



Maddog firmando revistas en el stand de Linux Pro en la LinuxFest Northwest 2009.

Ingeniería de software de calidad

SE BUSCA

Cuando se trata de ingeniería de software, siempre necesitamos más.

POR JON "MADDOG" HALL

Recientemente regresé de la fantástica LinuxFest Northwest 2009 en Bellingham, Washington, una pequeña ciudad al norte de Seattle y hogar de diverso tipo de gente, desde los auto-denominados "hippies antiguos" hasta los programadores que han huido de Redmond para tener una vida más tranquila. En mi vuelo de regreso, que partía del aeropuerto de Bellingham, me senté junto a un caballero "de mi edad". Cuando lo saludé, él respondió con un toque de acento escocés.

Nuestra charla derivó a nuestras ocupaciones: le hablé de mi trabajo "venta de Software Libre" y él me habló del suyo como ingeniero de sistemas para Chevron. A medida que la conversación avanzaba, comentó todos los esfuerzos que tenía que hacer para utilizar los productos de Microsoft y la cantidad de veces que se sentía "mareado" por culpa de eso. Su voz se hizo más cálida cuando habló sobre cómo los sistemas Unix y Linux son mucho más estables y cómo le gustaban mucho más.

Luego dijo algo que yo había oído hace mucho tiempo: "Por supuesto, para aplicaciones de misión crítica, las verdaderas apli-

caciones de misión crítica, el tipo de aplicaciones que tienen que funcionar sí o sí, nunca utilizaría software controlado por ordenadores. La hidráulica es el camino a seguir. El software es demasiado poco fiable".

Mi rostro enrojeció, después de todo mi vida gira en torno al software y a los ordenadores digitales. Los sistemas que he ayudado a crear han lanzado astronautas a la Luna, gestionado almacenes automatizados y realizado otros trabajos de "misión crítica". Pero mientras estaba ahí sentando escuchando sus historias sobre cómo un día Chevron perdió 60 millones de dólares porque alguien fue lo suficientemente descuidado como para no probar su código, pensé de nuevo en algunos de los proyectos en los que las pruebas tuvieron sentido "a posteriori", cuando ya habían aparecido los problemas.

Volverse Quisquilloso

Recuerdo que una vez recibí una copia de un software de prueba en campo e intenté instalarlo en mi ordenador, aunque no pude. Pensando que quizá era debido a mi particular configuración de hardware, miré

el código fuente del programa de instalación que, afortunadamente, estaba escrito en un lenguaje de guión, y vi que era imposible llegar a ningún sitio a través del código. En otras palabras, el ingeniero que lo había escrito no había intentado hacerlo funcionar ni una sola vez.

Inmediatamente, me dirigí a la oficina del ingeniero y le "di la charla" por haber puesto en peligro la totalidad de la prueba de campo del producto y, por tanto, todo el calendario previsto de entrega. Todas las personas de la empresa y sus medios de vida dependen de que nosotros cumplamos ese calendario, y aunque no quería enviar un producto defectuoso, era importante respetar los plazos.

En otra ocasión, se había descubierto – por causas ajenas a Digital – que 12.000 tarjetas de memoria tenían un chip defectuoso, lo que significaba que había que recuperarlas todas y rehacerlas. En aquel entonces, las memorias estaban cerca de los 1.000 dólares por mega byte, así que nos encontrábamos ante un retraso en el envío de un nuevo sistema, además de una pérdida potencial de 12 millones de dólares para la empresa.

Una posible solución era hacer un examen por software de la memoria cada pocos milisegundos; sin embargo, el software no pudo distinguir si una tarjeta en concreto tenía este defecto o si era una tarjeta de memoria que funcionaba con normalidad. Así que todos estos modelos en particular tendrían un examen de memoria mientras estuviesen funcionando.

Un ingeniero de hardware propuso que bastaría con que Ultrix (nuestro sistema Unix en ese momento) incluyera el software examen de memoria en el núcleo, y "problema solucionado". Señalé que el problema era del hardware y que no había ninguna garantía de que éste continuara ejecutando Ultrix. Algún día podría correr VAX/Eln, un verdadero sistema operativo en tiempo real utilizado para diversas operaciones de misión crítica.

Le dije que tal vez, un día en el futuro, cuando fuera necesario bajar las barras de control de un reactor nuclear, VAX/ELN haría una pausa de unos cuantos milisegundos por culpa del examen de la memoria y para cuando ya se decidiera a bajar completamente las varillas, el reactor nuclear sería un montón de cenizas.

El grupo de hardware volvió a fabricar las tarjetas de memoria.

La ingeniería de software de calidad es un trabajo serio. Nos hace falta más. ■