

# Costo de la Calidad en el Desarrollo de Software

María Teresa Ventura Miranda<sup>1</sup>

<sup>1</sup> Dirección de Sistemas, Dirección General de Servicios de Cómputo Académico-  
Universidad Nacional Autónoma de México. Edificio IIMAS Planta Baja, Ciudad  
Universitaria, México, D.F., C.P. 04510.  
tere@dcaa.unam.mx

**Resumen.** El desarrollo de software es una actividad muy compleja ya que éste es un producto intangible que depende principalmente del esfuerzo intelectual y creatividad de personas que lo realizan. Los errores humanos están presentes en todas las etapas de un proyecto de este tipo y puede llegar a ser muy costosa su corrección. La calidad no es casual. Existen actividades de aseguramiento de la calidad que permiten detectar defectos de manera temprana, las cuales también tienen un costo-beneficio asociado. En este trabajo se describen los tipos de costo de procurar o no la calidad del software y se expone la mejora de procesos como una alternativa para lograrla.

**Palabras:** aseguramiento de la calidad, calidad de software, costos de calidad, desarrollo de software, mejora de procesos,

## 1 Introducción

Este trabajo ofrece un marco general de referencia de los tipos de costos asociados a la calidad y no calidad del software. Resulta de interés para tomar conciencia de la situación especial del desarrollo de software con respecto a otras industrias, así como para comprender con mayor amplitud el concepto de calidad, proporcionando elementos que permitan tomar decisiones en relación a las inversiones de aseguramiento de calidad y la mejora de procesos.

## 2 Calidad en el desarrollo de software

### 2.1 Desarrollo de software

El software es definido como la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo (IEEE Std 729-1993). Bajo esta definición el concepto de software va más allá de los programas de cómputo en sus distintas formas: código fuente, binario o código ejecutable, además de su documentación.

El desarrollo de software es una actividad compleja que depende principalmente del esfuerzo intelectual de las personas, así como de los procesos y la tecnología con la que se desarrolla. Existen diversas metodologías y procesos para el desarrollo de software, algunas de las actividades típicas que se realizan son las siguientes:

1. **Requerimientos.** Identificar las necesidades del usuario y documentarlas en términos de los servicios que debe ofrecer el software.
2. **Análisis y diseño.** Desarrollar un modelo técnico conceptual de la solución que permita cumplir todos los requerimientos especificados y sirva como base para la construcción del software.
3. **Codificación.** Implementar el modelo conceptual en un lenguaje o herramienta de programación, es decir, generar el código fuente, además de realizar las pruebas unitarias a cada componente de software y la integración de estos.
4. **Pruebas de sistema.** Verificar que el software cumpla con los requerimientos especificados y no presente fallas.

## 2.1 Calidad de software

No existe una sola definición de la calidad de software. Las dimensiones clave de la calidad son las siguientes [9]:

1. **Nivel de satisfacción.** Grado en el cual los clientes o usuarios perciben que el producto de software cumple sus necesidades, requerimientos y expectativas.
2. **Valor del producto.** Grado en el cual un producto de software tiene valor para sus involucrados, en relación al ambiente competitivo.
3. **Atributos de calidad.** El grado en el cual un producto de software posee una combinación de propiedades deseadas (eficiencia, integridad, portabilidad, entre otras). Existen diversas taxonomías de los atributos de calidad, una de éstas es la que propone el Handbook of Software Quality Assurance [1] que considera: a) calidad de diseño (correcto, facilidad de mantenimiento, facilidad de verificación), b) calidad de desempeño (eficiencia, integridad, fiabilidad, facilidad de uso, facilidad de prueba), c) calidad de adaptación (facilidad de expansion, flexibilidad, portabilidad, reusabilidad, interoperabilidad. Esfuerzo necesario para acoplar el software sobre una plataforma con otro software y/o plataforma, intra-operabilidad).
4. **Defectos.** Grado en el cual el producto de software presenta fallas al utilizarse, atribuibles a éste.
5. **Calidad del proceso.** En relación al proceso de desarrollo por el cual fue construido el software.

Lograr todas estas dimensiones de la calidad es algo muy complejo, considerando además que el software es un producto intangible. Los defectos del software son una consecuencia indeseable del error humano. En cada etapa del desarrollo de software se cometen errores, es por ello que las pruebas y el aseguramiento de la calidad son actividades necesarias en los proyectos de desarrollo de software; y es importante

considerar que para el cumplimiento su objetivo, es necesario “invertir” recursos significativos.

Al contrastar el costo de encontrar errores en cada fase del desarrollo contra el costo incurrido de encontrar el error en una fase posterior, se demuestra que una inversión temprana en prevención y descubrimiento de errores, mediante aseguramiento de la calidad, estándares, revisiones, auditorías e inspecciones, permiten remover los defectos con un costo significativamente menor [14].

### **3 Costos de la calidad en el desarrollo de software**

De acuerdo con diversos autores [2], [8], [9], [11], [13], a continuación se presentan los diversos tipos de costos asociados con la calidad de software:

#### **a) Costos para asegurar la calidad o costos de conformidad**

##### **Costos de Prevención**

Costo de todos aquellos esfuerzos para asegurar la calidad del software y prevenir defectos en todas las fases del desarrollo de software. Por ejemplo: aseguramiento de la calidad (planeación de la calidad, mejora de procesos, definición de procesos, políticas y estándares, obtención, análisis y uso de datos sobre la calidad, análisis de causas raíces), requerimientos (especificaciones y prototipos), administración del proyecto (planeación, capacitación, recopilación de métricas), librería de reuso (capacitación y herramientas), administración de la configuración (capacitación y herramientas), consultoría.

##### **Costos de evaluación**

Costo del esfuerzo para descubrir la condición de la calidad del software (evaluaciones planeadas). Esto comprende por ejemplo: evaluación de proyectos (revisión de especificaciones de requerimientos, diseño y componentes, verificaciones y validaciones en general, inspecciones, pruebas unitarias, de integración y de sistema), auditorías de calidad del producto, evaluaciones externas, pruebas de productos adquiridos.

## **b) Costos por falta de calidad o costos de no conformidad**

### **Costos de fallas internas**

Costo del esfuerzo para detectar y corregir problemas previos a que el usuario los detecte, por ejemplo: corregir defectos y el retrabajo correctivo en todas las etapas.

### **Costos de fallas externas**

Costo del esfuerzo para corregir problemas que son detectados por el usuario. Esto incluye remoción de fallas (mantenimiento correctivo, retrabajo, codificación y prueba de defectos, compra de actualizaciones de productos), soporte (soporte técnico para responder por los defectos, preparación de apuntes de respuestas de soporte, investigación de las quejas del usuario), compensación (penalizaciones, reclamaciones, productos devueltos, reembolsos, descuentos, costos de garantía), otros (mala reputación, pérdida de ventas, usuario insatisfecho, pérdida de la confianza del cliente).

Sobre este enfoque Kaner menciona que el análisis del costo de calidad sólo considera los costos de la organización y no los costos del cliente, los cuales pueden ser por ejemplo: tiempo desperdiciado, pérdida de datos, pérdida de negocios, pérdida de clientes, desprestigio, salida de empleados, insatisfacción de clientes internos y externos, reemplazo del producto, re-configuración del sistema, soporte técnico, daños y perjuicios, quiebra [8].

Por otra parte, el argumento sobre que la calidad es gratis “quality is free” expresado por Crosby, Juran, Gryna y otros, significa que si el presupuesto destinado a la prevención de defectos es incrementado, los costos involuntarios de retrabajo decrecientan por mucho más que el incremento en los costos de prevención. El resultado neto es un costo más bajo, por lo tanto la calidad es gratis [16].

En este contexto, el costo de calidad son los costos totales que podrían desaparecer si no hubiera defectos. Es la suma de los costos asociados con la evaluación, prevención, fallas internas y externas [2].

Hay muy pocos costos a reducir en el ámbito de desarrollo de software. No puede reducirse el costo de materiales o inventarios, es poca la influencia del costo de la tecnología, y el costo de los expertos sigue aumentando. Las principales reducciones de costos están en relación con los defectos, encontrar y corregir las causas raíces de estos defectos, así como en medir y monitorear constantemente nuevas oportunidades de reducir costos de calidad. Prevenir defectos, reduce costos [11].

De acuerdo con el estudio de Krishnan, con una muestra de 27 proyectos de desarrollo: El costo total del desarrollo y soporte del software decrementa al incrementar la calidad. Inversiones en diseño y planeación, especialmente para proyectos grandes y complejos, permiten reducir sustancialmente el costo total [10].

Es importante medir la relación entre los cuatro tipos de costos de calidad para determinar qué parte se invierte en costos de prevención con respecto a los costos de corrección. Al agregar recursos para la prevención, los costos totales de la calidad son reducidos. Ésta es la razón por la cual la calidad es considerada gratis.

El costo de la calidad (CoQ), técnica contable introducida por Juran en 1951 utilizada en el ámbito de la manufactura, puede ser adaptada al software para proveer una justificación de las inversiones en mejora de procesos, así como para comprender los aspectos económicos involucrados en la entrega de software de buena calidad [3], [9].

#### **4 Mejora de procesos como alternativa para mejorar la calidad del software**

Para construir un sistema de software de alta calidad, debe asegurarse que cada una de sus partes sea de alta calidad. Para mejorar la calidad de un producto, debe mejorarse la calidad del proceso. Cuando la calidad del software es pobre, el proceso de desarrollo se enfoca a detectar y corregir defectos [4].

Las empresas de software requieren reducir costos, mejorar calidad y mejorar tiempos. Para lograr estos objetivos y mantenerse competitivas, deben mejorar sus procesos de desarrollo de software. Los desarrolladores de software están conscientes que todo el software potencialmente tiene defectos, esto hace que se realicen actividades para prevenir, encontrar y corregir errores en el software [13].

Los costos de la mejora de procesos comprenden a) costos iniciales (no recurrentes), costos de mantenimiento (recurrentes) y costos de optimización. En cada uno de ellos existen factores de costo como son: recursos humanos, tecnología y administración. El proceso inicial por ejemplo, empieza con el reconocimiento de la alta dirección sobre la necesidad de la organización en mejorar y termina cuando el proceso de ingeniería de software está suficientemente bien definido y practicado. El tiempo de este esfuerzo es de 4 a 6 años dependiendo de factores como el estado de la organización y la cantidad de apoyo de la administración para el esfuerzo de mejora. Algunos de los costos de esta fase están relacionados con: la evaluación inicial, implementar programa de métricas, la documentación procesos, la adquisición de herramientas y la capacitación inicial [15].

## 5 Conclusión

Es una necesidad inminente de la industria del software y de los usuarios que el software desarrollado cumpla con todas las dimensiones de la calidad. La calidad no es casual, sino resultado de las actividades y decisiones que se realizan para asegurarla. Las organizaciones deben invertir en actividades que permitan asegurar la calidad, el beneficio de hacerlo es significativamente mayor que no hacerlo.

## Referencias

1. Ashrafi N.: The Impact of Software Process Improvement on Quality: in Theory and Practice. Information & Management No. 40 (2003)
2. Demirörs O., Yildiz Ö., Selcuk G.: Using Cost of Software Quality for a Process Improvement Initiative. Proceedings of the 26th EUROMICRO Conference (EUROMICRO'00), IEEE (2000)
3. Houston D., Keats B.: Cost of Software Quality: A Means of Promoting Software Process Improvement (1998).
4. Humphrey W.: A Personal Commitment to Software Quality. Software Engineering Institute (1994)
5. Jeffery R., Ruhe M., Wiczorek I.: A comparative study of two software development cost modeling techniques using multi-organizational and company-specific data. Information and Software Technology No. 42 (2000)
6. Jones, C.: Activity-based software costing. Software Productivity Research Inc. (1996)
7. Jørgensen M., Moløkke K.: A Preliminary Checklist for Software Cost Management. Proceedings of the Third International Conference On Quality Software (QSIC'03), IEEE Computer Society (2003)
8. Kaner, C.: Quality Cost Analysis: Benefits and Risks. Software QA, Volume 3, No. 1 (1996).
9. Krasner H.: Using the Cost of Quality Approach for Software. CROSSTALK The Journal of Defense Software Engineering (1998)
10. Krishnan M.: Cost, Quality and User Satisfaction of Software Products : An Empirical Analysis. Carnegie Mellon University (1993)
11. Mandeville W.: Software Cost of Quality. IEEE Journal On Selected Areas in Communications, vol. 8, no 2 (1990)
12. Peslak A. Improving Software Quality: An Ethics Based Approach. ACM (2004)
13. Raffo D., Harrison W., Settle J., Eickelmann N.: Understanding the Role of Defect Potential in Assessing the Economic Value of Process Improvements. Proceedings of the Second Workshop on Economics-Driven Software Engineering Research (2000)
14. Rubey R., Browning L., Roberts A.: Cost Effectiveness of Software Quality Assurance. IEEE (1989)
15. Scott G., Hughes A.: The care and feeding costs of a maturing software process. IEEE (1991).
16. Slaughter S., Harter D., Krishnan M.: Evaluating the Cost of Software Quality. Communications of the ACM Vol. 41, No. 8 (1998)
17. Yang H.: Software quality management and ISO 9000 implementation. Industrial Management & Data System 101/7 (1998)