	4
9. Para asegurar una evaluación imparcial de cada entregable, es mandatorio que el “Evaluador” no haya formado parte del equipo que desarrolló en entregable a evaluar	4	4	5	5	5	4	5

III. FASE 1 DEL MÉTODO: ESTABLECER LOS REQUERIMIENTOS DE EVALUACIÓN							
Ítem	Respuestas						
10. Existe claridad en el contenido de la fase 1 del método	4	4	4	4	4	4	5
11. Es relevante el uso del modelo de calidad propuesto por el estándar ISO25010, para identificar las características y subcaracterísticas de calidad a evaluar en cada entregable	4	4	4	4	5	5	5
12. Es pertinente incluir el nivel de importancia de las características de calidad, para enfocar los recursos en las características más importantes.)	5	4	5	4	5	5	4
IV. FASE 2 DEL MÉTODO: ESPECIFICACIÓN DE LA EVALUACIÓN							
Ítem	Respuestas						
13. Existe claridad en el contenido de la fase 2 del método	4	4	4	4	4	5	4
14. Es relevante el uso de las medidas (métricas) de calidad definidos en la serie ISO25000.	4	4	4	4	5	5	3
15. Es importante detallar las medidas de calidad y los métodos de evaluación que se utilizarán en la evaluación, antes de ser ejecutada.	5	4	5	4	4	5	5
16. Es importante definir los criterios de decisión antes de ejecutar la evaluación de calidad.	5	5	5	4	5	5	5

V. FASE 3 DEL MÉTODO: DISEÑO DE LA EVALUACIÓN							
Ítem	Respuestas						
17. Existe claridad en el contenido de la fase 3 del método	4	4	4	4	4	4	5
18. Es relevante planificar las tareas de la evaluación de calidad antes de su ejecución	5	4	4	5	5	5	4
19. Es importante optimizar el plan de trabajo asegurándose de no tener tareas duplicadas	5	5	4	4	4	4	5
VI. FASE 4 DEL MÉTODO: EJECUCIÓN DE LA EVALUACIÓN							
Ítem	Respuestas						
20. Existe claridad en el contenido de la fase 4 del método	4	4	5	4	4	5	4
21. Es relevante detallar los componentes evaluados y las herramientas utilizadas en la ejecución de la evaluación	5	5	4	4	4	5	5
22. Es importante contar con una estructura del borrador de informe de evaluación de calidad para su redacción	4	5	5	4	4	5	4

VII. FASE 5 DEL MÉTODO. CONCLUSIÓN DE LA EVALUACIÓN							
Ítem	Respuestas						
23. Existe claridad en el contenido de la fase 5 del método	4	4	4	4	4	5	5
24. Es relevante revisar el borrador del informe de evaluación de calidad con el usuario antes de presentar el informe final.	5	4	4	4	5	5	5
25. Es importante definir el destino de los datos y documentos generados en la evaluación de calidad, según las políticas de seguridad de información de la empresa	4	4	5	5	5	5	5
26. Es relevante incluir los datos de los roles: (1) evaluador y (2) solicitante en el Reporte final de evaluación de calidad	5	5	4	5	5	5	5
VIII. EVALUACIÓN GENERAL							
Ítem	Respuestas						
27. Considera usted que la aplicación del método para la evaluación de calidad mejorará la calidad del producto software final	5	4	5	4	5	5	5
28. Considera que el método es pertinente para su aplicación en los proyectos de software	4	4	4	4	5	5	4
29. Considera relevante el método propuesto en el área de calidad de software	4	5	5	4	5	5	4

Elaboración: el autor