
Gestión de calidad en desarrollo de software

Management of quality in software development

Norberto Osorio Beltrán, Gloria Castro León

Universidad Nacional Mayor de San Marcos
Facultad de Ingeniería de Sistemas e Informática

nosorio30@hotmail.com; gcastrol@sistemas.edu.pe

RESUMEN

Durante los últimos años se ha mejorado sustancialmente la calidad de software, esto se debe en gran medida a que desarrollar un software no es lo mismo que la manufactura de un producto. Elaborar un producto de calidad implica que el producto desarrollado debe guardar similitud con su especificación. Sin embargo durante el proceso de desarrollo de software se presentan problemas entre la especificación de un software y su implementación, y en muchos casos no se sabe especificar características de calidad. Por otro lado, la ingeniería de requerimientos presenta limitaciones en la redacción de especificaciones concreta de software. Se presentan las métricas para medir el comportamiento del software: la competencia, calidad, desempeño y la complejidad del software.

Palabras clave: ingeniería de software, calidad de software

ABSTRACT

In recent years has substantially improved the quality of software, this is due in large measure to the use of techniques and methodologies in the software development process. Develop a software is not the same as the manufacture of a product. Develop a quality product means that the product developed must save similarity with the specification. However, during the software development process are presented problems between the specification of a software and its implementation, and in many cases it is not known specify features of quality. On the other hand the requirements engineering presents some limitations in the drafting of specifications specific software. Presented the metrics to measure the behavior of the software: the competence, quality, performance, and the complexity of the software.

Keywords: Software engineering, software quality

1. INTRODUCCIÓN

La calidad ha sido una preocupación desde hace mucho en los negocios, y debe serlo también para los ingenieros de software en el análisis y diseño de sistemas de información. El principal objetivo de los ingenieros del software es producir un sistema, aplicación o producto de alta calidad, para lo cual emplean métodos y herramientas efectivas dentro del contexto de un proceso maduro de desarrollo del software y además deben desarrollar mediciones que den como resultado sistemas de alta calidad. Para obtener esta evaluación, el ingeniero debe utilizar medidas técnicas, que evalúen la calidad con objetividad, no con subjetividad.

Es demasiado riesgoso realizar todo el proceso de análisis y diseño sin usar un enfoque de aseguramiento de la calidad. La planeación y el control son elementos esenciales de todo sistema exitoso. Se necesita un compromiso temprano de los ingenieros de software y de la empresa hacia la calidad para lograr el objetivo de la calidad.

2. CONTENIDO

2.1 estructura de la gestión de la calidad

La gestión de calidad de software se estructura en torno a tres actividades principales:

- Garantía de calidad
- Planificación de la calidad
- Control de la calidad

2.1.1 Garantía de calidad

La calidad es definida como el grado de relación que tiene el producto para satisfacer las necesidades del usuario. Un software que cumple con todos los requisitos con su usuario, y que sus procesos se ejecuten correctamente garantiza una buena Calidad¹

Para ello debe considerar tipos de estándar:

Estándares de producto. Estándares que se aplican sobre el producto software que se comienza a desarrollar. Incluye estándares de documentación, estándares de codificación.

Estándares de proceso. define los procesos que deben seguirse durante el desarrollo de software. Incluyen definiciones de procesos de especificación, diseño y validación, así como una descripción de los documentos que deben escribirse durante los procesos.

Importancia de los estándares de software

Los estándares de un producto guardan relación estrecha con los estándares de un proceso

Los estándares de producto se aplican a la salida del proceso de software, mientras que los estándares de proceso incluyen actividades de proceso que garantizan que se sigan los estándares de producto.

Los estándares de software:

- a) Se basan en el conocimiento de la mejor práctica de la empresa.
- b) Proveen el marco de trabajo alrededor del cual se implementa el proceso de garantía de la calidad.
- c) Permite que una persona continúe con facilidad el trabajo iniciado por otra. El utilizar buenas prácticas por todo el personal de la organización reduce el esfuerzo de aprendizaje cuando se inicia un nuevo proyecto.

2.1.2. Planificación de la calidad

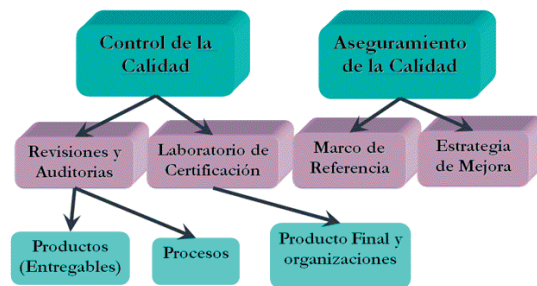


Fig. 1 planificación de la calidad.

La planificación de la calidad es el proceso en el cual se desarrolla un plan de calidad para un proyecto.

Define la calidad del software deseado y describe cómo debe valorarse

¹ Sommerville señala: La garantía de calidad es un proceso que define cómo lograr la calidad de software y cómo la organización de desarrollo conoce el nivel de calidad requerido de software. El uso de aseguramiento de calidad a lo largo del proceso es una forma de minimizar los riesgos y ayuda a asegurar que el sistema resultante es lo que se necesita y quiere, y que mejorará evidentemente algunos aspectos del desempeño del negocio. El proceso de garantía de la calidad (GC) se ocupa de definir o seleccionar los estándares que debe ser aplicados al proceso de desarrollo de software o al producto software.

Humphrey propone una estructura para un plan de calidad basado en los siguientes pasos:

Introducción del producto. Debe incluir la descripción del producto, el mercado al que se dirige y las expectativas de calidad

Planes de producto. Contiene las fechas y plazos de terminación de producto y las responsabilidades asignadas.

descripciones del proceso. Contiene los procesos de desarrollo y de servicio.

Metas de calidad. Contiene metas y planes de calidad para el producto, que deberán incluir la identificación de los atributos seleccionados como más relevantes.

Riesgos y gestión de riesgos. Contiene los riesgos clave que podrían afectar la calidad del producto.

Durante el proceso de planificación de la calidad debe considerarse los siguientes atributos:

Seguridad	Comprensión	Portabilidad	Protección	Experimentación
Usabilidad	Fiabilidad	Adaptabilidad	Reutilización	Flexibilidad
Modularidad	Eficacia	Robustez	Complejidad	Aprendizaje

Fig 2. Atributos de Calida

Generalmente, no es posible optimizar todos los atributos para un sistema, por tanto debe priorizarse los atributos más relevantes para un determinado producto a desarrollar.

2.1.3. Control de la calidad

El control de calidad considera la vigilancia del proceso de desarrollo de software para asegurar que se sigan los procedimientos y los estándares de garantía de calidad. El control de calidad incluye la comprobación de que las entregas cumplan los estándares definidos

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software. [1,2]

Los **requisitos del software** son la base de las medidas de calidad. La falta de concordancia con los requisitos es una falta de calidad.

Los estándares o **metodologías** definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se sigue ninguna metodología, siempre habrá falta de calidad.

Existen algunos **requisitos implícitos o expectativas** que a menudo no se mencionan, o se mencionan de forma incompleta (por ejemplo el deseo de un buen mantenimiento), que también pueden implicar una falta de calidad.

La política establecida debe estar sustentada sobre tres principios básicos: tecnológico, administrativo y ergonómico.

El **principio tecnológico** define las técnicas a utilizar en el proceso de desarrollo del *software*.

El **principio administrativo** contempla las funciones de planificación y control del desarrollo del *software*, así como la organización del ambiente o centro de ingeniería de *software*.

El **principio ergonómico** define la interfaz entre el usuario y el ambiente automatizado.

La adopción de una buena política contribuye en gran medida a lograr la calidad del software, pero no la asegura. Para el aseguramiento de la calidad es necesario su control o evaluación.

A partir del siguiente gráfico se observa la interrelación existente entre la gestión de la calidad, el aseguramiento de la calidad y el control de la calidad.

El control de calidad puede realizarse desde dos enfoques:

- **Revisión personal.** El proceso de revisión de calidad de software, documentación y los procesos están a cargo de un grupo de personas.
- **Revisión automática.** El proceso de revisión de calidad de software, documentación y los procesos es realizado por un programa utilizando para ello una medida cuantitativa de algunos atributos de software basados en métricas: inspecciones de diseño o programa, revisiones de progreso y revisiones de calidad.

El equipo de garantía de calidad debe crear un manual de estándares, el que debe incluir:

- 1 Estándares de producto
 - Formulario para revisión de diseño
 - Estructura del documento de requerimiento
 - Formato del encabezado de método
 - Formato de plan de proyecto
 - Estilo de programación en Java
 - Formulario de petición de cambios
- 2 Estándares de proceso
 - Conducto para la revisión de diseño
 - Sometimiento de documentos a CM
 - Proceso de entrega de versiones
 - Proceso de aprobación del plan de proyecto
 - Proceso de registro de pruebas

2.2. Estándares de documentación

Los estándares de documentación en un proyecto software son documentos muy importantes ya que son la única forma tangible de representar al software y su proceso. Los documentos deben ser fáciles de leer y de comprender².

Tipos de estándares de documentación:

- **estándares del proceso de documentación.** Define que proceso seguir para la producción del documento
- **estándares del documento.** Determinan la estructura y presentación de los documentos
- **estándares para el intercambio de documentos.** Permiten que todas las copias electrónicas de los documentos sean compatibles

2.3. Proceso de medición

Todo proceso de medición del software tiene como objetivo fundamental satisfacer necesidades de información a partir de las cuales se deben identificar las entidades y los atributos que deben ser medidos. El

proceso de medición de software consiste en derivar un valor numérico desde algún atributo del software o del proceso de software. Las mediciones se realizan para hacer predicciones generales acerca del sistema, o bien para identificar componentes anómalos. Para la medición se utilizan métricas, que son medidas relacionadas con un sistema, proceso o documentación de software. Las métricas pueden ser de control o de predicción[3].

2.3.1. Fases del proceso de medición

El proceso de medición se realiza a través de varias fases.

Elegir medidas a realizar. debe definirse claramente lo que se quiere medir. Para ello debe formularse las preguntas que la medición quiere responder y definir las mediciones requeridas para resolver las preguntas.

Seleccionar componentes a valorar. Se debe elegir un conjunto de componentes representativos, para la medición, no es necesario medir todos los componentes.

Medir características de los componentes. Se miden los componentes seleccionados y se calculan los valores de las métricas.

Identificar medidas anómalas. Luego de obtenidas las mediciones de los componentes, se comparan entre sí con las mediciones de una base de datos de mediciones.

Analizar componentes anómalos. Luego de identificados los componentes anómalos para las métricas particulares, se examinan estos componentes para decidir si los valores de las métricas indican que la calidad de los componentes está en peligro.

2.3.2. Clasificación de métricas

Con el fin de describir la conducta del software, se establecen las métricas que miden, entre otros aspectos, la competencia, calidad, desempeño y complejidad del software.

2 ISO 9000 constituye un conjunto de estándares internacionales que se pueden utilizar en el desarrollo de un sistema de gestión de calidad en todas las industrias. ISO 9001 define que estándares y procedimientos deben considerarse en una organización, los cuales deben documentarse en un manual de calidad organizacional. La norma ISO 9126 hacen referencia a la operación, transición y revisión del software y aunque no las dividen en estos tres grupos, señalan entre otras cosas la necesidad de lograr que el software opere correctamente y con el grado de exactitud requerido, que los usuarios sean capaces de entenderlo y usarlo, es decir que sea amigable con quienes interactúen con él, que sea capaz de responder correctamente ante fallos o cambios del entorno y que proporcione una ejecución o desempeño apropiado, teniendo en cuenta los recursos utilizados.

Presentamos a continuación la clasificación de las mismas:

- **Métricas de complejidad:** Definen la medición de la complejidad, tales como volumen, tamaño, anidaciones, costo (estimación) y configuración. Estas son los puntos críticos de la concepción, viabilidad, análisis y diseño de software.
- **Métricas de calidad:** Definen la medición de la calidad del software, tales como exactitud, estructuración o modularidad, pruebas, mantenimiento, reusabilidad, entre otras. Estos son los puntos críticos en el diseño, codificación, pruebas y mantenimiento.
- **Métricas de competencia:** Definen la valoración de las actividades de productividad de los programadores o practicantes con respecto a su certeza, rapidez, eficiencia y competencia.
- **Métricas de desempeño:** Definen la medición de la conducta de módulos y sistemas de un software, bajo la supervisión del sistema operativo o hardware. Generalmente tienen que ver con la eficiencia de ejecución, tiempo, almacenamiento, complejidad de algoritmos computacionales, etc.
- **Métricas estilizadas:** Definen los mecanismos para medir la experimentación y preferencia, por ejemplo, estilo de código, las convenciones denominando de datos, las limitaciones, etc. Estas no deben ser confundidas con las métricas de calidad o complejidad.

3. CONCLUSIONES

La industria de software viene desarrollándose en los últimos años y con ello la necesidad de producir software de calidad. Para lograrlo se tienen en cuenta numerosos factores entre los que se encuentran las métricas de software, una herramienta indiscutible para ayudar a mantener el control de los procesos y productos durante el desarrollo del software. La administración de la calidad es esencial a lo largo de todos los pasos de desarrollo de un sistema. Los elementos primarios de la gestión de calidad son significativos solamente cuando suceden en un contexto organizacional que de

soporte a un esfuerzo de calidad completo. Los planes de calidad pueden variar, en función del tamaño y el tipo del sistema a desarrollar. La redacción del plan de calidad debe ser lo más compacto posible, a fin de que los ingenieros lo lean y cumplan. Si son muy extensos y complicados, se corre el riesgo de no ser considerado y poniendo en riesgo la consecución de un plan de calidad.

Dependiendo del tipo de software requieren distintos procesos de desarrollo. Cada gestor de proyecto deberá decidir qué estándares de procesos utilizar, o modificarlos si es necesario.

ISO 9000 constituye un conjunto de estándares internacionales que se pueden utilizar en el desarrollo de un sistema de gestión de calidad en todas las industrias. ISO 9001 define qué estándares y procedimientos deben considerarse en una organización, los cuales deben documentarse en un manual de calidad organizacional. La norma ISO 9126 hace referencia a la operación, transición y revisión del software y aunque no las divide en estos tres grupos, señala entre otras cosas la necesidad de lograr que el software opere correctamente y con el grado de exactitud requerido, que los usuarios sean capaces de entenderlo y usarlo, es decir que sea amigable con quienes interactúen con él, que sea capaz de responder correctamente ante fallos o cambios del entorno y que proporcione una ejecución o desempeño apropiado, teniendo en cuenta los recursos utilizados.

4. REFERENCIAS BIBLIOGRÁFICAS

- [1] Sommerville Ian. 2006. *Ingeniería de software*. Madrid Edit Addison Wesley.
- [2] PRESSMAN ROGER, 2002 "Ingeniería de software. Un enfoque práctico". (Quinta Edición). Editorial Mc Graw - Hill Intes Americana de España S.A.U.
- [3] LAUDON KENNETH & LAUDON JANE, 2008 "Sistemas de información Gerencial. Administración de la empresa Digital". (Décima Edición). Editorial Pearson Education Prantice Hall México.
- [4] ISO 9001: 2008. "Gestión de la Calidad".