

El Contexto de la Prueba de Software

PROBAR PARA INCREMENTAR LA CALIDAD



El Profesor Doctorando **Luis Vinicio León Carrillo** es profesor investigador del Departamento de Electrónica, Sistemas e Informática del ITESO. Es Presidente y socio fundador de e-Quality, empresa especializada en prueba de software. Es cofundador del Capítulo Guadalajara de la AMCIS, así como su Secretario actual.

En países desarrollados encontramos toda una industria de prueba de software, constituida entre otras cosas por una buena cantidad de profesionales especializados, proveedores de herramientas, congresos, publicaciones periódicas, y múltiples alternativas de capacitación y certificación. La industria de software de esos países consume esa especialización, así como la objetividad e independencia con que ésta puede venir acompañada. Considero que en México la prueba no ha mostrado aún su potencial como catalizador en el incremento de la calidad de productos de software, así como en la disminución de los costos de desarrollo; esto en buena medida porque no se había ofrecido ese servicio de manera integral, especializada e independiente. En éste y los subsiguientes tres números, haré un recorrido por la disciplina de la prueba de software, brindando una perspectiva amplia de la misma.

El Contexto de la Prueba de Software

Inicialmente, la prueba fue confundida con el *debugging*, y se llevaba a cabo para ganar confianza en que el sistema sí “corría bien”. Los trabajos de G. Myers en los 70’s vienen a constituir un parteaguas: desde su punto de vista, la labor del *tester* es demostrar que el sistema “no satisface los requerimientos”.

Estos son algunos de los principios que Myers enunció al respecto:

- La organización desarrolladora debiera evitar probar sus propios sistemas (hay una barrera mental que dificulta que encontremos defectos en aquello que construimos con esmero y dedicación).
- No debe planearse un esfuerzo de prueba bajo el supuesto de que no se encontrarán defectos.
- Probar es una tarea creativa e intelectualmente demandante.

La prueba de software debiera verse como un proceso paralelo al de desarrollo de software, y que se realiza por el convencimiento de que todo sistema debe ser revisado con el objetivo de establecer si el nivel de calidad requerido es alcanzado. Esto aceptando limitantes prácticas que implican la imposibilidad de revisarlo de manera exhaustiva, pero aplicando técnicas ingenieriles para subsanar estas limitantes.

En ocasiones la prueba se ve como una *parte integrante* del Aseguramiento de la Calidad (QA). La actividad de ésta última gira alrededor de los procesos, bajo la hipótesis (heredada del área de la manufactura) de que la calidad del producto está determinada en buena medida por la calidad del proceso. Esa suposición es ampliamente debatida y, de hecho, el comentario de Hanna Oktaba en su columna del número anterior: “la evidencia de que [los procesos] realmente sirven es todavía débil”, resume acertadamente la contra-hipótesis. He visto que es más útil considerar la prueba de software como una *parte com-*

plementaria a QA que nos da información concreta acerca de la calidad del producto de software.

Problemática y Soluciones Propuestas

Es posible observar la siguiente problemática en la industria del software:

- Incremento en la demanda, originado no solamente por la necesidad de desarrollo de software “convencional” para una cantidad creciente de áreas de aplicación, sino por la asociada a los *legacy systems* y al *embedded software*.
- Incremento en la complejidad, debido a que los sistemas son cada vez más grandes, e involucran cada vez más la integración de software, firmware y hardware.
- Exigencia por la calidad, acentuada en buena medida por la globalización y el incremento en la “cultura informática”, pero también por la necesidad creciente de los denominados sistemas de misión crítica.

Estos son algunos enfoques que se han planteado para abatir cada uno de estos problemas:

- Incremento en la demanda: proveer herramientas y capacitación para desarrolladores y no-desarrolladores; facilitar el reuso de componentes, que permita construir software a partir de constituyentes maduros.
- Incremento en la complejidad: proveer ambientes de desarrollo que brinden mayor nivel de abstracción; desarrollar metodologías que faciliten la descomposición de problemas.
- Exigencia en la calidad: marcos como el *Total Quality Management for Software*, con fuerte énfasis en el incremento de la calidad por medio de la planeación estratégica, y el control estadístico de procesos; modelos de calidad para software (CMMI, MoProSoft, etc.), centrados en la definición y mejora de procesos en áreas clave del desarrollo de software; métodos formales, que por construcción, permitan desarrollar sistemas libres de defectos; prueba de software, que permita detectar anomalías en un producto de software.

La prueba de software es una actividad ingenieril que exige un perfil y una formación muy particular, en la que se utilizan técnicas y herramientas para detectar niveles inadecuados de calidad, aplicando una cantidad de recursos limitados (en especial tiempo y dinero) de forma tal que genere un valor agregado en el proceso de desarrollo de software. En los siguientes tres números abundaré en el tema. ©

- Luis Vinicio León

Referencias

- Myers, G. “The Art of Software Testing”. John Wiley & Sons; 1979
- “Conceptos de Prueba de Software”. www.e-quality.net/def.php