

LA IMPORTANCIA DE LA CALIDAD EN EL DESARROLLO DE PRODUCTOS DE SOFTWARE

Gérald Lomprey¹ y Saulo Hernandez²

Facultad de Ingeniería y Tecnología, Universidad de Montemorelos, México

1. lompreyg@gmail.com, 2. saulo.hernandez@gmail.com

Technical Report COMP-018-2008

ABSTRACT

This paper emphasizes the importance of quality in software development based on studies of various researchers. Also, it formulates solutions to avoid the lack of quality in the product according to well-established quality standards.

El artículo tiene como objetivo de enfatizar en la importancia de la calidad en el desarrollo de software. Basándose en los estudios hechos por varios investigadores analizándose las consecuencias. Entonces, formulara soluciones para corregir algunos fallos por el mejoramiento del producto. Porque, La industria de desarrollo de software no es la excepción, por lo que en los últimos años se han realizado intensos trabajos para aplicar los conceptos de calidad en el ámbito del software.

INTRODUCCIÓN

Muchos descubrimientos se han hecho a través del mundo hoy en día. El progreso de la tecnología figura casi en todas las noticias. Las necesidades aumentan en todos lados especialmente en la ciencia. Las tecnologías se mueven a gran velocidad. Todo va muy rápido.

Los desarrolladores por su parte, elaboran software usando los nuevos descubrimientos. Dado que el mercado es muy competitivo, el software es desarrollado con precipitación. A menudo, esto se considera como una de las razones por las cuales que no se cumplen las tareas y se respetan los factores y criterios de elaboración.

Conociendo la importancia del software, las necesidades del mercado y cuan exigente es el cliente. ¿Pueden los desarrolladores

permitirse desarrollarlo de cualquier manera? Las preguntas son numerosas como: ¿Es que se debe desarrollar un software solo por desarrollar sin darse cuenta de los estándares? ¿Qué tipo de software puede satisfacer el usuario? ¿Qué necesita el usuario? ¿Un software por poco tiempo? ¿Busca la calidad en el software? ¿Cómo desarrollar un producto de software de calidad? ¿Por qué enfatizar la calidad como parte muy importante en la elaboración y específicamente en el diseño? Son preguntas que el autor va tratar de responder basándose en el análisis de varios investigadores. Además, acentuar el rol de la calidad en el desarrollo de software e insistir en la garantía que necesita el usuario y su importancia.

El artículo habla del uso de los Estándares de Calidad ISO 9000 que describe un sistema de garantía de calidad en términos genéricos que se aplican a cualquier negocio sin importar los productos o servicios, ofreciendo un sistema de garantía de calidad, bien estructurado, organizacional, con responsabilidades y procedimientos.

El artículo propone un sistema para garantizar que el software sea representado de acuerdo con los estándares, y mejorar el proceso eliminando las causas originales de los defectos

así controlar el proceso para asegurar que no se repitan las causas de los mismos.

También se aborda el proceso a seguir para poder desarrollar un software de calidad. También tiene como objetivo enfatizar la importancia del producto. Basándose en las normas de estándares y los resultados de los investigadores. Con el objetivo de presentar un modelo adecuado al usuario. Porque la calidad del software es un problema cotidiano. Se lo encuentra de una forma u otra en nuestra vida diaria. Cuando en un cajero se pide una cantidad de dinero y recibe otra cantidad de otro tipo de billete, ¿Es que eso no puede ser un problema de software? Cuando en un restaurante se bloquean los terminales de cobro y anotación de pedidos, ¿No estamos ante un problema de calidad del software?

Así profundizaremos y analizaremos el impacto de los defectos en la calidad del software. Considerando igualmente los puntos anteriormente mencionados y propondremos algunos elementos de solución como veremos a continuación.

1. DEFINICIÓN DE LA CALIDAD DE SOFTWARE

Primeramente definimos la calidad relacionado al desarrollo de software. Según Pressman [1] y [2] la calidad del software es “*la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente*”.

Según el Departamento de Defensa [3] de los Estados Unidos es la capacidad de un producto software para satisfacer sus requerimientos específicos.

Se define como la capacidad del producto de software para permitirles a usuarios específicos lograr las metas [4] propuestas con eficacia, productividad, seguridad y satisfacción, en contextos especificados de uso.

Se considera como la totalidad de las características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades expresadas o implícitas [5] por Norma UNE 66-001-92 traducción de ISO 8402.

La calidad del software 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*.

Del mismo modo se define de las maneras siguientes:

1.- La totalidad de las funciones y características de un producto software que influyen en su capacidad de satisfacer determinadas necesidades; por ejemplo, el cumplimiento de las especificaciones.

2.- El grado en el que el software posee una combinación de atributos deseada.

3.- El grado en el que un cliente o usuario percibe que el software satisface sus expectativas globales.

4.- Aquellas características globales del software que determinan el grado en el que el software que se está utilizando satisfará las expectativas del cliente. Según [6] Enrique J. Abril Pérez.

La calidad del software puede ser entendida como el grado con el cual el usuario percibe que el software satisface sus expectativas [7] IEEE 729-83.

La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. [8] IEEE, Std. 610-1990.

También la calidad del *software* es el conjunto de cualidades que lo caracterizan y que determinan su [9] utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad

2. IMPACTO DE LA CALIDAD SOBRE EL PRODUCTO

Los fallos de software afectan a todos los sectores y a todos los países, como puede constatarse en la recopilación de incidentes documentados por [10] Peter Neumann desde 1987

Un ejemplo, las deficiencias en el software de manejo de equipajes retrasaron durante 16 meses la inauguración del nuevo Aeropuerto de Denver, desde octubre de 1993 a febrero de 1995. Como consecuencias, Según Jones, [11] presidente de la firma Software Productivity Research, el retraso costó a las líneas aéreas y a la ciudad de Denver un millón de dólares al día.

Otro ejemplo bien documentado se produjo con el lanzamiento en [12] y [13] 1996 del primer cohete Ariane 5 de la Agencia Espacial Europea. Este ingenio que costó 10 años y 7.000 millones de euros desarrollar, explotó por un defecto en el software de control interno antes de que pasara un minuto de vuelo. Cuando el ordenador intento convertir un dato de velocidad lateral del coche de 64 bits a 16 bits [14] y [15] hubo erro de overflow.

Es muy probable que hayamos sufrido los efectos de estos problemas de calidad en forma de retrasos, pérdidas de tiempo o dinero, etc. como simples ciudadanos. Lamentablemente, estos problemas pueden ser mucho más graves si afectan a sistemas críticos; es decir, aquellos cuyo fallo puede provocar graves pérdidas económicas o problemas ambientales o sociales e, incluso, la pérdida de vidas humanas.

Además, El impacto de los errores de software va mucho más allá del coste de su reparación. La pérdida de ingresos, las oportunidades de mercado perdidas, los pedidos tramitados incorrectamente y los errores de facturación, son factores todos ellos que hacen perder dinero a la empresa. Y hay que tener en cuenta también el coste de la insatisfacción de los clientes, intangible pero real.

3. ¿CUÁLES SON LAS PRINCIPALES CAUSAS DE LOS FALLOS DE UN SOFTWARE?

Como es por muchos conocidos, la presión de tiempo para las pruebas (se hacen al final de los proyectos, cuando se acumulan retrasos), la tentación de recortar en calidad cuando hay problemas de tiempo o de recursos, la falta de formación de profesionales y directivos, así como las deficiencias en esta área en la educación universitaria son los factores que más influyen en que exista esta situación.

¿De dónde proceden los problemas del software?

Requerimientos insuficientemente comprendidos: 50%

Diseño no comprendido o incorrectamente trasladado de los requerimientos: 30%

Codificación (error de programación o diseño mal comprendido): 20%
"Software Quality", por Mordechai Ben-Benachem y Cary S. Marliss [16]

Por otro lado, más de un 50% del personal técnico de software y un 70% de los directores de software tienen una formación

deficiente en control de calidad. Según, [17] Software Productivity Research.

4. ¿CÓMO IDENTIFICAR UN PROBLEMA DE CALIDAD DE SOFTWARE?

Administrar unas pruebas al software permite descubrir si es de calidad o no. Las Pruebas de software: un aspecto clave en la calidad.

De la misma forma, la pantalla azul que aparece en una computadora o un mensaje de error es la manifestación de un problema de calidad de software; Los mensajes de errores, las respuestas equivocadas son algunos factores indicativos de problema de calidad del software. También los ejemplos mencionados en otra parte del artículo se pueden considerar como defecto de calidad del producto.

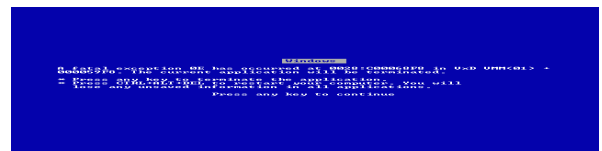


Figura 1. Pantalla azul. [18]

¿Cómo evaluar la calidad del software?

- Se evalúa el proceso con el que se desarrolla el software, mediante una auditoria de proceso frente a un modelo de buenas prácticas reconocidas por la industria o un estándar CMMI, SPICE o ISO 9000.

5. ¿CUÁLES SON LOS ESTÁNDARES DE ELABORACIÓN DE SOFTWARE?

Para [19] Ramiro Carballo, la solución para una mejora del proceso de software pasa por realizar una evaluación del Proceso de Desarrollo en base a un modelo de buenas prácticas reconocido, como CMMI, SPICE u otros.

5.1 ¿Cómo obtener un software con calidad?

La calidad del producto software viene determinada por la calidad del proceso con el que se desarrolla. Teniendo un proceso definido y aplicando mejoras sobre este proceso, podremos incrementar la calidad de nuestros productos continuamente.

5.2 ¿Cómo evaluar la calidad del software?

Se evalúa el proceso con el que se desarrolla el software, mediante una auditoria de proceso frente a un modelo de buenas prácticas reconocidas por la industria o un estándar [20] CMMI, SPICE o ISO 9000.

5.3 ¿Cómo controlar la calidad del software?

El control de la calidad es la fase tradicional de pruebas, en la que se intenta dejar libre de defectos el producto antes de su puesta en producción o su entrega al cliente. A diferencia del aseguramiento de la calidad, que intenta mejorar el proceso de producción para reducir el número de defectos introducidos, y realizar menos esfuerzo en las pruebas.

Hay otros estándares como los IEEE, IBM, HP_Merccury Compuxares, McCall, etc. Que se usa para el desarrollo de software de calidad. Por ejemplo un software de calidad debe ser desarrollado según el modelo de calidad.

5.4 Que es un modelo de calidad de software?

Es un conjunto [21] de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos.

6 ¿CÓMO DESARROLLAR UN PRODUCTO DE SOFTWARE DE CALIDAD?

Hablar de calidad del software implica la necesidad de contar con parámetros que permitan establecer los niveles mínimos que un producto de este tipo debe alcanzar para que se considere de calidad.

Los expertos en calidad sugieren adoptar los pasos siguientes para mejorar la calidad del software:

1 • Establecer un sistema de evaluación y medición para determinar en qué medida se está actuando correctamente. Elegir unos pocos proyectos típicos completados. Calcular el tiempo, el esfuerzo, los errores y el tiempo necesarios para la corrección.

2 • Documentar el proceso de desarrollo actual, por muy caótico que sea.

3 • Calcular el coste de corregir errores de software en términos de las mercancías o servicios vendidos por la compañía. Hacer público este coste en toda la organización informática.

4 • Hacer lo necesario para eliminar errores en requerimientos y diseño.

5 • Entrevistar cuidadosamente a usuarios y directores y verificar que se ha comprendido lo que necesitan. Mostrarles prototipos de diseño. Cualquier gasto adicional generado será compensado en el resto del proceso de desarrollo.

6 • Realizar pruebas (tests) pronto y con frecuencia. Iniciar la planificación de las pruebas (tests) tan pronto como se haya completado la definición de los requerimientos. Comprobar módulos según van siendo desarrollados, sin esperar a que estén integrados. Un buen conjunto de pruebas (tests) será aquel que demuestre que existen errores, no que no existen.

El mejorar la calidad del software no es algo que sucede una vez y queda ahí para siempre, ni tampoco es cuestión de realizar un drástico cambio en el proceso de desarrollo y pensar entonces que el problema de la calidad ha quedado definitivamente resuelto. Es algo

similar a cómo adquirir una forma física excelente. No basta con hacer ejercicio una vez y olvidarse de ello. Es un proceso continuo. Algo que se convierte en una forma de vida.

Bill Gates en [22] "Focus", una revista alemana declara: "No conozco un motivo menos relevante para una actualización que las correcciones de fallos. El motivo para las actualizaciones es presentar nuevas funciones y características."

7 ¿CUÁLES SON LOS FACTORES DE CALIDAD?

Los factores que determinan la calidad del software se clasifican en tres grupos:

7.1 Operaciones del producto: características operativas

– *Corrección* (¿Hace lo que se le pide?)

- El grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente

– *Fiabilidad* (¿Lo hace de forma fiable todo el tiempo?)

- El grado que se puede esperar de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida

– *Eficiencia* (¿Qué recursos hardware y software necesito?)

- La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados

– *Integridad* (¿Puedo controlar su uso?)

- El grado con que puede controlarse el acceso al software o a los datos a personal no autorizado

– *Facilidad de uso* (¿Es fácil y cómodo de manejar?)

- El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.

7.2 Revisión del producto: capacidad para soportar cambios

– *Facilidad de mantenimiento* (¿Puedo localizar los fallos?)

- El esfuerzo requerido para localizar y reparar errores

– *Flexibilidad* (¿Puedo añadir nuevas opciones?)

- El esfuerzo requerido para modificar una aplicación en funcionamiento

– *Facilidad de prueba* (¿Puedo probar todas las opciones?)

- El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos

7.3 Transición del producto: adaptabilidad a nuevos entornos

- *Portabilidad* (¿Podré usarlo en otra máquina?)
- El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo
 - *Reusabilidad* (¿Podré utilizar alguna parte del software en otra aplicación?)
- Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones
 - *Interoperabilidad* (¿Podrá comunicarse con otras aplicaciones o sistemas informáticos?)
- El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas Informáticos, “Juan Manuel” [23]

8 ¿CUALES SON LA IMPORTANCIA Y EL BENEFICIO DE DESARROLLAR UN SOFTWARE DE CALIDAD?

El control de calidad permite ahorrar la máxima cantidad de dinero cuando se realiza al comienzo del proceso de desarrollo de software. No es sorprendente que los errores detectados en los comienzos del proceso de desarrollo de software sean más fáciles de resolver y menos costosos que los que se detectan más adelante. En su libro [24] "Software Economics", publicado en 1981, Barry Boehm afirma que un defecto cuya corrección requiere una hora en el

momento en que se definen los requerimientos del sistema requerirá al menos 100 horas si no se detecta hasta que el sistema está en fase de producción. Este es un argumento muy poderoso a favor de aplicar un énfasis especial a la mejora de la calidad al comienzo del proceso.

En pocas palabras, educar a los profesionales informáticos sobre la importancia de la calidad no cuesta nada, y puede producir rápidamente resultados beneficiosos.

9 CONCLUSIÓN

A pesar de estudiar y de analizar los resultados de las investigaciones, concluimos que la calidad en el desarrollo de producto de software tiene una importancia clave.

Pues, la definición de Calidad del Software, entendida como la satisfacción de las necesidades del cliente en el plazo y presupuesto adecuado, evoluciona cada vez más hacia un concepto de valor global aportado al cliente en términos de servicio.

La calidad de un producto no es algo que se añade al final como si se pintara de un color su exterior, es algo que se cuida a lo largo de todo el proyecto de construcción. Y la falta de calidad puede causar problemas graves al

producto. También puede provocar pérdidas inestimables y consecuencias enormes. Entonces es mejor desarrollar el software con la calidad más alta que posible. La ausencia de defectos, la aptitud para el uso, la seguridad, la confiabilidad [25] y la reunión de especificaciones son elementos que están involucrados en el concepto de calidad del software. Sin embargo, la calidad del software debe ser construida desde el comienzo, no es algo que puede ser añadido después Humphrey.

La calidad es un término que ha adquirido gran relevancia con el paso del tiempo, ya que es considerada como una de los principales activos con los que cuenta un país para mejorar su posición competitiva global [26] Ivanisevich. Para conseguir una buena calidad del software es esencial establecer un programa de medidas a tomar con respecto a los proveedores. Es también importante utilizar los modelos y métodos apropiados para controlar el proceso de desarrollo del mismo.

Así que lograr el éxito en la producción de *software* es hacerlo con calidad y demostrar su buena calidad. Esto sólo es posible con la implantación de un Sistema para el Aseguramiento de la Calidad del *Software* directamente relacionado con la política establecida para su elaboración y que esté en

correspondencia con la definición internacional [27] ISO de calidad, amplia mente aceptada, y por los estándares del grupo ISO 9000.

BIBLIOGRAFÍA

[1] Luis E. Mendoza / Maria A. Perez / Anna C. Griman, *Prototipo de Modelo sistémico de calidad (MOSCA) del software Computación y Sistemas*, vol. 8, Num. 3, pp.196-217. Año 2005. Instituto Politecnico National. <http://redalyc.uaemex.mx/redalyc/pdf/615/61580304.pdf>

[2] Luis E. Mendoza / Maria A. Perez / Anna C. Griman. *Análisis del Impacto del Proceso de Desarrollo en las Características de Calidad del Software*. Universidad Simón Bolívar. http://www.lisi.usb.ve/publicaciones/02%20calidad%20sistemica/calidad_11.pdf

[3] Computerword. *La calidad compensa La mitad de la inversión en desarrollo de software se destina a solucionar fallos*. Sitio de web www.idg.es/computerworld

[4] Oficina Nacional de Gobierno Electrónico e Informática Presidencia del Consejo de Ministros. *Guía Técnica sobre Evaluación de Software para la Administración Pública*. Resolución ministerial n° 139-2004-pcm Lima, 27 de mayo de 2004. :

http://www.ongei.gob.pe/Bancos/Banco_Normas/archivos/Guia-Evaluacion-SW.pdf

[5] Maira R. Moreno Pino / Iliana Hernández Concepción / Sonia Lengster Gil / Omaira Moreno Pérez / Luis González. *Aproximación a un análisis epistemológico de la definición de calidad*. www.monografias.com

[6] Enrique J. Abril Pérez Procesos de las Aplicaciones Web – Informe sobre la “Calidad de las Aplicaciones Web” <http://zarza.usal.es/~fgarcia/doctorado/iweb/0507/Trabajos/CalidadAplicacionesWeb.pdf>

[7] Jesús María Minguet Melián – Juan Francisco Hernández Ballesteros. *La calidad del software y su medida*. Ed. Centros de estudios Ramón Areces S.A. 2003. <http://books.google.com.mx/books?id=DR74RkJIBTMC>

[8] Peláez J.I. Doña J.M / Dapozo G. N. Greiner C. L. *Un Entorno de Decisión para la Gestión y Evaluación de la Calidad en el Desarrollo del Software*. 9 de Julio 1449 (3400) Corrientes, Argentina. <http://www.cacic2007.unne.edu.ar/papers/051.pdf>

[9] Oscar M. Fernández Carrasco, Delba García León y Alfa Beltrán Benavides. *Un enfoque*

actual sobre la calidad del software. ACIMED, septiembre-diciembre, 1995.

[10] Peter G. Neumann. *Illustrative Risks to the Public in the Use of Computer Systems and Related Technology* desde 1987 de sitio de web <http://www.csl.sri.com/neumann/illustrative.html>

[11] Capers Jones . *La calidad compensa La mitad de la inversión en desarrollo de software se destina a solucionar fallos*. [23/10/1998]. Software productivity research <http://www.idg.es/computerworld/articulo.asp?id=58720>

[12] J. L. LIONS. *ARIANE 5: Flight 501 Failure*. Report by the Inquiry Board. Paris, 19 July 1996.

[13] Gérard Le Lann. *The Ariane 5 Flight 501 Failure - A Case Study in System Engineering for Computing Systems*. N° 3079, Décembre 1996. sitio de web <ftp://ftp.inria.fr/INRIA/publication/publication/pdf/RR/part3/RR-3079.pdf>

[14] Luis Fernández Sanz. *La Importancia De La Calidad Del Software*. Sitio de web. <http://www.cfired.org.ar/Default.aspx?nId=5436> Publicado por Baquia.com – 22/9/2008

[15] Luis Fernández Sanz, *Tutorial: Pruebas funcionales y trabajo en equipo*. Universidad

Europea de Madrid. Coordinador grupo de Calidad de software de ATI. 2007. Sitio de web <http://www.ati.es/IMG/pdf/Tutorial2007.pdf>

[16] Mordechai, Ben-Benachem y Cary, S. Marliss. *"Software Quality"*

[17] Mordechai Ben-Benachem y Cary S. Marliss. *La calidad compensa La mitad de la inversión en desarrollo de software se destina a solucionar fallos.* [23/10/1998]. Software productivity rechearch <http://www.idg.es/computerworld/impart.asp?id=58720>

[18] Figura 1
<http://img403.imageshack.us/img403/6822/bsodwe1.png>

[19] Ramiro, Carballo. *Los efectos de la calidad del software La mayoría del software de nuestro país se hace artesanalmente.* Computerworld [Nº:1045] [Pag:25] [04/02/2005] sitio de web <http://www.idg.es/computerworld/articulo.asp?id=165444>

[20] Liliana del S. Gómez Arenas Metodología Para Evaluar La Calidad De Los Sistemas De Información. <http://www.greensqa.com/archivos/Metodologia%20BaQEM.pdf>

[21] Ernesto Quiñones A. *Modelos de Calidad de Software y Software Libre* de web http://www.eqsoft.net/presentas/modelos_de_calidad_y_software_libre.pdf

[21] Juan Manuel Cueva Lovelle. *Calidad del Software.* Conferencia, 21 de Octubre de 1999 Grupo GIDIS. Universidad Nacional de la Pampa. http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF

[23] Rafael Gómez Sánchez. *Software libre vs. Software propietario: Programando nuestro futuro.* Universidad de Sevilla, Spain. Publicado: 15 Octubre 2003. Sitio de web <http://www.historia-actual.com/HAO/Volumes/Volume1/Issue2/eng/v1i2c10.pdf>

[24] Barry W. Boehm. *Software Engineering Economics.* (Englewood Cliffs, NJ : Prentice-Hall, 1981 ISBN 0-13-822122-7). Sitio de web http://en.wikipedia.org/wiki/Barry_Boehm#Software_economics

[25] Humphrey, W.S. *Bad Software: What To Do When Software Fails.* 1997

[26] Ivanisevich et al. *Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software,* 1997. <http://www.scribd.com/doc/208482/Prototipo->

de-Modelo-Sistemico-de-Calidad-MOSCA-del-Software

[27] Oscar M. Fernández Carrasco, Delba García León y Alfa Beltrán Benavides. *Un enfoque actual sobre la calidad del software*. ACIMED, septiembre-diciembre, 1995. http://www.bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm