

## Aprovechando los defectos para prevenir su repetición. Experiencia en una organización pequeña.

*ISC, LSCA Clara Min Poblete*  
Universidad Veracruzana, Xalapa, Ver. México.  
[claraminp@yahoo.com](mailto:claraminp@yahoo.com)

*Dr. Juan Manuel Fernández Peña*  
Universidad Veracruzana, Xalapa, Ver. México.  
[jfernandez@uv.mx](mailto:jfernandez@uv.mx)

*Dra. María de los Ángeles Sumano López*  
Universidad Veracruzana, Xalapa, Ver. México.  
[asumano@uv.mx](mailto:asumano@uv.mx)

**Área de participación:** *Sistemas Computacionales*

### Resumen

Una de las formas de mejorar la calidad de productos de software a partir de las pruebas es el análisis de los defectos identificados, buscando sus causas, para luego proponer mejoras al proceso de desarrollo y a las propias pruebas. El presente trabajo se desarrolla en una pequeña organización de gobierno, donde existe un reducido equipo de desarrollo de software y adquiere software *ad hoc*, con el propósito de mejorar la calidad del software desarrollado y verificar el que se adquiere de manera más adecuada.

Se reportan experiencias obtenidas en la clasificación de defectos usando el método de clasificación ortogonal y su posterior análisis causal, hasta proponer mejoras al proceso local y la verificación del software que se encarga. El aporte principal está en la adecuación de la clasificación ortogonal de los defectos y la identificación de sus causas básicas. La experiencia ganada, además de beneficiar a la organización donde se realizó, permitirá trasladarlo a otras de características similares.

**Palabras clave:** *Clasificación de Defectos, Análisis de los defectos, Prevención de defectos, Mejora de la calidad del software.*

### Introducción

La realización de pruebas en una organización que desarrolla y adquiere software sir para varios propósitos, inicialmente verificar sus productos y los que compra. A largo plazo resulta más relevante la prevención de los defectos, con la cual se mejora la calidad de los productos. Para prevenir los defectos debe conocerse el problema que se enfrenta, que depende del tipo de organización, la capacidad de su personal y la naturaleza del software desarrollado. Todo ello se puede resumir, de alguna manera, en que cada organización está más propensa a ciertos tipos de defectos. Así estudiando el pasado puede conocerse cómo provenir defectos futuros, siempre y cuando se tenga registro de los problemas encontrados.

Este trabajo busca precisamente analizar los defectos registrados en una organización pequeña que tiene como objetivo brindar el servicio de seguridad a personas físicas y morales del estado. Como en toda organización, se cuenta con áreas para la gestión y administración de recursos de la empresa, una de ellas tiene la responsabilidad de desarrollar y mantener sistemas de información por medio de terceros o por cuenta propia, asegurándose de su calidad. El software con el que cuenta la empresa tiene una vida corta (aproximadamente tiene tres años en operación) y han surgido fallos descubiertos en la operación del mismo originando, como consecuencia, un retraso en las operaciones, perjudicando las labores importantes y la toma de decisiones que dependen de la información que se procesa (por ejemplo cobro a clientes). En la organización se tienen dos tipos de desarrollo: los adquiridos por outsourcing (empresas dedicadas a ofrecer servicios de desarrollo de software) y los desarrollos internos. A partir de

2009, a raíz de observar estos problemas, se comenzó a registrar defectos para analizarlos, conocer su origen y su solución o las alternativas para evitarlos.

El objetivo principal del trabajo consiste en mejorar la calidad del software de la organización. Para lograrlo se partirá de una clasificación de los defectos registrados, para luego analizar las causas de los más frecuentes. Finalmente, se preparan sugerencias para prevenir los defectos detectados. Se espera que tener identificadas aquellas equivocaciones cometidas con anterioridad permitirá evitarlas en el mantenimiento o nuevo desarrollo de algún software, reduciendo la inversión del tiempo y costo que genera un defecto. En el caso del software adquirido fuera de la organización, el estudio de los defectos permitirá realizar un mejor proceso de aceptación, evitando fallas que puedan preverse.

Con la prevención de defectos se espera mejorar la calidad de los productos de software de la organización, Mejorando al mantenimiento, la adquisición del nuevo software, el proceso de las pruebas y el desarrollo interno del equipo con el que cuenta la organización.

Inicialmente se describen antecedentes teóricos relativos a la clasificación y análisis de defectos; luego se muestran los resultados observados a través de los registros de la organización y su análisis. Finalmente, se proponen algunas medidas para la prevención de defectos.

## Clasificación de Defectos

A lo largo del tiempo la realización de pruebas ha ido acompañada de la clasificación de defectos, generalmente de manera arbitraria y particular a cada empresa. Beizer propuso una manera de clasificar defectos que permite cierto grado de estandarización, pero dejando margen a la particularización de la misma (Beizer, 1990). Esta clasificación aún es muy empleada. Existen otras como la Norma IEEE 1044-93 (IEEE, 1994), la de Cem Kaner y sus colaboradores (Kaner et al, 1990) y alguna especializada en seguridad (Landwehr et al, 1996). Sin embargo en la década de 1990 se hizo evidente que las clasificaciones dejaban abiertos muchos problemas por ambigüedades y traslapes en las categorías usadas. Autores como Chillarege (Chillarege, 1994) y Demillo y Mathur, citados en (Ploski et al, 2007), propusieron formas de evitar las ambigüedades dadas. El método propuesto por Chillarege, conocido como *Clasificación Ortogonal de Defectos*, (ODC por sus siglas en inglés), continuó su desarrollo siendo muy empleado en IBM y otras empresas (Chillarege, 1999).

Muchos de los problemas para clasificar los defectos vienen de las diferentes concepciones acerca de su naturaleza. En la definición de ODC se parte de considerar un defecto como un cambio que se debe aplicar al software para corregir un funcionamiento inadecuado.

De acuerdo al método ODC se registra información en dos momentos: el primero (apertura) cuando ocurre una falla, anotándose las circunstancias que la provocaron y el impacto en los usuarios. El segundo (cierre), cuando un defecto ha sido reparado, encontrando su naturaleza exacta, anotándose el tipo de cambio necesario para su solución. Los defectos se clasifican en varias categorías mutuamente excluyentes pero complementarias, es decir ortogonales.

Las principales categorías sirven para identificar cada defecto. En la etapa de apertura se registra la actividad en que ocurrió (requerimientos, diseño, etc.), el impacto (en qué afecta, por ejemplo la integridad) y el disparador (situación que hace aparecer la falla). En el cierre, el sujeto del cambio necesario (modelo de diseño, código, etc.) y el tipo de la naturaleza del cambio (por ejemplo, asignación, inicialización, algoritmo).

En la Tabla 1 se muestran los criterios de clasificación de un defecto específico. El problema se identificó cuando varios usuarios realizaban las mismas operaciones de manera concurrente. Al buscar la causa se llegó a la conclusión de que fue un error en el diseño del algoritmo que no logra finalizar las operaciones cuando procesa varias solicitudes simultáneas. Para su corrección

de procedió a cambiar la forma en que se atiende las solicitudes, bloqueando el recurso, por medio de banderas para identificar cuándo se está ocupando éste.

Este método de clasificación abre las puertas a una amplia variedad de técnicas de análisis causal de defectos y de obtención de estadísticos acerca de ellos. Con el Análisis causal de defectos se identifican causas posibles y se buscan maneras de eliminar o disminuir éstas y de iniciar acciones orientadas a prevenir defectos en un futuro. El análisis es cualitativo y sólo está limitado por la capacidad y habilidad de los que la realizan (Chillarege et al, 1992).

**Tabla 1 Ejemplo de clasificación de un problema con ODC**

Etapa de apertura		Etapa de cierre	
Actividad	Pruebas de sistema	Origen	Diseño
Disparador	Cargas de trabajo o estrés	Tipo de defecto	Algoritmo implementado
Impacto para cliente	Fiabilidad de su operación	Calificador	Diseño incorrecto
		Fuente	Desarrollo interno
		Edad	Base

Para preparar el análisis causal se prepara un histograma con las frecuencias de los defectos, ordenados descendientemente y luego se seleccionan los tipos más frecuentes, aplicando el llamado Principio de Pareto. De este modo se atiende primero los defectos con mayor ocurrencia en la empresa.

Cuando el análisis de causas de defectos está completo, las acciones propuestas pueden incluir cambios en las prácticas empleadas, agregar herramientas, aplicar nuevos métodos y mejorar en la comunicación.

### Análisis de datos registrados

Con la intención de identificar cuáles defectos son los más frecuentes y así considerarlos en pruebas de proyectos futuros, se decidió estudiar la información disponible en la organización donde se aplica este estudio. Inicialmente se han registrado defectos, de manera informal, desde hace tiempo, en varios formatos. Tomando como referencia la clasificación sugerida en el método ODC, se procedió a clasificar los defectos y su solución para los datos registrados desde julio del 2009 hasta el mes de febrero del 2010. En ese período se encontraron y repararon 37 defectos, donde la mayoría corresponden a los sistemas que ya se encuentran implementados en alguna área de trabajo. El número quizá suene reducido, pero resultan importantes por los problemas que generan.

Para realizar la clasificación de defectos de esta institución se utilizó el siguiente proceso:

- Primero se registraron todos los defectos identificados en algún medio, para darle tratamiento y saber por qué y en qué momento ocurre la falla.
- Después, se procede a investigar el defecto para eliminarlo desde su raíz.
- Por último se registró la solución para conocer la causa con la clasificación ortogonal de defectos.

En cuanto a la clasificación de defectos, la Figura 1 muestra el impacto que provocan los defectos al cliente cuando estos producen una falla. Puede observarse que el impacto mayor se refleja en la integridad de la información del software. Como segundo lugar se observa que los fallos frecuentes son de diagnóstico en donde el software muestra un mensaje en forma de código el cual hace referencia al fallo interno; estos han sido y son fáciles de solucionar. También fueron detectados fallos que impactan la fiabilidad del producto para que el usuario pueda operarlo y los relacionados con la instalación del software. Éstos son importantes de mencionar

ya que originan problemas en la operación; su reducción mejorará la operatividad para el usuario final.

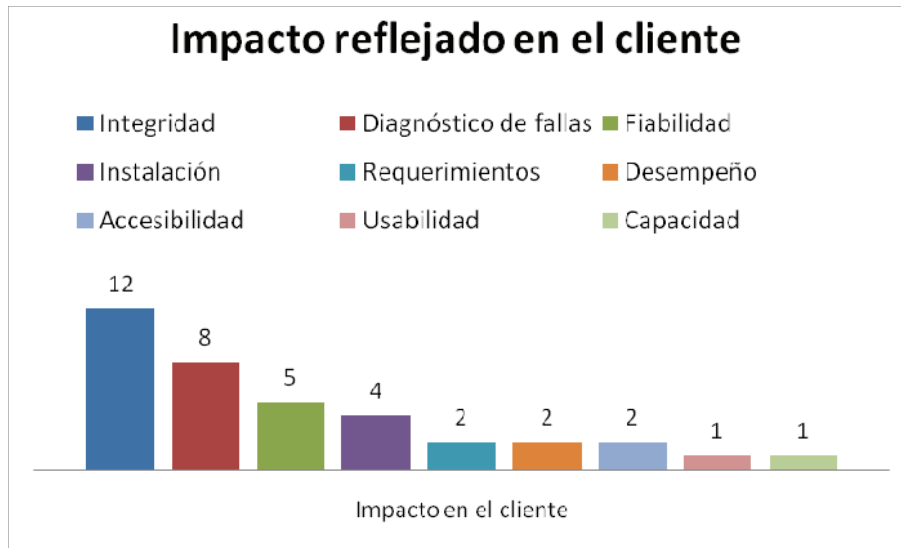


Figura 6 Gráfica que muestra el impacto de los defectos encontrados en el cliente.

En la Figura 2 los defectos fueron clasificados de acuerdo al artefacto que fue modificado, separándolos por tipo. Los resultados obtenidos muestran que existe un alto grado de fallos relacionados con comprobaciones. Otra de las observaciones el comportamiento de la Figura 2 es que existe un alto nivel de fallos en los algoritmos de interpretación del diseño al código y también en el propio código. Esto puede ayudar a ser más específicos cuando se realiza los diseños de software para minimizar los fallos en las comprobaciones. También puede observarse que el número de problemas que debieron corregirse en el diseño es igual al de aquellos que requirieron cambios en el código.

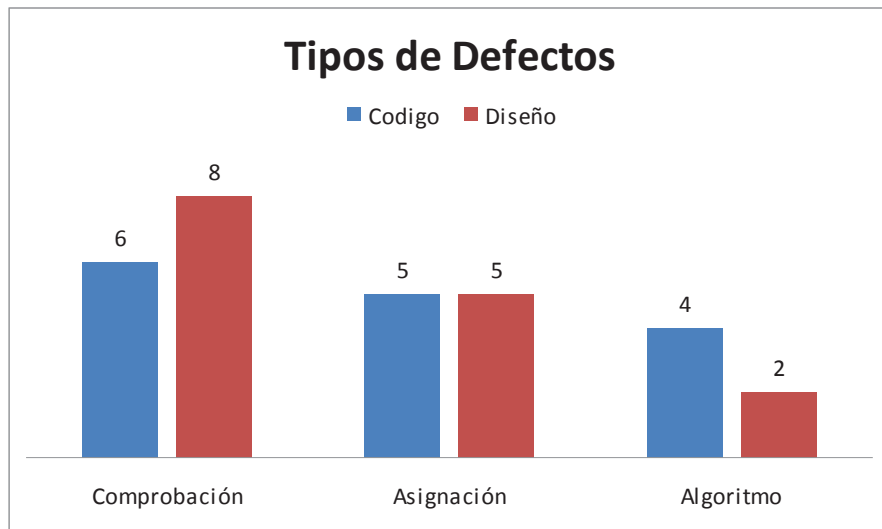


Figura 7 Tipos de defectos.

Dado que parte del software se adquiere, es importante identificar si hay diferencias de acuerdo al origen del software. La Figura 3 muestra la actividad donde se identificó el defecto separando los casos de software desarrollado localmente de los adquiridos. La mitad de los defectos es de Outsourcing: ocasionadas por diferentes situaciones como lo son la disponibilidad de acceso a la red, permisos o fallas internas provocadas por entradas de datos no válidos, entre otros. Se puede

observar que la mayoría de los defectos aparecen durante las pruebas de sistema o cuando el usuario está operando el software. Las inspecciones identifican pocos problemas.

De acuerdo a los resultados obtenidos de los defectos revisados en esta empresa puede identificarse las causas donde se debe trabajar para evitar los defectos: La identificación de los requerimientos y el medio ambiente en el que opera el software.

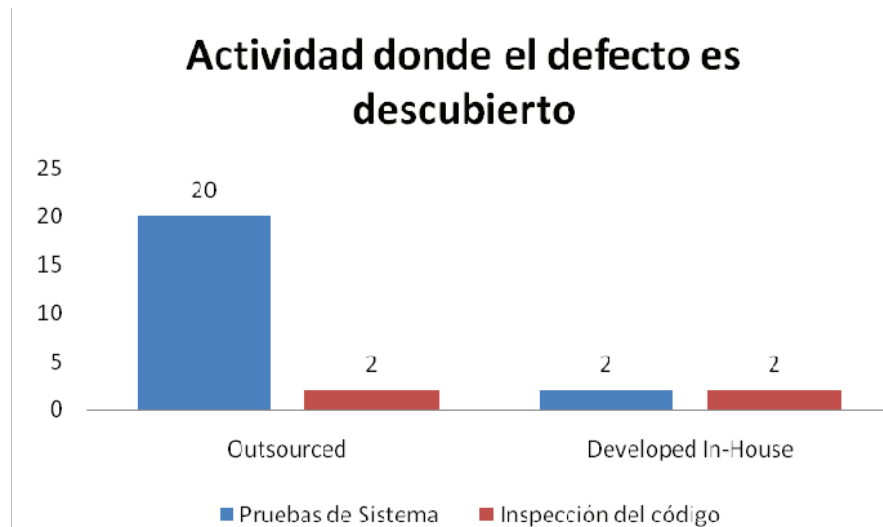


Figura 8 Actividades en donde se descubren los defectos.

### Propuesta

De los resultados obtenidos en la sección anterior, se propone utilizar en la empresa, permanentemente, la clasificación de defectos ortogonal. Esto tiene dos finalidades: la primera mejorar y medir la calidad del software utilizado en la empresa. La segunda, mejorar los productos y el proceso en la obtención o creación del software, por medio de guías que proporcionen un conjunto de reglas y pasos que ayudará a reducir los defectos en el ciclo de desarrollo de software.

Algunos aspectos que contemplan las guías son:

- Los desarrolladores tendrán un punto de referencia que identifiquen los fallos ya ocurridos para que no sean reflejados en el mantenimiento o nueva funcionalidad del software tanto de forma interna o externa.
- Se podrá planear pruebas utilizando varios escenarios en donde pueda ocurrir el fallo, teniendo como referencia el conocimiento e historia de todos los defectos del software.
- Mejorar la calidad del software adquirido, comunicando al proveedor las inconsistencias que deben ser eliminadas para que un nuevo producto o el mantenimiento de los existentes no repita los problema identificados.
- Informar a todos los desarrolladores sobre los fallos ocasionados en la organización para que sea considerado en las labores de producción del software.

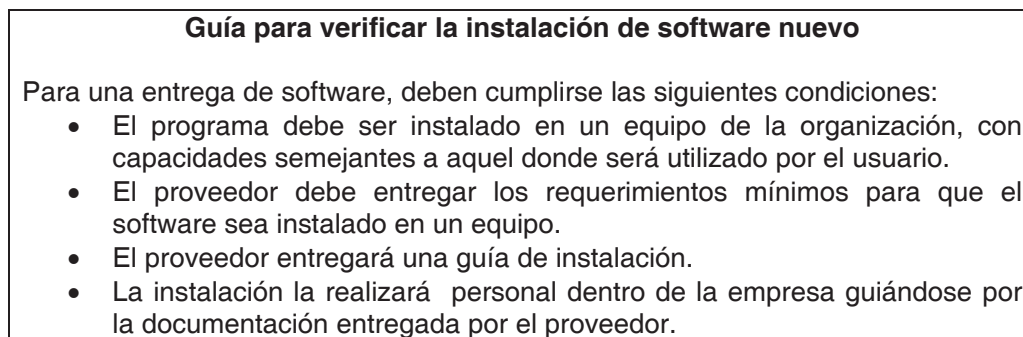
Algunas de las guías se organizan como listas de verificación sencillas, para causar poco retraso en el trabajo de los desarrolladores.

Otras prácticas que deben ser consideradas son:

Realizar reuniones de trabajo en el equipo para destacar los problemas obtenidos y cuáles han sido las soluciones. Ésto genera retroalimentación en el equipo.

Llevar un registro de los defectos identificados y sus causas y frecuencia. La identificación de un fallo y su solución se refleja en la satisfacción del usuario y la reducción del tiempo para arreglar el fallo. Con este registro muchos defectos serán identificados y prevenidos en el proceso de desarrollo.

Un ejemplo de guía es el que se muestra en la Figura 4. Se tomó el caso de la instalación de software adquirido a terceros, que usualmente se presenta en sus equipos, apareciendo después que son necesarias librerías ausentes.



**Figura 4 Fragmento de guía de aceptación de software**

## Conclusiones

Con la clasificación de defectos realizada se ha logrado lo siguiente:

- Fueron identificados los defectos más frecuentes en el software utilizado, logrando cuantificar en qué parte del proceso fue cometido el error y cómo perjudica al cliente.
- Se identificaron las fallas más comunes en el proceso de desarrollo.
- Ya se lleva un control de los defectos del software utilizado, tanto desarrollado de forma interna y de forma externa, en donde el personal que lleva el mantenimiento puede resolver con mayor rapidez.

Cabe mencionar que los resultados de un trabajo como el presentado también ayudan a terceras personas, como son proveedores y usuarios finales. Se refleja en la satisfacción que logran proveedor y cliente como resultado de una buena negociación.

Con la finalidad de disminuir la frecuencia de las fallas más comunes, se recomienda utilizar la clasificación de defectos en organizaciones donde exista personal que se encargue de dar soporte al software y tenga cierto contacto con usuarios finales.

Como continuación del presente trabajo se trabajará en mejorar las guías propuestas y en analizar los resultados de su aplicación en la prevención de defectos.

## Referencias

- <sup>1</sup> Beizer, B. *Software Testing Techniques*. Second Edition, Van Nostrand Reinhold New York, 1990.
- <sup>2</sup> Chillarege, R., Bhandari, I.S., Chaar, J.K., Halliday, M.J., Moebus, D.S., Ray, B.K., and Wong, M. (1992) Orthogonal Defect Classification - A Concept for In-Process Measurements, *IEEE Transactions on Software Engineering*, 18(11), pp 943-956
- <sup>3</sup> Chillarege, R., (1994) ODC for Process Measurement, Analysis and Control, *Proceedings, Fourth International Conference on Software Quality*, ASQC Software Division, Oct 3-5, 1994 McLean, VA



<sup>4</sup> Chillarege, R. (1999) *Software Testing Best Practice*, IBM Research- Technical Report RC 21457 Log 96856, April 26, 1999.

<sup>5</sup> (IEEE, 1994) *Norma IEEE 1044-93*. IEEE Std 1044-1993 IEEE Standard Classification for Software Anomalies. Description.[http://standards.ieee.org/reading/ieee/std\\_public/description/se/1044-1993\\_desc.html](http://standards.ieee.org/reading/ieee/std_public/description/se/1044-1993_desc.html)

<sup>6</sup> Kaner, C., Falk, J. and Nguyen, H.Q. *Testing Computer Software*, second edition, Wiley, 1990

<sup>7</sup> Landwehr, C.E., A.R. Bull, J.P. McDermott, W.S. Choi (1994) A taxonomy of computer program security flaws, *ACM Computing Surveys*, 26(3), pp 211-254.

<sup>8</sup> Ploski, J., Rohr, M., Schwenkenberg, P., and Hasselbring, W. (2007) Research Issues in Software Fault Categorization, *ACM Software Engineering Notes*, 32(6), pp 1-8

## Acerca de los autores

*Clara Min Poblete estudió la licenciatura en Ingeniería en Sistemas Computacionales en el Instituto Tecnológico Superior de Xalapa y la Licenciatura en Sistemas Computacionales en la Universidad Veracruzana. Actualmente estudia la Maestría en Ingeniería de Software en la Universidad Veracruzana en Xalapa, Veracruz.*

*El Dr. Juan Manuel Fernández Peña cursó sus estudios de licenciatura en Física en la Facultad de Ciencias de la UNAM (Universidad Autónoma de México), de Maestría en Investigación de Operaciones en el Instituto de Ingeniería de la UNAM y Doctorado en Ciencias de la Computación en el Centro de Investigación en Computación del IPN (Instituto Politécnico Nacional). Trabajó como investigador de 1971 a 1980 en el IIMAS (Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas) de la UNAM, como docente en diversas Universidades y desarrollado software a particulares. Actualmente es docente en la Universidad Veracruzana.*

*La Dra. María de los Ángeles Sumano López cursó sus estudios de licenciatura en Matemáticas en la Facultad de Ciencias de la UNAM (Universidad Autónoma de México), de Maestría en Ciencias de la Computación en el IIMAS de la UNAM y Doctorado en Ciencias de la Computación en el Centro de Investigación en Computación del IPN (Instituto Politécnico Nacional). Ha desarrollado software en varias dependencias de gobierno y de manera privada, ha sido docente en varias universidades. Actualmente es docente en la Universidad Veracruzana.*

## Autorización y renuncia

*Los autores del presente artículo autorizan al Instituto Tecnológico de Orizaba (ITO) para publicar el escrito en el libro electrónico del coloquio de investigación multidisciplinaria, en su edición 2010. El ITO o los editores no son responsables ni por el contenido ni por las implicaciones de lo que está expresado en el escrito.*