

CALIDAD DE SISTEMAS Y PROCESOS DE SOFTWARE

AUTORA: HELLEN CUBERO LEDEZMA

NOVIEMBRE: 2020



San Marcos

Introducción

La calidad de software se determina si cumple con todos los requerimientos que se determinaron en su inicio, además de esto se debe validar su funcionalidad, rendimiento, confiabilidad, seguridad y la disponibilidad de mantenimiento y la posibilidad de que se pueda actualizar. (Sommerville, 2011).

Existen diferentes métodos y normas para garantizar la calidad en los sistemas de software en todas sus etapas y de esta manera garantizar altos estándares y la satisfacción del cliente, además de proveer mejor documentación, incremento de eficiencia y eficacia al agilizar el tiempo de desarrollo.



Contenido

Concepto y factores de Calidad del software.....	3
Calidad de producto: norma ISO 9126 y calidad de la información.....	4
Calidad del proceso de desarrollo del software: ISO 90003, CMMI e ISO 15504	6
Métricas de los productos y procesos de software.....	13
Proceso GQM.....	13
Gestión de la calidad	14
Aseguramiento.....	14
Verificación	15
Validación del software	15
Revisiones, inspecciones y auditorías	15
Conclusiones y recomendaciones	17
Referencias bibliográficas.....	18

Concepto y factores de Calidad del software

Se conoce como SQA (Software Quality Assurance) o GCS (Gestión de la Calidad del Soft-ware).

A continuación, se relacionan algunas definiciones de calidad, propuestas por diferentes organizaciones: (León, 2017)

Organización	Concepto
ISO	"El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas".
Real Academia Española	"Propiedad o conjunto de propiedades inherentes a una cosa, que permiten apreciarla como igual, mejor o peor que las restantes de su especie".
IEEE Institute of Electrical and Electronics Engineers	"Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas".

*Ilustración 1: Tabla de conceptos de calidad
Fuente: León, 2017*

Gestión de la calidad del software	Determina y aplica las políticas de calidad.
Aseguramiento de la calidad del software	Conjunto de actividades que de una manera planificada y sistemática, permiten evaluar el proceso de desarrollo del producto y aportan para que el producto cumpla los objetivos establecidos para la calidad.
Control de calidad del software	Técnicas y actividades operativas que se utilizan para cumplir los requisitos de calidad; manteniendo un proceso bajo control y eliminando las causas de defectos que se puedan presentar durante el ciclo de vida del desarrollo
Verificación	Comprobar si los productos finales por cada etapa del ciclo de vida cumplen con los requisitos que fueron establecidos para la etapa inmediatamente anterior
Validación	Si el producto de software satisface los requisitos que ha dispuesto el usuario o cliente

*Ilustración 2: Conceptos relacionados con calidad
Fuente: León, 2017*

Principios de la calidad del software:

- Se requiere que la dirección del proyecto esté involucrada.
- Es un proceso que se desarrolla durante todo el tiempo que dure el ciclo de vida del proyecto
- Es alcanzable siempre y cuando las personas involucradas

- contribuyan y trabajen conjuntamente para cumplir los objetivos
- Prevenir los defectos implica una inversión de recursos
 - En las primeras fases del proyecto se debe priorizar y poner especial atención en la detección y eliminación de defectos. (León, 2017)

Calidad de producto: norma ISO 9126 y calidad de la información

El modelo ISO/IEC 9126 se enfoca en la evaluación de los productos de software. Indica características en cuanto a calidad, así como los lineamientos que se deben seguir para su implementación.

El marco conceptual sobre el que se fundamenta, tienen en cuenta la calidad del proceso, calidad del producto de software y la calidad en el uso del producto de software.

El modelo ISO/IEC 9126 se cimienta sobre 7 indicadores: funcionalidad, confiabilidad, utilidad, eficiencia, capacidad de mantenimiento, portabilidad y calidad en uso. Cada uno de ellos está compuesto por una serie de criterios, que permiten definir el nivel de calidad de software. (León, 2017)

Indicador	Criterio	Descripción
Funcionalidad	Adecuación	Suministrar las funciones correctas que puedan cumplir las tareas y objetivos que ha especificado el usuario.
	Exactitud	Realizar proceso procesos y generar resultados en forma precisa o de acuerdo a lo esperado.
	Interoperabilidad	Interactividad con otros sistemas específicos.
	Seguridad	Protección de información y datos Niveles de acceso de acuerdo a roles y funciones
	Conformidad de la funcionalidad	Cumplimiento de estándares de funcionalidad.
Confiabilidad	Madurez	Sortear fallas al encontrar errores. Ej.; cuando no hay espacio suficiente,

		notificaciones al usuario acerca de operaciones indebidas.
	Tolerancia a errores	Seguir funcionando aun cuando se presenten errores.
	Recuperabilidad	Recuperarse y restaurar datos afectados después de una falla.
	Conformidad de la fiabilidad	Cumplir estándares o normas enfocadas a la fiabilidad.
Usabilidad	Entendimiento	Que el usuario pueda comprender fácilmente su uso y funcionalidades, teniendo como soporte la documentación del software.
	Aprendizaje	Como el usuario puede aprender a manejar el software, teniendo como soporte la documentación del mismo.
	Operabilidad	Como el software puede ser operado y controlado por el usuario.
	Atracción	Cualidades que hacen que el software sea atractivo y agradable para el usuario (diseño gráfico)
	Conformidad de uso	Cumplimiento de estándares de usabilidad.
Eficiencia	Comportamiento de tiempos	Los tiempos de respuesta, procesamiento y rendimiento deben ser adecuados.
	Utilización de recursos	Optimización de cantidad y tipos de recursos cuando el software funciona bajo los estándares y requerimientos sobre los cuales fue diseñado.
	Conformidad de eficiencia	Cumplimiento de estándares relacionados a la eficiencia.
Capacidad de mantenimiento	Capacidad de ser analizado	Que permita la realización de diagnósticos acerca de deficiencias, causas de fallas o identificar secciones modificadas.

	Cambiabilidad	Poder implementar una modificación, incluyendo el diseño, codificación y documentación de los cambios realizados.
	Estabilidad	Evitar los efectos inesperados
	Facilidad de prueba	Proteger los datos cuando se realizan pruebas a modificaciones
	Conformidad de facilidad de mantenimiento	Cumplimiento de estándares de facilidad de mantenimiento.
	Portabilidad	Cambiar de entorno.
Calidad de uso	Eficacia	Que los usuarios puedan realizar procesos de forma exacta e integral
	Productividad	Utilizar recursos adecuados de tal forma que no afecte la productividad del empleado.
	Seguridad	Que no ponga en peligro la integridad de las personas, instituciones, software, propiedad intelectual o entorno.
	Satisfacción	Grado de satisfacción del usuario frente a la interacción con el software.

Ilustración 3: Indicadores para definir el nivel de calidad
Fuente: León, 2017

Calidad del proceso de desarrollo del software: ISO 90003, CMMI e ISO 15504

ISO 90003

Es una norma derivada de la norma 9001 para promover la adopción de un enfoque basado en procesos al desarrollar, implementar y mejorar la efectividad de un sistema de gestión de la calidad, para mejorar la satisfacción del cliente al cumplir los requisitos. (ISO, 2018) Según asgconsultora (s.f), la aplicación de ISO/IEC 90003 es

apropiada en software que es parte de un contrato comercial con otra organización, es un producto disponible en el mercado, es usado como un proceso de apoyo de una organización, es parte de un hardware, o está relacionado a servicios de software.

Además, señala que algunos de los beneficios obtenidos son los siguientes:

- Agiliza el tiempo de desarrollo de un sistema
- Produce un cambio cultural positivo
- Incrementa la eficacia y productividad
- Se reducen las auditorías de calidad por parte de los clientes
- Fortalece la percepción de calidad y satisfacción del cliente

Modelo CMMI

CMMI (Capability Maturity Model Integration) o modelo de madurez y capacidad integrado.

Este modelo permite:

- Hacer una descripción de los componentes del modelo y las relaciones existentes entre los mismos.
- Realizar una comprensión general de las áreas que compone el proceso.
- Identificar y localizar información relevante en el modelo.
- Aplicar en forma práctica los conocimientos sobre el entorno de trabajo y en el equipo encargado de la evaluación de componentes y relaciones del modelo.

La siguiente imagen muestra los niveles durante el proceso evolutivo del proyecto, siguiendo el modelo CMMI. (León, s.f)



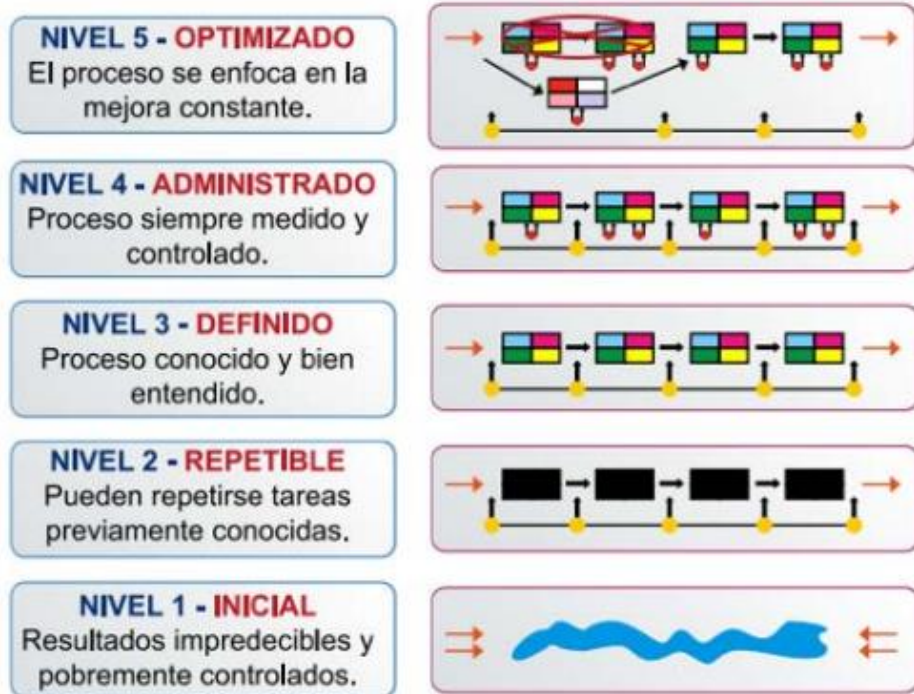


Ilustración 4: Niveles de proyecto siguiendo CMMI
Fuente: León, s.f

A continuación, se muestra la descripción de cada uno de los niveles:


Nivel	Descripción
Nivel 1	<ul style="list-style-type: none"> • Empresas sin procesos. • Presupuestos disparados. • Trabajos en tiempos extra para poder entregar el proyecto. • Entregas extemporáneas del proyecto. • Sin control acerca del estado del proyecto. • Futuro incierto del proyecto.
Nivel 2	<ul style="list-style-type: none"> • El proyecto ya es gestionado y controlado durante el proceso de desarrollo. • Se conoce del estado del proyecto en todo momento. • Se pueden repetir tareas ya realizadas. <p>Procesos a implantar para alcanzar este nivel:</p> 

Ilustración 5: Nivel 1 y 2
Fuente: León, s.f


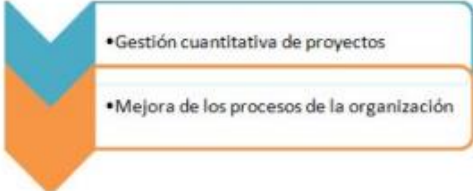
<p>Nivel 3</p>	<p>Se encuentra establecida y documentada la forma en que se deben desarrollar los proyectos. Existen métricas que permiten alcanzar los objetivos. Procesos a implantar para alcanzar este nivel:</p>  <ul style="list-style-type: none"> • Desarrollo de requisitos • Solución Técnica • Integración del producto • Verificación • Validación • Definición de los procesos de la organización • Desarrollo y mejora de procesos de la organización • Planificación de la formación • Gestión de riesgos • Análisis y resolución de toma de decisiones
<p>Nivel 4</p>	<p>Se usan objetivos medibles en los proyectos, enfocados a cumplir con las necesidades de cliente y la organización. Buen nivel en uso de métricas que permiten gestionar la organización. Procesos a implantar para alcanzar este nivel:</p>  <ul style="list-style-type: none"> • Gestión cuantitativa de proyectos • Mejora de los procesos de la organización

Ilustración 6: Nivel 3 y 4
Fuente: León, s.f

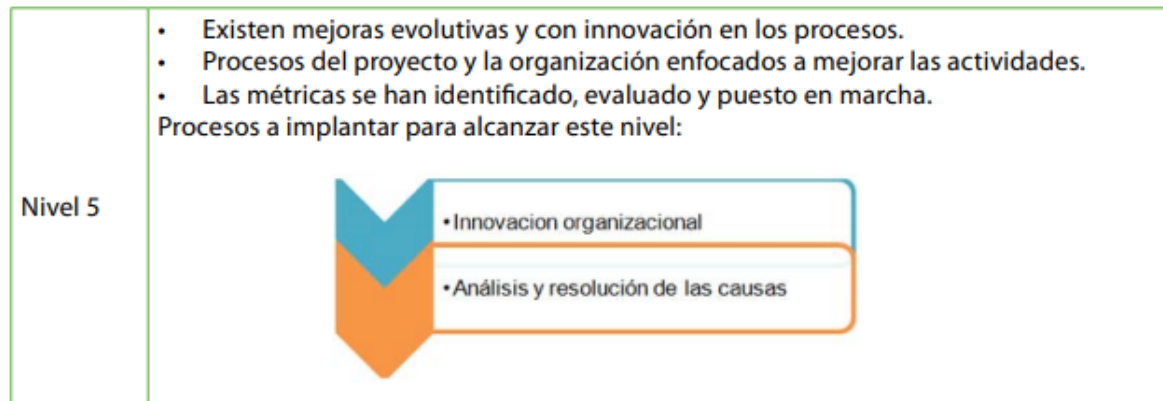


Ilustración 7: Nivel 5
Fuente: León, s.f

ISO 15504

- Se conoce como Software Process Improvement Capability Determinación (SPICE) o Determinación de la Capacidad de Mejora del Proceso de Software.
- Se enfoca en evaluar y mejorar la capacidad y calidad de los procesos de desarrollo y mantenimiento de sistemas de información y productos de software.
- Establece un marco de referencia y una serie de requisitos que deben ser implementados en los procesos de evaluación.
- Brinda información acerca de los requisitos necesarios para los modelos de evaluación de los procesos y de organizaciones.
- Brinda unas guías de referencia para definir las competencias que debe tener un evaluador de procesos.

Este modelo plantea 6 niveles de madurez:

- Nivel 0. Madurez 0.
- Nivel 1. Se realiza.
- Nivel 2. Se planifica.
- Nivel 3. Procesos definidos y desplegados.
- Nivel 4. Procesos medidos y analizados.
- Nivel 5. Mejora continua, innovación y experiencia.



A continuación, se muestra una infografía en la que se explican cada uno de los niveles de madurez existentes en el Modelo ISO/ IEC 15504. (León, s.f)



Ilustración 8: Niveles de madurez ISO 15504
Fuente: León, s.f

Métricas de los productos y procesos de software

Una métrica de software es una característica de un sistema de software, documentación de sistema o proceso de desarrollo que puede medirse de manera objetiva. Los ejemplos de métricas incluyen el tamaño de un producto en líneas de código; el índice Fog (Gunning, 1962), que es una medida de la legibilidad de un pasaje de texto escrito, el número de fallas reportadas en un producto de software entregado, y el número de días hombre requerido para desarrollar un componente de sistema. (Sommerville, 2011).

Métricas del producto

Son métricas de predicción usadas para medir los atributos internos de un sistema de software. Los ejemplos de las métricas de productos incluyen el tamaño del sistema, la medida en líneas de código o el número de métodos asociados con cada clase de objeto. (Sommerville, 2011)

Según Somerville (2011), las métricas del producto se dividen en dos clases:

- Métricas dinámicas: que se recopilan mediante mediciones hechas de un programa en ejecución, por ejemplo, el tiempo necesario para completar un cálculo.
- Métricas estáticas: se recopilan mediante mediciones hechas de representaciones del sistema, como el diseño, el programa o la documentación.

Para obtener más información, visite este [enlace](#)

Proceso GQM

Una dificultad fundamental en la medición del proceso es conocer qué información respecto al proceso debe recopilarse para apoyar la mejora de los procesos. Basili y Rombach (1988) proponen lo que llaman el paradigma GQM (Meta - Pregunta - Métrica, por las siglas de Goal-



Question-Metric), que se usa ampliamente en la medición software y procesos.

El paradigma GQM se usa en la mejora de procesos para ayudar a responder tres preguntas fundamentales:

1. ¿Por qué se introduce la mejora de procesos?
2. ¿Qué información se necesita para ayudar a identificar y valorar las mejoras?
3. ¿Qué mediciones de proceso y producto se requieren para obtener esta información?

Estas preguntas se relacionan directamente con las abstracciones (metas, preguntas, métricas) en el paradigma GQM:

1. **Metas:** una meta es un objetivo que la organización pretende lograr.
2. **Preguntas:** se trata de mejoras de las metas, en las que se identifican áreas específicas de incertidumbre relacionadas con las metas.
3. **Métricas:** se trata de mediciones que deben recopilarse para ayudar a responder las preguntas y confirmar si las mejoras del proceso lograron o no las metas deseadas.

La ventaja de usar el enfoque GQM en la mejora de procesos es que separa las preocupaciones de la organización (las metas) de preocupaciones específicas del proceso (las preguntas). (Sommerville, 2011, p.731).

Gestión de la calidad

La gestión de calidad del software implica los siguientes procesos:

Aseguramiento

El aseguramiento de la calidad (León, 2017), implica todas las actividades que requieren para que el producto de software pueda generar confianza y cumpla a cabalidad con los requerimientos planteados.

Dentro de estas actividades se pueden mencionar:

- Revisiones técnicas y de gestión, a fin de evaluar (su objetivo es la evaluación).

- Inspección, con el fin de verificar si se está construyendo el producto que ha solicitado el cliente.
- Pruebas y validación, para determinar si el software se está construyendo en forma correcta.
- Auditorías, para confirmar el cumplimiento de funciones.

Verificación

La finalidad de la verificación es comprobar que el software cumpla con su funcionalidad y con los requerimientos no funcionales establecidos. (Sommerville, 2011)

Validación del software

La meta de la validación es garantizar que el software cumpla con las expectativas del cliente. La validación es esencial, porque las especificaciones de requerimientos no siempre reflejan los deseos o necesidades reales de los clientes y usuarios del sistema. (Sommerville, 2011)

Revisiones, inspecciones y auditorías

Las revisiones e inspecciones son actividades QA que comprueban la calidad de los entregables del proyecto. Esto incluye examinar el software, su documentación y los registros del proceso para descubrir errores y omisiones, así como observar que se siguieron los estándares de calidad.

El propósito de las revisiones e inspecciones es mejorar la calidad del software, no de valorar el rendimiento de los miembros del equipo de desarrollo. (Sommerville, 2011).

El proceso de revisión

Según Sommerville (2011), aunque existen numerosas variaciones en los detalles de las revisiones, el proceso de revisión se estructura en tres fases:

- **Actividades previas a la revisión:** se trata de actividades



preparatorias esenciales para que sea efectiva la revisión. Por lo general estas actividades son la planeación y preparación de la revisión. La planeación consiste en establecer un equipo de revisión, organizar un tiempo, destinar un lugar para la revisión y distribuir los documentos a revisar.

- **La reunión de revisión:** durante la reunión de revisión un autor del documento o programa a revisar debe repasar el documento con el equipo de revisión. Un miembro del equipo dirige la revisión y otro registra formalmente todas las decisiones y acciones de revisión a tomar.
- **Actividades posteriores a la revisión:** después de finalizada la sesión, deben tratarse los conflictos y problemas surgidos durante la revisión. Esto puede implicar corregir bugs de software, refactorizar el software de modo que esté conforme con los estándares de calidad, o reescribir los documentos.

Inspecciones del programa

Somerville (2011), describe el proceso como “revisiones de pares” en las que los miembros del equipo colaboran para encontrar bugs en el programa en desarrollo. Las inspecciones pueden ser parte de los procesos de verificación y validación del software. Además, permiten identificar problemas con las pruebas, y así, mejorar la efectividad de dichas pruebas en la detección de bugs de programa.

Las inspecciones de programa incluyen a miembros del equipo con diferentes antecedentes que realizan una cuidadosa revisión, línea por línea, del código fuente del programa, buscan defectos y problemas, y los informan en una reunión de inspección.

Auditorías

Las auditorías establecen si el producto ha sido construido de acuerdo a los requerimientos y que el software esté realmente representado por la documentación que le acompaña. (Soto, s.f).

Conclusiones y recomendaciones

La gestión de calidad del software se ocupa de garantizar que el software tenga un número menor de defectos y que alcance los estándares requeridos de mantenibilidad, fiabilidad, portabilidad, etcétera. Incluye definir estándares para procesos y productos, y establecer procesos para comprobar que se siguieron dichos estándares. Además, el grupo de gestión de calidad debe ser independiente del equipo de desarrollo para que pueda tener una perspectiva objetiva del software, preferiblemente no debe asociarse con algún grupo de desarrollo particular; sin embargo, tiene la responsabilidad ante toda la organización por la administración de la calidad. A nivel jerárquico debe ser independiente y ubicarse sobre el nivel del administrador del proyecto. (Sommerville, 2011).

Por otra parte, los estándares establecidos para garantizar la calidad en el software buscan cumplir correctamente los requerimientos y la satisfacción del cliente.

A modo de recomendación, observe los enlaces indicados en algunas secciones para obtener más información relevante asociada al tema.



Referencias bibliográficas

Sommerville, I. (2011). Ingeniería de software (9a. ed.). Pearson Educación. <https://elibro.net/es/ereader/usanmarcos/37857?page=224>

Soto, E. (s.f). Aseguramiento de la calidad. Recuperado de <https://cutt.ly/7htnHRa>

León, F. (2017). Ingeniería de software I. Recuperado de <https://digitk.areandina.edu.co/bitstream/handle/areandina/1217/Ingenier%C3%ADa%20de%20software%201.pdf?sequence=1&isAllowed=y>

León, F. (s.f). Ingeniería de software II.

ISO. (s.f). ISO/IEC/IEEE 90003:2018. Recuperado de <https://www.iso.org/standard/74348.html>

asgconsultora. (s.f). Calidad en software ISO/IEC 90003. Recuperado de <http://www.asgconsultora.com/docs/asg-consultora-calidad-software-iso-90003.pdf>



www.usanmarcos.ac.cr

San José, Costa Rica