

## De hacker a programador

# PLENITUD

La diferencia entre el hacking y el desarrollo de software podría estar en la calidad de las comprobaciones y la documentación de los proyectos.

POR JON "MADDOG" HALL

Un amigo mío que dirige una empresa de consultoría de software libre se lamentaba de que, cuando pregunta a sus ingenieros si un proyecto está "terminado", con frecuencia le responden que "sí". Sin embargo, cuando vuelve a comprobar el trabajo antes de facturárselo al cliente, se encuentra que no tiene documentación o ésta es inadecuada, no hay pruebas de control de calidad y no hay comunicación con los desarrolladores del código fuente original para incluir los cambios en el software. En resumen, el trabajo se parece más a un "hack" rápido que a un encargo terminado.

Desgraciadamente, esto sucede mucho con los desarrolladores de software y, a menudo, con aquellos que son contratados desde "la comunidad". Acostumbrados a trabajar en pequeños proyectos en los que "el código es la documentación" o para usuarios finales que no pagan mucho dinero por el desarrollo de software, no tienen la experiencia o el rigor necesario para proyectos más grandes o la producción de código de calidad comercial.

Con los mejores programadores, el proyecto comienza antes de que el código sea escrito, asegurándose que los requisitos de los clientes son entendidos perfectamente. Se realizan líneas de tiempo detallando el desarrollo del trabajo, las cuales incluyen el tiempo y los recursos para el aseguramiento de la calidad y la documentación. Después se considera el diseño, se escribe el código y luego comienza el "trabajo de verdad".

## Control de Calidad

El control de calidad (QA) ha de hacerse incluso para el programa o el cambio más pequeño. La mayoría de las veces es mejor que este tipo de pruebas sea realizado por alguien distinto a quien escribió el código, porque las pruebas de control de calidad realizadas por los propios programadores están contaminadas por sus puntos de vista de lo que hay que hacer y que no necesita probarse, y su confianza en sus habilidades para programar. Además, una persona diferente

puede crear casos de prueba muy lejos de los que crean los programadores para su propio código.

## Documentación

La documentación es otro ejemplo en el que otra persona con habilidades y puntos de vista diferentes podría hacer un trabajo mejor que el programador original en la creación de la documentación utilizable para el nuevo código (o cambios al código viejo). Documentar lo "intuitivamente obvio" es a menudo lo que marca la diferencia entre documentación útil y documentación que no sirve para nada en absoluto.

En el caso del control de calidad y la documentación, el programador tiene que crear listas detalladas de "puntos clave" para indicar al personal del control de calidad y de documentación qué ha cambiado y cómo afectarían esos cambios al código anterior y sugerir lo que debe ser probado y documentado. Más tarde, el programador y líder del proyecto debe revisar las pruebas de control de calidad y verificar los resultados, además de los borradores de la documentación.

En proyectos y empresas pequeñas, la misma persona puede encargarse de la programación, la documentación y el control de calidad, pero éstas deben ser realizadas. Por otra parte, aquí es donde realmente el "Software Libre" puede aprovecharse de los miembros de la comunidad.

## Llamada a la Comunidad

Muchas veces he oído el lamento de los usuarios finales diciendo: "no puedo ayudar porque no soy programador". Sin embargo, estas mismas personas podrían ser buenos escritores o (suponiendo que los desarrolladores les indiquen lo que hay que probar) buenos probadores.

Con el uso de modernas herramientas de colaboración en línea como los wikis, por ejemplo, no hay ninguna razón por la cual los usuarios finales de Software Libre no puedan contribuir a realizar una buena documenta-



ción. Si usted lee una parte de la documentación en una wiki y no la entiende, una vez que averigüe cómo funciona el software, intente editar la documentación en línea para hacerla más clara.

Si, como cliente o usuario de software libre, está siguiendo un proyecto en particular y cómo lo están realizando los programadores, intente probar las nuevas versiones del código para concentrarse en probar lo que los programadores han cambiado, así como la funcionalidad general de la sistema. Sí, esto puede que requiera aprender a utilizar un sistema de seguimiento y notificación de problemas, pero el aprendizaje del sistema de seguimiento y notificación de problemas es parte de su contribución al Software Libre.

Curiosamente, cuando se tienen acuerdos con algunos "profesionales" de las compañías de código cerrado, hay que pagar por el "privilegio" (a veces hasta 500 dólares o más) de notificar un informe de error y más para conseguir que se elimine u obtener asesoramiento para resolverlo.

Por supuesto, los líderes de equipo o los directores de proyectos son los responsables en última instancia de la calidad global y de la "integridad" del programa. Deben explicar las reglas básicas para el desarrollo de código dentro de sus proyectos, y las normas han de incluir un alto nivel de pruebas y documentación.

Le dije a mi amigo todo esto mientras seguía quejándose sobre su propia empresa. Dijo que iba a tener una conversación con sus programadores sobre la creación de software de clase mundial. Le mencioné que me gustaría escribir este artículo y así, tal vez, facilitar su trabajo. ■